

המדריך השלם

Access 2000

VBA

Microsoft® Press

יש להתעלם מכל מה שנכתב על התקליטור
את קוד המקור ניתן להוריד מאתר הود-עמי
בתיקיה "קבצי תרגול לספרים"

עורך ראשי ומקצועי: **יצחק עמיhood**

תרגומים: **במיליטס אחירותת תיעוד בע"מ**

עריכה לשונית ועיצוב: **ענת קדם צבי, שרה עמיhood**

עריכה מקצועית: **ארז ירונ**

אל גוטليب, מנהל מרכז התמיכה של מיקרוסופט ישראל

איןדקס: **זהר עמיhood**

עיצוב עטיפה: **סטודיו מצגר**

שמות מסחריים

שמות המוצרים והשירותים המוזכרים בספר הינם שמות מסחריים רשומים של חברותיהם. הוצאת Microsoft Press והוצאה הוד-עמי עשו כמיטב יכולתן למסור מידע אודוט השמות המסחריים המוזכרים בספר זה ולציין את שמות החברות, המוצרים והשירותים. שמות מסחריים רשומים (registered trademarks) המוזכרים בספר צוינו בהתאם. Microsoft Windows ו-Office הם מותרים רשומים של חברת

הودעה

ספר זה מיועד לתת מידע אודוט מוצרים שונים. נעשו מאמצים רבים לגורם לכך שהספר יהיה שלם ואמין ככל שניתנו, אך אין משתמשת מכך כל אחריותה.

המידע ניתן "כਮות שהוא" ("as is"). הוצאה הוד-עמי אין אחריותה כלפי יחיד או ארגון עבור כל אובדן או נזק אשר ייגרם, אם ייגרם, מהמידע בספר זה, או מהתקליטור המצורף לו.

לשם שטף הקריאה כתוב ספר זה בלשון זכר בלבד. ספר זה מיועד לגברים ונשים כאחד ואיןכוונו להפלות או לפגוע בזכיר המסתמשים/ות.

□ טלפון: **09-9564716**

□ פקס: **09-9571582**

□ דואר אלקטרוני: **info@hod-ami.co.il**

□ אתר באינטרנט: **www.hod-ami.co.il**

המדריך השלים

Access 2000

VBA

רייק דובסון

Designed for



Microsoft®
Windows NT®
Windows®98

Microsoft Press

הוצאת הود-עמי
לספרי מחשבים



Programming Microsoft Access 2000

By Rick Dobson

Published by **Microsoft Press**

Editor: I. Amihud

Copyright © 1999 by Microsoft Corporation

Original English Language edition Copyright 1999 by Rick Dobson.

All rights published by arrangement with the original publisher, Microsoft Press,
a division of Microsoft Corporation, Redmond, Washington, U.S.A

Hebrew language edition published by

Hod-Ami Ltd. Copyright © 2000

(C)

כל הזכויות שמורות

**הוצאת הוד-עמי
לספרי מחשבים בע"מ
ת.ד. 6108 הרצליה 46160
טלפון : 09-9571582 פקס : 09-9564716
info@hod-ami.co.il**

אין להעתיק או לשדר בכל אמצעי שהוא ספר זה או קטעים ממנו בשום צורה ובשום אמצעי
אלקטרוני או מכני, לרבות צילום והקלטה, אמצעי אחסון והפצת מידע, ללא אישור בכתב
מאת ההוצאה, אלא לשם ציטוט קטעים קצרים בציון שם המקור.

הודפס בישראל 2000
(כולל התאמה לעברית)

All Rights Reserved
HOD-AMI Ltd.
P.O.B. 6108, Herzliya
ISRAEL, 2000

ISBN 965-361-206-9 מסת"ב

תוכן עניינים מקוצר

הקדמה.....	15
פרק 1 : היכרות עם VBA	21
פרק 2 : מודלי גישה לנוטונים.....	85
פרק 3 : עיצוב טבלאות	155
פרק 4 : טיפול בנוטונים באמצעות שאלות	199
פרק 5 : בניית ממשק משתמש באמצעות טפסים.....	269
פרק 6 : ייצור דוחות.....	301
פרק 7 : מודלי חלוקה, טופס ודוח.....	335
פרק 8 : אובייקטי Microsoft Office	369
פרק 9 : שילוב Access עם יישומי Office אחרים	409
פרק 10 : עבודה עם מסדי נתונים מרובי-משתמשים	439
פרק 11 : שכפול מסדי נתונים	467
פרק 12 : בניית פתרונות עם MSDE ופרויקטי Access	497
פרק 13 : Web Access 2000 וה-Access 2000	537
פרק 14 : מהדרת Office 2000 למפתחים (ODE)	567
אינדקס	581

תוכן עניינים

הקדמה	15
על Access 2000	15
רכיבים חדשים ב- Access 2000	15
על הספר	19
קהל היעד	19
תוכניות וקוד המקור בספר	19
תקליטור המוצר	19
מוסכמות בהן נעשה שימוש בספר זה	20
פרק 1 : היכרות עם VBA	21
אוספים, אובייקטים, מאפיינים, שיטות ואיורים	22
אוספים ואובייקטים	22
איורים	25
שגרות ומודולים	27
שגרות משנה	27
שגרות פונקציה	30
שגרות מאפיין	34
מודולים	35
משמעות VBE	36
chlonoת VBE	36
סורק האובייקטים	42
Jet, סוגים נתוניים והצורות	43
Jet	43
סוגי נתונים	45
הצורות	49
לוגיקה של התנויות ומבנה לולאה	50
If...Then	50
Select Case	54
For...Next	55
With...End With - ו For Each...Next	57

61	Do...Loop
63	פונקציות מוכללות.
67	תקציר פונקציות נבחרות.
73	ণיפוי באגים ולכידת שגיאות.
74	לכידת שגיאות – תחביר בסיסי.
75	דוגמאות ללבידת שגיאות.
79	העלאת שגיאות.
81	פקודות מאקרו.
81	תכנון פקודות מאקרו.
83	לעומת פקודות מאקרו.
85	פרק 2 : מודלי גישה לנדרונים.
86	סקירה כללית על DAO.
86	סביבות העבודה של Jet.
86	סביבות העבודה של ODBCDirect.
87	אובייקטים המשותפים לשביבות העבודה של Jet ו-ODBCDirect.
93	אובייקטי סביבת העבודה של Jet.
96	אובייקטי סביבת העבודה של ODBCDirect.
98	סקירה כללית על ADO.
100	הספריה ADODB.
134	הספריה ADOX.

155	פרק 3 : עיצוב טבלאות .
156.....	טבלאות ומסדי נתונים יחסיים.
156.....	נормול.
159.....	קשרי גומלין בין טבלאות.
162.....	יצירת טבלאות באמצעות אשיים.
162.....	אשיי מסד הנתונים.
163.....	ASF הטבלאות.
164.....	בונה השדות.
165.....	יצירת טבלה באופן ידני.
165.....	סוגי נתונים.
172.....	AIMOT נתונים.
174.....	יצירת אינדקסים.
178.....	יצירה וניהול של טבלאות באמצעות קוד.
178.....	יצירת טבלה.
183.....	עבודה עם אינדקסים.
188.....	אכלוס טבלה באופן דינמי.
192.....	עבודה עם נתונים בתבניות אחרות.

פרק 4 : טיפול בנתונים באמצעות שאלות 199

200.....	סקירת סוגים של שאלות
200.....	שאילנות בחירה (Select)
202.....	שאילנות פעולה (Action).
204.....	סוגי שאלתה אחרות...
205.....	פעולות להגדלת נתונים.
206.....	עובדת עם מקורות נתונים מרוחקים...
207.....	עיצוב שאלות באופן ידני.
207.....	שימוש באשפים.
216.....	שימוש בתצוגה עיצוב.
228.....	שאילנות בחירה מיוחדות.
228.....	שאילנות פרמטר.
234.....	שאילנות פעולה.
234.....	שאילנות עדכון.
237.....	שאילנות הוספה.
239.....	שאילנות מחיקה.
242.....	שאילנות יצירתיות טבלה.
243.....	תכנות שאלות באמצעות SQL ו-ADO
244.....	משמעותי SELECT
249.....	פונקציות הגדלת נתונים.
255.....	זוגות ושורות מאוחסנות.
261.....	הפעלת שאלות על מקורות נתונים מרוחקים.
261.....	הפעלת שאלות על מקורות ODBC מוקשרים.
263.....	הפעלת שאלות בפרויקט נתונים של Access
265.....	תכנות שאלות עברו מסדי נתונים מרוחקים.

פרק 5 : בניית ממשק משתמש באמצעות טפסים..... 269

270.....	טופס מסך פתיחה.
270.....	יצירת מסך פתיחה.
271.....	בקרה על מסך התצוגה.
272.....	טופס מסך ניווט.
272.....	ניווט בעזרת היפר-קישורים.
274.....	ניווט באמצעות קוד.
276.....	קישור טפסים לנתונים.
276.....	שימוש באשר הטפסים האוטומטיים.
277.....	עיצוב מותנה.
279.....	טפסי משנה.
281.....	בדיקה מיידית והציג נתונים.
281.....	יצירת טופס בבדיקה מיידית.
287.....	הציג מייד דינמית.

289.....	יצירת תרשימים מקבוצות משנה של נתונים.....
291.....	טיפול בקבצים באמצעות VBA.....
291.....	מספר טפסים ופקדים
293.....	הסתרה והצגה של טפסים
294.....	מספר טפסים בפרויקט אחר.....
295.....	שימוש במחלקות טופס
295.....	הציג מאפיינים ושיטות של מחלקת טופס
296.....	טיפול במחלקות טופס
298.....	הਪניות למופעי מחלקת טופס.....

פרק 6 : יצירת דוחות.....

302.....	כיצד ליצור דוח
302.....	שימוש באשף הדוחות האוטומטיים
302.....	שימוש באשפים נוספים
303.....	יצירה ידנית של דוח בתצוגת עיצוב
306.....	מקרה לדוגמה : ספר האורחים של FrontPage
306.....	ייבוא נתונים
307.....	המרת מבנה הנתונים.....
310.....	יצירת תוויות מען.....
311.....	יצירת מכתב אחד.....
314.....	מיון, קיבוץ וחישוב.....
315.....	יצירת דוחות מרובי עמודות
316.....	שימוש באשף הדוחות
317.....	יצירת דוח מותאם אישית
320.....	ഫצת דוחות באמצעות תמונות
320.....	יצירת תמונה
320.....	הציג תמונה (Snapshot)
322.....	שימושים נוספים בתמונות
323.....	הפיקת דוח לדינמי
323.....	עיצוב והוספת תוכן
326.....	סיכום ערכי דף
328.....	עדכון דוח באופן דינמי.....
330.....	טיפול בדוחות ובפקדי דוח באמצעות קוד
330.....	מספר דוחות
332.....	שינויי מאפייני פקד דוח
333.....	שלוח תמונות

פרק 7 : מודולי מחלוקת, טופס ודוח

335.....	סוגי מודולים.
336.....	מודולי מחלוקת.
336.....	פונקציות מאפיין ושיטות מותאמות אישית.
337.....	יצירת מופעים של מחלוקת.
337.....	מחלקות וAIRUIM מותאמים אישית.
338.....	שיטות ומאפיינים מותאמים אישית.
338.....	חטיבת מאפיינים באמצעות משתנה ציבורית.
339.....	חטיבת מאפיינים באמצעות פונקציית מאפיין.
340.....	משתנים ציבוריים לעומת פונקציות מאפיין.
340.....	מודולי מחלוקת ומקורות נתונים.
343.....	מקרה לדוגמה : תכונות משק כניסה בסיסי.
343.....	טופס כניסה הראשון.
346.....	טופס כניסה השני.
353.....	תכונות AIRUIM לתוכן מחלוקת מותאמות אישית.
354.....	שתי מחלוקות AIRUIM מוכללות.
354.....	שימוש במילת המפתח WithEvents לכלידת AIRUIM שהופצו.
355.....	מודולים סטנדרטיים הגורמים ל-AIRUIM.
357.....	שימוש באוסף ALL.
357.....	מאפייני AccessObject.
357.....	האוספים All.
359.....	מספר חברי האוסף All.
360.....	התאמת לשוגי הקבצים *.adp ו- *.mdb.
360.....	AllModules ו- AllForms.
362.....	עריכת מודולים באמצעות קוד.
362.....	גישה עERICA.
363.....	הוספת טקסט למודולים.
364.....	מחיקת טקסט ממודולים.
365.....	הוספת טקסט למודול מחלוקת טופס.
366.....	מחיקת טקסט ממודול מחלוקת טופס.

פרק 8 : אובייקטי Microsoft Office

369.....	שימוש באובייקטים המשותפים של Office.
372.....	האובייקט FileSearch.
373.....	ניהול חיפוש קובץ בסיסי.
374.....	מיון הקבוצה המוחזרת.
375.....	חיפוש המבוסס על תוכן הקובץ.
377.....	הכללת קרייטיריוני חיפוש מרובים.
379.....	האובייקט Assistant.
380.....	מסיעים.

387	בלוניים
394	האוובייקט CommandBar
394	ספרת האלמנטים בסרגלי פקודות
397	טיפול בסרגלי פקודות מוכללים
401	יצירת סרגלי פקודות מותאמים אישית
403	שינויי סרגלי פקודות מותאמים אישית
404	יצירת סרגלי פקודות מוקפצים
407	מחיקת סרגלי פקודות מותאמים אישית

פרק 9 : שילוב Access עם יישומי Office אחרים 409

410	קישור Access ליישומי Office אחרים
410	מנהל ISAM ברי התקנה
411	השיטה OpenDataSource
412	אוטומציה
415	עבודה עם Matrox Access
415	עבודה עם ערכיהם מתוך גליונות עבודה של Excel
418	יצירה דינמית של טבלאות Access המבוססות על גליונות עבודה של Excel
421	הפעלת שגרות Excel מתוך שגרות Access
424	עבודה עם Outlook מתוך Access
424	ספרת פריטים בתיקיה אנשי קשר
425	הוספת פריט לתיקיה אנשי קשר
426	מחיקת פריט מהתיקיה אנשי קשר
427	הוספת פריטים רבים לתיקיה אנשי קשר
428	מחיקת פריטים רבים מהתיקיה אנשי קשר
430	עבודה עם Word מתוך Access
430	אוטומציה של Word מתוך Access
432	הפקת תוויות מען
435	הפקת מכתב אחד

פרק 10 : עבודה עם מסדי נתונים מרובי-משתמשים 439

440	שיתוף קבצים
440	המאפיין Connection Control
441	רישימת המשתמשים
442	שיתוף טפסים
443	נעילת רשות באופן ידני
443	רعنון ערכי שדה
445	נעילת רשומות באמצעות קוד
447	שיתוף ערכות רשומות
447	נעילה ברמת השורה
449	נעילה ברמת עמוד

450.....	אבטחה
451.....	חולפות לאבטחה ברמת-משתמש.....
454.....	בקהה על אבטחה ברמת-משתמש באמצעות קוד
465.....	טרנוזקציות

פרק 11 : שכפול מסדי נתונים

467.....	כיצד עובד השכפול
469.....	שכפול בעזרת סמל המזودה.....
469.....	פקודת השכפול
470.....	תכנות OJN.....
471.....	מנהל השכפול
471.....	סינכרון דרך האינטרנט
472.....	שינויים בעיצוב השכפול
476.....	חידושים בתחום השכפול ב- Access 2000
476.....	שכפול דו-כיווני ב- Jet-SQL Server
477.....	עדכונים ברמת-עמודה.....
477.....	רמות החשיפה של עותק משוכפל
478.....	פתרונות התങשיות המבוסס על קידימות
478.....	שכלולים שונים
480.....	טכניקות פיתוח ב- JRN
480.....	הפיקת מסד נתונים לבר-שכפול
483.....	יצירת עותקים מלאים נוספים
484.....	יצירת עותק משוכפל חלקי, וمسננים
488.....	סינכרון עותקים משוכפלים
489.....	עובדת עם עותקים משוכפלים הנמנעים מחיקות
492.....	עובדת עם מאפייני עותק משוכפל
495.....	דחישה והצפנה של עותקים משוכפלים

פרק 12 : בניית פתרונות עם MSDE ופרויקטי Access

498.....	מנגנון הנתונים של Microsoft (MSDE)
498.....	לעומת Jet MSDE
501.....	פרויקט Access
501.....	חיבור פרויקט Access עם מסד נתונים
503.....	ليمוד מסדי הנתונים NorthwindCS ו- Pubs
508.....	דיagramות וטבלאות של מסדי נתונים
508.....	מייפוי קשרי גומלין באמצעות דיagramות של מסדי נתונים
510.....	ניהול דיagramות של מסדי נתונים מרוביים
512.....	עריכת טבלה בחלוון של דיagramת מסד נתונים
512.....	עריכה ויצירה של טבלאות באמצעות גילוון עבודה
513.....	תציגות ושגרות מאוחסנות
514.....	שימוש בתציגות

516.....	שימוש בשגרות מאוחסנות.....
519.....	דוחות וטפסים.....
520.....	מיון ועיצוב באמצעות דוחות, ועוד.....
521.....	הוספת היפר-קישורים
522.....	עריכה והציג של נתונים באמצעות טפסים
524.....	סינכרון חדש של טופס באמצעות נתוני יחיד לרבים.....
527.....	סוגיות של תוכנות.....
527.....	עובדת עם טפסים.....
530.....	עובדת עם מודולים עצמאיים.....

פרק 13 : Web וה-Access 2000

538.....	גישה מסורתית.....
538.....	פרסום גליונות נתונים.....
544.....	שימוש בטפסי HTML.....
548.....	שימוש בתמונות FTP עם דפדפני Netscape
549.....	שימוש בהיפר-קישורים
549.....	סוגים של היפר-קישורים
549.....	סוג הנתונים Hyperlink
550.....	הוספה ועריכה של היפר-קישורים
552.....	דוגמאות של היפר-קישורים
553.....	יצירת דפי גישה לנתונים והשימוש בהם
554.....	יצירת דף גישה לנתונים
555.....	יצירת דף טורי פשוט והשימוש בו
557.....	קידוץ רשות
559.....	רכיבי Web של Office 2000 בדף גישה לנתונים
563.....	נושאי תוכנות הקשורים בדף גישה לנתונים

פרק 14 : מהדורות Office 2000 למפתחים (ODE)

568.....	סקירה כללית על ODE
568.....	כלי VBA לפירון עבודה
569.....	גישה לנתונים
570.....	הטמעה וניהול
571.....	ספרן הקוד
574.....	פתרונות אריזה והטמעה
576.....	גישה לנתונים ביישומים שאינם Access
578.....	חתימה דיגיטלית של פרויקטי VBA

581 אינדקס

הקדמה

על Access 2000

הישום Access הוא מערכת פיתוח פופולרית נפוצה ביותר, עקב היותו חלק מקבוצת Microsoft Office. לתוכה רבים מעוניינים שיישומי Access יפעלו בשיטוף עם שאר יישומי Office בזרה וקלת תחזקה, ללא צורך בתמיכת איש הפיתוח. הטכניקות המתוארות בספר יסיעו לך לענות על ציפיות אלו (לקבלת תמיכה נוספת, בקר בשני אתרים www.microsoft.com/office/ : Microsoft העוסק ביכולות הכלליות של Access, ו- www.microsoft.com/officedev, הסוקר את רכיבי הפיתוח של המוצר).

רכיבים חדשים ב- Access 2000

Access 2000 מתקדם באופן משמעותי בתחוםים רבים. Microsoft יקרה מוצר הכלול חידושים רבים, אך מתנהג עדין כמו הגרסה המוכרת של Access. הספר מדגיש את החדשושים בחמשة תחומיים עיקריים :

- ↳ אובייקטי נתונים של ActiveX (ActiveX Data Objects) ADO – ActiveX
- ↳ תפקודיות מורחבת של SQL Server בסביבות מחשב שונות,
- ↳ (packaging enhancements) Visual Basic for Applications ורחבות אריזה (VBA)
- ↳ שיפורים מנגן Jet,
- ↳ תפקודיות Web משופרת בסביבות מחשב שונות.

אובייקטי נתונים של ActiveX

אובייקטי נתונים של ActiveX - **ADO**, מחליפים כמעט את כל פונקציות הגישה לנתונים שbowcuו באמצעות אובייקטי גישה לנתונים (Data Access Objects - DAO). Access 2000 מציע תפקודיותADO באמצעות שלוש ספריות: ADO, ADODB ו-JRO.

הספריה **2.1 ADODB ActiveX Data Objects** כוללת פונקציות עיבוד גישה לנתוני **Recordset**, **Connection** ו-**Command**. אפשר להשתמש באובייקטים אלה, במאפיינים ובשיטות (methods) שלהם, כדי להתחבר ולטפל במקור נתונים. אובייקט Connection משך לטכנולוגיה חדשה של ספק DB OLE. לטכנולוגיה זו חשיבות חיונית עבור גישה אוניברסלית לנתונים – **UDA** – Microsoft Universal Data Access (Microsoft Universal Data Access) המספקת גישה עיתרת ביצועים למגוון מבנים נתונים (יחסיות ולאיחסיות כאחד). טכנולוגיית UDA מאפשרת עיבוד משולב של מקורות נתונים מקובלים, כגון Jet ו-SQL Server, עם מקורות פחות מקובלים, כגון דואר, ספריות קבצים וAPILO וידאו. UDA מייצגת התקומות הפתוחות מעבר למשקי הנתונים המקובלים כיום, כגון קישוריות מסד נתונים פתוחה (Open DataBase Connectivity - ODBC), אובייקטי נתונים מרוחקים (DAO ו-RDO – Remote Data Objects).

הספריה **2.1 Microsoft ADO Ext. for DDL and Security (ADOX)** מציעה גישה מבוססת-אובייקט להגדרת נתונים ולאבטחה ברמת-משתמש. היא מספקת אוסף **משתמשים וקבוצות** (Users and Groups) בرمת משתמש מקובל של Jet. היא גם קוסרת הרשות בקבץ מידע של קבוצת עבודה (Members) באוסף **משתמשים וקבוצות** שבקובץ מידע של קבוצת עבודה (Workgroup information file). מודול ADOX של ספריה זו מטפל במטלות הגדרת נתונים באמצעות אובייקטים כגון **טבלאות** (Tables), **עמודות** (Columns), **אינדקסים** (Indexes), **מפתחות** (Keys), **צוגות** (Views) ו**שגרות** (Procedures). תוכל לנצל אובייקטים אלה כדי להגדיר בקורס דינמית טבלאות, אינדקסים וקשרי גומלין חדשים בין טבלאות. תוכל גם להגדיר שאילות על טבלאות.

הספריה **2.1 Microsoft Jet and Replication Objects (JRO)** מספקת עיקר שירותים שכפול (Replication) של מגנון Jet באמצעות משק ADO. המודול החדש ADO מאפשר לנצל את היתרונו של רכיבי השכפול המתוכנתים של מסד נתונים Jet. בנוסף, המודול כולל פונקציות של מגנון Jet, כגון דחיסת מסדי נתונים ורענון המטמון.

הגברת הפעולות המשולבת של SQL Server

Access 2000 מאפשר לטפל במסדי נתונים מסווג Enterprise באותה קלות בה מטפלים במסדי נתונים מבוססי Jet, קישוריות ADO היא חלק מהסיבה לכך. סוג הקובץ החדש פרויקט Access (.adp), מאפשר גמ השתלבות הדוקה עם גרסאות 6.5 ו-7 של SQL Server. סוג קובץ זה עובד עם מסדי נתונים מבוססי SQL Server ו-MSDE (Microsoft Data Engine) בדיקוק כפי שקבעי mdbאפשרים לטפל במסדי

נתונים מבוססי Jet. MSDE הוא מנגנון מסד נתונים חדש שמבוסס על מודל SQL Server. הוא מיועד לפתרונות עבור קבוצות משתמשים, ומשלים את תפקודו המוגבל Jet. תוכן לבחור לפתח פתרונות בעזרת Jet או MSDE.

הפרויקטים של Access מגלים תוכנות ושגרות במסגרת המוכרת של מסד הנתונים. תוכל להתחבר מיידית למסדי SQL מרוחקים תוך שימוש במשק הגרפי הפשט והמוכר מסדי נתונים מסווג Jet. תוכל גם לנצל את נתוני SQL Server בעובדה עם טפסים ודווחות של Access (בדוק كيف שאתה נהוג עם נתונים Jet).

לשווותך גם יכולת פעולה שקופה של OLE DB עם SQL Server ומקורות נתונים עורפיים (Back-End) בסביבות מחשב שונות. השימוש באובייקט Connection של ADO ובספק DB OLE מאפשר להתחבר למקורות נתונים מרוחקים ולהפנות (reference) אליהם באמצעות קוד עבור יישומים מותאמים אישית.

VBA והרחבות האריזה

Access 2000 ממשיך בהתאם גרסת VBA שלו ל-VBA של שאר יישומי Office 2000 כולל עורץ VBE – Visual Basic Editor) שמשק המשמש שלו זהה לזה של PowerPoint, Word ו-Excel. ניתן להעיר את מיעוטיות ניהול הקוד והפיתוח ישירות אל שאר היישומים, ועל ידי כך להעשיר אותו בפתרונות גישה לנתונים.

מהדורות הפיתוח של Office 2000 כוללת אפשרויות אריזה והפעלה משופרות. לדוגמה, ניתן להפעיל פתרונות באוצרות MSDE ופתרונות המתבססים על מסדי נתונים דמיי-SQL Server. סידור זה מאפשר לקבל את עשר התצוגות והשגרות של המשק הגרפי וממשק התכנות. הדבר חשוב במיוחד במקרה של יישום קטן שפותח וזוקק מעטה ליכולות של מסד נתונים.

- Access 2000 יכולה הטמעה חדשה המאפשרת להפיץ חבילות התקינה מותאמות אישית (custom setup packages) באוצרות האינטרנט. כך תוכל להרחיב במידה רבה את תחום לקוחותך. מاجر הלוקחות הפוטנציאליים שלך יגדל ויקיף את כל מי שמחובר לאינטרנט, גם אם הוא נמצא בקצה העולם.

шиפורים במנגןון Jet

Access 2000 כולל את גירסה 4 של מנגנון מסד הנתונים Jet, המציע שיפורים בתחוםים תפקודיים אחדים. מצודדת במיוחד היא זמינות נעילת דף בرمת השורה (row-level page locking). הנעה הנמוכה ביותר של גרסאות קודמות של Access הייתה בرمת הדף. אחת הסיבות להקלת נעילה בرمת השורה היא זמינות תמיכת Unicode בתווי טקסט. ב-Office 2000 ניתן ליצג נתונים טקסט בשפות שונות ביישומים רב-לשוניים. שיטת הקידוד החדשה של שדות טקסט מגדילה את הזיכרון הדרוש לאחסוןתו בודד מבית אחד לשני בתים, והדף גדול מ-2kb ל-4kb. מאז העליה בגודל הדף, Microsoft אפשרה לנעה בرمת השורה לצמצם את אפשרות נעילת דף בו-זמנית ביישומים מרובי משתמשים.

גם בשכפול מסד נתונים חלו מספר שיפורים, אחד מהם הוא זמינות **שכפול בرمת העמודה** (column-level replication). גרסאות Jet קודמות זיהו התàngשיות בرمת השורה, שני עותקים התגשו אפלו אם שינוי שדות שונים באותה רשומה. שכפול בرمת העמודה משפר את הביצועים על ידי הסרת התàngשיות אלו. Access 2000 כולל גם שכפול דו-כיווני בין מסדי הנתונים Jet ו-SQL Server. הגירסה הקודמת אפשרה שכפול חד-כיווני בלבד, מ-SQL Server בלבד.

עוד שיפור ראוי לצוון הוא בקרה תקונית שבאמצעותה משנים את ערך שדות המספר האוטומטי (AutoNumber). בעת ייצור טבלה תוכל להגדיר את הערך ההתחלתי ואת ערך הצעד (Step) של שדות מספר אוטומטי. תוכל גם לשנות ערכים אלה לרשותה הבאה. בפרויקט Access ניתן להגדיר שדות מספר אוטומטי מתצוגת **עיצוב טבלה** (Table Design). אפשר גם לשנות ערכים אלה לאחר הייצהר הראשונית של הטבלה.

Jet אף מציע גישה ברמת SQL לagetions ושורות. שיפור SQL של Jet מאפשרים ליצור או לשנות את שני הסוגים של מודולי אובייקט מסד הנתונים.

הגברת הפעולות המשולבת ב-Web

אחת התכונות החדשנות והמשמעות של Access 2000 הוא דפי גישה לנתונים, המתנהגים כמו טפסים ודוחות Web-access. תוכל לעצב דפי Web הקשורים ישירות לנתונים מסוים מסוג Jet או SQL Server. דפים הפעילים בטפסים אפשרים למשתמש לעזרך, להוסיף ולבטל רשומות בצהרה גרפית מתוך הדף. תוכל להשתמש בכל זמן-עיצוב (design-time) כדי לבקר באופן תקוני תוכנות אלו, וגם כדי לנצל את יכולות המיוון והסינון. הדפים אינם אפשרים ייצרת טפסי-משנה, אך ניתן ליצור דפים מקובצים של גישה לנתונים (grouped data access pages) שגדלים באופן מותנה בהתאם לקלט המשתמש.

דף גישה לנתונים יכולם לשמש גם כמארכיים של רכיבי אינטרנט חדשים, ונitinן להשתמש בהם לייצרת דפים המכילים גליונות אינטראקטיביים, תרשימים דינמיים וטבלאות ציר (pivot tables). ניתן גם לחבר את רכיבי ה-Web של גליונות ותרשיימים לנתונים המוצגים בדפים מקובצים או בלתי מקובצים של גישה לנתונים. משמעות הדבר היא שבאפשרות להציג חישובים ותרשיימים המשתנים בצהורה דינמית בעת המעבר מרשותה אחת לבאה אחרת. דף גישה לנתונים המכילים טבלאות ציר אינם מתקשרים עם מקורות נתונים אחרים בדף, אלא מספקים "סידור בטבלת ציר" (pivoting) בסגנון Excel, על ידי הנעה גרפית של חלק נתונים בתצוגה נפרדת. בנוסף, ניתן לנצל טבלאות ציר לניטוח סוג נתונים רבים, לרבות מקורות נתונים של Jet, SQL Server ועיבוד ניתוחי מקוון - OLAP (OnLine Analytical Processing).

על הספר

סגנון הלימוד של הספר מיועד להעניק לך יכולת מעשית כאיש פיתוח, ולהציג תפוקה גובהה בעבודה עם הרכיבים החדשניים. הלימוד הוא בסגנון **הבט, ראה, فعل!** דוגמאות הקוד הרבות שתמצוא בספר ממחישות את התפיסות שניתנו בשלב ביישומים באופן מיידי. התיחס אליהן כאל מתכנים לביצוע מטלות פיתוח מוגדרות. נסה להפעילו "כמו שהן", ולאחר כך התאמס אותן לעובדה עם הנתונים שלך ובהקשר של היישום שיצרת. הדוגמאות פשוטות למדי, קלות להבנה ולשימוש חוזר, עובדות שתדרבנה אותך לישמן.

קהל היעד

ספר זה מיועד למפתחי מסדי נתונים המתוכננים לפתח יישומים מותאמים אישית באמצעות Microsoft Access 2000. הספר דן ברכיבים החופכים את Microsoft Access לכלים המועדף על מפתחים, ובחדושים המיוחדים של Access 2000. הספר מציג דוגמאות קוד רבות המדגישות את טכניקות הפיתוח העיקריות, וכליים לפיתוח יישומים מהיר. פשtuות העיצוב נשמרת בקפנות, כדי לאפשר לך לאמצץ את קטיעי הקוד שבספר ליצירת יישומים מותאמים אישית, או לישום הטכניקות השונות במהלך העבודה. ~~בתיקיון המצורף בספר מכיל את כל דוגמאות הקוד בספר.~~

הספר מיועד לשפר מענה לצרכי מגוון קוראים. אנשי פיתוח וティקיס ימצאו בו מידע חיווני על הרכיבים המתקדמים הכלולים ב- Access 2000. קהל יעד נוסף הוא אנשי פיתוח שעובדים בסביבה שונה מזו של Microsoft, כגון dBase או Paradox. אלה מたちאים במושגי פיתוח, אך אינם יודעים בהכרח כיצד לישם את המושגים בסביבת Access. קבוצה נוספת היא של משתמשים מקצוענים השואפים ללמידה לפתוח פתרונות באופן עצמאי במינימום זמן.

תוכניות וקוד המקור בספר

כל קטיעי הקוד המופיעים בספר מסופקים בתקליטור המצורף. תוכל להשתמש בהם בשלהותם כדי לחסוך זמן, ולמנוע טויות הקלדה במהלך העבודה.

~~התקליטור המצורף~~

בתקליטור המצורף תמצא את תיكيית קבצי הקוד הרלוונטיים בספר זה - תייקה בשם **59237** הנמצאת בתיקיה **Books Programming Access**. צור תייקה בשם **C** בכוון **59237** ולאחר כך העתק אליה את תיikitות הפרקים שבתיקיה 59237 (קטיעי הקוד הרלוונטיים בפרק מסוים, מופיעים בתיקיה בשם שם הפרק).

להרחבה על תכונות התקליטור ואופן השימוש בו, פנה לקובץ **ONCD** שבתקליטור.

מוסכמות בהן נעשה שימוש בספר זה

הספר משתמש במספר מוסכמות ואלמנטים מיוחדים כדי להציג נושאים מסוימים, ולהקל על הקריאה והשימוש. מומלץ להתבונן בהםטרם תחל בקריאה.

הרשימה שלහן מפרטת את המוסכמות בהן עושה הספר שימוש:

↳ מילים ומשפטים המדגישים את כוונת המחבר, כמו גם מונחים חדשים, מופיעים **באותיות מודגשות**.

↳ קטעי קוד ופקודות לדוגמה מופיעים בגוף **Tahoma** וברקע אפור.

↳ חלקים שונים בקטעי הקוד מופיעים מדי פעם **בגוף נטוי** כדי לציין שאלה חלקים אשר עליך להוסיף או לשנות (למשל פקודות או שמות משתנים). שיטה זו מאפשרת לך לראות מה עליך להוסיף או לשנות על פי ההקשר.

↳ בשל מגבלות הרוחב של עמודי הספר, נאלצנו במקרים מסוימים לשבור שורות קוד ארוכות במיוחד למספר חלקים. במקרים אלה, מופיע בסוף השורה השבורה סימן המשך שורה (...). בעת העתקת הקוד למחשב תוכל להשمي את סימני המשך וכותוב את הקוד במלואו בשורה בודדת או לחילופין - להעתיקו כפי שהוא מופיע בספר. **Access** יידע לפרש את הקוד בכל אחת מהצורות שתבחר.

הערה:

כל התמונות המופיעות בספר לקוחות מהגירסה האנגלית של 2000 Office. בדיקות נכונות המידע בוצעו על הגירסה העברית, עם התיחסות לגירסה האנגלית בסוגרים.



הערה נוספת: כאשר תמצא התיחסות ל-8 Excel, הקונה היא ל-97-95.

היכרות עם VBA

לפני יותר מעשור שנים הציג ביל גייטס שפט מאקרו אוניברסלית לישומים שולחניים. שפט Visual Basic for Applications (VBA) היא הגשמת החלום וייתר מזה. VBA של Microsoft Access 2000 משותף לכל יישומי Office, ואפשר למצאו אותו גם במודרי תוכנה של יצרנים נוספים. VBA עומד גם בכל כללי התchapיר של שפת התכונות העצמאית Visual Basic. מאפשר לפתחים להשתמש בשפת תכונות יחידה בעשרות הקשיים, כשהכל שעלייהם לעשות הוא למדו מודול אובייקט חדש. VBA הוא ה"דבק" של יישום Access: הוא מחזик הכל יחד ומעניק ליישום צורה.

- Access 2000 ממשק חדש - וורץ (VBE), המופיע ביישומי Office. אך בוודאי תמצא עצמן מוסף קוד VBA לטפסי המוכרים, ולא לטפסי המשמש בשימוש יישומי Office אחרים. זיוג טכנולוגיות זה ייראה לך טבעי מאד.

פרק זה נציג את VBA כפי שהוא מופיע ב- Access 2000, ונסקור את עקרונותיו הבסיסיים בהקשר לפיתוח יישומים. כמו כן נציג את חידושי VBA העיקריים ונדגים טכניקות תמייה בטפסים באמצעות קוד. אומנם מדובר בסוגיית פיתוח מקובלת ב-Access, אך קיימים מספר שינויים באופן בו מ ממשים אותה באמצעות VBA.

הפרק ידון בשבועה היבטי VBA ב-Access, ויסתאים בתיאור קצר של פקודות מאקרו.

↳ **אובייקטים** (Objects), **אפשרויות** (Properties), **פעולות** (Methods) ו**אירועים** (Events)

↳ **שגרות** (Procedures) ו**מודולים** (Modules)

↳ ממשק **VBE**,

↳ **Jet**, סוגים נתוניים והצורות,

↳ לוגיקה מותנית ומבנה לולאה,

↳ פונקציות מוכללות,

↳ ניפוי ולכידת שגיאות.

אוסףים, אובייקטים, מאפיינים, שיטות וארועים

Access 2000 תומך ב-VBA, מה שמאפשר פיתוח מונחה-אובייקטיבים. בסעיפים הבאים נעסק בפיתוח מונחה-אובייקטיבים בהקשר של VBA ו- Access 2000. המידע מיועד למשתמשים מתקדמים העוברים לפיתוח יישומים מתוכנת, ולמשתמשים ברמה ביןונית המונחיינים לסקור את נושא התכונות המונחה-אובייקטיבים באמצעות VBA.

אוסףים ואובייקטים

Access 2000 מהויה סביבת פיתוח מונחה-אובייקטיבים. חלון מסך הנתונים שלו מאפשר למשתמש לגשת לטבלאות, שאילתות, טפסים, דוחות, מודולים ופוקודות מאקרו. VBA הופך את כל הרכיבים האלה ל恣ניים יחד עם מגוון רחב של תכונות מתוכנות, כגון **ערוכות רשותות** (recordsets) ואובייקטים של **TableDef**. כדי להפיק מ-VBA את מרבית התועלות ב-Access, עליך להבין את נושא האובייקטיבים ומושגים נוספים הקשורים לנושה זה.

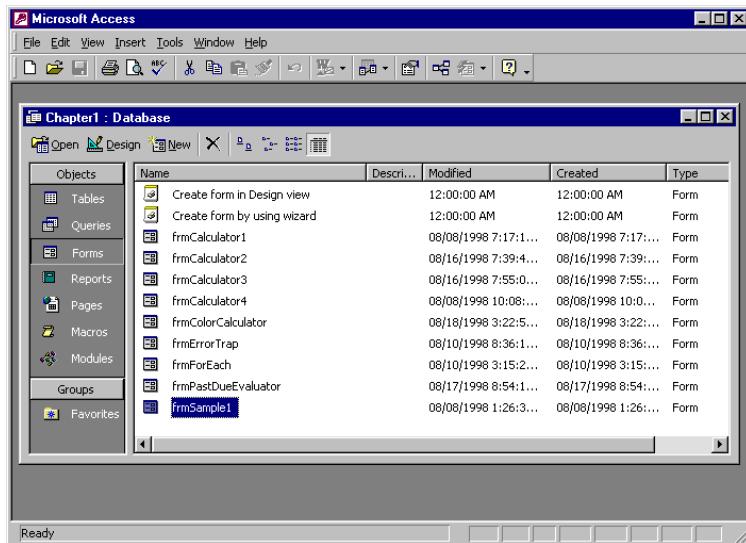
אובייקט (object) הוא עצם, כגון מכונית, טלפון או מכשיר וידאו. לכל האובייקטיבים יש מאפיינים. מכוניות למשל, מוגדרות באמצעות מאפייני הצבע, הדלותות, המנווע ומאפיינים נוספים. מאפיינים יכולים להגדיר מופעים של אובייקטיבים כלליים. התנהוגות מונחה-אובייקטיבים מאפשרת להגדיר מופעים ייחודיים של אובייקטיבים בהתאם למאפייניהם. לדוגמה, מכונית אדומה ומכונית שחורה הן שני מופעים ייחודיים של האובייקט "מכונית".

מאפייני האובייקט משתנים בהתאם למחלקה האובייקט אליה הם מתייחסים. למכונית יש קבוצת מאפיינים שונה מזו של מכשיר טלפון. שני האובייקטיבים מאפיינים באמצעות צבע, אך לטלפון יכול להיות גם מאפיין רמקול. למכוניות לעומת זאת, יש מנועים בנחאים שונים. יש אובייקטים שמהווים **מכלות** (containers) עבור אובייקטים אחרים. גם לאובייקטים המוכלים יש מאפיינים משלהם. מנועים מיוצרים במיגון נחחים ותצורות, בעוד לרמקולים יש בקריעות ומאפייני איות צלילים. מאפיינים יכולים להגדיר גם מופעים ייחודיים של מחלקות אובייקטיבים מוכלים. טלפון עם רמקול (speaker) קבוצת מאפיינים שונה מזו של מכשיר טלפון רגיל.

בנוסף למאפיינים, אובייקטיבים כוללים גם שיטות. שיטות של אובייקט הן הפעולות שהוא מסוגל לבצע: מכשיר הטלפון משתמש להתקשרות, מכונית נוסעת. אובייקטיבים רבים כוללים שיטות רבות. לדוגמה, מכשיר הטלפון מאפשר לבצע שיחות מקומיות ושיחות בין-עירוניות.

פתחים בסביבת Access אינם מטפלים באובייקטיבים פיסיים, אלא בתכונות מתוכנות, כגון טפסים, טבלאות ושאלות שיכולים לייצג אובייקטיבים ופעולות בהם מבצעים. החלון **מסך נתוניות** (Database) של Access 2000 מציג חלק ממחלקות האובייקטיבים של מסך הנתונים בסרגל הכלים המזמין את זה של Outlook (ראה

תרשים 1.1. לחיצה על לחץ **טפסים** (Forms) שברגל הכלים פותחת תצוגה של אובייקטי טופס ומציגת שתי אפשרויות ליצור טפסים חדשים. אובייקטי טופס מסווגים להכיל אובייקטים נוספים שנקראים **פקדים** (Controls). אובייקטים מוכלים מגדירים אובייקט בדיקת כפי שפקדים בטופס מגדירים את מראהו והתנהגותו.



תרשים 1.1: החלון **מסך נתונים** בו אוסף של אובייקטי טופס ושתי אפשרויות ליצור טפסים חדשים (שים לב, התרשימים לקוח מהגירסה האנגלית)

תרשים 1.1 מציג אובייקטי טופס אחדים. אובייקטים אלה מהווים **אוסף** (collection). בפתיחת טיפוסי, יישומי Access כוללים אוסף טפסים, טבלאות, שאלות וowiיקיטים נוספים. החלון **מסך נתונים** עצמו ממילוי אובייקטים לפי מחלקות. לחיצה על סרגל הכלים בחלק הימני (או השמאלי בגרסה האנגלית) מציגה בחלון את כל האובייקטים שבאוסף.

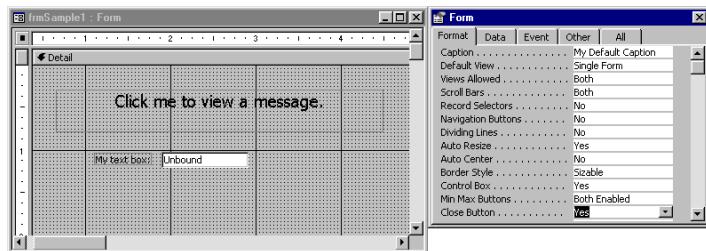
אוספים דומים מאוד לאובייקטים. לכל אוסף Access יש מאפיין Count (מונה) שמנדר את מספר המופיעים באוסף. Item (פריט) הוא מאפיין נוסף של אוסף. תוכל לנצל את המאפיין **Item** לקריאה בלבד, כדי להחזיר טופס יחיד מתוך האוסף AllForms (כל הטפסים). האוסף הם אובייקטים ייחדים, ולכן אין להם מאפיין Count. אובייקטי אוסף משמשים למטרות שונות. מכוניות משפחתיות בעלת גג מתאפשר יכולת למלא פונקציות שונות מלאו של מכוניות משפחתיות וגיליה, אך שתיהן יכולות להשתייך לאוסף המכוניות של משפחה מסוימת.

מאפיינים ושיטות

המאפיינים והשיטות מאפיינים את מראה והתנהגות האובייקטים. תחביר ההפנייה למאפיינים ושיטות הוא object.property ו-object.methods, בהתאם. המונח object (אובייקט) יכול להתייחס לאובייקט יחיד או לאוסף אובייקטים. לדוגמה,

מצין את מאפיין צבע הרקע של תיבת טקסט בטופס, ואילו AllForms.Items(0) מתייחס לטופס הראשון באוסף הטפסים. אם שם הטופס הראשון AllForms.Item("frmSample1"), ניתן להפנות אליו בתור ("frmSample1

אפשר להציג את מאפייני אובייקט מסד הנתונים על ידי בחרתו בתצוגת **עיצוב** (Design) ולחיצה על לחץ **מאפיינים** (Properties) שבסרגל הכלים. תרשימים 1.2 מציג טופס בתצוגת **עיצוב** עם גליון המאפיינים שלו. גליון המאפיינים מציג ערך מותאם אישית, My Default Caption (Caption). המאפיין **לחוץ סגירה** (Close Button) נבחר. תוכל לחוץ על שורת המאפיין **לחוץ סגירה** ולחזור לא (No). שינוי הערך של המאפיין יציג באפור את הלחוץ סגירה כאשר הטופס יופיע בתצוגת **טופס** (Form). שים לב שגליון המאפיינים מכיל כמה דפים (כרטיסיות). תרשימים 1.2 מציג את הכרטיסיה **מבנה** (Format) שבגליון המאפיינים. הכרטיסיות מסדרות את המאפיינים בקבוצות, מה שמאפשר לשיפה מהירה.



תרשים 1.2: טופס פשוט בתצוגת **עיצוב** יחד עם גליון המאפיינים שלו

הערה:



ב-2000 גליון המאפיינים זמין גם בתצוגת **טופס**, ולא רק בתצוגת **עיצוב**. משמעות הדבר היא שניתן לשנות ולשפר את מראה הטופס בצורה קלה ופושטה תוך כדי הצגת **טופס**.

האובייקט DoCmd הוא מקור שיטות עשיר למפתחי יישומי Access בכל הרמות, אך מפתחים מתחילה ימכו בו עוזר רב בעובודה בסיסית עם שיטות. אובייקט זה כולל שיטות רבות, RunCommand, FindRecord, GoToControl, OpenForm, Close, RunCommand ו-FindRecord. שיטות רבות של אובייקט זה דורשות ארגומנטים המציינים את אופן הביצוע שלו. שיטות אחרות כוללות ארגומנטים נדרשים או אופציונליים. אם אין מציינים ערכים של ארגומנט אופציונלי, השיטה מנצלת את הגדרות ברירת המחדל. השיטה RunCommand מחייבת לפחות את הגדרות ברירת המחדל. השיטה Close מחייבת לפחות את הפקודות הזמינות בתפריטים ובסרגל הכלים של Access.

ב-Access אפשר לסגור טופס באמצעות השיטה Close של האובייקט DoCmd. לשיטה זו שני ארגומנטים נדרשים ואחד אופציונלי. הארגומנט הנדרש הראשון מצין את סוג האובייקט המיועד לסגירה. לסגור טופס המשתמש ב-*acForm* (*acForm*) הוא קבוע מוכפל של שערכו מורה לשיטה Close כי ברצונך לסגור טופס. לקבלת מידע

נוסף על קבועים מוכללים ב-Access, עיין בסעיף "סורק האובייקטים"). הארגומנט השני הוא שם הטופס. ערך זה מופיע במאפיין **שם** (Name) שבגילוון המאפיינים של הטופס. הציב את השם בין מרכאות. הארגומנט האופציונלי מורה ל-Access אם לשומר שינויים כלשהם בטופס. בירית המוחלט היא לבקש את אישור המשתמש לביצוע הפעולה. השתמש בשיטות acSaveYes או acSaveNo לסגירת הטופס תוך שמירת השינויים או ללא שמירתם. להלן תחביר המשתמש בשיטה Close לסגירת טופס:

```
DoCmd.Close acForm, "formname", acSaveNo
```

שיטות רבות של DoCmd חלות יישור על אובייקטים ייחדים. לדוגמה, השיטה GoToControl מעבירה את המיקוד לפקד מסוים בטופס. ניתן להשיג זאת גם באמצעות השיטה SetFocus שבוחרת את הפקד. נוח להפעיל את שתי השיטות כאשר על היישום להעביר את המיקוד לשם קליטת נתונים חדשים, או לתיקון מידע שגוי.

אירועים

אירועים הם מרכיבים חשובים מאוד בתכנות VBA. ניתן לנצל אותם כדי ליצור יישומים דינמיים ו互動ראקטיביים. אירועים של אובייקטים ואוספים משמשים מעין נקודות הפעלה לקוד מותאם אישית שייצר איש הפיתוח. במהלך העבודה עם טפסים תוכל לנצל אירועים לביצוע מטלות, כגון אימוטות נתונים, הפיכת פקידים לזמןניים או בלתי זמינים, העברת המיקוד לפקד אחר, ופתחה וסגירה של טופס.

עליך להבין את תזמון האירועים וגם את סדר הפעולות. פתיחת טופס מביאה לשורת אירועים : Current-Resize, Load, Open. האירוע Open מתתרחש בעת שתופס מתחילה להיפתח, אך בטרם הוצגו רשותות כלשהן. האירוע Load מתתרחש לאחר האירוע Open במאכזות השיטות MoveSize, Minimize, Maximize או של האובייקט DoCmd. מפעיל את האירוע Current הוא האחרון שמתתרחש בדרך כלל בעת פתיחת טופס. אירוע זה מסמן את הרגע בו רשומה מסויימת הופכת לרשותה נוכחית או לרשותה זמינה. האירוע מופעל גם כשהמשתמש מנוט לרשומה חדשה, מפעיל שאלתה, או מרענן טופס.

ニיגשים לאירועי טופס על ידי בחירת הטופס או הפקד בתצוגה **עיצוב** ולאחר מכן בוחרים בכרטיסיה **אירוע** (Event) שבגילוון המאפיינים. לחיצה על הלחצן **בנייה** (Build) שבשורת כל אחד ממאפייני אירוע, גורמת לפתיחת תיבת דו-שיכון המאפשרת לפתח את קוד שמאחורי הטופס. בחרה ב**בבונה קוד** (Code Builder), אשר גורמת לפתיחת שגרת אירוע ב-VBE. שגרת אירוע מקבלת את השם Objectname_Eventname. הקוד, אשר בוחרים טופס ולוחצים על לחץ **בנייה** להפעלת האירוע Close (בעת סגירה), שגרת אירוע תקבל את השם Form_Close. אם יוצרים שגרת אירוע עבור האירוע OnClick (בעת לחיצה) של תווית הנקראת lbTitle_Click, היא תקבל אוטומטית את השם lbTitle_Click.

לפניך שלוש שגרות אירוע המתייחסות לטופס המוצג בתרשים 1.2 : Form_Open Form_Load IbTitle_Click. בעת פתיחת הטופס בפעם הראשונה בתצוגת טופס, מופיעה תיבת הודעה ובה הודעה "The form opened" (הטופס נפתח). אישור ההודעה גורם להציג הודעה נוספת, "The form loaded" (הטופס טוען). בתום טיעינת הטופס, לחיצה על התווית גורמת לפתיחה תיבת הודעה שלישית, "Hello from the label".

```

Private Sub Form_Open(Cancel As Integer)
    MsgBox "The form opened.", vbInformation, _
        "Programming Microsoft Access 2000"
End Sub
Private Sub Form_Load()
' This is a simple statement.
    MsgBox "The form loaded.", vbInformation, _
        "Programming Microsoft Access 2000"
' This sets a property.
    Me.Caption = "New Caption"
' Here are two methods for giving a control focus.
' DoCmd.GoToControl "txtMyTextBox"
    Me.txtMyTextBox.SetFocus
' Now that the method worked, VBA sets a property.
    Me.txtMyTextBox.Text = "Hi, there!"
End Sub
Private Sub IbTitle_Click()
    MsgBox "Hello from the label.", vbInformation, _
        "Programming Microsoft Access 2000"
End Sub

```

שגרות האירוע גורמות להציג תיבות הודעה. לחיצה על תווית מפעילה את שגרת האירוע IbTitle_Click. בשיגרה זו משפט יחיד שמציג את תיבת הודעה (הקו התחתון בסוף השורה הראשונה של המשפט, מצין שהפקודה ממשיכת בשורה הבאה). שגרת האירוע הראשונה של המשפט, כוללת אף היא משפט אחד. שגרת האירוע Form_Load כוללת משפטים אחדים נוספים זהה שמציג את תיבת הודעה. שגרת אירוע זו מגדרה בצורה דינמית את כתוב הטופס, דבר זה שימושי במיוחד לטופס שמלא שני תפקדים או יותר באותו יישום. השגירה גם מעבירה את המיקוד לטיבת הטקסט שנקראת txtMyTextBox ולאחר כך מציבה את המחרוזת "Hi, there!" במאפיין Text של הפקד. שגרת אירוע זו מדגימה שתי טכניקות נפרדות לקביעת מיקוד. הראשונה מבוססת על השיטה SetFocus והשנייה על השיטה GoToControl. הגרש בתחילת שורות מסוימות מצין שהשורות הן העורות המיועדות לאנשים הקוראים את הקוד ו-Access אינו מתייחס אליהם. שם לב כי אחת הטכניקות מופיעה בשורת הערא.

שגרות ומודולים

שגרות הן מכולות (containers) של קוד VBA. קיימים שלושה סוגי מכולות: שגרות משנה (subprocedures), שגרות פונקציה (function procedures) ושגרות מאפיין (property procedures). למרות החפירה במספר תפקודים, לכל שגירה מטרה מוגדרת וייחודית לה בלבד. Access כולל שני סוגי מכולות בסיסיים: מודולים סטנדרטיים ומודולי מחלקות. מודולי מחלקות יכולים להיות מחלקות מותאמות אישית עבור טפסים ודוחות. תוכל לנצל אותם כדי להגדיר עצמאך מחלקות, ולפשט את השימוש החזר בקוד של מטלות שגרתיות כגון הוספת עובד חדש, ביצוע הפקדה לחשבון או משיכה ממנו.

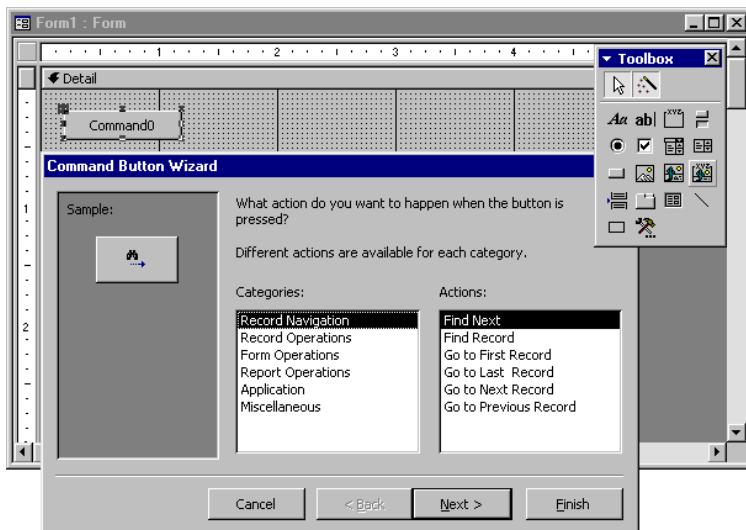
שגרות משנה

שגרות משנה יכולות לבצע פעולות, לחשב ערכאים, לעדכן ולשנות הגדרות מאפיין מוכללות (לעתים שגרת משנה נקראת בקיצור "שיגרה"). כפי שלמדנו, Access 2000 מפעיל אוטומטית שגרות כאשר מתרחשים אירועים, אך תוכל להרחיב את השימוש בשגרות אלו. שגרות עלוליםstin מחשירות ערכאים וגם אין מוגדרות מאפיינים מותאמים אישית של טופס, דוח או מודול מחלקה.

שיגרה מכילה סדרת משפטי VBA בין המשפטים **Sub** ו- **End Sub** התוחמים אותה. המשפט **Sub** חייב להציג על שם שיגרה. שגרות אירוע מקבלות שמות מסווגנים מאוד (כגון **object_event**), אך שמות של שגרות נקבעים לפי כללי מתן שמות לשנתנים. עליהם להתחילה, אורכם אינם יכול לעלות על 255 תוים, אין לכלול בתוכם סימני פיסוק או רווחים וגם לא מילוט מפתח, שמות פונקציות או שמות אופרטורים של VBA. שגרות יכולות לקבל ארגומנטים הבאים לאחר שם השיגרה. אם השיגרה כוללת יותר מארגומנט אחד, יש להפריד את הארגומנטים בפסיקים.

אחת הדרכים להציג טוב יותר את גושה השגרות היא השימוש באשף להצני פקידות (Command Button wizard), אשר כותב קוד VBA למיניהם 30 פונקציות. כל שימוש לעשות הוא לבחור הגדרות חדשות בתיבות הדיו-שיכון. האשף לחבר שגרות פשוטות למדוי, ולכנן הן מהוות מקור יעיל ללימוד הנושא. גם מפתחים ברמה בינונית ופתחים متקדמים יצאו נשכרים משימוש באשף, מכיוון שהקוד שהוא יוצר יכול לשמש כשלד בסיסי שעליו אפשר להוסיף קוד מפורט יותר. מפתחים מתחילה יכולים לנצל את האשף כדי להפוך תהליכיים לאוטומטיים, כמו ניוט בין רשומות, תחזוקת מסד נתונים, טיפול כללי בטופס ובדוח, הפעלת יישומים נוספים ומטילות שונות נוספים, כגון הפעלת שאילתת או חיבור מספר טלפון.

את האשף מפעילים מארגו הכלים שבतצוגת **עיצוב טופס**. בחר בלחץן אשפי בקרה (Control Wizards) ואחר כך בחר לחוץ וצייר את הפיקד על הטופס. תיפתח תיבת הדיו-שיכון המוצגת בתరשימים 1.3. בכל קטגוריה מוצעות מספר פעולות. לאחר שתסייעים את המעבר על תיבות הדיו-שיכון, תוכל להציג את הקוד ב-VBE. לחץ על לחוץ קוד (Code) שב프로그램 הכלים **עיצוב טופס** (Form Design) כדי לעבור אליו.

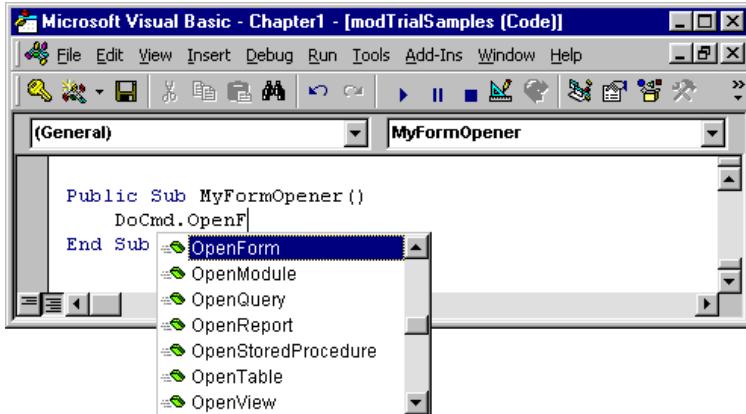


תרשים 1.3: אשיף להציג פקודות מאפשר ליצור שגרת אירוע מלאה תוך שימוש בתיבות

דו-שיח אחדות

פתחים רבים מעדיפים כתוב שגרות מהתחלת (ambilי להתבסס על שיגרה קיימת). תוכל לפתח את חלון הקוד וליצור שגרת אירוע כפי שתואר לעיל, או ליצור שיגרה רגילה. קיימות שתי דרכים להתחילה שיגרה כזו, זאת בהתאם למקום בו תציב אותה. אם השיגרה מיועדת לטופס או לדוח, לחץ על הלחצן **קוד** (Code) שברגל הכללים **שימוש**. אם הקוד יופיע במודול סטנדרטי שאינו מיועד לטופס או לדוח מסוימים,בחר **כלים** (Tools), **מאקרו** (Macro), **Visual Basic Editor**, או לחילופין הקש **Alt+F11**. חלון VBE יפתח. בחר **Insert** (הוסףה), **Procedure** (שיגרה), הקלד שם בתיבת הדו-שיח **Add Procedure** ואשר את בחירת ברירת המחדל של להציג האפשרות **Sub**. כך תיווצר מעטפת לשיגרה, ובה המשפטים **Sub** ו- **End Sub**.icut תוכל להוסיף לשיגרה קוד.

השתמש בידע שרכשת על מודל האובייקט של Access וכותב מספר פעולות. זכר כי האובייקט **DoCmd** כולל שיטות רבות. הקלד **DoCmd** ואחר כן הקלד נקודה. תופיע תיבת רשימה נפתחת ובה כל הערכים החוקיים שיכולים להופיע לאחר **DoCmd**. במהלך הקלדת הערך, תוצג רשימת ערכים אפשריים התואמים להקלדה (תרשים 1.4). אם איןך בטוח בערך שעלייך להקליד, גלול את רשימת הערכים ואתר את השיטה המבוקשת. טכניקה זו מתאימה לכל האובייקטים ולא רק ל- **DoCmd**. קוראת לתכונה זו **IntelliSense**, מכיוון שהיא מזזה בצורה נבונה את התשובות האפשריות. התכונה **IntelliSense** מבצעת למעשה שתי פעולות: מציגה את השיטות והמאפיינים החוקיים בכל שלב של הרכבת משפט VBA, ומספקת נתוני תחביר על תכונות השדות הדרושים למשפט VBA שבבנייה. פעולות אלו מוצמצמות באופן נicer את שגייאות התחביר בקוד, ומסייעות לך להתחילה במהירות.



תרשים 1.4: התוכנה מסייעת להשלים מושפט VBA באובייקט DoCmd בהקלדת תחילת המשפט. IntelliSense מציגה רשימת ערכאים חוקיים התואמים לטעסט שהוקלד

להלן שיגרה פשוטה הכוללת שלוש שורות. השורה הראשונה מקצת מקום בזיכרונו לתוכה מחושבת. השנייה מחברת שני קבועים. השורה השלישית מדפסה את התוצאה בחלון **Immediate** (מיידי). חלון זה משמש כפנקס לאחסון תוצאות ביניים במהלך בדיקת הקוד. כדי להציגו בחר **View** (תצוגה), **Immediate Window**. ניתן להפעיל את השיגרה מותוך ערך ערך VB על ידי לחיצה באותו מקום כלשהו בשיגרה ולאחר כך לחיצה על הלחצן **Run Sub/UserForm** שבסדרל הכללים הרגיל.

```
Sub MyFirstCalculator()
    Dim Result
    Result = 1 + 2
    Debug.Print Result
End Sub
```

במצבים אופייניים יותר, תוכל לקרוא לשיגרה באמצעות משתי דרכים. תוכל לכלול את שמה בשורה עצמאית; אם השיגרה כוללת ארגומנטים כלשהם, תוכל לכלול אותם לאחר שמה, כשם מופרדים בפסיקים. לחילופין, תוכל להקדים את מילת המפתח **Call** לשם השיגרה. Call היא מילת מפתח של VBA המשמשת לקריאה לשיגרה. אם תשימוש במילה Call, תחום בסוגרים את הארגומנטים המופיעים לאחר שם הפונקציה.

להלן גירסה גמישה מעט יותר של פונקציית המחשבון הבסיסית. בדוגמה מופיעות שתי שגרות. השיגרה שנקראת MySecondCalculator מחברת שני מספרים, ומדפסה את התוצאה בחלון Immediate. השיגירה קולטת את המספרים באמצעות שני ארגומנטים שנשלחים אליה. השיגרה השנייה קוראת לשיגרה שמבצעת את פעולה החיבור. תוכל לשנות את המספרים שהשיגרה מסכמת, על ידי שינוי ערכי הארגומנטים בשיגרה הראשונה. בגירסה מותאמת מעט יותר של יישום זה, אפשר לקשור את ערכי הארגומנטים למשתנים או לשדות טופס.

```

Sub CallSecondCalculator()
    MySecondCalculator 1, 3
End Sub

Sub MySecondCalculator(First, Second)
Dim Result
    Result = First + Second
    Debug.Print Result
End Sub

```

שגרות פונקציה

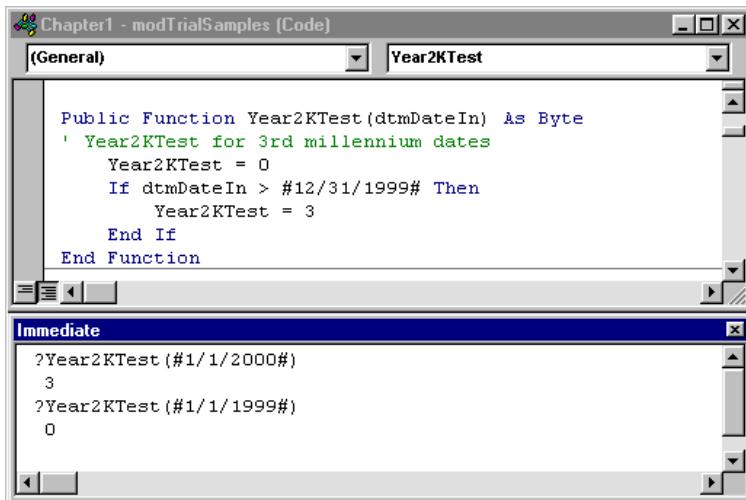
שגרות פונקציה, הנקראות בדרך כלל **פונקציות** (Functions), שונות משלגות רגילות במספר מבנים. ראשית הן מחזירות ערך, ולכן ניתן לנצלן בביטויים כאילו היו משתנים. שנית, הן אינן משמשות לשגרות אירוע. שגרות ופונקציות יכולות לבצע מטלות. למעט הנקודות שצינו, ניתן לנצל את שני הסוגים לביצוע מטלות זהות רבות.

פונקציה היא אוסף משפטי VBA, תחומים בין משפטי **Function** ו-**End Function**. פונקציה יכולה לקבל ארגומנטים בדיק כmo שירה. היא יכולה להכיל ביטוי אחד או יותר, שלפחות אחד מהם יכול לקבוע את הערך שיוחזר בשם הפונקציה. אפשר לסייע לפונקציה לאחר קביעת ערכה, באמצעות המשפט **Exit Function**. כל פונקציה יכולה להכיל כמה משפטי **Exit Function**.

פונקציה מסוגלת להחזיר ערך, אך אינה חייבת לעשות כך. פונקציה יכולה להיות אוסף משפטי המפעילים שיטות ומגדירים מאפיינים, מבלי להחזיר ערך. זו אחת התכונות המשותפות לפונקציות ולשגרות.

אתה מתחלפ פונקציה ושירה באותה דרך, אך בתיבת הדיו-שים **Add Procedure**, יש לבחור באפשרות **Function** במקום באפשרות **Sub**. ניתן להפעיל פונקציה על ידי לחיצה על הלחצן Run Sub/UserForm הכללים Standard של VBE. תוכל להפעיל פונקציה גם מתוך החלון **Immediate**: הקלד סימן שאלה ואחריו שם פונקציה. אם הפונקציה כוללת ארגומנטים, יש להציבם בסוגרים ולהפרידם בפסיקים. מתוך חלון זה תוכל להפעיל פונקציות שכבתבת עצמן וכן פונקציות מוכללות של Access.

תרשים 1.5 מציג את VBE עם פונקציה פשוטה הקובעת אם התאריך המופיע כargonmant, חל במילניום השלישי. כדי לעמוד בכללים המקובלים, נתייחס למילניום זה כאילו הוא מתחליל בשנת 2000 במקום בשנת 2001. הפונקציה Year2KTest קולעת תאריך ומחייבת 3 אם התאריך חל במילניום השלישי, או 0 בכל מקרה אחר. החלון Immediate המוצג מתחת לחלון Code, מציג את תוצאת הפעלת הפונקציה על שני תאריכים שונים. החלון Immediate שבתרשים 1.5 מאשר את התוצאה על ידי החזרת הערך 3 לארגומנט שהוא היום הראשוני בשנת 2000, ואת הערך 0 עבור היום הראשון בשנת 1999. שים לב כי עליך לתחום את התאריך בין תווים #.



תרשים 1.5: פונקציה פשוטה שמופעלת מתוך החלון Immediate

הדוגמה הבאה מציגה גישה מותאמת יותר לזיהוי המילניום. הפונקציה פותחת בהצהרה על טווח התאריכים המדוייק. לאחר מכן היא בודקת את הארגומנט נגד שני תאריכי ציון. אם התאריך אינו חל בימי נסחף הראשון או השני, הפונקציה מניחה שהוא חל בימי נסחף השלישי. משפט תיבת ההזדעה התחלה מצין את טווח התאריכים שבו הפונקציה מחזירה תוצאות מדוייקות.

```

Public Function PopularMillenium(dtmDateIn) As Byte
    MsgBox "This works for dates after 12/31/0099 and before 1/1/3000.", _
        vbInformation, "Programming Microsoft Access 2000"

    If dtmDateIn <= #12/31/999# Then
        PopularMillenium = 1

    ElseIf dtmDateIn <= #12/31/1999# Then
        PopularMillenium = 2

    Else
        PopularMillenium = 3
    End If

End Function

```

טווח התאריכים החוקי של Access 2000 הוא 1/1/100 עד 31/12/2999. טווח זה מספק את דרישות מרבית היישומים השולחניים. אם תידרש לטפל בתאריכים החורגים מטווח זה, שקול תכנות תאריכים בנפרד מערכות התאריך הסדרתי של Access.



Access 2000 תומך לשנת 2000. כמו גרסאותיו הקודומות, הוא מחזק את ערך השנה באربע ספרות. היישום גם מטפל בצורה תקינה בשנים מעוברות: שנה שמתחלקת ב-4 היא שנה מעוברת, אלא אם היא מתחלקת ב-100. לעומת זאת, שנה שמתחלקת ב-400 היא כן שנה מעוברת. שנת 2000 מתחלקת ב-400, ולכן היא שנה מעוברת. ככל זה הוא קרטיסי כשמחשבים את ההפרש בין שני תאריכים. תבנית התאריך הכללי ותבנית התאריך הקצר של Access מנצלות את אפשרות בניית התאריך הקצר של מערכת הפעלה, כדי לקבוע כיצד להציג את התאריך בצורה תקינה. אם תנסה את ההגדרות האזרחיות שבלוח היבקרה להציג השנה באמצעות התבנית. אربע ספורות, כל תבניות התאריך הכללי תציגנה את השנה כך.

חוקרים שתוארו אינם קופים עליך לפתח יישומים באופן שמנטיח אותם מפני בעיות שנות 2000. למרות ש- Access 2000 תומך לשנת 2000, כל יישום שלו מועד לבעה. בקר באתר Microsoft (www.microsoft.com/technet/topics/year2k/default.htm) קיבלת סקירה על הנושא, ומארמים ייחודיים על מוצרים והתקאמם לשנת 2000. אתר FMS (www.fmsinc.com/tpapers/index.html#Year 2000 Papers) מציג מידע שונה (באנגלית) מזו של Total Access Inspector שאמור לזהות בעיות בסוגיה זו. FMS משוקת מוצר שנקרא Access 2000 Biennial Maintenance Pack. תאיםות שנות 2000 ביישומים מבוססי Access.

תוכל להשתמש בשגרות ובפונקציות כדי לפתח פתרון לבעה. תרשימים 1.6 מציג טופס שմבוסס על שגורות ופונקציות. הטופס מאפשר לקליד ערכים בתיבות הטקסט שתווויותיהן 1 Number 1 ו- 2 Number 2. לחיצה על לחצני /, *, -, או +, גורמת לחישוב התוצאה והציגתה בתיבת הטקסט Result.

תרשימים 1.6: טופס זה משמש כמחשבון פשוט. פונקציות VBA מאפשרות שימוש בלחצני הטופס ומציגות את התוצאה בתיבת הטקסט Result בהתאם לערכים שבשתי תיבות הטקסט האחרות.

קוד VBA של הטופס שבתרשימים 1.6 מאנצל רק ארבעה זוגות שגורות, כפי שמתואר בהמשך. ארבע שגורות מטלחות באירועי הלחיצה על לחצני האופרטורים לחישוב. הטיפול באירועים ממוצה בקריאה לפונקציה ששולפת ערכים מתוך שתי תיבות

טקסט, מבצעת פעולה חישוב באמצעות האופרטור שמצוין על הלחץ שנלחץ, ומחזירה את הערך שחוושב לשגרת האירוע. בתגובה, מזכיבה שגרת האירוע את הערך המוחזר מהפונקציה בתיבת הטקסט השלישי בטופס. שים לב לשימוש האופציוני בקידומת Me לפני שמות תיבות הטקסט. משום שהקוד הינו קוד הטופס שכולל את תיבת הטקסט, אין חובה לציין את שם הטופס. שגורות האירוע מנצלות את הקידומת Me במקום שם המחלקה הארוך והרשמי של הטופס, `.Form_frmCalculator1`.

```

Option Compare Database
Option Explicit
Dim dblResult As Double
Private Sub cmdAddition_Click()
    Me.txtResult = MyAdder
End Sub
Private Function MyAdder()
    dblResult = CDbl(txtNumber1) + CDbl(txtNumber2)
    MyAdder = dblResult
End Function

Private Sub cmdSubtraction_Click()
    Me.txtResult = MySubtractor
End Sub
Private Function MySubtractor()
    dblResult = CDbl(txtNumber1) - CDbl(txtNumber2)
    MySubtractor = dblResult
End Function

Private Sub cmdMultiplication_Click()
    Me.txtResult = MyMultiplier
End Sub
Private Function MyMultiplier()
    dblResult = CDbl(txtNumber1) * CDbl(txtNumber2)
    MyMultiplier = dblResult
End Function

Private Sub cmdDivision_Click()
    Me.txtResult = MyDivider
End Sub
Private Function MyDivider()
    dblResult = CDbl(txtNumber1) / CDbl(txtNumber2)
    MyDivider = dblResult
End Function

```

הפונקציות והשגורות נמצאות במודול הקוד של הטופס. המשפט Dim שבראש המודול מצהיר על משתנה שכל שגורות המודול יכולות לגשת אליו. המשתנים יכולים להחז על לחץ אופרטור אחד בלבד בזמן נתון, ולכן ניתן לשתף את `dblResult`. המשפט

כופה הצהרה על משתנים בטרם שימוש בהם, כדי להימנע משגיאות דפוס המהוות מקור שכיח לשגיאות בתוכנית. **Option Compare Database** הוא מפרט ברמת-מודול המקצה משתני מחזות ממויינים בסדר שנקבע בלוח הבדיקה.

זוג השגרות הבא מדגים את רוב הנושאים הקשורים לשגרות, פונקציות ושיטות בהם עוסנו בתחילת הפרק. הטופס frmCalculator2 כולל שני פקדים בלבד: תיבת טקסט שנקראת txtInput ולחצן פקודה שנקרא cmdSquarer. כשהמשתמש לחץ על הלחצן, הטופס מחשב את השורש הריבועי של הערך שבתיבת הטקסט. השיגרה מציגה את התוצאה בתיבת הودעה.

```
Option Compare Database
Option Explicit
Dim dblResult As Double
Private Sub cmdSquarer_Click()
    MySquarer Form_frmCalculator2.txtInput
End Sub
Public Sub MySquarer(MyOtherNumber As Double)
    dblResult = MyOtherNumber * MyOtherNumber
    MsgBox dblResult
' Optional statements illustrating the use of methods
' DoCmd.GoToControl "txtInput"
' txtInput.SetFocus
' DoCmd.Close acForm, "frmCalculator2", acSaveNo
End Sub
```

שגרת האירוע cmdSquarer_Click מפעילה את השיגרה MySquarer ומעבירה לה כารוגמנט את תוכן תיבת הטקסט txtInput מחשבת את התוצאה ומציג אותה בתיבת הודעה.

שלוש שורות הערה האחראות לפעולות נוספות לbijoux. השורה הקוראת לשיטה GoToControl מותאמת כיצד להעביר את המיקוד מהלחצן אל תיבת הטקסט. השורה הבאה (SetFocus) מציעה דרך חלופית להשיג תוצאה זהה. השיטה Close מתארת סגירת טופס. שים לב שורה זו מכינית שם נוסף frmCalculator2, באמצעותו היא מפנה לטופס. יש חשיבות לשימוש קבוע acSaveNo מכיוון שהוא מאפשר לסגור טופס ללא בקשה אישור שמירה מהמשתמש.

שגרות מאפיין

שגרות מאפיין משמשות להגדרת מאפיינים מותאמים אישית לטפסים, דוחות ומודולי מחולקה. בסעיף הבא נעסק במודולי מחולקה, בפרק 7 נציג דוגמאות לשגרות מאפיין.

יש שלושה סוגי משפטים מאפיין: **Property Set**, **Property Get** ו- **Property Let**.
טוכל לנצל משפטי אלה להוספה מאפיינים מיוחדים לטופס. המשפט Get בדוק אם מגדרים ובן זוגו End Property יכולים להחזיר ערך, בדוק כפי שעשו פונקציה.

מאפיין בעזרת Method Get בלבד, המאפיין יהיה לקריאה בלבד. מאפיין זה נוח לשימוש אם אתה יכול להציג ערך כלשהו, אך לא לשנותו. למשל ציוני מבחנים.

קיימים מאפיינים שלא ניתן לקרוא, אך יש חשיבות ליכולת לשנותם. לדוגמה, מנהלי אבטחת מסדי נתונים אינם יכולים לקראו סיסטמות משתמשים; הם רק צריכים להיות מסוגלים לדרוס אותן כשהמשתמשים שוכחים את סיסמת הכניסה שלהם. המשפטים `Property Let` ו- `Property Get` יתאפשרו להגדרת סיסמה.

המשפט `Property Set` דומה למשפט `Property Let` בכך שהוא משמש להגדרת מאפיין. המשפט `Property Let` מגדר מאפיין מסווג נתונים, כגון מחרוזות (string) או מספר שלם (integer). המשפט `Property Set` מגדר מאפיין מסווג להפניה לאובייקט. משתמשים במשפט `Property Set` לצורך הפניות לאובייקטים, כגון הפניה לטופס או לדוח.

מאפיינים רבים הם לקריאה ולכתחיה גם יחד, וכך לעיתים קרובות משתמשים במשפט `Property Get` יחד עם המשפט `Property Let`. במקרה זה, על זוג משפטי להיות בעלי שם זהה, כדי שיוכלו להפנות אותו מאפיין.

מודולים

מודול הוא מכולה של שגרות והצהרות, כגון `Explicit` ו-`Dim`. קיימים שני סוגי מודול בסיסיים. הסוג הראשון הוא מודולים סטנדרטיים המוצגים ברשימה לאחר לחיצה על הלחן **מודולים** (Modules) שבחלון **בסיס נתונים** (Database). השגרות במודול סטנדרטי אינם תלויים באובייקטים המקוריים בקובץ מסד נתונים של Access. שימושה הדבר, שאין הפניות אל Me או שמות פקדים ללא קידומות מתאימות של הפקדים. לעומת זאת, יישומים יכולים לפנות לשגרות במודולים סטנדרטיים מתוך כל אובייקט אחר.

הסוג השני הוא **מודול מחלקה** (Class Module) ובו שלושה טיפוסים בסיסיים: **מודולי מחלקה טופס**, **מודולי מחלקה דוח** ו**מודולי מחלקה מותאמים אישית**. עקרונית, מודולים יכולים לגשת לשגרות שנמצאות במודולים אחרים. ניתן להוציא שירה מחוץ לטווח ההכרה (scope) הכללי על ידי השימוש במילת המפתח **Private** (פרט) בעת הגדרת השירה (תרשים 1.7 מציג דוגמה לתחריב). תוכל גם להציג מפורשות על שגורות כבעלות טווח הכרה גלובלי על ידי שימוש במילת המפתח **Public** (ציבורי).

מודול טופס הוא מודול של טופס כלשהו שככל הצהרה או שירה אחת לפחות. יצירת שגרת אירוע של טופס או פקד בטופס, גורמת לייצרת מודול מחלקה טופס. מודולי מחלקה דוח פועלים בצורה דומה למודולי מחלקה טופס, אך אירועי דוח שונים מairyuei טופס, ובדרך כלל לא יופיע צירוף פקדים זהה בטופס ובdoch. תוכל לייצר מודולי מחלקה מותאמים אישית המכילים פונקציות שיטה ופונקציות שירה של אובייקט כלשהו, כגון עובד או חשבון בנק. תוכל ליחס את השיטות והמאפיינים למודולי מחלקה מותאמים אישית, כפי שניתן לעשות עם מחלקות מוכפלות של Access.

השימוש במודולי מחלקה מותאמים אישית דומה לשימוש בשבלונה ומטרתו ליצור מופעים (instances) חדשים של מחלקה. Access מאפשר לעשות זאת בשתי דרכים. הראשונה, באמצעות משפט Dim אחד שמנדר את המחלקה וגם יוצר מופיע חדש של המחלקה שהוגדרה. להלן תחביר המשפט:

```
Dim objInstance As New objClass
```

הגישה השנייה מבוססת על שני משפטיים. הראשון מצהיר על מופיע האובייקט. השני מגדר הפנייה אל האובייקט. להלן תחביר המשפטים:

```
Dim objInstance as objClass  
Set objInstance = New objClass
```

השם objClass פונה למודול מחלקה הכלול שגורות מאפיין ופונקציות שיטה ציבוריות. פונקציות שיטה אלו פועלות כשייטות מחלקה, בדיקת כמי שגורות מאפיין משמשות להגדרת מאפיינים. פרק 7 מתאר כיצד יוצרים ומשתמשים במודולי מחלקה מותאמים אישית.

牒בוק VBE

השינוי הבולט ביותר בסביבת הפיתוח של Access הוא牒בוק החדש של VBE. מנוקדת המבט של牒בוק הפיתוח,牒בוק VBE מביאר את התאימות בין Access 2000 ל宦游 Office האחרים (Word, Excel ו-PowerPoint). בסעיף זה נבחן לעומק את תכילת החלונות ופריסטים וכייד משתמשים בהם לניפוי שגיאות ופעולות נוספות. כמו כן נסקור את השימוש בסורק האובייקטים (Object Browser).

תבניות VBE

Access 2000 מציע לפחות שלושה נתיבים אל牒בוק VBE של מודולים שאינם תומכים בטופס או בדוח. מי שיש לו ניסיון בפיתוח באמצעות רכבי Office אחרים, יכול לבחור בחלון **מסד נתונים** (Database) את תפריט **כלי** (Tools), **מקרו** (Macro), **Macro** (Macro). **Visual Basic Editor**. קיצור הדריך מהמקלדת הוא **Alt+F11**. מקשי הקיצור מאפשרים גם להחליף בין עורך VB לחלון מסד הנתונים. אפשרות נוספים, אם ברשותך מודולים סטנדרטיים כלשהם, לחוץ מודולים (Modules) שבחלון **מסד נתונים**, ואחר כך לחוץ להיצאה כפולה על מודול סטנדרטי שברצונך להציג. חלון VBE יפתח כשהוא מציג את המודול שהרצה. כדי ליצור מודול סטנדרטי חדש, יש לחוץ על לחוץ מודולים בחלון **מסד נתונים** ולאחר מכן על **חדש** (New) בסרגל הכלים. פעולה זו תפתח מודול חדש וריק.

הערה:



צוות הפיתוח של Office עובד על פיתוח נטיב הפעלה רביעי של VBE שייהי מוכר לחלק מהפתחים לסייע בAccessibility. הנטיב החדש יאפשר לבחור, טופס או דוח, או מודול סטנדרטי בחילון **מסד נתונים** וללחוץ על לחץ **קוד** (Code). לחץ זה יפתח את VBE כשהוא מציג את הקוד התואם לאובייקט הנבחר.

כדי הגיעו אל מודול התומך בטופס או דוח, عليك לפתח תחילה את האובייקט בתצוגת **עיצוב** (Design). תוכל לחוץ על לחץ **קוד** (Code) שבירוג הכלים **עיצוב** (Design). חילון VBE ייפתח ויכיג את ראש המודול. כדי לעבור לשגרת האירוע המטיפול באובייקט מוגדר בטופס או דוח, לחץ על לחץ **בניה** (Build) בשורת המאפיין הרצוי, בכרטיסיה **אירוע** שבגיליון המאפיינים. אם לא קיימת שגרת אירוע הרצוי של האובייקט, הלחיצה על לחץ **בנייה** (Code Builder) תפתח שגרת אירוע ריקה.

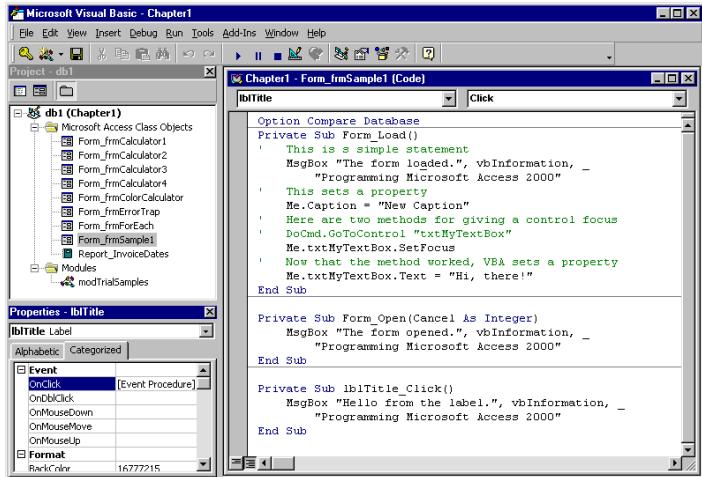
כשתגיעו ל-VBE, תרצה בוודאי לפתח את החלונות **Project** (המכונה גם Project Explorer) ו-**Properties**. חלונות אלה מאפשרים לפתח ולבחון מודולים נוספים של היישום בצורה נוחה. החילון **Project** מציג מודולים שאינם תומכים בטופס או בדוח ונמצאים בתיקיה **Modules**. מודולים שתומכים בטפסים ובדוחות נמצאים בתיקיה **Microsoft Access Class Objects**. כדי להציג ולהגדיר את האובייקטים הקשורים למחלקה, תוכל לבחור תיקיה הקשורה לטופס או לדוח. לפיתוח חלונות אלה, בחר בפקודה המתאימה מתוך **תפריט תפריט View** (תצוגה), או השתמש במקלדת או בלחיצני סרגל הכלים.

הערה:



כדי שהחלון **Properties** יוכל להציג את האובייקטים של מחלקת אובייקט Access, אובייקט זה חייב להיות פתוח בתצוגת **עיצוב** (Design). לחיצה כפולה על מודול או על מחלקה בחילון **Project** מציגה את ההווראות והছזרות המתאימות בחילון **קוד** התואם, אך האובייקטים מופיעים רק בתנאי שאובייקט המחלקה גם פתוח בתצוגת **עיצוב**.

תרשים 1.7 מציג את קוד מסד הנתונים המדגמי של פרק זה, לאחר טעינתו ל-VBE. חילון **Project** מציג את המחלקה Form_frmSample1 שselected. תחת חילון זה ניתן לבחון בחילון **Properties** המציג את פקד התווית של כוורת הדוח שנבחר. החילון מראה גם שלפקד התווית שיגרה הקשורה לאירוע **onClick**. החילון **Code** שמימין החלונות **Properties** ו-**Project** מציג את קוד שגרת אירוע. באפשרותן להזיז, לשנות גודל או לעגן כל אחד מחלונות אלה.

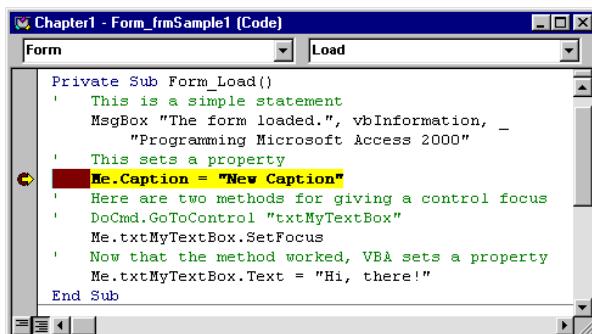


תרשים 1.7: החלונות **Properties**-**Project** ,**Code** מציגים את קוד מסד הנתונים המדגמי של פרק זה

לחיצה כפולה על מחלקה אחרת או אובייקט Module כלשהו של Access בחלון **Project** פותחת את חלון הקוד התואם. החלון מאפשר לבדוק, לעורוך או להעתיק את הקוד. חלון **Code** מעוצב בצורה מוכרת, בחלקו העליון תיבותו הרשימה הנפתחת **Object**-**Procedure**. ניתן לנצל אותו כדי לנווט במודול גדול, או כדי לפתוח שורות חדשות במודול קיים. החלון **Properties** מציג את מאפייני המחלקה או המודול הנוכחי של Access. החלון מאפשר לעורוך את מאפייני האובייקט, אולם קל ונוח יותר לעורוך מאפייני אובייקט של טופס או דוח בתוצאות **עיצוב**.

ניפוי שגיאות

תוכל לנצל את חלון **Code** לבדיקת הקוד ולניפוי שגיאות. תוכל להוסיף או לבטל נקודת עצירה (breakpoint) במשפט, על ידי לחיצה על השוליים שלושה. מסמן את נקודות העצירה בדרכן המקובלת, ככלומר נקודת השולאים השמאליים.



תרשים 1.8: חלון **Code** של אירוע טיענת המחלקה Form_frmSample1 בעת עצירה. החץ מצביע על שורת הקוד הבאה לביצוע

תרשים 1.8 מציג את מטרת הבאים בעצירה על השורה השנייה מתוך 4 שורות הקוד בשירה ; החץ שבשוליהם השמאליים מצביע על שורת הקוד הבאה לביצוע.

Run לחידוש ביצוע הקוד לאחר העצירה, לחץ על **Continue** (המשך) בתפריט הפעלה). תרשים 1.9 מציג את תוצאה ביצוע שאר חלקי הקוד של השירה Form_frmSample1. שים לב שהכיתוב בטופס הוא "New Caption". הטקסט בתיבת הטקסט הוא ."Hi, there!"



תרשים 1.9 : תוצאה הפעלה של שלוש שורות הקוד הנותרות של תרשים 1.8

תוכל להשתמש בחץ שבתרשים 1.8 כדי לדלג על שורת קוד אחת או יותר בנתיב הביצוע. לדוגמה, תרשים 1.10 מציג את תוצאות גיררת החץ לשורה שפעילה את השיטה SetFocus ובחירה בפקודה **Continue** מותוך התפריט Run. השירה מדלגת על הצבת המחרוזת "New Caption" (כיתוב) של הטופס, ולכנו הטופס נראה כמעט זהה לזה שבתרשים 9, למעט הכיתוב שלו, "My Default Caption".



תרשים 1.10 : תוצאה הדילוג על השורה השנייה הנותרת בתרשים 1.8 וההמשך הביצוע

תרשים 1.11 מציג את תוצאת גיררת החץ בתרשים 1.8 אל שורת הקוד האחרון שמבצעת את ההצבה בתיבת הטקסט. פועלה זו תגרום כמובן, להציג הודעה שגיאיה, עקב הניסיון להציב ערך במאפיין מבלי להעביר את המיקוד לאובייקט שלו.

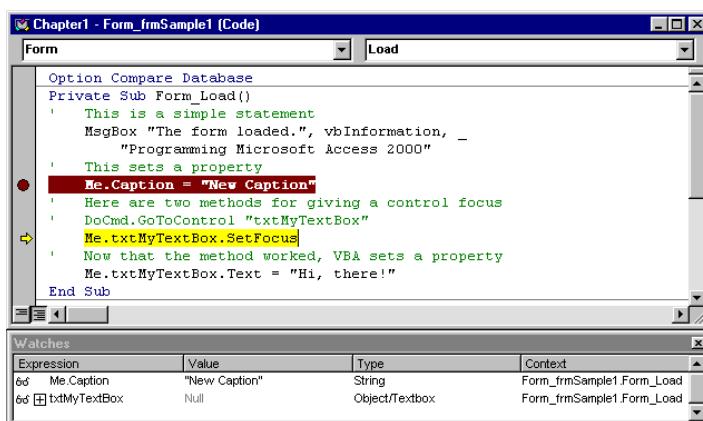


תרשים 1.11: תוצאת הדילוג אל השורה الأخيرة של תרשيم 1.8 והמשך ביצוע הקוד

כולל את רוב הרכיבים התפקודיים של חלון איתור הבאים של Access 97, Access, שהכיל חלון Watch וחלוון Locals. כל חלון הכליל מסך מפוץל שככל את החלון המיידי (Immediate). Access 2000 מציג את שלושת החלונות, Locals, Watch ו-Immediate בזירה נפרדת. ב-VBE פותחים את החלונות על ידי שימוש בתפריט View (תצוגה). תוכל לגרור, לשחרר ולשנות גודל של כל אחד מהחלונות לצד החלונות (תצוגה). בחר את Tools (כליים), Options (אפשרויות) והשתמש ב-caretיסיה Docking (עיגון) כדי לקבוע את החלונות שברצונך לעגן.

תוכל לנצל את חלון Watch (צפייה) למעקב אחר ערכי ביטויים, משתנים ואובייקטים במהלך ביצוע הקוד. לאחר פתיחת חלון Watch, תוכל להוסיף משתנים לצפייה על ידי בחירה ב-Add Watch, Debug. בטרם הפעיל את הפקודה, בחר משתנה למעקב. כשפתחת תיבת הדו-שיח Add Watch, בחר סוג לצפייה ולאחר על OK כדי לסגור את תיבת הדו-שיח. ערך המשתנה מוצג בחלון הצפייה במהלך ביצוע הקוד. אם תבצע את הקוד במצב עצירה (break mode), תוכל לאמת ערך של משתנים חיוניים בכל שלב ביצוע.

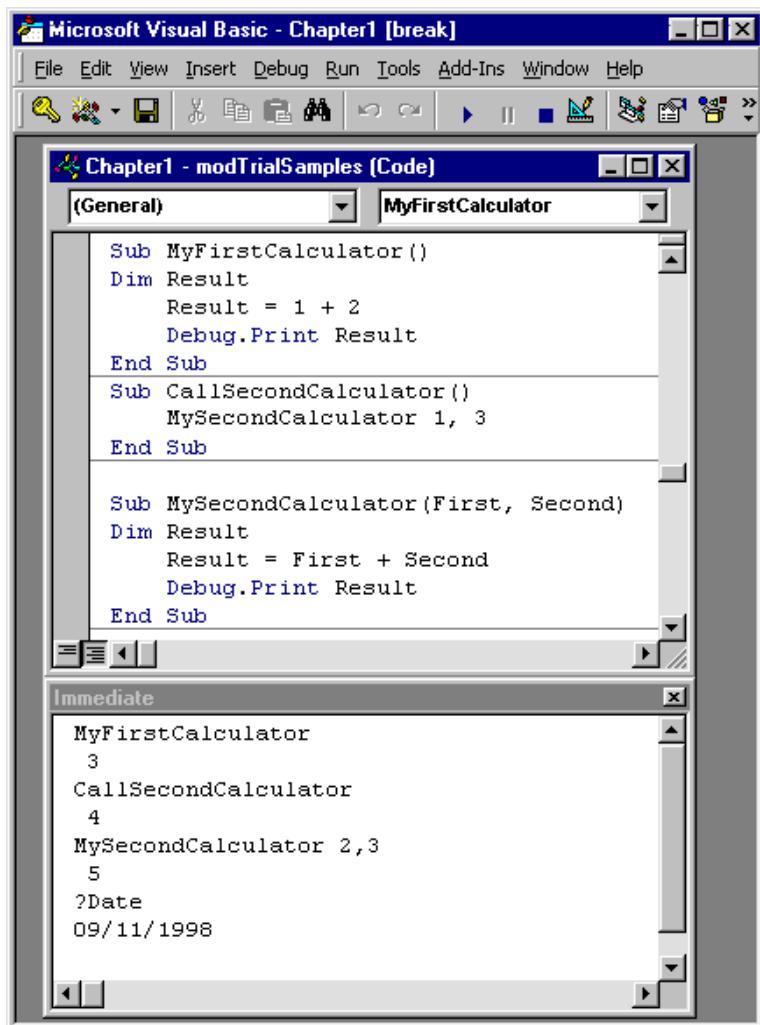
תרשים 1.12 מציג את ערך המאפיין Caption של frmSample1 של Caption, מייד לאחר שהקוד שינה אותו מ-"New Caption" ל-"My Default Value". חלון Watch מציג את ערך txtMyTextBox בתור Null, מכיוון שהקוד לא ביצע עדין את משפט ההצבה.



תרשים 1.12: חלון Watch המציג את מצב הביטויים בעת ביצוע שגרת אירוז

החלון Locals יכול להציג את כל המשתנים שבתווחת ההכרה, בעת שהקוד מתבצע במצב עכירה. כמשמעותו נקראות העצירה, החלון מכיל אובייקט Me. תוכל להרחיב את Me ואת רכיביו שנבחרו, בזה אחר זה, כדי לגלו את ערכיו כל המאפיינים והמשתנים. הדבר יסייע לך בשיש צורך לחקור את מחלק התוכנית לעומק.

החלון Immediate עשוי להיות הכלי הנוח ביותר לפיתוח קוד וניסוי באגים. החלון מאפשר להפעיל פונקציה או שגרת משנה כלשהי שנמצאת בטווח ההכרה. תוכל גם להעריך ביטויים ולבדוק דרכים שונות לכתיבת פונקציות. בגרסאות שלא מאפשר להציג ביטויי צפיה, החלון המיידי היה מקום נוח להדפסת תוצאות ביניהם במהלך ביצוע תוכנית במצב עכירה. ניתן עדין לנצל חלון זה בסביבות מיוחדות.



תרשים 1.13: כך נראה חלון Immediate בעת הפעלת שגרות ופונקציות מוכללות

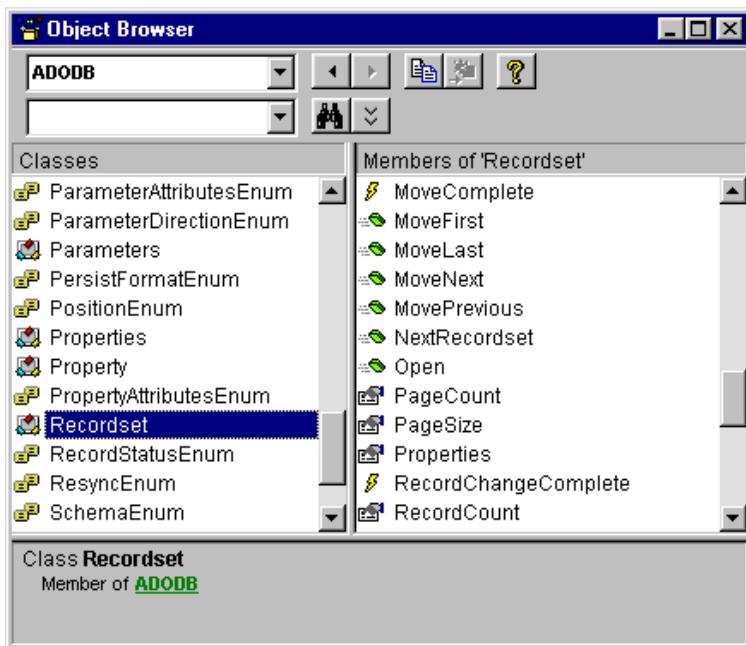
תרשים 1.13 מציג שלושה שימושים אפשריים של החלון **Immediate**. עסקנו בפונקציות אלו בתחלת הדיוון בשגרות, אך לא הצגנו את הפלט שלון בחולון **Immediate**. הקלדת Enter MyFirstCalculator והקשת CallSecondCalculator מפעילה שיגרה שנקראת בשם זה. מופעלת הפונקציה שמדפסה את התוצאה 3 בחולון. כך גם בשיגרה שנקראת בשם זה. מופעלת הפונקציה שמדפסה את התוצאה 4 הדוגמה האחורונה מציגה כיצד מציינים ארגומנטים עבור הפונקציה MySecondCalculator עליה לבצע הפעולות `LET`, `ROW` ואת הארגומנטים כשם מופרדים בפסיקום. לדוגמה זו חשובות מכרעת, כיון שהיא ממחישה את אופן השימוש בארגומנטים המועברים לשיגרה או פונקציה כדי לחשב ערך החזרה.

תוכל גם לנצל את חלון **Immediate** להפעלת פונקציות מוכללות וモთאמות אישית. בעבודה עם פונקציה, עלייך לצין פקודת הדפסה כדי להחזיר את תוצאת הפונקציה. באפשרותך להקליד את מילת המפתח Print, ללחוץ על לחץ ? או להקליד את שם הפונקציה ולסייעים בהשאלה על Enter. הדוגמה האחורונה שבחלון **Immediate** מציגה כיצד קוראים לפונקציה המוכללת Date.

סורך האובייקטים

תרשים 1.14 (סורך האובייקטים) המוצג בתמונות המציג מודלים אובייקט של Object Browser (סורך האובייקטים) המוצג בתמונות 1.14, הוא כלי רב-עוצמה ללימוד נושא מודלי אובייקט. הכליל חשוב במיוחד למפתחים בסביבת Access, עקב החידושים הרבים שהוא כולל בנושא מודל האובייקט. לדוגמה, Access 2000 כולל שפת פיתוח חדשה לגישה לנתונים – ADO (ActiveX Data Objects) – שתדחק בסופו של דבר את מקומה של DAO (Data Access Objects) ממוקמה. מימוש DAO ב-2000 מבוסס על לפחות שלושה מודלים אובייקט.

טרם תציג את מחלקות האובייקט של ADO בסורך האובייקטים, عليك להגדיר הפניה אל הספרייה המתאימה. בחר Tools (כלי), References (הפניות) כדי לאמת או ליצור קישורים אל ספריות מסווג ADO. פתח את סורך האובייקטים על ידי לחיצה על הלחצן Object Browser בسؤال הכלים הרגיל של VBE. ניתן לפתוח אותו גם באמצעות הטריט View או הקשה על מקש הקיצור F2. אחת הספריות ADO נקראת ADODB. בחר אותה מתוך הרשימה הנפתחת Project/Library בראש סורך האובייקטים. בחרית הספרייה משנה את תוכן הרשימות Classes (מחלקות) ו-Members (חברים) של הסורך. תרשים 1.14 מציג את המחלקה Recordset שנבחרה מתוך הרשימה members ורשימה מעורבת של מאפיינים, שיטות ואירועים המוצגים ברשימה Members. בחר ערך של חבר ולחץ על לחץ ? כדי לקבל עזרה מפורטת בנושא. הודות למנגנון החיפוש שלו, סורך האובייקטים מסוגל גם לחפש מחלקות וחברים מוגדרים. השתמש בתיבת הרשימה הנפתחת השנייה כדי לצין קירטוריוני חישוב. לעיתים מקבלים כך מחלקות וחברים נוספים שנמצאים בספריות אחרות. סורך האובייקטים מאפשר לבדוק אותם בנפרד.



תרשים 1.14: מחלקת ערכת הרשומות של ADODB ומבחן מחבריה, כפי שהם נראים בתצוגת Object Browser

ט, סוגים נתונים והצורות

ב-Access עליך לציין סוגים נתונים במקומות רבים, לרבות משתנים בשגרות וטבלאות נתונים שמאוחסנות בהתקני אחסון. הטעיפים הבאים עוסקים בנושא פיתוח הקשורים לנתונים שמקורם ב- Jet ובסימוש במשתנים בשגרות.

Jet

Access 2000 תומך ביסודות בשני מנגנוני מסד נתונים. במקור, Access היה קשור למנגנון מסד הנתונים Jet. Jet 7 כולל גם תמיכה מוגבלת במסד הנתונים SQL Server. פרק 12 עוסק בהיבטי התאמיות של Access 2000 עם SQL Server. הסעיף הנוכחי עוסק בהשפעות חידושי Jet 4 על עיצוב יישום 2000 Access. חידושים אלה מסוגלים להשפיע על גודל קבצי מסד הנתונים ועל נעלית רשותות, על אופן הטיפול בסוגי נתונים שדות שנבחרים (כולל שדות המפתח Memo (תזכיר), Hyperlink (היפר- קישור) ו- auto-incrementing (מספר אוטומטי בהפרשים קבועים) וקישורים אל מקורות נתונים חיצוניים.

Access 2000 מאחסן את כל נתוני הטקסט וההערות בתבנית Unicode. תבנית זו מחייבת את מערך התווים מרובה הבתים – MBCS (multi-byte character set) שהייתה מקובלת בגרסאות קודמות של Access, לטיפול בשפות כגון יפנית, סינית ועוד. הוא תקן-משנה של ISO 10646, המתאר את שיטת הקידוד הדיגיטלית של כל השפות הכתובות. תקן חדש זה מיציג כל TWO במציאות שני בתים (Bytes) ולא באמצעות בית בודד, כפי שהוא נהוג בשיטה הקודמת. לכן, מסדי נתונים SMBOSSIMS במידה רבה על שדות מבוססי-טו, עלולים להכפיל את גודלם. Jet מתגבר על הבעיה על ידי דחיסת הנתונים ופרישתם מחדש לפי הצורך. הוא דוחס שדות תזכיר (Memo) באורך 4,000 תוים או פחות, כך חלק משורות שדה נזקירות נדחיסות, אם כי לא כולם. Access וגם ADO דוחסים אוטומטית תוים בקידוד Unicode, אך DAO אינו תומך בדחיסת נתונים מסווג מחורזות. מתכנתים שפתחים ב-SQL יוכלו לבצע באפשרות WITH COMPRESSION בעט הגדרת טבלאות.

נוסף לתבנית הייצוג של סוגי הנתונים, גודל הדף הוכפל ומעתה גודלו 4KB, דבר שמצמצם את הפעולות הבו-זמןיות שמקורן בהתנגשויות נעילת דף (page-lock). Jet פותר את הבעיה באמצעות נעילת שורה יחידה. ניתן להקטין את בעיית הבו-זמןיות על ידי נעילת רשומות בזדדות במקומות נעילת דפים שלמים. Access 2000 מאפשר למשתמשים לעדכן שתי רשומות בו-זמןית באותו דף.

הערה:



גודל הדף החדש מגיל את הגודל המקסימלי של מסד הנתונים מ-1.07 ג"ב

עד 2.14 ג"ב.

נעילה ברמת שורה בזדדת היא אפשרות בירית המחדל הכללי, אך משתמשים ומפתחים יכולים לנצל את הנעילה ברמת הדף, כפי שהוא מקובל בגרסאות קודמות. שדות תזכיר ואיינדקסים אינם תומכים בעילה ברמת שורה בזדדת. גישה לנ נתונים באמצעות DAO או Jet SQL מאפשרת לחזור לנעילה המקובלת ברמת הדף. גישה לנתונים באמצעות טפסי Access ו-DAO מתחכמת תמיד באמצעות נעילה ברמת שורה בזדדת.

נעילה ברמת שורה בזדדת חיונית לצמצום התנגשויות בו-זמןיות, אך היא כרוכה בתקורת גבואה של הגדרה והסרת נעילות רשומה. השפעת התקורת תלולה בגודל הרשימה יחסית לגודל הדף, ולמספר הנעילות הדורשות במשימה אחת.

המעבר ל-Unicode מאפשר לבצע מיון תואם NT Microsoft Windows, מה שמשמעותו התאימות והביצועים, מכיוון ש-2000 מסוגל לבצע מיון עקי תחת מערכות הפעלה Windows 95 ו-Windows NT אחד. הדבר אפשרי מכיוון ש-Windows 95 תומכת במילוי בשפת בירית המחדל של המערכת, בעוד ש-NT Windows תומכת במילוי תקין בשפות רבות.

בנוסף, Visual Basic 6 ו-SQL Server 7 תומכות בתקן מיון זהה, וכך ניתן לתקן מוצרי תוכנה רבים. ברוב השפות 2000 מבצע מיון במהירות גדולה ב-50%, אך בשפות מסוימות (כגון תאילנדית) הוא עושה זאת במהירות גדולה אף יותר.

4 Jet תומך ביצירת אינדקס של 255 התחומים הראשוניים של נתונים מסוג תזכיר (Memo). גרסאות מוקדמות של Jet Access ו-Jet לא תמכדו ביצירת אינדקס של שדות תזכיר. האורך המוגבל של האינדקס אינו עונה על הדרישות של יישומי שדות תזכיר, אך יש לו שימושים מיוחדים עבור נתונים מסוג היפר- קישור, שנוצר מסוג הנתונים תזכיר. שיטת יצירה האינדקס החדשה מסוגלת לשפר את המיון והחיפוש של שדות היפר- קישור.

4 Jet גם מאפשר למפתחים לציין ערך התחלתי וערך צעד (step value) עבור שדות מספור אוטומטי בהפרשים קבועים (auto-incrementing). בנוסף, המשפט החדש ALTER TABLE מאפשר למפתחי SQL Jet לאפס את ערכי ההתחלה והצעד. כדי לשזר את הערך החדש של עמודת המספר האוטומטי בוחרים משפט SELECT @@IDENTITY החדש. יש להעביר את ערך משפט SQL המיעוד עבור השיטה Open של אובייקט ערכת הרשומות (recordset) במשפט DAO. DAO אינה תומכת במשפט החדש.

4 Jet כולל את הטכנולוגיה המשופרת הניתנת להתקנה ISAM, בכמה תחומים. ISAM עשוייה להשפיע על הזדמנויות הפתיחה ב-Access כשייש להשתמש במקרים מסוימים חיצוניים. הגירסה החדשה של ISAM מאפשרת לקרוא מסמכים Web שבוססים על תבנית Exchange ISAM .Unicode כולל שיפורים בתחומים אחדים. ראשית, הוא קורא אינדקסים מתרץ Exchange Server, דבר שמאיצז בצורה דרמטית את תהליכי החיפוש אחר רשומות במקומות נתונים של Exchange. בנוסף, הגירסה החדשה של ISAM תומכת בפנקס **ה כתובות של Windows** מוגדרי-התאמת אישית וגם של ללקוח Outlook Express (Windows Address Book) של Microsoft. מבוסקים קטעים שבtems הדואר מיועד בעיקרו לשלוח חיצוני ולא פנימי, יש בכך יתרונות משמעותיים אם כי בתחום הפריון האישי בלבד. שיפור נוסף הוא יכולת Jet לאחזר ל��וח מוגדרי-התאמת אישית וכן של Outlook תחת Exchange server משתמשי מסדי הנתונים dBase ו-Paradox יכולים להמשיך לגשת למסדי נתונים אלה עם זכויות קראיה וכתיבה באמצעות Jet. משתמשים ה Zukotkim לתמייה בגירסה עדכנית יותר, חייבים להציג עותק של Borland Database Engine מיצרנו תוכנה אחר.

סוגי נתונים

VBA משתמש במסתנים לשימור תוצאות חישוב, להגדרת מאפיינים, שליחת ארגומנטים לשיטות ולהעברת ערכים בין שגרות. כדי שיתפקיד בצוrah יעליה, VBA מבסס משתנים אלה על קבוצת סוג נתונים. בסעיף זה נתמקד בסוגי נתונים של משתנים. סוגים נוספים של נתונים דומים באופן כללי, אך אפשר להשתמשם בוגני נתונים לשימה.

הטבלה שלפניך מציגה את סוגי הנתונים העיקריים ודרישות האחסון שלהם בתוכניות VBA. מפתחים חזוקים לרמות דיק גבוחות בחישוביהם המספריים יכולים לנצל לשם כך את סוג הנתונים **Decimal** (עשרות) בטור קבוע-משנה של סוג הנתונים **Variant**. סוגי הנתונים **Decimal** הם מספרים שלמים לא חתומים של 12 בתים שקנה מידיה שלהם מצין את המספר הרabi של ספרות עשרוניות שנייה לאחסן מימין לנקודה העשורה. לא ניתן להציג על סוג נתונים **Decimal** באמצעות משפט **Dim**, אך אפשר לאחסן סוג נתונים כזה על ידי הפיכת **Variant** באמצעות הפונקציה **CDec**. כללית, נדרש לשימוש בסוג הנתונים הקטן ביותר האפשרי, כדי להוtier מקום בזיכרון עבור משתנים נוספים ופקודות ביצוע של יישומים. יחד עם זאת, כאשר חזוקים לדיק גדול בעולות חשיבות, ניתן לשימוש בסוג הנתונים **Decimal**.

אם לא תצהיר על משתנה בטור סוג מבין המוגדרים, Access יקצת לו את סוג הנתונים **Variant**. סוג נתונים זה הוא גמיש, מכיוון שהוא מטפל בערכים מספריים ובערכי מחרוזות כאחד. סוג הנתונים **Variant** יכול לאחסן את כל שאר סוגי הנתונים חוץ מסוגים מוגדרי-משתמש (שנגורים מכל שאר הסוגים). כש-**VBA** מבצע פעולה על שני נתונים מסוג **Variant**, עליו לאחזר תחילה את תוכן הנתון השני ולהמיר את התוצאה לשוג המשנה של ה-**Variant** הראשון. תהליך זה מאט את הטיפול במשתני **Variant** לעומת מנגנון משתנים אחרים. בנוסף, סוג המשנה של הנתונים המוחזר באמצעות **Variant** עלול להיות שגוי בהקשר של היישום.

באפשרות לשימוש בפונקציה **VarType** כדי לקבוע את סוג המשנה של סוג הנתונים **Variant**. הפונקציה קולטת נתון מהסוג **Variant** ומחזיריה קבוע **VBA**, שמצוין את סוג המשנה של הערך בטור סוג נתונים. לדוגמה, **Variant** **varMyVariable** מכיל את הערך "Hi, there!" הפונקציה **VarType(varMyVariable)** מחזיר אותה(#8). באופן דומה, אם **varMyVariable** מכיל את התאריך "#1/1/2000", הפונקציה **VarType(varMyVariable)** תחזיר את(#7). עיין בתיעוד המקוון או ב-**Object Browser** (סורך האובייקטים) לקבלת רשימה מלאה של קבועי סוג-משנה של **Variant**.

פונקציות המירה של **VBA** מאפשרות להמיר סוג נתונים **Variant** לסוג-משנה מוגדר, וגם לציין את סוג התוצאה לה מצלפים. לדוגמה, **CDbl(varMyVariable)** מחזיר את תוכן המשתנה **varMyVariable** בתור משתנה מהסוג **Double**.

על אףesi כי קיבלת טעות והוצרך בעיבוד נוספים, **Variant** הוא סוג נתונים מקובל. סוג ברירת המחדל של נתונים טבלה, שאלתה, דוח ושדה טופס, הוא **Variant**. אם תשאיר בטבלה שדה לא מוגדר, הוא יחויר ערך ריק (**Null**) ברגע שתפעיל עליו שאלתה (בהתחלת שלא נשלחו אליו נתונים), שהוא אחד משני ערכי הנתונים המיוחדים של **Variant**. אם הערך **Null** מצין נתון חסר, לא ידוע, או לא ישים. הפונקציה **IsNull** של **VBA** בודקת אם הערך שנבחר הוא **Null**, והאופרטור **Is Null** בקריטריון שאלתה עשויה פועלה זהה בשדה. טיפול בזיהירות במשתנים המכילים **Null**, מכיוון שערכי **Null** נוטים להתפשט. צירוף של משתנה שווה ל-**Null** עם משתנה כלשהו אחר, יחויר תמיד ערך **Null**.

סוגי נתונים של משתנים

שם	מספר בתים	טווח
(Byte)	1	0-255
(Boolean)	2	False או True
(Integer)	2	-32,767 עד -32,768
(Long)	4	-2,147,483,648 עד -2,147,483,648
(Single)	4	1.401298E-45 עד 3.402823E38 ערבים שליליים; ערבים חיוביים עד 1.401298E-45 ערבים חיוביים עד 3.402823E38 עבור
(Double)	8	1.79769313486232E308 עד -1.79769313486232E308 -14.94065645841247E-324 עד 4.94065645841247E-324 ערבים שליליים; ערבים חיוביים עד 1.79769313486232E308 עבור ערכים חיוביים
(Currency)	8	-922,337,203,685,477.5808 עד 922,337,203,685,477.5807
(Date)	8	1 בינואר, 100 עד 31 בדצמבר 9999
(Object)	4	הפניה לאובייקט (ראה מושפט Set בעזרת המזקונת).
(Fixed String)	אורך המחרוזת עד 64,000 תווים	מחרוזת קבועה
(Variable String)	אורך המחרוזת עד כ-2 מיליארד תווים	מחרוזת משתנה
(Variant)	16	כמו כפול
(Variant)	22+אורך המחרוזת	כמו כפול
מוגדר משתמש	תליי באלמנטים	סך האלמנטים המרכיבים את סוג הנתונים המותאם אישית

הערה:



מהחר שרך נתונים מהסוגים Variant יכולם לקבל ערכי Null, עליך להשתמש בסוג זה בכל מקרה שהשימוש שלך יכול לצאת נשכר מכך. ערכי Null מבטלים את הצורך בערכים שירוטיים לציין נתונים חסרים בשדה או בסוג משתנה.

מילת המפתח Empty (ריק) משמשת בתור סוג-משנה ומיצגת ערך משתנה שלא אוטחל. ב-VBA משתמשים בפונקציה IsEmpty כדי לקבוע אם משתנה הוא בעל ערך כזה. Null, 0 ומחרוזת ריקה מתווים ("") שונים זה מזה. כ-Sh VBA הופך משתנה מסוג Variant שערכו Empty, הוא הופך את המשתנה ל-0 או - "", בהתאם לסוג הערך ההולם ביותר במקרה זה, מספר או מחרוזת.

כשעליך לקרוא לפונקציית Windows באמצעות ספריות קישור דינמי (DLL), לעיתים תזדקק למשתנה מוגדר-משתמש. عليك להציג על המשתנה מוגדרי משתמש בין המשפטים Type Type ו- End Type. השורות הבאות בין שני משפטיים אלה צריכה להגדיר את האלמנטים של סוג המשתנה המותאם אישית. למשל, אפשר להתייחס בספר בתור אוסף אלמנטים, כולל ISBN (מספר ספר סטנדרטי בינלאומי), שם הספר, שם המחבר (או המחברים), המו"ל ומספר דפי הספר. באפשרות להגדיר את סוג הנתונים המותאם אישית כך:

```
Type Book
```

```
    ISBN as Long  
    Title as String  
    Authors as String  
    Publisher as String  
    Pages as Integer
```

```
End Type
```

```
Sub MyBook
```

```
Dim udvMyBook as Book  
    udvMyBook.ISBN = 1234567890  
    udvMyBook.Title = "Programming Microsoft Access 2000"  
    udvMyBook.Authors = "Rick Dobson"  
    udvMyBook.Publisher = "Microsoft Press"  
    udvMyBook.Pages = 550
```

```
End Sub
```

מוסיפים את זוג המשפטים End Type ו-Type-Inside באזורי התחזרות הכללי במודול. השיגרה MyBook יוצרת מופיע (instance) של משתנה מוגדר-משתמש שנקרא Book.

הצחרות

ניתן להציג על משתנים וקבועים (קבוע הוא משתנה מיוחד שלא ניתן לשנות את ערכו לאחר שקבע במשפט). להצירה שני תפקדים: היא קובעת את טווח ההכרה (scope) של המשתנה (התחום שבו הישום מכיר את המשתנה), ומגדירה את סוג הנתונים של המשתנה. אלו שתי סיבות טובות להציג על משתנים בטרם מתחילה לטפל בהם. באמצעות המשפט Option Explicit שבאזור General (כללי) של המודול תוכל לדרוש הצירה על המשתנה.

המשפט Public (ציבורי) באזורי Declaration (הצחרות) של מודול משמש כדי להציג על משתנים שהישום יכול לטפל בהם מתוך שיגרה כלשהו שנמצא ביחסו, כולל אלו שנמצאות במודולים אחרים. אפשרו זאת להשמש במשפט Private (פרט) כדי להציג בפורש משתנה בעל טווח הכרה מקומי במודול (הדבר אינו הכרחי, כיוון שהמשפטים Dim ו-Static מצהירים על משתנים שהם פרטיים במודול, בברירת מחדל).

הצירה Dim שומרת מקום בזיכרון עבור משתנה רק עד סיום השיגרה שבה מופיעה הצירה על המשתנה. משמעו של הדבר, משתנים שהוצגו באמצעות המשפט Dim מאבדים את ערכם לאחר הקראיה לשיגרה, עד הקראיה הבאה. משתנים שהוצגו בעזרת המשפט Static נשמרים לאורך חיי המודול או עד אתחול היחסום או הפעלו מחדש (אחד השימושים של הצירה על משתנים סטטיים הוא עリכת סיכומים או קביעת מספר הפעמים שהשיגרה הופעלה).

הערה:



באפשרות לנוקות את הערכים מהמשתנים הסטטיים בשיגרה על ידי בחירה

באפשרות Reset מתפריט Run של VBE.

ניתן להשתמש במילת המפתח Static בהצירות על פונקציות ושרגות וגם בהצירות על משתנים. מילת המפתח פועלת בזרה זהה עבור פונקציות ושרגות, בכך שהיא לשמורת את כל המשתנים המקומיים בפונקציה או בשיגרה, לכל אורך חיי המודול.

מוסיפים את מילת המפתח As להצירה Dim או Static, כדי לציין סוג נתונים של המשתנה. המשפט Dim inMyNumber As Integer מציין על סוג הנתונים של המשתנה inMyNumber בתור מספר שלם.

מערכות

Access מאפשר להציג על **מערכות** (arrays). מערכים הם משתנים המכילים רשימות ערכים מסוימו סוג. מערך יכול להיות רב-מימדי, ומספר המימדים שלו עשוי להגיע ל-60. לדוגמה, המשפט Dim aryScores(2, 1) As Integer, מזכה מערך דו-מימדי של שישה אלמנטים: שלוש שורות ושתי עמודות של מספרים שלמים (הספרה מתחילה מ-0).

כברית מחדל, מערכים הם מבוססי-אפס. המשפט Dim aryMyName(1) כולל שני אלמנטים, שהקוד יכול להפנות אליהם בתור aryMyName(0) ו-aryMyName(1). השימוש במשפט Option Base 1 Declarations כדי לציין מערך כבסיס-אחד. השגירה שלפניך מצהירה על מערך בן שני אלמנטים, מגדרה את ערכיו ומדפסה את האלמנטים שלו ואת סכום האלמנטים בחلون Immediate.

```
Sub ArrayTest()
Dim aryMyExpenses(1) As Currency, Total As Currency
    aryMyExpenses(0) = 10.5
    aryMyExpenses(1) = 22.25
    Total = aryMyExpenses(0) + aryMyExpenses(1)
    Debug.Print aryMyExpenses(0) & " + " & aryMyExpenses(1) & " = " & Total
End Sub
```

הערך aryMyExpenses מכיל שני אלמנטים שסוג הנתונים שלהם הוא Currency (מطبع). סוג נתונים זה כולל שמות בטאים, ולכן המערך aryMyExpenses כולל 16 בתים. מאחר שבאפשרותם להציג מעריכים בעלי מספר רב של מימדים ומחוזות באורך משתנה, משאבי הזיכרון עלולים להתכלות במהירות.

אתה משתמש במילוט המפתח Dim, Public, Static ו- Private להגדרת טווח ההכרה של מערך, כמו עברו משתנים סקלריים. מילת המפתח ReDim משמשת לשינוי דינמי של מימדים במערך רב-מימדי.

לוגיקה של התנויות ומבנה לולאה

ביצוע מותנה של קוד ולולאות מהוות מרכיב בסיסי בפתרונות מבוססי-קוד רבים. VBA כולל מבחר אפשרויות גדול ליישום תהליכיים אלה. הסעיפים הבאים סוקרים את סוגים המשפטים העיקריים המאפשרים ביצוע מותנה של תוכנית וכוללים דוגמאות מעשיות.

If...Then

שגרות רבות אין ממלאות את תפקידן על ידי ביצוע סדרתי של שורות הקוד. לעיתים רוצים להתנות את ביצוע הקוד – לדג על שורות מסוימות ולבצע שורות אחרות. אחת הדרכים הגמישות והבטוחות לעשות זאת בשגורות VBA היא באמצעות משפט If...Then משפט בסיסי זה כולל למעשה שלוש וריאציות בסיסיות. הראשונה היא ביצוע מותנה של קטע קוד פשוט. כך נראה התחביר שלו:

```
If condition Then
    ' Statements
End If
```

קטע מהסוג If...End If יכול להכיל בתוכו משפט אחד או יותר. המשפט If מצין את סוף הקטע. VBA מבצע את המשפטים שבתוך הקטע, רק כאשר תנאי מסוים מתממש (כלומר, מקבל ערך True). ניתןukan קטעי If...End If רבים זה בתוך זה.

הווריאציה השנייה של משפט If...Then מאפשרת לקוד לבצע אחד מתוך שני קטעי קוד. כך נראה התחביר שלו:

```
If condition Then  
    Statements1  
  
Else  
    Statements2  
  
End If
```

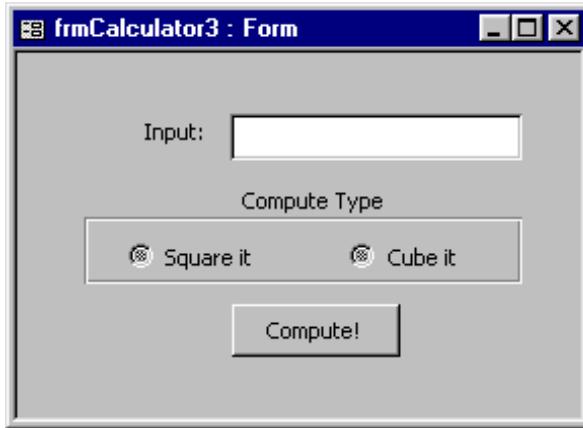
סוג משפט זה מבצע קוד אחד מתוך שניים. כמשמעותו, מtbodyים משפטים הקבועה הראשונה. אחרת, מtbodyים משפטים הקטע השני.

יעיצוב זה גמיש יותר, אך הוא עדין מגביל אותנו לשתי אפשרויות בלבד. למעשה של דבר, באפשרותך לנקן משפט If...Then כך שמספר האפשרויות יגדל, אך VBA כולל דרך נוספת ש解脱ת ביצוע מותנה של קטע קוד אחד או יותר, עם התחביר הבא:

```
If condition1 Then  
    ' Statements  
  
ElseIf condition2 Then  
    ' Statements  
  
Else  
    ' Statements  
  
End If
```

כורה זו של משפט If...Then משלבת תנאים רבים ושלוש קבוצות משפטיים, או יותר. באפשרותך להוסיף קבוצות משפטיים ותנאים חדשים על ידי הוספה נוספת IfElseIf, ובهم התניות ומשפטים שלהם. דרך זו מזכה אפשרויות רבות יותר מקודמתה, לא זו בלבד שהיא מאפשרת מספר גדול יותר של תנאים, אלא שהיא גם מגבילה את הביצוע של כל קטע קוד (למעט האחרון) במקרה שבו תנאי הבדיקה מוגדר במלואו. בקטע If...End הקודם, משפט If...Then מבצע את קטע המשפטים השני כל פעם שהתנאי שבמשפט הראשון אינו מוגדר. כאשר אי-מיולי התנאי הראשון אינו מהוות אוטומטיות סיבה לבצע את קוד השני, יש צורך בתבנית התניה נוספת.

עיין בטופס שבתרשים 1.15. הטופס כולל תיבת טקסט, קבוצת לחצני אפשרויות ולהציג פקודת. קבוצת לחצני האפשרויות מאפשרת להשתמש בהעלות את הערך שבתיבת הטקסט בריבוע או בחזקת 3. כדי לחשב ריבוע של מספר, יש להקליד את המספר בתיבת הטקסט, לבחור אפשרות **Square** ולבסוף להזע על לחץ **Compute!**



תרשים 1.15: טופס לחישוב חזקות ריבועיות או חזקות שלישיות של מספרים

הקוד שלפניך מציג את שתי השגרות שמאפשרות לבצע הולאה בריבוע. שגרת האירוע cmdComputer_Click מגיבה להחיצה. אם קבוצת לחצני האפשרויות (opgComputerType) מקבלת את הערך 1, סימן שהמשתמש בחר באפשרות Square it (לחצן האפשרות הולאה בריבוע). אם קבוצת לחצני האפשרויות שווה ל-1, השיגרה קוראת לשירה MySquarer. אחרת, היא מסתiyaמת מבליל לבצע חישובים כלשהם.

```
Sub cmdComputer_Click()
    If opgComputeType = 1 Then
        MySquarer txtInput.Value
    End If
End Sub

Sub MySquarer(MyOtherNumber As Double)
    dblResult = MyOtherNumber * MyOtherNumber
    MsgBox dblResult, vbInformation, "Programming Microsoft Access 2000"
End Sub
```

השירה cmdComputer_Click מנצלת את מבנה If...Then. פוללה העיקרית היא ביצוע מותנה של קטע קוד אחד. במקרה זה, הקטע מכיל שורה אחת בלבד. אם המשמש בוחר באפשרות MySquarer, Square it את הערך שבתיבת הטקסט他自己, ומציג אותה בתיבת הודעה.

אם המשמש בוחר באפשרות Cube It, בטרם לחץ על לחצן הפוקודה, הקוד לא יבצע פעולה כלשהי. קבוצת האפשרויות מחזירה את הערך 2, אך הקוד אינו כולל תנאי ליזהו מצב זה. כאמור של דבר, הקוד אינו מכיר קבוצת אפשרויות אחרת מ-1. הקוד הבא מציג גירסה חדשה של שגרת אירוע ייחד עם הקוד שמתפל במצב בו המשמש אינו בוחר באפשרות Square it.

```

Sub cmdComputer_Click()
    If opgComputeType = 1 Then
        MySquerer txtInput.Value
    Else
        MyCuber txtInput.Value
    End If
End Sub

Sub MyCuber(MyOtherNumber As Double)
Dim dblResult As Double
    dblResult = MyOtherNumber ^ 3
    MsgBox dblResult, vbInformation, "Programming Microsoft Access 2000"
End Sub

```

השיגרה cmdComputer_Click מונצלת את התבנית השנייה של משפט If...Then. היא קוראת לפונקציה MySquerer אם opgComputeType שווה ל-1, אך בכל מקרה אחר היא קוראת ל-MyCuber. הקוד פועל בצורה תקינה כשהמשתמש בוחר באפשרות Cube. It. לעומת זאת, נותרה לפחות בעיה אחת: הטופס נפתח מבלי שאפשרות פעולה כלשהי נבחרה מראש.

אם המשתמש מקליד ערך בתיבת ולאחר מכן לחוץ על לחץ הפקודה, הטופס יחזיר את החזקה השלישי של הערך על אף העובדה שהמשתמש לא בחר באפשרויות. הבעיה נועוצה בתכנון המשפט If...Then. אנו זוקקים לויריאציה השלישי עם שני תנאים נפרדים: אחד להעלאה בריבוע והآخر להעלאה בחזקת שלוש. הקוד שלפניך מדגים זאת:

```

Sub cmdComputer_Click()
    If opgComputeType = 1 Then
        MySquerer txtInput.Value
    Else if opgComputeType = 2 Then
        MyCuber txtInput.Value
    Else
        msgbox "Please select computing option"
    End If
End Sub

```

דוגמה זו כוללת שני תנאים. האחד בודק אם קבועת האפשרות שווה ל-1, והאחר בודק אם היא שווה ל-2. אם תוצאות הבדיקה אינם עונות על אף אחד מהתנאים, השיגרה תציג תיבת הودעה ובה הוראה למשתמש לבחור את סוג החישוב. קל ופשוט יהיה עתה להוסיף לחצני אפשרות חדשים קבועת האפשרות ועל ידי כך להגדיל את מגוון החישובים. כל שעליך לעשות הוא להוסיף משפט ElseIf חדש ובו תנאי מיוחד עבור כל לחץ שהוספה.

כפי שתוכל לראות, המשפט If...Then מצטיין בغمישותו. באפשרותך לנצל אותו לטיפול באפשרויות בחירה רבות, אך הוא מטפל בפריט אחד או שניים ביותר הצלחה. תחבר המשפט אף הוא משתנה קמעה בהתאם למשימה שברצונך לבצע.

Select Case

המשפט Select Case מבצע פעולה אחת: הערכת ביטוי וביצוע מותנה של קטע קוד אחד. כשמטפליים ביותר אפשרות מותנית אחת או שתיים, כל יותר להגדר ולתזוז Select Case לעומת אוסף משפטי If...Then. התחבר הכללי של Select Case הוא:

```
Select Case test expression
  Case expression list-1
    ' Statements
  Case expression list-2
    ' Statements
  Case Else
    ' Statements
End Select
```

משפט Select Case מעיריך ביטוי בדיקה התחלתי. הביטוי יכול להיות פשוט כמו פרמטר שהועבר, או מורכב כמו ביטוי המשמש בהנדסת טילים. ו- expression list-2 הם טווחים (או ערכיהם מוגדרים) עבור ביטוי הבדיקה. הגדרת רשימות ביטויים יכולה לכלול, קבועים, סדרת פריטים מופרדים, או ביטויי אמת או שקר. כאשר תוצאת רשימת ביטויים היא אמת מכיוון שהיא תואמת לביטוי הבדיקה, יבוצע קטע משפטיים מתאימים. דוגמת הת לחבר שהוצאה מציגת שתי רשימות בלבד, אך תוכל להגדילה על ידי שימוש Case נוספים. משפט Case Else הוא אופציוני, אך הוספנו תורמת לעמידה בכללי התקנים. אפשרות זו לוודת ערכי ביטוי בדיקה שלא נלכדו על ידי משפטי Case קודמים. ניתןukan משפטי Case זה בזוז; כל מופיע חייב להתחיל במשפט Select Case ולהסתティם במשפט Case .End Case

תרשים 1.16 מציג המשפט Select Case המשמש בשגרת פונקציה. שיגרה זו מטפלת בזוג מספרים שהועברו אליה באמצעות אחת מאربע פעולות מספוריות. ארגומנט מחרוזת מצינו את הפעולה, והחלו **Immediate** (מיידי) מציג את תוצאת הפעולה שנבחרה. כפי שניתן לראות בחלוון המיידי, הבדיקות אין רגישות לאוთיות רישיות.

פעולה שאויתה בצורה שגوية גורמת לתוכנית הגיעו למשפט Case Else. במקרה זה, הפונקציה תדפיס הודעה בחלוון המיידי, בה ייאמר כי הפעולה שנבחרה אינה חוקית.

The screenshot shows the Microsoft Visual Basic Editor interface. The top window is titled 'Chapter1 - modTrialSamples [Code]' and contains the following VBA code:

```

Public Function Computer0(db1Number1, db1Number2, Operation As String)
    Select Case Operation
        Case "Addition"
            Computer0 = db1Number1 + db1Number2
        Case "Subtraction"
            Computer0 = db1Number1 - db1Number2
        Case "Multiplication"
            Computer0 = db1Number1 * db1Number2
        Case "Division"
            Computer0 = db1Number1 / db1Number2
        Case Else
            Debug.Print "Operation is not valid."
    End Select
End Function

```

Below it is the 'Immediate' window, which displays the results of running the function with various operations:

```

?computer0 (1,?computer0 (1,2,"addition")
3
?computer0 (1,2,"subtraction")
-1
?computer0 (1,2,"multiplication")
2
?computer0 (1,2,"division")
0.5
?computer0 (1,2,"additon")
Operation is not valid.

```

תרשים 1.16: שגרת פונקציה והפלט שלה ממחישים את השימוש במשפט Select Case

For...Next

המשפט For...Next משמש בעיקר לביצוע קטע קוד בצורה חוזרת (לולאה), מספר פעמים ידוע. ניתן לצאת מתוך הלולאה באמצעות המפט Exit For, ולכון ניתן לנצל את המשפט For...Next כשלא יודעים בדיקם כמה פעמים יש לבצע את הקוד. אך יש לציין את מספר הlolאות המקסימלי. התחביר הכללי של משפט Next הוא :

```

For counter = start To stop Step step
    ' Statements
    If condition Then
        Exit For
    End If
    ' Statements
Next counter

```

lolאת For מתחילה בשורה For counter ומסתיימת בשורה Next counter. המשפט For...Next מתחילה את counter לערך start של counter ולאחר מכן מבצע את המשפטים שבין For ו-Next. מהלך הביצוע חוזר תמיד בראש counter, כש-step גדל בערך האופציונלי של step (ערך ברירת המחדל של step הוא 1).

בדוגמה, משפטי לולאת For מבוצעים פעם אחר פעם, עד שערך של counter עבר את זה של stop (או כאשר counter קטן מ-step שלילי). בשלב זה השיטה עבורת למשפט הבא מייד לאחר Next counter. המשפט האופציונלי Exit For גורם לולאה להסתיים לפניו של stop עבר את הערך של stop.

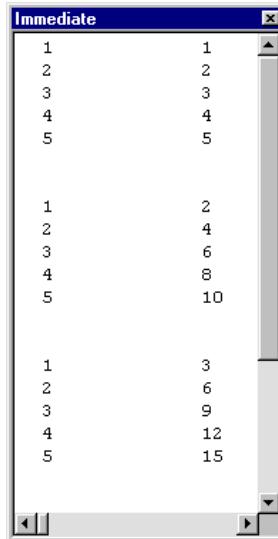
ניתן לנו משפטי Next...For זה בזה. לולאה פנימית מעבירה את השיטה לולאה חיצונית שהערך של counter של stop עבר את זה של stop. VBA מסוגל לגרום לשגיאת זמן ריצה, אם הוא נתקל במשפט Next לא בן זוגו For, אך סביר יותר שלפני כן תקבל שגיאת תחביר.

השגרה הבאה מיישמת את הלוגיקה של לולאת For...Next. וボ-זמןן מבססת את טכניקות העבודה עם מערכים והצורות Static (המערך aryMyArray כולל חמישה אלמנטים. כזכור, אינדקס של מערך מתחילה ב-0, אלא אם כן הקוד מצין אחרת במפורש). הקוד מכיל הצהרה על משתנה מסוג Integer שיישמש כמונה. המשפט CountFor מציב באופן עוקב את הערכים 1-5 במשתנה המונה, intIndex.

שני משפטיים מותבצעים לכל מעבר בלולאה. ראשית, ערך אלמנט במערך aryMyArray מחליף את הערך הנוכחי של intIndex לערך הנוכחי שלו. הקוד מצהיר על aryMyArrayStatic, ולכן האלמנטים של המערך שומרים על ערכיהם גם באמצעות משפט intIndex. לאחר הביצוע הראשון של השגרה, האלמנטים במערך זהים לערכים המקוריים של intIndex. לאחר הביצוע הבא, ערכי האלמנטים גדולים פי שניים מלהلة של intIndex וכן הלאה בכל ביצוע מוצלח של השגרה. המשפט השני מדפיס את הערך הנוכחי של intIndex והאלמנט הקשור אליו מתוך המערך. וכך כי אפשרותך לאותה מחדש את ערכי האלמנטים במערך ל-0, על ידי בחירת האפשרות **Run** מתוך **Reset** ב-VBE.

```
Sub CountFor()
Static aryMyArray(4) As Integer
Dim intIndex As Integer
    For intIndex = 1 To 5
        aryMyArray(intIndex - 1) = _
            aryMyArray(intIndex - 1) + intIndex
        Debug.Print intIndex, aryMyArray(intIndex - 1)
    Next intIndex
    Debug.Print vbLf
End Sub
```

תרשים 1.17 מציג את פלט החילון **Immediate** בשלושה ביצועים עוקבים של השגרה. העמודה הראשונה מציגה את הערכים המקוריים של intIndex, והשניה מציגה את הערכים המתאימים של האלמנטים במערך. בביטוי השגרה הראשון, האלמנטים של המערך תואמים בבדיקה את אלה של intIndex. בביטויים השני והשלישי, האלמנטים גדולים פי שניים ופי שלושה בהתאמה. תוכנת הסכימה ממחישה את השפעת ההצהרה על משתנה כ-Static. אם תנסה את מילת המפתח ל-Dim, ערכי intIndex וערכי aryMyArray יתאיימו בבדיקה אלה לאלה בכל ביצועי השגרה.



תרשים 1.17: החalon **Immediate** מציג את תוצאות ביצוע השירה CountFor שלוש פעמים רצופות

With...End With - ו- For Each...Next

המשפטים Next...For Each...Next ו- End With...With יכולים להשתלב יחד היטב בפעולות אימיות נתוניות ומשימות ספירה מקובלות. For Each...Next מבצע הילך חזרה דרך אוסף כלשהו (כגון פקידים בטופס) או מערך. אין דרישת לדעת את מספר האלמנטים באוסף או במערך. המשפט Next...For Each...Next יכול להשלים את End With על ידי פישוט אופן הקידוד של פקדים רבים בקטע קוד.

המשפט Next...For Each...Next דומה וגם מורכב פחות מהמשפט For...Next. הדמיון בין השניים נובע מכך שבשנייהם הלולאה פותחת בשורה For ומסתיימת בשורה Next. יחד עם זאת, עיצוב השורות For ו-Next שונה For...Next שונה בשתי גרסאות לולאת For. המשפט Next...For Each...Next פשוט יותר מהמשפט For...Next...Next, מכיוון שהוא פוטר אותו מה צורך לעקוב אחר שלושה פרמטרים נפרדים (start, stop, counter) או מההתייחסות לערך חובי או שלילי של step. המשפט תמיד תמיד בתחלת האוסף או המערך, ומתרבע בלולאה עד הגיעו לאלמנט האחרון. תחביר המשפט נראה כך:

```
For Each element In group
    ' Statements
    If condition Then
        Exit For
    End If
    ' Statements
Next element
```

המונה group بشורה הראשונה של המשפט For Each...Next מתיחס לשם האוסף או המערך. element בשורה הראשונה ובשורה האחורה מצין אובייקטים בודדים באוסף או אלמנטים במערך.

המשפט For Each...Next מבצע בצורה חוזרת את משפטי גוף הקוד על כל אלמנט באוסף או במערך שעלייהם הוא פועל. לעיתים תרצה לשלב את המשפט Exit For או משפט אחר שביצוע מותנה, במקומות כלשהו בגוף הלולאה For. כך תאפשר לקוד להגיב בצורה דינמית לאיורע מיוחד בסביבת המחשב. בדיקת התנאי מזזה את האירוע המיעוד, ו- Exit For או משפטיים אחרים שביצועם מותנה, יהולו רק בכפוף להתרחשויות האירועים.

בדומה למשפט Next...For, ניתןukan משפטי For Each...Next זה זהה. בסיום לולאת For Each...Next, השיליטה עוברת למשפט הרASON שבא לאחר הלולאה.

המשפט With...End With מפשט את ההתייחסות לשיטות או מאפיינים שונים של אותו אובייקט. בתחלת השורה With ציין את האובייקט שברצונך להתייחס למאפייניו או שיטותיו, וסגור את הפניה לאובייקט בשורה End With שבסוף הקטע. בין השורות ו- End With אפשר לגשת למאפיינים או לשיטות האובייקט, מוביל לצין את שלו. :

With...End With

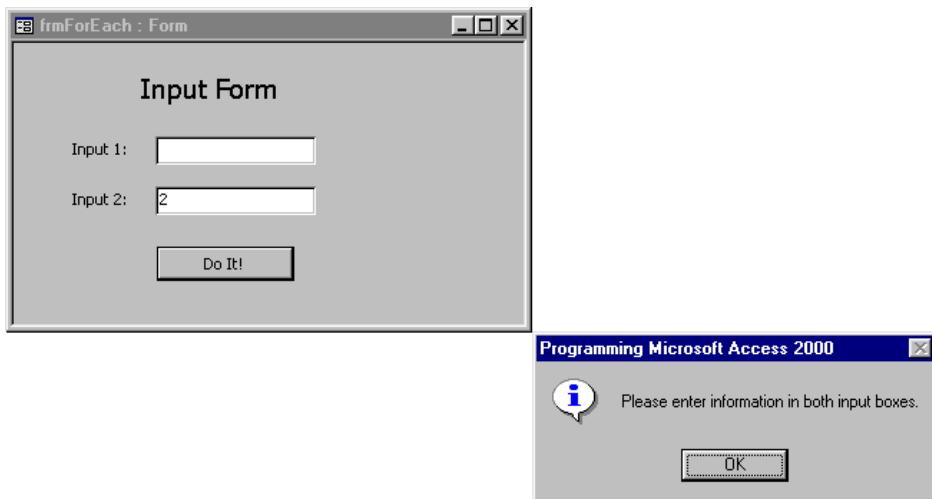
```
With object
    .propertynname1 = "new value1"
    .propertynname2 = "new value 2"
    .method1
    .method2
End With
```

object הוא שם האובייקט, הפניה לאובייקט או שם מערך.(propertynname1 ו- propertynname2 הם מאפייני האובייקט ו- method2 ו- method1 ו- method1 ו- method2). כדי שניתן לראות, המשפט With...End With מאפשר לגשת למאפיינים ולשיטות האובייקט.

תרשים 1.18 ו- 1.19 מציגים טופס אימות נתונים בפועלה. למרות שהטופס מכיל רק זוג תיבות טקסט הדורשות אימות, קוד הטופס מנצל לולאת For Each...Next שנitin להרחביה כך שתכיל פקד תיבת טקסט נוספים. באפשרות לעורך שינוי כל כולל סוגים פקדים נוספים בשגרת האימות. תרשים 1.18 מציג עיצוב של טופס קליטה בסיסי הכלול זוג תיבות טקסט ולחצן פקודה.

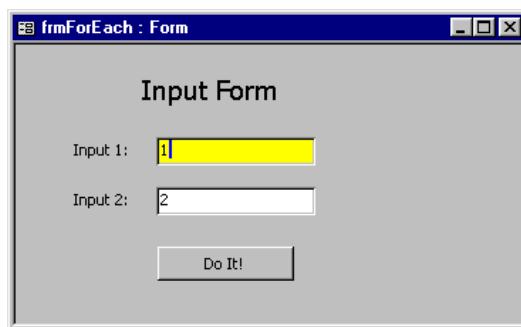
לחיצה על לחץ הפוקודה! So, מפעילה שגרת אירוע שבודקת את תיבות הטקסט כדי להבטיח שאין מכילות ערכי Null. אם פקד כלשהו מכיל ערך Null, שגרת האירוע תציג תיבת הוועדה שתזכיר למשתמש להזין נתונים בשתי תיבות הטקסט.

שגרת האירוע קוראת גם לשיגרה שמשנה את הרקע של כל תיבת טקסט שמכילה ערך Null מלבד צהוב. השיגרה גם מעבירה את המיקוד לטיבת הטקסט האחורה שמכילה ערך Null. הרקע נשאר צהוב, עד שהמשתמש מעדכן את הנתונים בתיבת הטקסט.

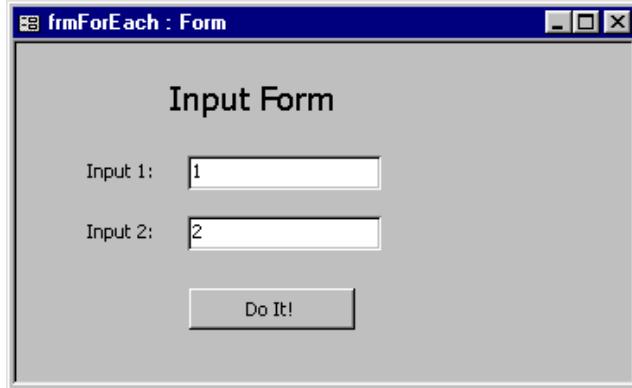


תרשים 1.18: תוצאה של החיצה על להזין הפקודה `Do It!` כאשר תיבת טקסט אחת לפחות מכילה ערך `Null`

תרשים 1.19 מציג תיבת טקסט שהכילה קודם ערך `Null`, אך כתעת מכילה 1. צבע הרקע של תיבת הטקסט משתנה חזרה לבן ברגע שהמשתמש מזין בה נתונים כלשהם ומעביר את המיקוד מתיבת הטקסט למקום אחר. תרשים 1.20 מציג את הטופס לאחר שהערך החדש מעדכן את תוכן תיבת הטקסט.



תרשים 1.19: תיבת טקסט מסומנת בצהוב שהכילה קודם لكن ערך `Null`, ועכשו מכילה 1 זוג השגרות הבאות – `MarkFieldsToEdit`-`cmdSubmit_Click` – בוחן את תיבות הטקסט ומAIR בצהוב תיבת טקסט כלשהו שמכילה `Null`. שגרת האירוע `Click` של להזין הפקודה עברת בלולאה על כל פקדי הטופס – בין שהם פקד תיבות טקסט וכאליה אינם. שגרת האירוע מנצלת את מילת המפתח `TypeOf` כדי לזהות פקדי תיבת טקסט בין הפקדים. אם לא תעשה זאת, תיגרם שגיאת זמן ריצחה, מכיוון שלא לכל הפקדים יש `Value`. אם שגרת האירוע מזזה פקד תיבת רשימה, היא משתמשת את המאפיין `Value` של הפקד כדי לקבוע אם הוא מכיל ערך `Null`. פקד שמכיל ערך `Null` יוזם את הפעלת הקוד הכלול במשפט `If...Then`. קוד זה מציג את תיבת ההודעה וקורא לשיגרה להאייר את הפקד שחרר לו ערך.



תרשים 1.20: לאחר שהמשתמש מזין מידע ומעביר את המיקוד, צבע הרקע של תיבת הtekסט חוזר להיות לבן

```
Private Sub cmdSubmit_Click()
' Check for valid entries.

For Each ctl In Screen.ActiveForm.Controls
    If TypeOf ctl Is TextBox Then
        If IsNull(ctl.Value) Then
            MsgBox "Please enter information in both input boxes.", _
                vbInformation, "Programming Microsoft Access 2000"
            MarkFieldsToEdit
            Exit For
        End If
    End If
    Next ctl

End Sub

Public Sub MarkFieldsToEdit()
    For Each ctl In Screen.ActiveForm.Controls
        If TypeOf ctl Is TextBox Then
            If IsNull(ctl.Value) Then
                With ctl
                    .BackColor = RGB(255, 255, 0)
                    .SetFocus
                End With
            End If
        End If
    Next ctl
End Sub
```

גם השגירה MarkFieldsToDelete מונצלת את מילת המפתח TypeOf כדי לזרות תיבות טקסט. כשהיא מזזה תיבת טקסט שמכילה ערך Null, היא משתמש במשפט With...End With כדי לשנות את צבע הרקע של הפקד ומעבירה את המיקוד אל הפקד. על ידי כך היא מבטיחה שתיבת הטקסט האחרונה שמכילה ערך Null תהיה בנקודתBT בatoms השגירה.

כל אחת משגרות האירוע בקוד שלפניך מופעלת בתגובה לאירוע AfterUpdate (לאחר עדכון). כל שגירה מונצלת קטע מסוג With...End With כדי לשנות את צבע הרקע של הפקד המטופל לבן, אם צבע הרקע הנוכחי שלו צהוב. האירוע AfterUpdate מתרחש באופן עצמאי עבור שתי תיבות הטקסט, אך הקוד שבכל אחת מהשגרות זהה, למעט שם האובייקט המטופל (txtInput1 או txtInput2).

```
Private Sub txtInput1_AfterUpdate()
    With txtInput1
        If .BackColor = RGB(255, 255, 0) Then
            .BackColor = RGB(255, 255, 255)
        End If
    End With
End Sub

Private Sub txtInput2_AfterUpdate()
    With txtInput2
        If .BackColor = RGB(255, 255, 0) Then
            .BackColor = RGB(255, 255, 255)
        End If
    End With
End Sub
```

Do...Loop

המשפט Do...Loop הוא סוג נוסף של משפט לולאה של יישומי VBA. משפט זה מהווה חלופה גמישה יותר של משפט הלולאה While...Wend – הוא כולל את כל הרכיבים התפקידיים של While...Wend ומוסיף עליהם. VBA ממשיך לתמוך במשפט הלולאה While...Wend משיקולי תאימות לגרסאות קודמות.

באפשרותך לנצל את לולאת Do כדי לבצע בצורה חזרת קבוצת משפטי עד שתתנאי מוגדר יתמלא (כโลמר, קיבל ערך True או False). לחבר המשפט Do...Loop תומך במפורש בביטוי בדיקת התנאי טרם ביצוע קטע קוד או מיד לאחר מכן. בדומה למשפט לולאה אחרים, קיימים גם משפטי מיוחד מיוחד ליציאה מקוד הלולאה במהלך הביצוע. לפניו שתי גרסאות לחבר המשפט Do...Loop.

```

Do {While | Until} condition
  ' Statements

  If condition Then
    Exit Do
  End If

  ' Statements

Loop

```

הדוגמה השנייה :

```

Do
  ' Statements

  If condition Then
    Exit Do
  End If

  ' Statements

Loop {While | Until} condition

```

הגירסה הראשונה בודקת את התנאי טרם ביצוע משפט הולאה, והשנייה עושה זאת לאחר מכן. בדיקת התנאי יכולה לנצל את מילת המפתח While, או את מילת המפתח Until. מילת המפתח While גורמת לולאה לשוב ולהתבצע כל עוד התנאי מתקיים (אמת); מילת המפתח Until גורמת לולאה להתבצע כל עוד התנאי אינו מתקיים (שקר). מפתחים מנוסים בסביבת Access כתבו בוודאי לולאות So רבות לניות בראשומות בערך רשות עד זיהוי הסימן EOF (סוף קובץ) או BOF (התחלה קובץ). כמו המשפטים לולאה אחרים של VBA, גם כאן ניתן לצאת מlolata So לפני תום המחוור על ידי שימוש במשפט Exit Do.

תרשים 1.21 מציג זוג שגורות פונקציה ואות הפלט שלhn, כדי להמחיש תכונות לולאות So. הפונקציה DaysToNextMonth קולטת תאריך ומחזירה את מספר הימים מאותו תאריך עד ליום הראשון בחודש הבא אחריו. אם תקליד את התאריך הנוכחי, השגירה תמנה אותו כאחד הימים בתוך מספר הימים הכלול עד החודש הבא. כך, לדוגמה, הפרש הימים בין התאריך 28/2/2000 ל-28/2/2000 (29/2/2000-1). הפונקציה פותחת בחישוב dtmnNextMonth(28/2/2000-1), שהיא תאריך הקלט בתוספת חודש אחד (בדוגמה שבתרשים 1.21, התאריך הוא 03/2000). אחר כך השגירה משתמשת בשם הפונקציה כМОנה שערכו התחלה הוא 0. השורה השלישי מפעילה לולאת Do שהתנאי שלה בודק את חוסר השווון בין החודש הבא לבין החודש המתkeletal מתאריך הקלט בתוספת המונה. כל עוד אין שוויון, הלולאה מבצעת את קטע הקוד שכולל משפט אחד – הגדלת ערך הפונקציה ב-1.

The screenshot shows the Microsoft Visual Basic Editor interface. The top window is titled "Chapter1 - modTrialSamples (Code)" and contains two functions: "DaysToNextMonth" and "DaysFromFirstOfLastMonth". The "DaysToNextMonth" function calculates the number of days from the input date to the first day of the next month, including the current date. The "DaysFromFirstOfLastMonth" function calculates the number of days from the input date to the first day of the previous month, excluding the current date. The bottom window is titled "Immediate" and shows the results of running these functions with the date #2/28/2000#.

```

Public Function DaysToNextMonth(dtmDateIn)
' Including current date
    dtmNextMonth = DateAdd("m", 1, dtmDateIn)
    DaysToNextMonth = 0
    Do While Month(dtmNextMonth) <> Month(dtmDateIn + DaysToNextMonth)
        DaysToNextMonth = DaysToNextMonth + 1
    Loop
End Function
Public Function DaysFromFirstOfLastMonth(dtmDateIn)
' Not including current date
    dtmPreviousMonth = DateAdd("m", -1, dtmDateIn)
    dtmFirstPreviousMonth =
        DateValue(Month(dtmPreviousMonth) & "/1/" & Year(dtmPreviousMonth))
    DaysFromFirstOfLastMonth = 0
    Do
        DaysFromFirstOfLastMonth = DaysFromFirstOfLastMonth + 1
    Loop While dtmFirstPreviousMonth <> (dtmDateIn - DaysFromFirstOfLastMonth)
End Function

```

```

?DaysToNextMonth(#2/28/2000#)
2
?DaysFromFirstOfLastMonth(#2/28/2000#)
58

```

תרשים 1.21: זוג שגרות פונקציה שימושת בבדיקות תנאי בתחילת ובסיום לולאות Do

הfonקציה DaysFromFirstOfLastMonth מראה את השפעת בדיקת התנאי שמתבצעת בסיום לולאת Do. הפלט הוא מספר הימים בין 28/2/2000 לבין היום הראשון של החודש הקודם (1/1/2000), שהוא 58 (28 ימי פברואר ו-30 הימים האחרונים בינוואר). פונקציה זו מתחילה בחישוב החודש הקודם עד היום שצווין. לאחר מכן היא מעבירה חודש זה אל הפונקציה DateValue כדי לקבוע את היום הראשון בחודש הקודם. לפני תחילת לולאת Do, הקוד מספק ל-DaysFromFirstOfLastMonth ערך 0. הכניסה ללולאה אינה מותנית בבדיקה כלשהי, ורק-DaysFromFirstOfLastMonth מוגדלת ב-1. בדיקת התנאי בשורה Loop מאפשרת ביצוע מחזור לפחות נסף, כל עוד הפרש בין תאריך הקלט לבין DaysFromFirstOfLastMonth קטן מהיום הראשון לחודש הקודם. באופן זה הפונקציה מונה 58 ימים מתאריך הקלט עד, אך לא כולל, היום הראשון של החודש הקודם.

פונקציות מוכללות

חלק גדול מהדוגמאות שהציגנו כולל ספרייה עשירה של פונקציות מוכללות של Access. הפונקציות יכולות לסייע לזרז תהליכי פיתוח של פתרונות מותאמים אישית על ידי מצומם ופישוט הקוד המותאם אישית. באפשרות לנצל את הפונקציות המוכללות בדיקות כדי שאתא מנגנון את הפונקציות שאתה כותב במוחך בהזדמנויות רבות, כגון שאלות, טפסים ודוחות. אם תלמד להכיר את השימושים האפשריים

בפונקציות מוכללות, תוכל לגלוות אופקים חדשים ויצירתיים ליישום פונקציות מסוימות אישית שתכתחוב בעצמך.

סעיף זה כולל שלוש דוגמאות המשמשות בפונקציות מוכללות: אחת לשאלתה, אחת לטופס ואחת לדוח. באפשרות (ויתכן שתעשה זאת) למזוג את כל שלוש הgiooth ביישומיך. הדוגמה השלישיית מראה כיצד לגרום לפקד בזוזה לתפקיד בטור מקור עבור קוד VBA במודול של דוח על ידי שימוש בפונקציות מוכללות. כל שלוש הדוגמאות מבוססות על טבלה קטנה המכילה נתונים להפקת חשבונות, ודוגמאות דרכי דיווח על חשבונות שמועד פרעוןן חלף.

תרשים 1.22 מציג את הטבלה המקורי InvoiceDates בצד שמאל, ובצד ימין לשאלתה ובה תוצאה של הטבלה. השאלתה בוחרת שתיים מתוך עשר שורות הטבלה (התרשימים Past Due Days ומהשנתה עמודה חדשה שנקראת InvoiceDates).(12/9/98

	InvoicelD	InvoiceDate
	1	06/15/1998
	2	07/01/1998
▶	3	08/17/1998
	4	08/18/1998
	5	08/20/1998
	6	09/01/1998
	7	09/08/1998
	8	09/10/1998
	9	09/12/1998
	10	09/14/1998

	InvoicelD	InvoiceDate	Past Due Days
	1	06/15/1998	59
▶	2	07/01/1998	43
	(AutoNumber)		

תרשים 1.22: הטבלה InvoiceDates ותוכנת שאלתה שבודקת שורות ומחשבת שנות חדשות באמצעות פונקציות מוכללות

באפשרות למצוא את היום הנוכחי באמצעות פונקציה מוכללת אחת או יותר: הפונקציה Now והפונקציה Date (ערך החזרה של Now כולל גם את השעה הנוכחיית). אם ברצונך לטפל בתאריכים בלבד, אין זה משנה באיזו פונקציה תשתמש).

השאלתה כוללת את שתי העמודות של הטבלה InvoiceDates. הקритריון של העמודה InvoiceDates הוא <Date>-30. התאריך הנוכחי בזמן ביצוע השאלתה היה 12/9/98, ולכן מועד פירעונה של חשבונית מתאריך כלשהו שקדם לו 13/8/98. מכאן שהשאלתה תחזיר חשבונות מהתאריכים 1/7/98 ו- 15/6/98.

ערכים Past Due Days ראויים לעמודה נפרדת. שדה מחושב זה משתמש בזוג פונקציות מוכללות: DateDiff ו-Now. הפונקציה DateDiff אידיאלית לחישוב ההפרש בין שני תאריכים. באפשרות לציין את ההפרש בימים, חודשים, רביעונים, שנים ויחידות זמן שונות נוספת. במקרה של העמודה Past Due Days, הפונקציה DateDiff מחזירה את ההפרש בימים. DatesDiff("d", [InvoiceDate]+30, Now) הוא הביטוי שמחשב את ערך השדה. ביטוי זהקובע את התאריך הנוכחי באמצעות הפונקציה Now (למרות

שהפונקציה Date יכולה לעשות זאת בצורה זהה). המונח [InvoiceDate] מפנה לטליה המשמשת כמקור הרשותות עבור השאלה. התוספת 30+ משמשת לציין תאריך התשלום האחרון המותר, מלבדו שהדבר יחווש כאיתור.

השאילתת יכולה להיות נקודת פטיחה להפקת שימושים עסקיים מעשיים, אך יש לה כמה חסרונות. אחד החסرونויות הבולטים הוא שתצוגת גילוון הנתונים אינה מציגה את התאריך הנוכחי, מה שulton לעורר ספק באשר לבסיס החישוב של חשבוניות שמועד פירעונים חלף. אובייקט הטופס frmPastDueEvaluator שבתרשים 1.23 פותר בעיה זו על ידי הצגת הנתונים בתצוגת הטופס הידידותית. בתצוגה זו, שלושת שדות הטופס מספקים את המידע החשוב ביותר. השדה InvoiceDate מציג את יום הכתנת החשבוניות. השדה הבא אחראי מציג את התאריך הנוכחי. השדה הבא מחשב את מספר הימים שחלפו ממועד שנקבע לפירעון החשבונית. שדה זה יכול את הערך 0, אם החשבונית נוצרה לפני פחות מ-30 יום.

InvoiceID:	1
Invoice Date:	06/15/1998
Today's Date:	09/12/1998
Days late:	59

Record: 1 of 10

תרשים 1.23: טופס לחישוב מספר הימים שחלפו ממועד הפירעון של חשבוניות

שני השדות האחרונים בטופס משתמשים בפונקציות מוכללות לחישוב ערכים. השדה Today's Date (תאריך נכון) משתמש בפונקציה Date. במקרה זה, עדיפה על Now, מכיוון Sh-Sow מציגה את התאריך ו גם את השעה, אם אין מעכבים את השדה אחרת. אם תיתן למאפיין Control Source (מקור פקד) של השדה את הערך (=Date(), Immediate If IIF)). אין צורך לעצבו במיוחד. השדה Days Late מנצל פונקציה מסווג IIF (If).

הפונקציה משמשת כתבנית ביטויים מובנים ומוכרת גם למפתחי נליונות אלקטרוניים. לדוגמה הביטוי שבסדה האחרון בטופס הוא :

`=IIF([txtInvoiceDate]>30,[Date()-[txtInvoiceDate]]<=30,0)`. הביטוי כולל שלושה ארגומנטים, בדומה לפונקציית If של הגילוון האלקטרוני. הארגומנט הראשון הוא תנאי שיכול לקבל ערך True או False. אם התנאי אינו מתקיים (שקר), פונקציית IIF מחזירה את ערך הארגומנט השני; אם התנאי אינו מתקיים (אמת), פונקציית IIF מחזירה את ערך הארגומנט השלישי. מכאן שאם התאריך שבתבנית הטקסט של תאריך החשבונית, txtInvoiceDate, מוקדם ב-30 יום או יותר מהתאריך הנוכחי, פונקציית IIF תחזיר את מספר הימים שמעבר ל-30 ימי האורכה; אחרת, פונקציית IIF תחזיר את הערך 0.

הדוח שבתרשים 1.24 מציג גישה נוספת לסטיקום חשבונות שמועד פירעון עבר. הדוח מכיל שלוש עמודות. העמודה האחורונה מחשבת את מספר הימים שהלפו ממועד פירעון החשבונית. בכל שורה נתונה של חשבונית, עמודה זו תהיה ריקה אם מועד פירעון החשבונית לא חלף. כוורת הדוח מזכירה למשתמש את תאריך הדוח, וכן ניתן לדעת את התאריך שמנדר את החשבונות שמועד פירעון חלף.

The screenshot shows a Microsoft Access report window titled "InvoiceDates". The report title is "30 Day Past Due Report". It displays a table with three columns: "InvoiceID", "Invoice Date", and "Past Due Days". The data shows seven invoices, all of which are past due by more than 30 days. The "Past Due Days" column contains the values 59, 43, 08/17/1998, 08/18/1998, 08/20/1998, 09/01/1998, and 09/08/1998 respectively. The report includes a "Today's Date is:" label and a date field showing "09/12/1998". At the bottom, there is a page navigation bar with buttons for first, previous, next, last, and last pages, and a current page indicator showing "Page: 1".

InvoiceID	Invoice Date	Past Due Days
1	06/15/1998	59
2	07/01/1998	43
3	08/17/1998	
4	08/18/1998	
5	08/20/1998	
6	09/01/1998	
7	09/08/1998	

תרשים 1.24: דוח שמציג בצורה מותנית את מספר הימים שהלפו ממועד פירעון חשבונית. אם החשבונית נכתבת 30 ימים או פחות ממועד הנוכחי, העמודה האחורונה תהיה ריקה

ערך עמודת הדוח האחורונה מחושב באמצעות נוסחה. תוכל להתביס על שתי הדוגמאות הקודמות כדי לכתוב את הנוסחה. ייחודה של הדוח הוא בכך שהעמודה האחורונה תהיה לפעמים ריקה. שגרת האירוע OnFormat (בעת עיצוב) של המקטע פירוט (Detail) של הטופס מאפשרת זאת על ידי הגדרה מותנית, בזמן ריצת, של המאפיין Visible (גלווי) של פקד העמודה האחורונה. לפני כן שגרת האירוע.

```

Private Sub Detail_Format(Cancel As Integer, FormatCount As Integer)
Dim ctlID As Control

For Each ctlID In Me.Detail.Controls

    If ctlID.Name = "txtPastDueDays" Then
        If ctlID.Value >= 0 Then
            ctlID.Visible = True
        Else
            ctlID.Visible = False
        End If
    End If

    Next ctlID

End Sub

```

שגרת האירוע מנצלת לולאת For Each...Next כדי לעבור בלאה על הפקדים של כל רשומה בחיפוש אחר תיבת הטקסט txtPastDueDays. תיבת טקסט זו מכילה את מספר הימים שהלפו מאז מועד פירעון החשבונית. אם הערך חיובי, מועד הפירעון חלף, והקוד יגדיר את המאפיין Visible של הפקד כאמת. אחרת, מועד פירעון החשבונית לא חלף והמאפיין Visible של הפקד יקבל ערך שקר. אם כן, מספר מצומצם של שורות קוד יכול ליצור עיצוב דוח גמייש בזמן ריצה.

תקציר פונקציות נבחרות

Access 2000 מכיל מעל 170 פונקציות מוכללות המבצעות מגוון משימות רחבות, לרבות המרה בין סוגי נתונים, עיבוד תאריך/שעה, ניתוח סטטיסטי של נתונים טבלאות ושאילות, פעולות מתמטיות וטיפול בטקסט. תוכל להרחב את ידיותך אודוות פונקציות אם תלמד להכיר מה ניתן לעשות באמצעותן. Microsoft יקרה פונקציות רבות, ולכן אין צורך כתוב אותן מחדש. לאחר מכן תוכל לעמוד על הדרכן שהפונקציות משלימות זו את זו. באפשרות גם ללמוד על תכונות הערכיכים המוחזרים. לדוגמה, הפונקציה Hex מחזירה מחזורת שמיצגת את השוקול הhexadecimal של מספר עשרוני, והפונקציה Oct מבצעת המרה דומה על מספרים אוקטליים. פلت שתי הפונקציות מתאים לשדרור ולהשוואת מחזוריות, אך לא לפעולות חשבוניות.

באפשרות גם לחקור את השפעת ארגומנט אופציוני של פונקציה על ערך ההחזרה של הפונקציה. אי-הבנה של ארגומנטים אלה עלולה לגרום לתוצאות מוטעות או מבלבלות. לדוגמה, הפונקציה strComp משווה שתי מחזוריות זו לזו. שני הארגומנטים הראשונים הם המחרוזות המיועדות להשוואה. הארגומנט השלישי הוא אופציוני ומצין את סוג ההשוואה שיש לבצע. באפשרות לציין השוואת תלויות-רישיות, השוואת שאינה תלואה ברישיות או לחילוף השוואת המבוססת על האפשרות סדר מיון מס' נתונים חדש (New Database Sort Order) בכרטיסיה **כללי** (General) של תיבת הדוח-שים אפשרויות (Options) של Access 2000. אם לא תציין את ההשוואה

האופציונלית, הפונקציה תנקוט את שיטת ההשוואה המצוינת במשפט Option Compare באזור החרזרות (declaration) של המודול (אם לא צוינה שיטה כלשהי, Access יבצע השוואה בגיןירות).

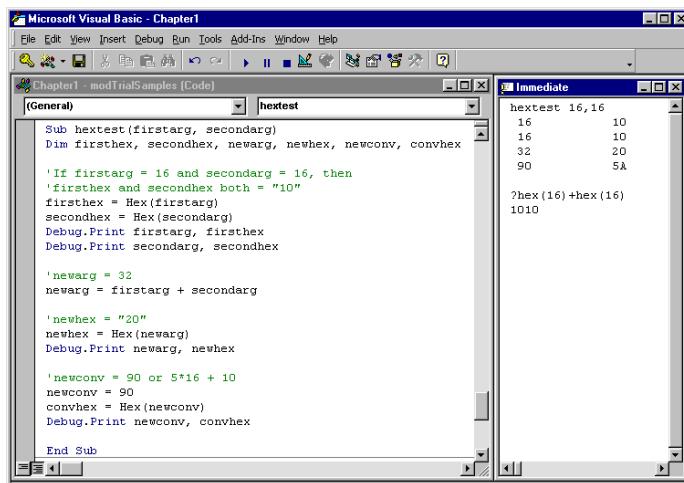
הטבלה שלפניך מציגה מבחר קטגוריות פונקציות עם תיאור יעודה של הקטgorיה. הטבלה אינה כוללת את כל הקטגוריות, וכל קטgorיה מכילה בדרך כלל יותר פונקציות. סיווג הפונקציות בקטgorיה כזו או אחרת הוא מעשה שרירוטי, מכיוון שפונקציות רבות מתאימות לכמה קטgorיות. לדוגמה, הפונקציה DateSerial המטפלת בתאריכים, מחזירה מספרים סידוריים של תאריך בהתבסס על ארגומנטים שאיןם מסווג תאריך/שעה, אך באפשרות גם להמיר תאריכי מהרוות למספרים סידוריים. לפיכך הפונקציה יכולה להימנות בו-זמנית עם הקטגוריות תאריך/שעה והמרה.

קטגוריות נבחרות של פונקציות

יעוד	פונקציות מייצגות	קטgorיה פונקציה
קביעת סוג הפעולה החשבונית שביטוי מבצע והמרה בין סוגים נתונים	CDbl, CSng, CInt, CCur, CDec, Str, Val, Hex, Oct	המרה
ביטוי, חישוב, חילוץ ערכי תאריך ושעה מתוך מספרים סידוריים של תאריך ויזוגי מחזרות של תאריכים ושעות.	Date, Now, DateAdd, DatePart, DateDiff, Year, Month, Day, Hour, Minute, Second, Weekday, DateValue, DateSerial, MonthName, WeekdayName, FormatDateTime	תאריך שעה
חישוב ערכים סטטיסטיים, כגון ספירה, עבר ותחום (לדוגמה, טבלה או שאלתה)	DLookup, DCount, DSum, DAvg, DVar	ציבור המופעלת על תחום
לכידת שגיאות וקודם שגיאות מותאמים אישית	Error, CVErr, IsError	טיפול בשגיאות
הערכת סוגים נתונים באמצעות קוד	IsDate, IsTime, IsNumeric, IsNull, IsEmpty, VarType	בדיקה
ביצוע פעולות מתמטיות	Rnd, Sqr, Exp, Log, Sin, Cos, Tan	מתמטי
הפקת הודעות ואיסוף נתונים קלט מהמקלדת	MsgBox, InputBox	הודעות
עיבוד מחזרות	Left, Right, Mid, Trim, UCase, LCaseStrReverse, Replace, InStr, InStrReverse	טקסט
ציבור ערכים סטטיסטיים, קביעת מאפייני צבע, חישוב ערכים פוננסיים, יצירה או קילית הפניות אל אובייקט ActiveX, הצגת מדדי התקדמות ועוד	Sum, Count, RGB, FV, NPV, CreateObject, GetObject, SysCm	שונות

פונקציות אלו מבצעות מיגון משימות. פונקציות מחרוזת יכולות לחץ, להמיר ולעגל חלקים לא רצויים של מחרוזת טקסט, אך באפשרות גם לנצל את הפונקציות המוכנות לחישוב תוצאות פיננסיות או ערכאים מתמטיים. פונקציות בדיקה שימושיות לקביעת סוגים נתוניים לפני חישוב הביטויים. שימוש מושכל בפונקציות אלו ובפונקציות המירה יאפשר לך להימנע משגיאותAi-התאמה אפשריות בין סוגים נתוניים (טיפול בשגיאות הוא חלק חיוני של פתרון כולל כלשהו. בצדיו הפתורנות שתיצור לא יוכל ללכוד שגיאות זמן ריצה, או לעורר שגיאות מותאמות אישית של זמן ריצה). פונקציות צבירה המופעלות על תחומי (domain aggregate functions) אינן מצטיינות בביטויים מהירים מעולים, אך ביכולתך להחילף שאילתת הצבירה לתחום SQL מורכב יותר. לדוגמה, ישומיך יכולם לנצל את פונקציות הצבירה לתחום, Sum ו-Count ו-Count בשאלות ובפתרונות. פונקציות צבירה ל-SQL, VBA וגם שגיאות של מקורות רשומות. פונקציות אלו אינן ישימות בהיקף רחב כמו פונקציות צבירה לתחום. לדוגמה, לא ניתן להשתמש בהן בשגרות VBA.

תרשים 1.25 מציג שגירה ואת החלון **Immediate** כדי להמחיש את התנהגות הפונקציה Hex. השגירה הופכת את שני הארגומנטים למחרוזות הקסדצימליות ולאחר מכן מדפסה את התוצאות בחולון **Immediate**.



The screenshot shows the Microsoft Visual Basic Editor interface. On the left is the code editor window titled 'Chapter1 - modTrialSamples (Code)' containing the following VBA code:

```

Sub hextest(firstarg, secondarg)
Dim firsthex, secondhex, newarg, newhex, newconv, convhex

If firstarg = 16 and secondarg = 16, then
    firsthex and secondhex both = "10"
    firsthex = Hex(firstarg)
    secondhex = Hex(secondarg)
    Debug.Print firstarg, firsthex
    Debug.Print secondarg, secondhex

    newarg = 32
    newarg = firstarg + secondarg

    newhex = "20"
    newhex = Hex(newarg)
    Debug.Print newarg, newhex

    newconv = 90 or 5*16 + 10
    newconv = 90
    convhex = Hex(newconv)
    Debug.Print newconv, convhex

End Sub

```

To the right of the code editor is the 'Immediate' window titled 'Immediate' which displays the output of the code execution:

hextest 16,16	
16	10
16	10
32	20
90	5A
?hex(16)+hex(16)	
1010	

תרשים 1.25: שגירה שבודקת את הפונקציה Hex

לא ניתן לבצע טיפול מתמטי במחרוזת המוחזרת על ידי הפונקציה Hex. הפונקציה hextest מתגברת על הבעיה על ידי ביצוע הפעולה החשבונית באמצעות הארגומנטים העשרוניים שלה, והמרת התוצאה שמתאפשרת באמצעות הפונקציה Hex. שורת הקוד בהמשך השגירה מדגימות גישה זו: הן מחברות את firstarg ואת secondarg ו-newarg ו-newhex ו-convhex את התוצאה ב-g-arg. לאחר מכן, השגירה הופכת את newarg למחרוזת הקסדצימלית ומ אחסנת אותה ב-newhex. לבסוף, היא מדפסה את newhex ואת .Immediate

5A. שורות הערה הקודמות לפעולות המרורה מציינות את העבודה ש-90 הוא תוצאת הביטוי $10 + 16 \times 5$.

זוג השורות האחרון בחילון **Immediate** מראה שהפונקציה Hex מחזירה ערכי מחרוזות. למעשה Hex מחזירה נתון מסווג variant, אך סוג המשנה שלו הוא מחרוזות. לחבר המספריםhextext והערך 16 + 16 היה אמור לתת תוצאה של 20 הקסדצימלי, אך בחילון **Immediate** מקבלים 1010. תוצאה זו מעידה שישמן החיבור בשורה `newarg = firstarg + secondarg` ביצוע שרשור מחרוזות במקום חיבור אריתמטי של מספרים.

עליך לבחש בהתמדה דרכים לצמצום זמן העיבוד. לדבר חשיבות מיוחדת בפתרונות שנוצעו לשרת את משתמשים תקופה ארוכה, או כאלה שמבצעים משימות עסקיות חיוניות. תרשימים 1.26 מציגו שיגרה שיכולה לאמוד משך זמן ביצוע של קוד. השיגרה מחשבת את הזמן הדרוש לביצוע לולאה מספר קבוע של פעמים. באפשרותך להרחיב את התפקוד הבסיסי ולהוסיף לו השוואות זמניות של שני פתרונות תכונות. כפי שניתן לראות בחילון **Immediate**,قطع הקוד הפשטוט פועל במתירות. מחשב הבדיקה שלו ביצע את הלולאה שני מיליון פעם בשתי דקות.

The screenshot shows the Microsoft Visual Basic Editor interface. The top window is titled "Microsoft Visual Basic - Chapter1". Below it is a toolbar with various icons. The main workspace window is titled "Chapter1 - modTrialSamples (Code)" and contains the following VBA code:

```

Sub TimedLoop(intLoops As Long)
    Dim dtmStartTime As Date, dtmEndTime As Date
    Dim intCounter As Long, intDummy As Currency
    Dim intSeconds
    dtmStartTime = Now()
    For intCounter = 1 To intLoops
        intDummy = intCounter + intCounter + intCounter
        intDummy = intDummy - (3 * intCounter)
    Next intCounter
    dtmEndTime = Now()
    intSeconds = DateDiff("s", dtmStartTime, dtmEndTime)
    Debug.Print Format(intLoops, "##,##0") -
        & " loops took " & intSeconds & -
        " seconds to complete."
End Sub

```

Below the workspace is the "Immediate" window, which displays the output of the code execution:

```

timedloop 2000000
2,000,000 loops took 2 seconds to complete.

```

תרשים 1.26: הערכת זמן ביצוע של משימה. השיגרה TimeLoop מציגה, בין היתר, כיצד פועלות פונקציות מוכללות אחדות

השיגרה TimeLoop קולטת ארגומנט ייחיד שקובע את מספר הפעמים שיש לבצע את הלולאה. ביצוע חזר של לולאה מספר פעמים ידוע, מאפשר לעיתים קרובות להבחן בהבדלי ביצועים בין מחשבים. הלולאה פשוטה שלנו כוללת שני משתנים קבועים בלבד. שים לב שלפני שהשיגרה מתחילה ביצוע הלולאה, היא שומרת את השעה הנוכחית (Now) ב-dtStartTime. הפונקציה DateDiff מחשבת את ההפרש בין שני הערכים זמן הסיום ב-dtStartTime. הפונקציה DateDiff מטפלת ביחידות זמן שונות, כולל שנים, חודשים, שבועות, ימים. הפונקציה מטפלת ביחידות זמן שונות, כולל שנים, חודשים, שבועות, ימים, ועוד (סוף שבוע). הפונקציה DateDiff מטפלת בהמרה לשניות ובכל התופעות המזוררות המתעוררות כתוצאה מהמעבר של חוות הלילה, ומחלצת את הערכים הרלוונטיים של מועד ההתחלת והסיום.

בדיקה זו של משך זמן מtabסת על מספר גדול מאוד של חוות, ולכן נוח במיוחד לבדוק את מספן באמצעות פסיקים. הפונקציה Format מאפשרת לבצע שימושה זו.

פונקציות צבירת התחום מהוות קבוצה ייחודית. הן מחזירות נתונים סטטיסטיים ונתוני חישוב על תחום (קבוצת רשותם). הרשותות יכולות להימצא בטבלה אחרת, או להיות תוכאה של שאלת בחירה. הפונקציות יכולות לחזור נתונים שונים על תחום נתון, כגון: ספירה, סכום, ממוצע, שונות וסטטיסטיקה. תוכל לנצל את פונקציות צבירת התחום DMin ו-DMax כדי לחזור את הערך הקטן או הגדל ביותר בעמודה שבתחום. פונקציות צבירת התחום DFirst ו-DFirst מחזירות רשומה אקרואית במקומות רשומה ראשונה או אחרונה מתוך רשומות הממוינות לפי שדה מסוים. פונקציית צבירת התחום DLookup היא פונקציה מקובלת מאוד; היאמחזירה רשומה אחת או יותר שתואמת לкрיטריון נתון. לכל פונקציות הצבירה בתחום ישנה תבנית הדומה לתבנית הבאה:

```
DFunctionName("fieldexpression", "domainname", "criteria")
```

כל השמות של פונקציות צבירה המופעלות על תחום מתחילה ב-"D". שים לב שככל שלושת הארגומנטים של הפונקציה מופיעים בין מרכאות כפולות. ניתן, אך לא חובה, להציב את שם השדה של הארגומנט הראשון והשלישי בין סוגרים. שני הארגומנטים הראשונים הם חובה, והשלישי – אופציוני. הארגומנט fieldexpression מצין שדה שעליו יש לדוחות. הארגומנט domainname הוא שם מקור הרשותות. הארגומנט האופציוני השלישי מצין משפט קרייטריון כדי לציין את האלמנטים שברצונך לכלול בסדרת הרשותות המוחזרת (קבוצת החזרה). כדי לחזור את שם החברה מהרשומה הראשונה בטבלה Shippers שבבסיס הנתונים Northwind, הפעלת הפונקציה DLookup如下:

```
DLookup("CompanyName", "Shippers", "ShipperID = 1")
```

הקרייטריון בדוגמה זו מצין את הרשימה בטבלה שבה שדה ShipperID מכיל את הערך 1. זהו שדה מספור אוטומטי המכיל נתונים מסוג Long Integer. אם קרייטריון מסוג זה מתאים לצרכיך, הדבר יפשט את השימוש בפונקציות צבירה לתוךם. גירסה

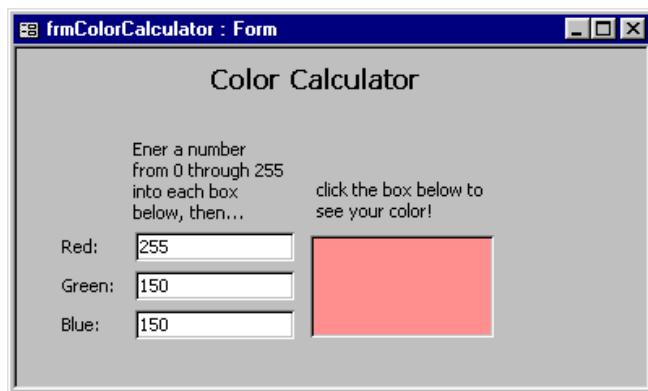
זו ממחישה את ערך השדה מתוך הרשומה היחידה התואמת לкрיטריון. אם קיימות כמה רשומות התואמות לкрיטריון, הפונקציה תחזיר את הרשומה התואמת הראשונה.

פונקציות צבירה נוספת לתchrom, כגון DCount, DSum ו-Avg מחשבות תוכאות המבוססות על רשומה אחת או יותר. משפט VBA שלפנינו מדפיס את תוכאת ספירת החשבונות שמועד פירעון חלף בחלון Immediate בהתאם לתוכנית החישוב של הפונקציה : DCount

```
Debug.Print DCount("InvoiceDate", "InvoiceDates", "InvoiceDate < Now() - 30")
```

דוגמא זו מחשבת את מספר הרשומות שמועד פירעון חלף ומדפסה את תוכאת החישוב. כמשמעותו, במשפטים אוטה על הטבלה InvoiceDates שבקובץ הדוגמה בפרק 1, בתאריך 12/9/98, הפונקציה DCount מחזירה שתי רשומות. ייתכן שתרצה להוציא רשומות חדשות לטבלה, או לשנות תאריכים כלשהם כדי לקבל תוכאה אחרת של ספירת הרשומות עבור דוגמה זו.

תרשים 1.27 מציג טופס שמאפשר למשתמש לקבע את ערכי הגוונים אדום, ירוק וכחול של צבע ולאחר מכן להציג את הצבע המתתקבל כשמשתמש לווח על תיבת הטקסט שבעצם ימיון. גווני האדום, הירוק והכחול של הצבע יכולים לקבל ערכים בטוחה בין 0 ל-255. אם כל שלושת הערכים הם 0, יתקבל צבע שחור; אם כלם 255, יתקבל צבע לבן. קביעת אחד הגוונים בתור 255会使 השני הגוונים האחרים 0, נותרת צבע חזק של הגוון שקיבלה את הערך המקסימלי. האדרות הגוון במחשב שלו מציגות את הצבע המוגדר בתרשים 1.27 בתור אדום בהיר או ורוד.



תרשים 1.27: טופס המראה את מציג הצבע

שגרת האירוע שלפניך השicket לתיבת הטקסט הימנית בתרשים 1.27, מבצעת את תהליך הטיפול הפשטוט במציג הצבע. מלבד בדיקת ערכים ריקים בתיבות הטקסט של גווני הצבע, השגורה מכילה משפט בודד. המשפט מנצל את הפונקציה RGB כדי להגדיר את המאפיין BackColor (צבע רקע) של תיבת הטקסט הגדולה בחלקו הימני של הטופס.

```

Private Sub txtColorMe_Click()
'If null color field, set a default value.

If IsNull(Me.txtRed.Value) Then
    Me.txtRed.Value = 0
End If

If IsNull(Me.txtGreen.Value) Then
    Me.txtGreen.Value = 0
End If

If IsNull(Me.txtBlue.Value) Then
    Me.txtBlue.Value = 0
End If

'Set BackColor property.
Me.txtColorMe.BackColor = RGB(Me.txtRed.Value, Me.txtGreen.Value, _
    Me.txtBlue.Value)
End Sub

```

בבדיקה ערכים ריקים (Null) חיונית, כיון שערכים כאלה יגרמו לפונקציה RGB להיכשל. ניתן לתקן את הבעיה בדרכים רבות, וקוד השיגרה txtColorMe_Click הוא רק אחד מהן. לדוגמה, קביעת ערך בירית המהבדל בתווך 0 במקרה שלא צוין ערך כלשהו, היא קביעת שרירותית בעלייל; הערך יכול להיות 255 או מספר כלשהו בין 0 ל-255. אפשרות נוספת נספהת היא להימנע מחישוב הפונקציה RGB, ובמקרה זאת לבקש מהמשמש למלא ערכים בכל שלוש תיבות הטקסט. והדרך האחרונה – לאפשר התרחשויות של שגיאות זמן ריצה עקב ערך Null, לכלוד את השגיאה ואז לטפל בה בהתאם. בסעיף הבא נציג דרכי למימוש גישה זו.

ניפוי באגים ולכידת שגיאות

אין קוד נקי משגיאות. מלבד שגיאות תחביר ושגיאות לוגיות, נחלת כל פתרון תוכנה בשלבי הפיתוח, גם קוד שאמור לתפקיד כהלה "מניב" לעיתים שגיאות זמן ריצה. שגיאות אלו עשויות להיות תוצאה של אי-הקפdet המשמש להזין ערך, או הזנת ערך מוחוץ בתחום. הפעולות נסיוניות אפשרות לפחות ארת רוב הבעיות מסווג זה, אך איןך יכול למנוע מהמשמש להזין ערך בלתי צפוי כלשהו, או לוחוץ על לחוץ בטרם עט.

לכידת שגיאות הוא תהליך "תפיסת" השגיאות המתחרשות בזמן ריצה, בטרם תגרומנה לסיום לא מבוקר של היישום. לכידת שגיאות מעnickה לך שליטה בעניינים. הקוד יכול לבדוק את השגיאה ולויטים אפילו להתגבר על הבעיה מבלי שהמשמש יצטרך לעשות כלום. באפשרותן לקבוע את נחיצות התערבותה המשמש ולהציג לו הودעה בהתאם, ואף לקבל ממנו קלט חדש באמצעות תיבות דו-שים מותאמות אישית ולאחר מכן להמשיך את פעולות היישום. גם במצבים בהם לא ניתן להתאושש בצורה מסודרת ממצב שגיאה, ניתן עדין לסייע את התוכנית בתנאים שנקבעו מראש.

הפתרון יכול לשמר מידע על משפט הקוד בו התרחש הכשל, לצורך קוד השגיאה, תיאור וכל פרטי המידע הרלוונטיים. לאחר כך תוכל להורות למשתמש לדוח על התקלה לנורם המתאים.

마חר ש-Access מאפשר לפתחים להגיב לשגיאות זמן ריצה, מומלץ שתחשוב על לכידת מצבים שלגיאה ביחסים שלך, כגון הזנת ערכיהם לא חוקיים. למעשה של דבר, Access מאפשר ליצור שגיאות מותאמות אישית בדרכים שונות. הדבר מאפשר לך להתחושים מהשגיאות המותאמות אישית שיצרת באותו אופן שהוא נוהג בשגיאות רגילות של Access. קרובה לוודאי שתדע טוב יותר כיצד להגיב לשגיאות המותאמות אישית, מכיוון שאתה יצרת אותן.

לכידת שגיאות – תחביר בסיסי

כדי למשם מנגנון לכידת שגיאות, عليك להבין את המשפט `Error On`, את האובייקט Err ואת המשפט `Resume`. כל אלה פועלים יחד כדי לאפשר לכידת שגיאות ולסייע לך לטפל בשגיאות לאחר לכידתן.

קיימות שלוש גרסאות למשפט `Error On`, אך הגירסה שלפניך מאפשרת לכידת שגיאות קרובה לוודאי שתבחר להשתמש בה:

`On Error GoTo linelabel`

בש-Access מזזה שגיאה, משפט זה מעביר את השליטה במצב לתוכניתך שלך, במקום לחתת ל-Access לטפל בשגיאה לפי ברירת המחדל שלו. הארגומנט `linelabel` הוא שם השורה שאליה ברצונך להעביר את השליטה. במקום ש-Access יבצע את תהליך הטיפול המוגדר מראש, יבצע הקוד המקורי בשורה ששם צוין במשפט. על הקוד לקבוע ולפענה את מס' השגיאה, כדי שהתוכנית תנקוט פעולה הולמת. אם אין אפשרות לתקן את התקלה, תוכל לפחות לבקש מידע כלשהו אודותיה בטרם תסתיים התוכנית בזורה מסודרת.

האובייקט Err מלמד על השגיאה. לאובייקט שני מאפיינים. המאפיין `Number` מגדר את השגיאה במדויק. המספרים המאוחסנים במאפיין `Number` אינם משתנים מגירסה לגירסה של Access. משמעות הדבר היא שנitinן לנצל את המאפיין בתור מזזה שגיאות עמיד. מידע נוסף ניתן להפיק מהמאפיין `Description` של האובייקט Err שמכיל תיאור מילולי של השגיאה. عليك להציג את תיאורי השגיאות בחלוון המיידי או בתיבת טקסט.

כאשר מתכוונים תוכנית הראשית, תיאורי האובייקט Err יכולים לסייע לשפר את התכנון ולפתח תగיות הולומות לשגיאות זמן ריצה שגרתיות, כגון ערך קלט מחוץ לתחום משתנה. גירסה מוכנה לשימוש של התוכנית יכולה גם לשמר תיאורי שגיאה בקובץ יומן. הדבר מאפשר לתכנן תיקוני קוד מהירים לאורך חיי היחס. יחד עם זאת, בדרך כלל לא תרצהゾהות שגיאות על פי תיאורייהן שהם ארכויים במספריהם (בנוסף, Microsoft שעווה לנוכח החדש תיאורי שגיאה, למורות במספריהם אינם משתנים).

המשפט Resume משמש לצוון נקודת המשך הפעלת היישום לאחר זיהוי וטיפול בשגיאה. אם ברצונך להעביר את השיטה למשפט שגרם לשגיאה, השתמש במילה Resume ללא ארגומנטים כלשהם. גישה זו עשויה להתאים במקרים בהם אתה מתקן את התקלה שגרמה לשגיאה מלבתיחה, כגון ערך שדה חסר. אם מגנוון הטיפול בשגיאות אינו מסוגל להתגבר על השגיאה, ייתכן שתרצה למורות זאת להמשך בביוזו התוכנית שורה אחת אחרי זו שגרמה לשגיאה. במצב זה השתמש במשפט Resume Next. כל עוד השגיאה אינה משפיעה על תקופות שורות הקוד הנותרות, גישה זו באה בחשבון. במצבים שהיישום מגלח שגיאה שאינו מסוגל לטפל בה, ייתכן שתרצה פשוט להפסיק את פעולהו באופן מיידי. במקרים אלה העבר את השיטה למשפט linelabel. בעת לכידת שגיאות, תכלול לעיתים קרובות המשפט Resume linelabel או משפט Exit Sub בצוירוף תווית.noch להשתמש בתווית זו בטור Exit Function ארגומנט המשפט Resume linelabel במרקם שברצונך לצאת מהשיגרה לאחר שגיאה.

דוגמאות לכלcit שגיאות

סעיף זה מציג שתי דוגמאות של לכידת שגיאות. הראשונה שביניהן היא יישום בסיסי של לוגיקת לכידת שגיאות. הדוגמה מציגה לכידת שגיאות של שורת פונקציה בתיקיה Module. היא מציגה גם כיצד לעורר קוד שגיאות מותאמים אישית. הדוגמה הראשונה ממחישה את יכולות הטיפול בשגיאות היישום הפנימיות באמצעות לוגיקה זהה זו המשמשת לכידת שגיאות ב-Access. הדוגמה השנייה מנצלת לוגיקת לכידת שגיאות בקוד התומך בטופס. התכונה הבולטת של יישום זה היא תגובתו השונה לכל אחת משלקיות של שגיאות.

שורת הפונקציה שלפניך משלבת לכידת שגיאה בדוגמה הפשוטה יותר שהוצגה קודם לכן בתרשים 1.16. שים לב כי שגירה זו מבצעת את מאربع פעולות חשבוניות. מעבירים לשגירה שלושה ארגומנטים: שני אופרנדים ומחרוזת המציינת את הפעולה החשבוןית שיש לבצע על המספרים שהועברו אל הפונקציה.

שתי שגיאות לפחות עלולות להתעורר מפעולות הקוד בזמן ריצה. הראשונה, המשמשת עלול לנסוט חלק מסטר-0. פעולה זו יוצרת שגיאה מסטר-11. שגיאה שנייה, הפעולה עלולה ליצור תוכאה גדולה או קטנה מזו שמאפשר סוג הנתונים. סוג שגיאה זה נקרא במוני Access **שגיאת גלישה** (overflow error) ומספרה הוא 6. מגנוון לכידת השגיאות של השגירה שלפניך לווד את שתי השגיאות הללו בפורש, ובאופן משתמש מניח מלבדות לכל סוג אחר של שגיאה. קטע הטיפול בשגיאה מסיע למשתמש, על ידי הצעת תיקון לחילוק ב-0, וגם מסביר מה טיבה של שגיאת גלישה בשפה יידיתוית יותר מזו ש-Access משתמש. הויל ולעתים קרובות כתוב יישומים עבורי קהיל משתמשים מוכר לך, תוכל לפנות אליהם באופן ישיר יותר מאשר תוכנית כללית (כמו Access).

```
Public Function Computer2(db1Number1, db1Number2, Operation As String)
On Error GoTo Computer2Handler
```

```
Select Case Operation
Case "Addition"
    Computer2 = db1Number1 + db1Number2
Case "Subtraction"
    Computer2 = db1Number1 - db1Number2
Case "Multiplication"
    Computer2 = db1Number1 * db1Number2
Case "Division"
    Computer2 = db1Number1 / db1Number2
End Select
```

```
Computer2Exit:
    Exit Function
```

```
Computer2Handler:
```

```
If Err.Number = 11 Then
    MsgBox "Can't divide by zero. Change second number from 0.", _
        vbInformation, "Programming Microsoft Access 2000"
ElseIf Err.Number = 6 Then
    MsgBox "Result exceeds data type value limits.", vbInformation, _
        "Programming Microsoft Access 2000"
Else
    MsgBox Err.Number & ":" & Err.Description, vbInformation, _
        "Programming Microsoft Access 2000"
End If
Resume Computer2Exit
End Function
```

המשפט On Error מופיע מיד לאחר תחילת התוכנית. הוא ממוקם שם כדי לכלוד שגיאות מוקדם ככל האפשר. אם פונקציה אחרת קוראת לתוכנית הנוכחית, יתכן שכדי להשתמש בლכידת שגיאות באוטה תוכנית, כך שהיא תוכל להשתמש בლכידת שגיאות עבור הארגומנטים שלה המועברים לפונקציה Computer2.

אם תקליד ("Enter ?Computer2(2, 4, "multiplication")" ותקיש בחלון Immediate ?Computer2(2, 4, "multiplication")) והערך הפונקציה תחזיר את הערך 8. החילف את פעולה הכפל בפעול חילוק (division), והערך החזר יהיה 0.5.icut החילף את הארגומנט השני ב-0. ללא שימוש מנגןן לביידת שגיאות, התוכנית הייתה "נויפلت" בצירוף הودעת מערכת מתאימה. מנגןן הטיפול בשגיאות מעביר את השליטה ל- Computer2Handler ברגע שנעשה ניסיון לחילך ב-0. הרוטינה בזדקה את שגיאה מספר 11, ומכיון שזו השגיאה שאירעה, היא מציגה למשתמש הودעה שעליו לשנות את גורם החילוק למספר שונה מ-0. הקלד

פעולת הכפל מנסה ליצור תוצאה גדולה מזו ש-Access מסוגל ליחס. גם במקרה זה מועברת השליטה ל- Computer2Handler מייד לאחר שהפעולה החשבונית נכשלה. לאחר שנקבע כי קוד השגיאה אינו 11,ilogיקת לכידת השגיאות בודקת את קוד השגיאה כנגד 6. התאמת הקוד גורמת להציגת תיבת הودעה שמסבירה את סיבת השגיאה.

קטע הטיפול בשגיאות של השיגרה משתמש במשפט If...Then...ElseIf...Else. הפסקוקית Else מציעה דרך לכידת שגיאות שהקוד אינו Locator באופן מפורש. בעת אתחול היישום, איןך מסוגל לבדוק אילו שגיאות תתרחשנה. במקרה זה השתמש במשפט If...Then...ElseIf...Else. הפסקוקית Else מציגה מספרי שגיאות בצירוף תיאוריה כשהן מתרחשות בשלב הבדיקה. באפשרות לנצל מידע זה כדי לפתוח מנגנון לכידה מפורטים עבור שגיאות מסוימים יחד עם פתרונות זמינים למצבים אלה. במהלך בדיקת היישום ייתכן שתהיאנה לכך הזדמנויות נוספות לשככל את אוסף הפסוקיות Else של מנגנון הטיפול בשגיאות.

השורה האחרונה בקטע הטיפול בשגיאות היא המשפט Resume. משפט זה מעביר את השליטה לשורה שמננה ווצאים מהשיגרה. באפשרות גם להשתמש במשפט Resume נפרדים לכל אחת ממילכות השגיאות. גישה זו מתאימה למקרים בהם היישום מחיב אותך לנקודות פעולות שונות לכל סוג של מלכות שגיאות.

מנגנון הלכידה לא יכול סוג שגיאות אחדים. לדוגמה, הוא לא יכול שגיאה שמקורה באירוע שגוי של הפעולה החשבונית, כמו MULTIPLICATION במקומות multiplication. שגיאת מפעיל מעין זו עברת דרך משפט Select Case מבלי לעצור כדי לבצע פעולה חשבונית כלשהי, עד שהיא מוגיעה למשפט Exit Function. יש צורך לכלוד שגיאה זו ולהודיעו למשתמש מה עליו לעשות. הפתרון הוא לזרות את הבעיה באמצעות הפסוקית Case Else במסגרת המשפט Select Case. לאחר מכן מועררים שגיאה מותאמת אישית שמודיעת למשתמש כיצד לנוהג.ilogיקת הלכידה עמידה מאוד, ולכן ניתן לשנות רק את המשפט Select Case. הקטע הבא מותוך הקוד המתוקן מציג גירסה חדשה, הכוללת רק שני משפטיים נוספים:

```
Select Case Operation
Case "Addition"
    Computer2 = dblNumber1 + dblNumber2
Case "Subtraction"
    Computer2 = dblNumber1 - dblNumber2
Case "Multiplication"
    Computer2 = dblNumber1 * dblNumber2
Case "Division"
    Computer2 = dblNumber1 / dblNumber2
Case Else
    Err.Raise 1, , "Wrong operation."
End Select
```

השורה שלאחר Case Else מציגה את התchapir לייזום קוד שגיאת מותאמת אישית. שגיאת מס' 1 אינה מקבלת הצבה-DAO או ADO. לכן, כאשר המשפט Case Else לוכד פעולה שסמה מאוית בצורה שגوية, הוא מעורר שגיאת מותאמת אישית וזו מעבירה את השליטה למשפט Computer2Handler. הפסוקית Else בקטע משפטים זה מזזה את קוד השגיאת 1 ויוצרת תיבת הودעה ובها קוד השגיאת והתייאור המותאם אישית "Wrong operation" (פעולה בלתי חוקית). לאחר שנסкор את תחילה לכידת שגיאות בעזרת טופס, נחזור לנושא בחירת השגיאת המותאמת אישית.

הקוד התומך בטופס שモצג בתרשיים 1.20 עד 1.20 מתבסס על משפטי For Each...Next מקוונים כדי לטפל בתגובהות לטופס. כזכור, שגרת אירוע הלחיצה עבר לבחן הפוקודה עוברת בלולאה על כל הפקדים בטופס כדי למצוא את שתי תיבות הטקסט. הדבר נעשה כדי להימנע משגיאת המתרכשת במקרה שהפונקציה IsNull מופעלת על פקד, כגון תווית או לחץ פקודה, שאינו תומך בפונקציה זו.

גישה חלופית היא לאפשר לשגיאת להתרחש, למכוד אותה ולהתאושש מהבעיה. הפעלת הפונקציה IsNull על פקד שאינו מתאים לה גורמת לשגיאת מס' 438. לפניהם .cmdSubmit_Click

```
Private Sub cmdSubmit_Click()
On Error GoTo SubmitErrorTrap
' Check for valid entries.
    For Each ctl In Screen.ActiveForm.Controls
        If IsNull(ctl.Value) Then
            MsgBox "Please enter information in both input boxes.", _
                vbInformation, "Programming Microsoft Access 2000"
            MarkFieldsToDelete
            Exit For
        End If
    SubmitNextCtl:
        Next ctl

    SubmitExit:
        Exit Sub

SubmitErrorTrap:
    If Err.Number = 438 Then
        Resume SubmitNextCtl
    Else
        MsgBox Err.Number & ": " & Err.Description, vbInformation, _
            " Programming Microsoft Access 2000"
        Resume SubmitExit
    End If
End Sub
```

שים לב כי הלוגיקה של לכידת השגיאות מאריכה את הקוד, למרות שהיא מאפשרת להסיר אחד משני משפטי Next...For Each. הלוגיקה החדשה לכידת שגיאות עמידה יותר. לולאות For המוקנוןת נמנעות מסוג שגיאה אחד בלבד – שגיאה מס' 438. להלכה, החלופה החדש יכולה לטפל בכל סוג שגיאה. בנוסף, היא מסוגלת להגביל לשגיאות בשתי דרכיים שונים. שים לב כי שגיאה 438 מגיבה בהעברת השליטה אל המשפט Next בחלק שנותר בלולאת For. הדבר מאפשר לתוכנית להמשיך לטפל בפקדים נוספים שתואימים לפונקציהIsNull. שגיאה מסווג אחר כלשהו גורמת לשגרת האירוע להסתיים לאחר הצגת תיבת הודעה ובה מספר השגיאה הבלתי צפוייה ותיאורה.

העלאת שגיאות

הפונקציה Raise (העלאה) של האובייקט Err עמידה, אך היא מחייבת לאפשר את הטיפול בשגיאות. הוסרת רוטינות טיפול בשגיאות עלולה להאריך את הקוד. אם אין לך ניסיון רב בכיניות קוד וברשותך שיגרה קצחה שיש לה פוטנציאלי קטן לגורם לשגיאות, ייתכן שתרצה דרך פשוטה יותר להחזיר קוד שגיאה.

השיגרה שלפניך מבצעת זאת בעזרת הפונקציה CVErr. שיטה זו אינה עמידה לתקלות כמו מנגןון שלם לטיפול בשגיאות, מכיוון שהיא לוקדת שגיאות מסווג אחד בלבד. הויל ולא קיימים מנגןון לטיפול בשגיאות, היישום יכול עלול לקורס אם מתרכשת שגיאת זמן ריצה, אך קל ופשוט להשתמש בפונקציה CVErr. הפונקציה אינה מתנגשת עם קוד שגיאה מוכלים, מכיוון שהיא מחזירה קוד שגיאה באותו מסלול. עליך לשקל את יתרונות וחסרונות השימוש בפונקציה CVErr במקרים המחייבים החזרת קוד שגיאה.

```
Public Function Computer1(db1Number1, db1Number2, Operation As String)
```

```
    Select Case Operation
        Case "Addition"
            Computer1 = db1Number1 + db1Number2
        Case "Subtraction"
            Computer1 = db1Number1 - db1Number2
        Case "Multiplication"
            Computer1 = db1Number1 * db1Number2
        Case "Division"
            Computer1 = db1Number1 / db1Number2
        Case Else
            Computer1 = CVErr(2002)
    End Select
End Function
```

הפונקציה לוקדת שגיאה של פעולה שאויהה בצורה שגوية או כזו שאינה נתמכת בקוד. המשפט Select Case מבודד בעיות אלו באמצעות הפסוקית Case Else. כשההמשתמש מקליד פעולה שגوية, הפונקציהמחזירה נתון מסווג Variant שסוג המשנה שלו Error

מכיל קוד שגיאה מס' 2002. גירסה זו אלגנטית יותר מהשיגרה Computer2 שהוצאה קודם לכן בפרק. יחד עם זאת, Computer2 לוconda בפורש שגיאות מסווגי חלוקה באפס וגלישה, גם כשהיא מוחזירה שנויות מסווגים אחרים. השיגרה Computer1 קצרה בהרבה ממנה ולוכדת את הבעיה היחידה של פעללה שאויתה בצורה שגיאה, או צו שאיינה נתמכת בקוד.

הfonקציה CVErr אינה מושפעת מהתנשויות בין מספרי שגיאה של VBA ו-Access, זוג השגורות שלפניך מפשט בצורה פשוטה את תהליך איתור קוד שגיאה פנוים שתוכל לנצל לשימוש המותאם אישית שלך. הפונקציה הראשונה, VBADAOUsedErrorList, מציגה את קוד השגיאה בטוחה המספרים בשימוש DAO. עליך לציין את המספר הפתוח והמספר המסיים בעת הקראיה לשיגרה. המספרים הנדרדים מהפלט זמינים לשימוש בתור קוד שגיאה מותאמים אישית. השיגרה השנייה מאפשרת אף היא לציין מספר פתוח ומספר מסיים של הטוח שבו ניתן לאתר קוד שגיאה. שיגרה זו, לעומת זאת, מדפסה בחולון המידע את הקוד השגיאה שאינו שמורים על ידי VBA או DAO. חלק גדול מקוד השגיאה של ADO הם מספרים שליליים גדולים.

```
Sub VBADAOUsedErrorList(intStart, intEnd)
Dim intErrorCode As Long, strAccessError As String

For intErrorCode = intStart To intEnd
    strAccessError = AccessError(intErrorCode)
    If strAccessError <> "" Then
        If strAccessError <> _
            "Application-defined or object-defined error" Then
            Debug.Print intErrorCode, strAccessError
        End If
    End If
    Next intErrorCode

End Sub

Sub VBADAOUnUsedErrorList(intStart, intEnd)
Dim intErrorCode As Long, strAccessError As String

For intErrorCode = intStart To intEnd
    strAccessError = AccessError(intErrorCode)
    If strAccessError = "" Or strAccessError = _
        "Application-defined or object-defined error" Then
        Debug.Print intErrorCode, strAccessError
    End If
    Next intErrorCode

End Sub
```

פקודות מאקרו

פקודות מאקרו אינן חלק מ-VBA, אך הן מאפשרות להפוך יישומי Access לאוטומטיים בצורה קלה ו פשוטה. אחד היתרונות הגדולים של פקודות מאקרו הוא שאין צורך לזכור את התחביר שלהם; Access מאפשר לבחור בקלות פעולה מאקרו מתוך תיבת רשימה נפתחת. לעומת זאת, טכניקות ניפוי הבאים של מאקרו אינן עשירות כמו אלו של VBA. חיסרונו נוסף – פקודות מאקרו מאוחסנות במקולת מאקרו נפרדת, ולא אחורי טפסים, מה שعلול לגרום במשך הזמן לבעיות תחזקה. בנוסף, פקודות מאקרו של Access שונות מאוד של Word, Excel ו-PowerPoint. למעשה של דבר, פקודות מאקרו הולכות והופכות למיאושות.

אם השתמשת עד כה בגרסה קודמת של Access, יש לך בוודאי ניסיון מעשי בעבודה עם פקודות מאקרו. במהדורות Access קודמות, פקודות מאקרו היו האמצעי היחיד ליצירת אפשרויות אתחול מיוחדות ותפריטים מותאמים אישית. מעטה תוכל לנחל תפריטים באמצעות אובייקט האוסף CommandBars (סריגי פקודות) של VBA. בנוסף, באפשרותך לנצל את תיבת הדו-שיח **הפעלה** (Startup) לשיליטה במיגון החיבטים של אתחול יישום Access, כגון הצגת טופס בסיום תחיליך אתחול, צוין אם המשתמש יכול לעורך שינויים בתפריטים ובسرיגי כלים, ובכיהה אם יוצג חלון מסד הנתונים.

תכון פקודות מאקרו

כדי לעבוד עם פקודות מאקרו, عليك לדעת כיצד להשתמש במסמך המאקרו וגם להכיר בצורה בסיסית את פעולות המאקרו. תרשימים 1.28 מציג את חלון המאקרו לאחר טעינת מסד הנתונים Northwind.

כל מאקרו של Access מורכבת משורה אחת או יותר בחלון **macro** (Macro). כל שורה מצינית פעולה (כגון העברת מיקוד לפקד), תנאי אופציונלי עבור הפעולה והערה אופציונלית. האזoor **ארגוניים של הפעולה** (Action Arguments) בתחריט החלון (מושג בעת בחירת פעולה) מאפשר לציין את האפשרויות עבור כל פעולה ומציג עזרה עבור הפריט שנבחר. בנוסף, השורה הראשונה בכל מאקרו מצינית את שם המאקרו (בחר **שמות מאקרו** (Macro Names) מהתפריט **תצוגה** (View) של Access אם איןך רואה שמות מאקרו).

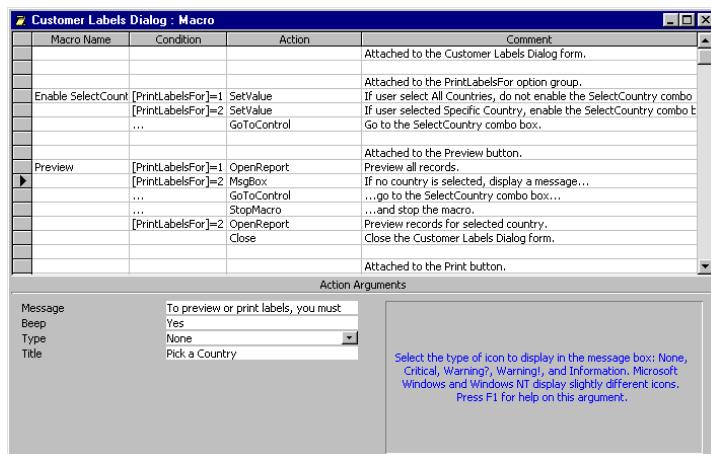
פעולות המאקרו רבות כוללות אפשרות שניית לבחור בהן באמצעות לחיצה פשוטה. לדוגמה, השורה השנייה של המאקרו Preview בתרשימים 1.28 (השורה בעלת המשולש שקודקודה פונה ימינה) מציגה תיבת הودעה. באזoor **ארגוניים של הפעולה** בחלון ניתן לבחור בארגומנטים של תיבת ההודעה מתוך תיבות רשימה נפתחות. לדוגמה, תוכל להשתמש בתיבת הרשימה הנפתחת **סוג** (Type) כדי לבחור בסמל שיציג בתיבת ההודעה.

בגרסאות קודמות, הבחירה בעורצת הצבעה ולחיצה העניקה לפקודות המאקרו יתרון. Intellisense מוחלט על קוד VBA מבחינת נוחות השימוש. כיום, VBA כולל את הרכיב

אומנם כל לשימוש בפקודות מאקרו עקם יכולות הבחירה של ארגומנטים, אך הן אין מיציאות את הממשק הגרפי שמאפשר להציג בו-זמנית את הקוד (לרבוט הארגומנטים) של פקודת מאקרו בעלת שורות רבות.

העמודה החשובה ביותר ביחסו בחלון **מאקרו** היא העמודה **פעולה** (Action). אין צורך להקליד פעולות בעמודה זו; באפשרות ללחוץ על שורה כלשהי בעמודה כדי לקבל תפריט מלא של פעולות מאקרו. בעמודה **תנאי** (Condition), משמאלי לעמודה **פעולה**, תוכל להגיד קרייטריונים ששולטים באופן מותנה בבחירה הפעולה אותה שורה.

תוכל לאחסן פקודות מאקרו רבות באובייקט מאקרו יחיד. זו דרך אפקטיבית לאחסון פקודות מאקרו ורבות המשותפות ביניהן תפקוד כללי. השימוש בעמודה **שם המacro** (Macro Name) כדי להציג שמות לפקודות המacro שנמצאות באובייקט המacro. בעט הקצתת פקודת מאקרו לפעולה על פק' בוטפס או בדוח, מצינים את המacro, באמצעות שם המרכיב שני חלקים: החלק הראשון הוא שם אובייקט המacro, והחלק השני הוא שם פקודת המacro המצוין בעמודה **שם המacro** (הפרד את שני חלקים השם בטו נקודה, כמו בסיסם `.MyMacros.DisplayHelp`).



תרשים 1.28: המרכיבים העיקריים של משק המשמש בחלון **מאקרו** (Macro)

כלל שגרת המרה מוכلالת לתרגום פקודות מאקרו לקוד VBA. הפעלת רכיבי המירה משתנה במקצת אם היישום קורא לפקודות המacro מתוך טופס (או דוח), או מתוך חלון מסך הנתונים. במרקחה הראשונית, בחר באפשרות **macros** מאקרו (Macro) מתוך תפריט **כלים** (Tools) ולאחר מכן בחר באפשרות **המרת פקודות מאקרו של טופס ל-** (Convert Form's Macros To Visual Basic) **Visual Basic** או באפשרות **המרת פקודות מאקרו של דוח ל-** (Convert Report's Macros To Visual Basic) **Visual Basic**. כדי להמיר מאקרו מתוך חלון מסך הנתונים, בחר אותו כרטיסייה **מאקרו**. בחר בתפריט **קובץ** (File), בחר **שמירה باسم** (Save As) ובתייה הדו-שייח **שם**, בחר **מודול** (Module) מתייבת הרשימה הנפתחת **בסוג** (As).

לעומת פקודות מאקרו

VBA הולך ומשתפר במידה רבה ממהדורה למהדורה, בעוד שפקודות מאקרו נותרו כמעט ללא שינוי. זהה היא הסיבה העיקרית לעבור פקודות מאקרו לו-*VBA*, אך קיימות סיבות נוספות:

- ↳ מנגנון לכידת השגיאות המכלול ב- VBA ורכיבי ניפוי באגים גמישים נוספים מקטיננס באופן דרמטי את עלות מחזור החיים ואת עלות הבעלות על יישומי Access, עקב הקטנת עלויות התחזוקה שלהם.
- ↳ VBA מציע רכיבי עיבוד מתקדמים רבים, כגון אפשרות עיבוד בלולה, תפקודיות אינטרנט מעוללה ו קישורי DB OLE למקורות נתונים חיצוניים.
- ↳ הטופס המותאם אישית ומחלקות הדוח של VBA מפשטים את השימוש החזר בקוד.
- ↳ VBA מאפשר להציג בו-זמן נית ארוגומנטים של כל השורות - לא רק של שורה אחת.
- ↳ VBA מאפשר להפעיל פונקציות API של Windows.

מודלי גישה לנוטונים

DAO Microsoft Access 2000 תומך בשני מודלים לגישה לנוטונים: המודל המקורי DAO - ActiveX Data Objects (DAO - Data Access Objects) אוvíיקטי גישה לנוטונים) ו-ADO (ActiveX). DAO מיועד למנגנון מסד הנתונים Jet, ומאפשר תכונות פשוט ומהיר של מסד הנתונים. Access 2000 הוא הגירסה הראשונה של Access שתומכת גם ב-ADO בכל הקשור לטיפול במסדי נתונים תומכי-Jet. במקום להתבסס על מנגנון מסד נתונים יחיד, DAO מנצח מודל תכונות מסווג כדי לספק גישה לנוטונים אוניברסליים, כאשר הקישורים הבסיסיים למקורות נתונים הוא מקבל מספקי OLE DB. טכנולוגיות OLE DB תדקה בסופו של דבר את טכנולוגיות ODBC שקדמו להן, כפי שמודול DAO יחליף את מודול DAO. אם תלמד לעובד עם ADO כבר עתה, תוכל לאמץ במהירות את השיפורים העתידיים בתחום הגישה לנוטונים של גרסאות Access שבתואנה בהמשך, כולל האפשרות לנצל מקורות נתונים נוספים ושוניים.

בפרק זה נסקור את מודלי הגישה לנוטונים DAO ו-ADO, תוך שימוש חדש מיוחד על DAO תכונות. הסקירה התמציתית על DAO מציגה את תפיסות הפיתוח העיקריות שלו ומבט היסטורי על גישה לנוטונים באמצעות Access. מודול DAO לא ימלא תפקיד מרכזי בפרקים הבאים, ולכן בפרק זה נציג את השימוש ב-Jet ובמסדי נתונים מרוחקים. לקבלת מידע על קוד DAO, עיין בעזרה המקוונת של Access ובקර באתר התמיכה המקוונת של Microsoft (support.microsoft.com/support). האתר מתעד בעיות טיפוסיות ודרכי התגברות עליהם. מאמרים רבים באתר כוללים דוגמאות קוד.

פרק הנוכחי מתמקד בעיקר במודלי האובייקט DAO עבור Jet ביחסם ובספריות ADODB ו-ADOX. נציג דוגמאות קוד מקיפות שמהווים כיצד לבצע מטלות מסד נתונים אופייניות. הפרקים האחרים בספר מתבוססים על הפרק הנוכחי וועוסקים בהיבטי ADO נוספים, כגון שכפול מסד נתונים, גישה למסד נתונים מרוחק ואבטחה בסביבה רבת-משתמשים.

סקירה כללית על DAO

Access 2000 כולל את גרסת 3.6 של ספריית DAO, שהיא גרסת שדרוג תחזוקה של גרסת 3.5 שشوוקה עם Access 97 (קיים דמיון רב בין הארכיטקטורה הבסיסית והרכיבים התפקודיים של שתי הגרסאות). DAO מתבסס על מודל אובייקט סביבת העבודה לסוגי גישה לנוטונים. אובייקט סביבת העבודה יכול להכיל מידע הפעלה (session), אבטחה וטרנסקציה (אובייקט סביבת עבודה מגדר את האינטראקציה בין היישום והנתונים).

קיימים שני סוגי סביבת עבודה: סביבות עבודה של Microsoft Jet וסביבות עבודה של ODBC Direct.

סביבות העבודה של Jet

סביבות העבודה של Jet (Jet workspaces) מיועדות למקורות נתונים נתוני Jet, ODBC הקשור ל-Jet ומקורות נתונים ISAM הנחוצים להתקנה. מקורות נתונים ODBC הקשור ל-Jet מאפשרים להתחבר אל מקורות נתונים מרוחקים בסביבת DAO המוכרת. לרובה הצער, סוג חיבור זה מהיבש שימוש במודול DAO במלואו, והוא טוען את Jet גם כאשר צורך בגישה לנוטונים. מקורות נתונים ISAM הנחוצים להתקנה, משוקרים במיגון Tabnios, כגון 1-2-3, Lotus Paradox וכו'.

לסביבות העבודה המקבילות של Jet יתרונות אחדים:

- ﴿ עדכון נתונים באובייקטים של ערכת רשומות,
- ﴿ צירוף טבלאות מקורות נתונים שונים לערכת רשומות אחת,
- ﴿ יצירת טבלאות המבוססות על שיטות DAO מוכROT, במקומות על מוסכמות שפת DDL, SQL (Data Definition Language)
- ﴿ איגוד נתונים לטפסים ולדווחות.

לסביבות העבודה של Jet אובייקטים של אוסף Groups (קבוצות) ו-Users (משתמשים), סביבות העבודה של ODBC Direct אינן כוללות אובייקטים כאלה. הסיבה היא שמקורות מסדי נתונים מרוחקים, כגון Microsoft SQL Server יכולים לנצל ביצועם את האבטחה שלהם.

סביבות העבודה של ODBC Direct

סביבות העבודה של ODBC Direct (ODBC Direct workspaces) היא טכנולוגיית DAO חדשה יחסית שהוצגה לראשונה ב- Access 97 וב- DAO 3.5. משום שבעבודה עם מקורות נתונים מרוחקים של ODBC אין שימוש באובייקטי האבטחה של DAO, ואובייקטי DAO אחרים פועלים בצורה הטובה ביותר עם מקורות נתונים שמחוברים בצורה מקומית, Microsoft יצרה את מודל האובייקט ODBC Direct, הזמין מסוג סביבת עבודה נפרד. סביבת עבודה זו מאפשרת גישה מהירה וישראל אל מקורות נתונים מסוג

ODBC (כגון SQL Server) תוך עקיפת הוצרך לשימוש ב-Jet. איןך מאבד את האפשרויות הרבות הטמונה במודול אובייקט, ואיןך חייב להסתמך באופן בלעדי על פקודות SQL כמו במקרה של שאילתות מעבר של SQL.

להלן מספר היתרונות העיקריים של סביבת העבודה :

- ↳ שימוש במקרים מסוימים מרוחקים ללא טעינה מוגנון Jet,
- ↳ שאילתות אסינכרוניות,
- ↳ גישה משופרת לתפקודיות מסד נתונים מרוחק, כולל סמנים ושרותי מאוחסנות (Stored Procedures)
- ↳ עדכון אצווות של מקורות מרוחקים מתוך מטמון מקומי,
- ↳ החזרת ערכות תוצאות מורובות באמצעות שאילתת בודדת.

בסביבות העבודה של ODBC Direct ספריית סמנים עשירה יותר מזו של סביבות העבודה של Jet, והן توומכות בסמנים דינמיים (dynamic cursors) ובسمני עדכון אצווה (batch update cursors) שאינם זמינים בסביבות העבודה של Jet. הסמן הדינמי מאפשר להפעלה (session) להציג שינויים שנערכו על ידי משתמשים אחרים, מבלי לבצע מחדש שאליתה על מקור הנתונים. סמן עדכון האצווה מאפשר לעדכן אסינכרונית מקור נתונים חיצוני, דבר התורם לשיפור הביצועים, משום שאין צורך לנעול רשותות.

הערה:

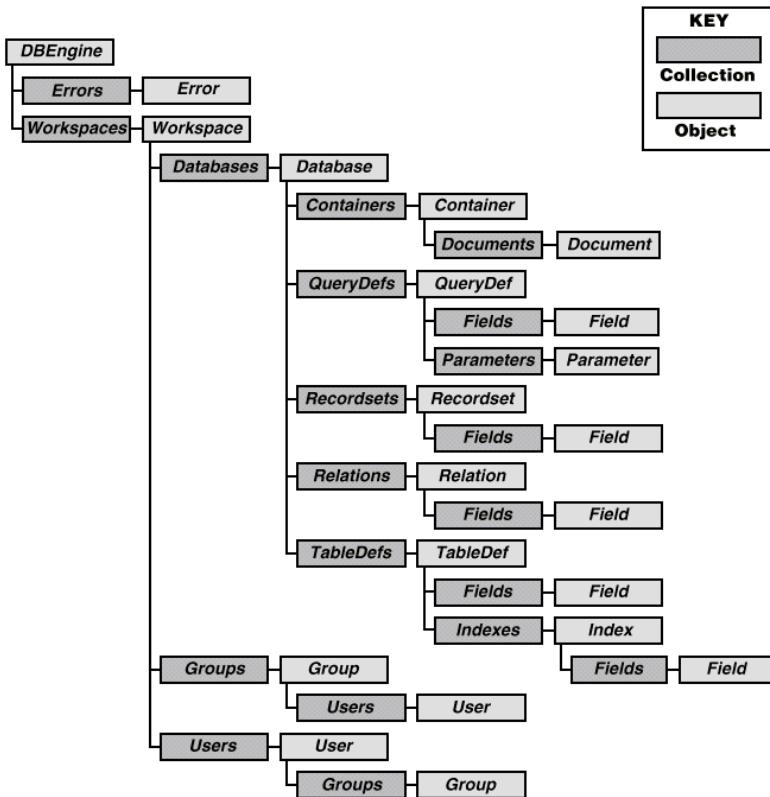


סמנים מגדרים את סוג ומיקום הגישה למקור נתונים. הגדרת סמן עשויה לכלול מספר מאפייני מקור נתונים. לדוגמה, סמן יכול להיות בר-עדכון או לא. סמן יכול לאפשר תנוצה קדימה בלבד או תנוצה דו-כיוונית. הם יכולים להתעדכן באופן אוטומטי כדי לשקר שינויים או הודעות במסד נתונים, והם יכולים לחיבר פעולות רענון מפורשת כדי להציג גרסה עדכנית ביותר של מסד הנתונים. תוכל ליעיד סמן שפועל בשורת מסד נתונים מרוחק, או בתחנת עבודה מקומית.

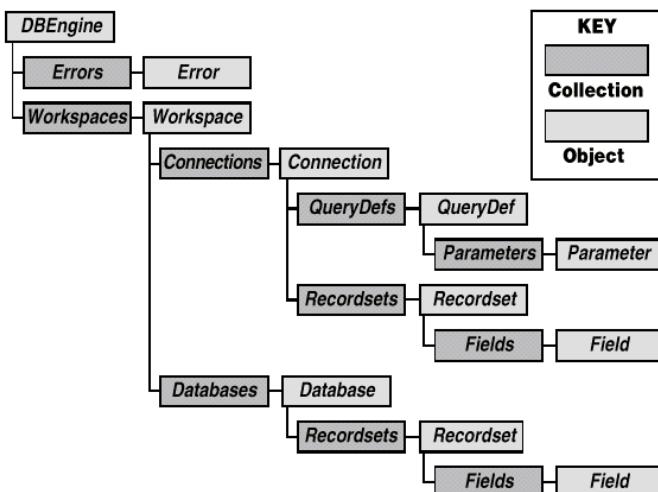
כדי להפיק את מירב התועלות משתי סביבות העבודה, עובוד עם סביבות עבודה מרובות של שני הסוגים כשהן פתרות בו-זמנית. פועלה זו מאפשרת לנצל היבט את פשטות הטעסים המאוגדים לניטונים שמקורם בסביבת העבודה של ODBC Direct. כל שעיליך לעשות לשם כך הוא להשתמש ברשומות המוחזרות בסביבת העבודה של Jet.

אובייקטים המשותפים לסביבות העבודה של Jet ו-ODBC Direct

בשני סוגי סביבות העבודה, אובייקטים של DAO מאורגנים בדרך כלל בצורה היררכית. תרשימים 2.1 מציג את ההיררכיה של האוסףים והאובייקטים של DAO בסביבות עבודה של Jet. תרשימים 2.2 מציג את ההיררכיה של האוסףים והאובייקטים של DAO בסביבות עבודה של ODBC Direct.



תרשים 2.1: אובייקטים וऑופים של DAO בסביבות עבודה של Jet



תרשים 2.2: אובייקטים וऑופים של DAO בסביבות עבודה של ODBC Direct

האובייקט DBEngine

האובייקט **DBEngine** הוא DAO בرمמה הגבולה ביוטר המועד לשתי סביבות העבודה. השיטה **CreateWorkspace** (יצירת סביבת עבודה) של האובייקט זה, משמשת לפתחת **הפעלה** (session). Type (סוג) הוא ארגומנט אופציוני של השיטה, שמאפשר להגדיר סביבת עבודה של Jet או סביבת עבודה של ODBC Direct. תוכל גם לקבוע את המאפיין DefaultType (סוג ברירת מחדל) של האובייקט DBEngine כך שאחת מסביבות העבודה תיפתח במקרה של העדר הגדרה מסוימת לשיטה **CreateWorkspace**. סביבת העבודה מסוג Jet היא ברירת המחדל הטבעית. הגדרה כלשהי של הארגומנט Type השיקס לشيخ CreateWorkspace תדרוך את המאפיין DefaultType, בין אם הוגדר מפורש, ובין אם לא.

המאפיינים והשיטות של DBEngine הזמינים לשני סוגי סביבות העבודה כולם קבוצת פונקציות DAO בסיסיות. השתמש ב-DBEngine ליצירת מסדי נתונים וניהולם.

השיטה **CreateDatabase** (יצירת מסד נתונים) יוצרת סביבות עבודה חדשות. תוכל להפעיל עס ארגומנטים כדי להגדיר את סדר המיוון ומצב הצפנה של מסד הנתונים. תוכל גם להגדיר את תבנית הגירסה של מסד נתונים כדי ליצור מסדי נתונים בצורה מתוכנתת, התואמים לgresאות קודמות של Access. השיטה **OpenDatabase** (פתחת מסד נתונים) פותחת מסד נתונים נוכחי באובייקט של סביבת עבודה. לאחר יצירת מסד נתונים תוכל להחיל עליו את השיטות **Compact** (דחסיה) ו-**Repair** (תיקון) כדי לנחל אותו.

DAO מאפשר לעבד **טרנזקציות** (Transaction) באמצעות שלוש שיטות של האובייקט **RollBack**, **CommitTrans**, **BeginTrans**: DBEngine. טרנזקציה היא קבוצת פעולות שמתבצעות בצורת הכל-או-לא-כלום. אם נכשל אחד השלבים בשרשראת פעולות מסד נתונים, ניתן לבטל את כל הפעולות (Roll back). לדוגמה, אם הבנק מעביר סכום כסף מחשבו עו"ש לתוכנית חישכון, שתי הפעולות יחד - חיוב העו"ש וזיכוי חשבונו החישכון - חייבות להסתיים בהצלחה כדי שהרישום בספרים יתאוזן. אם אחת הפעולות אינה מסתירה בהצלחה, יש לשחרר את המצב שקדם לפעולות בשני החשבונות. טרנזקציות יכולות להאיץ את תהליכי העיבוד במסדי הנתונים על ידי ארגון פעולות הכתיבה לדיסק בaczות. ניתןukan טרנזקציות עד חמיש רמות.

חלק מהשיטות והמאפיינים הזמינים תלויים בסוג סביבת העבודה, אך בכל סביבת עבודה, תהא אשר תהיה, DBEngine יכול את האוספים Errors (שגיאות) ו-Workspaces (סביבות עבודה).

האוף Errors

האוף **Errors** משמש לטיפול בשגיאות המתרחשות בעת גישה לנתונים. האוסף מכיל מספרי שגיאות וטיור המשפט האחרון שנכשל. בעבודה עם מקורות נתונים מסווג ODBC אפשר לקבל שגיאות רבות, מכיוון שכבות שונות של ODBC שעשוות לדוח על שגיאות שמקורן באוטה תקלת, כגון מקור מסד נתונים מרוחק שאינו פעיל.

האוסף Errors, בדומה לאוסף DAO אחרים, הוא מבוסס-אפס. המאפיין Count (מונה) מציין את מספר השגיאות באוסף, וכל שגיאה נושאת מספר מ-0 ועד -1. הערך Microsoft Visual Basic for Errors מקביל לאובייקט Err של VBA Applications.

במהלך ניפוי הungeons מהיישום, ייתכן שתגלה כי כדי לפשט את החיפוש אחר מקור השגיאה או פתרון הבעיה, כדאי למסוף את האוסף.

האוסף Workspace

המאפיינים והשירותות של האוסף **Workspace** משמשים כדי להפנות להפעולות סביבות עבודה בודדות. האוסף Workspace זמין תמיד Access, יכול לקיים הפעולות רבות בו-זמןית, וגם היישומים יכולים לפתח ולנהל צירופים שונים של סביבות Jet ו-ODBCDirect. הפעולות אלו אינן מתאפשרות מעבר להפעלת הכניסה, אך הן מסוגלות להתקיים לכל אורך הפעלה. באמצעות התבניות הבאות תוכל לנצל את הארגומנט Name (שם) של השיטה **CreateWorkspace** כדי להפנות בצורה ייחודית לסביבות עבודה מסוימות באוסף:

- DBEngine.Workspaces(0)
- DBEngine.Workspaces("Name")
- DBEngine.Workspaces![Name]

כל אובייקטי DAO מבוססים על תחביר הפניה זהה לתבניות שבדוגמה. שני הסוגנונות הראשונים עומדים בכללי ADO. בתחלת דרכך בתכונות DAO, عليك לאמץ אחת מהן כדי לשפר את מיומנותך ההגירה שלך.

אובייקטי Workspace משתמשות חשובות אחדות עם האובייקט DBEngine, כגון CreateUser, BeginTrans, OpenDatabase, CreateDatabase ו-Workspace.CreateGroup. שתי השירותות הללו מסייעות לנו לאבטחה ברמת-משתמש בתוך סביבת העבודה.

האוסף Database

באמצעות קוד אפשר לפתח מסדי נתונים רבים מתוך סביבת עבודה כלשהי. השימוש ב-DAO בשילוב עם VBA מעניק יתרון ברור על פני שימוש המשמש שמאפשר פתיחת מסד נתונים יחיד בזמן נתון. ניתן להציג את הגישה על ידי שימוש בשמות משתנים לצורך הפניה לאובייקטים של מסד נתונים. גישה זו מושפרת את מהירות הגישה לאובייקטי DAO רבים.

המאפיין Name של מסד נתונים בודד בסביבת עבודה של Jet, הוא הנתיב אל קובץ מסד הנתונים. קוד VBA מתייחס למסד הנתונים בפרויקט הנוכחי באמצעות הפונקציה CurrentDb. CurrentDb תומך גם בתחבר חלופי של: (0)(0) DBEngine מסד הנתונים הנוכחי. תחביר CurrentDb יוצר מופיע נוסף של פרויקט מסד הנתונים הנוכחי, אך תחביר DBEngine מתייחס לעותק הפתוח של מסד הנתונים הנוכחי. תוכל לפתח

אובייקטים של מסד נתונים במקורות נתונים שונים מ-Jet, כגון ISAM (לדוגמה, dBase או Microsoft FoxPro). בדומה זו ניתן להשתמש במקור נתונים ODBC, אך ההצעים ישתפרו כמשמעותם בסביבת העבודה ODBC (כפי שתואר בסעיף **אובייקטי סביבת העבודה של ODBC**).

האוסף Recordset

אובייקט **Recordset** (ערכת רשומות) מייצג רשומות בטבלה או את אלו שהתקבלו בתוצאה משאיילת החזרת שורות (row-returning). ערכת רשומות חדשה נוצרת באמצעות השיטה **OpenRecordset**. אפשר להפעיל את השיטה מתוך אובייקטים אחדים, לרבות אובייקטי מסד נתונים ו-TableDef, וליצור עזרתה ערכת רשומות. עם האובייקטים הנוספים הכלולים את השיטה **OpenRecordset**, נמנית אובייקטי QueryDef ו-Applet ערכות נוספות נוספות. שיטה זו מוסיפה ערכת רשומות חדשה לאוסף ערכות הרשומות. אפשר ליצור 5 סוגים ערכות רשומות, כאמור בטבלה הבאה.

אובייקטים של ערכת רשומות

סוג	תיאור
Table	סוג זה מתיחס לרשומה בטבלה, כגון זו שיוצרים באמצעות השיטה <code>CreateTableDef</code> . סוג מתיחס רק לטבלה יחידה של Jet. ניתן לעדכן ערכיה שדה, להוסיף ולמחוק רשומות. אין סמן ODBC מקביל
Dynaset	סוג זה הוא אוסף רשומות דינמי עשוי להיות תוצאה של טבלה אחת או יותר. השדות הנבחרים ניתנים לעדכן, וכן ניתן להוסיף, למחוק ולשנות רשומות. המאפיין <code>DataUpdatable</code> משמש לקביעה אם שדה כלשהו הוא בר-עדכן. במסד נתונים מרובה-משתמשים, ניתן להציג שינויים נבחרים שנערכו על ידי משתמשים אחרים. סוג ערכת רשומות זה מקביל לסמן keyset של ODBC
Snapshot	סוג זה מאפשר לבחון רשומות המבוססות על טבלה אחת או יותר, אך אינו מאפשר לשנות אותן. ברגע שתוועדים תמונה (<code>Snapshot</code>) לזכרון, היא אינה משקפת שינויים נוספים ככל שהם שנערכו בטבלאות. סוג ערכת רשומות זה מקביל לסמן static של ODBC
Forward-only	סוג זה זכה עקרונית לערכת הרשומות <code>Snapshot</code> , אך ניתן לגילוי אותו קדימה בלבד. סוג זה מקביל לסמן forward-only של ODBC
Dynamic	סוג זה מייצג תוצאה שאילתת המבוססת על טבלה אחת או יותר. משתמשים יכולים לעדכן את ערכת הרשומות על ידי הוספה, מחיקה ושינוי רשומות. סוג זה גם מציג שינויים שנערכו על ידי משתמשים נוספים בסביבה מרובה-משתמשים. סוג זה זמין רק בסביבות עבודה של ODBC ו-מקביל לסמן dynamic של ODBC

השיטה OpenRecordset. שיטה זו זוקקה לארגומנט מקור שמצוין את מקור תוצאת ערכת הרשומות. כפי שהוזכר, ארגומנט טיפוסי הוא שם טבלה, שם שאללה או משפט SQL. ניתן לציין גם את סוג ערכת הרשומות; אם אין מציינים, DAO מחזיר ערכת רשותות מסוג `dynaset`, `table` או `forward-only`, בהתאם למקור הנתונים.

ארגומנט אפשריות (options argument) מאפשר לציין אחת מספר תכונות של ערכת רשותות, כגון מניעת כתיבה לערכת רשותות או קריאה ממנה. ארגומנט סופי (final argument) מאפשר לקבוע את המאפיין LockEdits של ערכת רשותות. מאפיין זה מציין את סוג הנעילה התקיפה בעת שימוש מעדן, מושך או מוחק רשותות מתוך ערכת רשותות. בסביבות עבודה של Jet, ההגדירות שניתן לספק לארגומנט זה תהינה בדרך כלל `optimistic locks`, `read-only` ו- `pessimistic locks`. אם ההגדירה היא `Edit`, `read-only`, לא ניתן לעדכן את ערכת רשותות. שימושים אחרים יכולים לנעול את הדף שמכיל את הרשותה. במצב `optimistic locks`, `pessimistic locks` ו- `Update`. סידור זה יכול לשפר את עדכן את הרשותה עד שהשימוש מפעיל את השיטה `AddNew`. ביצועו של גורם `Update` יאפשר ערכות רשותות, אך עלול גם לגרום להתנגשויות בין פעולות עדכון.

שיטות Recordset. אפשר לשנות את הרשותות בערכת רשותות באמצעות `Update`-`Delete`, `AddNew`, `Edit`. השיטות `Edit` ו- `Update` משמשות במשולב לעדכן `Update` הערךים בערכת רשותות. השיטה `Edit` פותחת רשותה לבסיס נתונים לצורך עריכה, והשיטה `Update` מחלילה את הערךים החדים על הטבלאות שבבסיס נתונים. השיטות `AddNew` ו- `Update` פועלות במשולב. הקוד מודיע על הכוונה להוסיף רשותה על ידי הפעלת השיטה `AddNew`. לאחר מכן שומרים את הרשותה החדשה באמצעות השיטה `Update`. השיטה `Delete` מסירה את הרשותה הנוכחיית מערכת רשותות. לאחר `Delete` שרשומה נמחקת היא ממשיכה להתקיים עד למעבר לרשומה חדשה. השיטה `Update` אינה מחייבת את השימוש בשיטה `Edit`.

בעת הוספת ערכת רשותות לאוסף `Recordset` אפשר להשתמש בקבוצת השיטות **Move** (מעבר אל) כדי לנוטו אותה. השיטה **MoveNext** מנוטת אל הרשותה הבאה, והשיטה **MovePrevious** מנוטת אל הרשותה הקודמת. שנמצאים ברשותה הראשונה ומפעילים את השיטה `MovePrevious` DAO, `MovePrevious` מחזיר סמן **BOF** (Beginning Of File) ו- `EOF` (End Of File). ניסיון לנוט מעבר לסטמנים `BOF` או `EOF` יוצר שגיאת זמן ריצה (Run-Time Error). השיטה `Move` מאפשרת לציין מספר שורות קבוע מעבר. אפשר גם לקבוע מיקום התחלתי שונה מהרשומה הנוכחיית. השיטות **MoveLast** ו- **MoveFirst** עוברות ישירות לרשותה הראשונה או האخונה בערכת רשותות, בהתאם. בערכות רשותות גדולות מאוד, עשוי לחול השעה משמעותית במהלך הגיעו לרשותות הקצה שבערכת רשותות.

קבוצת שיטות נוספת מנוטת אל רשותה חדשה שעונה על קריטריונים מוגדרים. שיטות אלו הן **FindPrevious**, **FindNext**, **FindLast**, **FindFirst**. קריטריוני השיטה מסווג `Find` (חישוב) נקבעים באמצעות תחביר זהה לזה של הפסוקית WHERE שבמשפט SQL. אם לא קיימת ערכת רשותות שעונה על הקריטריונים שצוינו, שיטה אלו

תספוקנה למאפיין NoMatch של ערכות הרשומות את הערך True. אחרת, הן פשות תüberונה אל הרשימה שעונה על הקriterיוונים. השיטות FindPrevious ו-FindNext, FindFirst ו-FindLast מתחילה את החיפוש מהרשומה הנוכחית. השיטות FindFirst ו-FindLast, TableDef, Seek, Find, Move, OpenRecordset ו-SQL. בשימוש בשיטות Find, Seek ו-Move, רצוי להגדיר את המאפיין Index של ערכות רשומות, כדי לאפשר ארגון הרשומות בסדר הנכון שנקבע על ידי שדות האינדקס.

אובייקטי סביבת העבודה של Jet

אובייקטים של מסד נתונים בסביבות העבודה של Jet כוללים אלמנטים נבחרים של סכימת מסד נתונים, ובאפשרותם גם להפעיל אותם. לדוגמה, באפשרות לפתח ערכות רשומות לטיפול או להפעיל שאילתות פעולה שמעודכנות, מצרפות או מוחקות רשומות. שיטות של מסד נתונים אף מאפשרות ליצור ולנהל עותקים. לפניך חמשת האוספים ההיררכיים של מסד נתונים. האובייקט Database כולל שיטות להוספת אלמנטים חדשים לכל האוספים הללו.

- TableDefs
- Recordsets
- QueryDefs
- Relations
- Containers

האוסף TableDefs

אוסף זה מאפשר לגשת לאובייקטים בודדים של TableDef במסד נתונים. אובייקטים אלה מכילים את האוספים Fields ו-Indexes, ולכך הם מאפשרים להגדיר טבלה באמצעות אובייקט TableDef. באפשרות להשתמש בשיטות CreateField, CreateIndex, CreateIndex כדי להרכיב את הגדרת הטבלה. בעת השימוש בשיטה CreateField, מגדרים תחילת השדה על ידי ציון שמו, סוגו וגודלו. לאחר מכן מפעלים את השיטה Append כדי להוסף את השדה החדש לאוסף TableDef עבור אובייקט TableDef. אם אוסף מכיל כבר שדה בשם צוין כארוגומנט, הקוד שכתבת ייצור שגיאת זמן ריצה שנייתן לכוד אותה. תוכל לנצל זאת לניהול האובייקט TableDef למשל, על ידי הסרת השדה הישן באמצעות השיטה Delete.

בעת יצירת אינדקסים, מפעלים את השיטה CreateIndex ומיצרים לאינדקס שדה אחד או יותר. לאחר מכן מוסיפים את האינדקס החדש לאוסף Indexes עבור האובייקט TableDef. אם קיימים כבר אינדקס בשם זה, קיבל שגיאת זמן ריצה. תוכל לנצל שגיאות אלו כדי לנהל את תהליכי יצירת האינדקס של האובייקט TableDef.

האובייקט TableDef יכול לנתח קישורים אל טבלאות במקורות הנתונים ISAM ו-ODBC. הקוד דורש את המאפיינים Connect ו-SourceTableName ו-Connect והשיטה CreateTableDef. השיטה CreateTableDef מופעלת כדי להגדיר הפניה במשתנה לטבלה המקורית. אחר כך מגדרים את המאפיינים Connect ו-SourceTableName של המשתנה. המאפיין מצין את סוג מקור הנתונים, כגון 5.0 או x.5.x Paradox, ואת הנתיב אל מקור הנתונים הספציפי שלו את הקישור. המאפיין SourceTableName הוא שם הטבלה המקורית. לאחר קביעת מאפיינים אלה, משלימים את התהליך על ידי צירוף האובייקט TableDef לאוסף TableDefs.

ה老子 QueryDefs

אוסף זה מכחstan את האובייקטים הבודדים מסוג QueryDef במסד נתונים. אובייקט QueryDef הוא משפט SQL שמחזיר באופן טיפוסי קבוצת שורות או מבצע פעולה, כגון UNION, הוספה או מחיקת רשומות בערך רשות. כמשמעותם של אובייקט SQL של QueryDef מחזיר שורות, יכול להיות לו אוסף Fields ובו שדות ייחדים. אובייקט QueryDef יכול להכיל אוסף Parameters, אם משפט SQL שכבתה מקבל ארגומנטים שמציעים את הקריטריונים שלו בזמן ריצעה. כשהאובייקט QueryDef פועל, באפשרותן לציין פרמטרים אלה בזרה מתוכנת (באמצעות קוד), או לאפשר למשתמש לעשות זאת בזמן ריצעה באמצעות תיבת דו-שיה.

תוכל ליצור אובייקטים חדשים מסוג QueryDef באמצעות השיטה CreateQueryDef או עבור האובייקט DataBase בסביבת העבודה של Jet, או האובייקטים Connection ODBCDirect בנסיבות שאורכה שונה מאפס, DAO. יוסיף את השם אוטומטית לאוסף QueryDefs וישמור את האובייקט QueryDef באופן קבוע בדיסק יחד עם מסד הנתוניים. השתמש בשיטה Delete כדי להסיר פריט מהאוסף QueryDefs. כל אובייקט QueryDef שהמאפיין Name שלו הוא מחוץ באורך אפס, הוא אובייקט זמני, ו-DAO אינו מקיים אותו לאורך זמן. אובייקטים QueryDef זמינים נוחים לשימוש כהישומים צריכים ליצור אובייקטים כאלה באופן דינמי.

שתי שיטותאפשרות להפעיל אובייקט OpenRecordset מחזירה את השורות שבאובייקט QueryDef באמצעות המשפט SELECT, והשיטה Execute. השיטה dbFailOnError מאפשרת לאפשר לישום קבוע אם מבצע שאילתת פועלה. האפשרות dbFailOnError יכולה לאפשר לרשומות העומדות בדרישות אובייקט QueryDef נכשל בבחירה ועלתו המיוועדת על כל הרשומות העומדות בדרישות הקריטריוניים. כל עוד אובייקט QueryDef מצית לכללי התchapיר, הוא לא יגרום לשגיאה, גם במקרה שייכשל בבחירה פעולתו. האפשרות dbFailOnError מחייבת את המצב לקדומו (roll back), אם האובייקט QueryDef אינו מסוגל לבצע את כל השינויים. אפשרות זו אף יוצרת שגיאת זמן-ריצה כדי לסייע לבצע את פעולות העיבוד הקשורות בתהליכי, כגון הצגת הودעות למשתמש.

האוסף Relations

האוסף Relations (קשרי גומלין) ומערכת קשרי הגומלין שלו משמשים להגדרת קישורים (links) בין טבלאות באמצעות קוד. השיטה CreateRelation של האובייקט Database יכולה לאותל קשרי גומלין; היא מאפשרת ליישום להגדיר קשרי גומלין referential (מושג יחיד-ליקיד או יחיד-לרבים בין שתי טבלאות, קבועו של מושם קשרים integrity) ולארגן מחיקות ועדכונים על פי היררכיות קשרים (פרק 4 מרחיב בנושא). מגדירים קשרי גומלין בין טבלאות בהתאם לשדות משתפים בשתי הטבלאות. Relations Relation כולן אוסף Fields התומך בפונקציה זו. האוסף Relations ואובייקטיםבודדים מסוג Relation ייחודיים לסייעות העבודה של Jet. הם אינם זמינים בסביבות העבודה של ODBC Direct, מכיוון שמנגנוני מסדי נתונים מרוחקים שומרים באופן טיפוסי בעצםם על קשרי גומלין בין טבלאות.

האוסף Containers

אוסף זה מגדיר קבוצת אובייקטים של מכולה עבור מסמכים מסד נתונים. חלק מהאובייקטים מגעים מחלון מסד נתונים (Database: **טפסים, דוחות, Scripts** של מאקרו ומודולים. כל אלה הם אובייקטים של Access ולא אובייקטים של מסד הנתונים Jet. האובייקטים הנוספים של המכולה הם אובייקטים מבוסטי-Jet: **מסדי נתונים, טבלאות וקשרי גומלין**. אובייקט המכולה **טבלאות** כולל מידע על טבלאות ושאלות אחת. אובייקט מכולה נוסף מכיל נתונים מתארא של קשרי גומלין שנשמרו.

אובייקטי מכולה אחסנים מסמכים המבוססים על כל הרכיבים השמורים של סוג, כגון טפסים וקשרי גומלין. מסמכים אלה מספקים מידע מינהלי (ולא תוכן) על האובייקטים שבאובייקט מכולה. מאפיינים נחקרים של מסמכים כוללים תاريichi יצירה ועדכון של המסמך ואת שם בעליו והרשאות הטיפול בו. Jet משתמש במידע הכלול במסמך לניהול אבטחת האובייקטים של Access, ושל הטבלאות והשאלות שלו עצמו.

חשוב להבין כי מסמכים באובייקטי מכולה שונים מאלמנטים של אוסף. מסמכים מכילים אובייקטים שמורות – בין אם הם פתוחים ובין אם לאו. אוספים הם קבוצות אובייקטים פתוחים. אם אובייקט אינו פתוח, הוא אינו מהווים חלק מאוסף, אך הוא עשוי להשתיך לאובייקט מכולה. בנוסף, מסמכים מכילים מידע מינהלי על אובייקטים, אך אלמנטים באוסף מכילים מידע על תוכן, עיצוב ואלמנטים מנויים של האובייקטים שבאוסף.

האוספים Users (משתמשים) ו-Groups (קבוצות) מכילים אובייקטים תואמים שלהם. אוספים ואובייקטים אלה משלימים את מסמכי המכולה כדי לסייע ל-Jet להניל אבטחה ברמת-משתמש. מסמכים כוללים הרשות. משתמשים שייכים לקבוצות. הרשות מסמך מתארות רמות גישה של משתמשים וקבוצות. פרק 10 עוסק בהרחבה באבטחה ברמת-משתמש, לרבות משתמשים, קבוצות והרשאות.

אובייקט סביבת העבודה של ODBC

כשמשווים את תרשימים 2.2 ו-2.1 לתרשים 2.2, ניתן להבחין כי מודול ODBC Direct חסכו יותר, מכיוון ששביבות העבודה של ODBC Direct מעבירות לשרתים מסדי נתונים מרוחקים חלק מהפונקציות שמנהל Jet. לדוגמה, שרתים מסדי נתונים מרוחקים מנהלים את האבטחה בעצם, ולכן אין צורך באוטופים Users ו-Groups. בנוסף, לא קיים אוסף TableDefs, מכיוון שהשרתים מסדי נתונים מרוחקים מנהלים את טבלאותיהם בעצם. כך הדבר גם לגבי קשרים.

ישנם הבדלים נוספים בין שני מודלי סביבות העבודה: מודול ODBC Direct כולל אוסף Connections חדש בעל אובייקטים מקבילים שלו. אובייקט Database ממלא תפקיד אחר במודול ODBC Direct, אובייקטי QueryDef אינם מכילים אוסף Fields. ניתן לגזרו אובייקט Recordset מאובייקט QueryDef באמצעות השיטה OpenRecordset.

למרות שתתיי סביבות העבודה כוללות אובייקט מסוג Database, התנהגות האובייקט בסביבת העבודה של ODBC Direct שונה במידת-מה מהתנהגותו בסביבת העבודה של Jet. בסביבת העבודה של ODBC Direct, אובייקט זה מכיל את המאפיין Count שמחזיר הפניה אל אובייקט Connection. האובייקט Connection מכיל את המאפיין Database שמחזיר הפניה אל אובייקט DAO. במודול DAO, האובייקטים Database ו-Connection מחוונים אל אובייקט Database של Jet אל סביבות העבודה של ODBC Direct וחזרה.

האוסף Connections

האוסף Connections של סביבת העבודה והאובייקטים שלו מהווים גורמים קריטיים בעת ניהול נתונים מרוחקים. בסביבות העבודה של ODBC Direct מנצלים את השיטה OpenConnection כדי להקים חיבור (connection) אל מסד נתונים מרוחק. הדבר מחייב ארבעה ארגומנטים, שלושה מהם אופציונליים. הארגומנט הנדרש הוא השם. מתן שם לחיבור מוסיף אותו לאוסף Connections. שלושת הארגומנטים האחרים מגדירים את טיבו של החיבור. לאחר שהם אופציונליים, ניתן להגדירם בעת יצירת החיבור או לאחר מכן. לעומת זאת, יש להגדיר חיבור בטרם ניתן יהיה להשתמש בו לשילפת נתונים לביצוע פעולה אחרת כלשהי במקור הנתונים המרוחק. לארוגומנט Connect חשיבות מיוחדת, בהיותו הארגומנט שקובע את המאפיין Connect של האובייקט Connection. ארגומנט זה מגדיר את מחורזות החיבור. הוא מתחיל ב-ODBC ותו נקודת-פסיק (;) ולאחר מכן נתונים החיבור הדורשים לקישור אל מקור הנתונים המרוחק. נתונים אלה יכולים לכלול שם מסד נתונים ושם קישור B-ODBC ומהם גם יכולים להכיל שם משתמש וסיסמה תואמת. מפרידים בין סוגים המידע השונים של הארגומנט באמצעות נקודת-פסיק. יישום יכול לבדוק ולתחל את מחורזות החיבור באמצעות המאפיין Connect של האובייקט Connection.

פרמטר אופציוני נוסף של השיטה OpenConnection שולט בשני תפקודים נפרדים: אופן התגובה של חיבור למיידע לא שלם של מחורזות חיבור, וצורת הפתיחה האсинכרונית של חיבור. במקרה של מחורזות חיבור שהמידע שבה אינו שלם, ניתן

לאפשר לחברו להיכשל וליצור שגיאת זמן ריצה, או לכלכוד את השגיאה ולבקש מהמשתמש להשלים את המידיע. באפשרות גם לנצל פרטמר זה כדי לקבע פתיחה אסינכרונית של חיבור. היחסום יכול לפתח את החיבור ואחר כך לעבור למשימות אחרות. יישום Access יכול לשרת את המשתמש המקומי על ידיפתיחה טפסים או אפילו קיום חילופי תקשורת עם המשתמש. באותו זמן ממש, שרת מסד הנתונים המרוחק מטפל בבקשת הקמת החיבור החדש. המאפיין StillExecuting של האובייקט Connection מאפשר ליישום לדגום את החיבור כדי לקבע את זמינות השימוש שלו.

האובייקט Connection כולל חמיש שיטות:

◀ **OpenRecordset**. שיטה זו מחזירה קבוצת שורות מקור נתונים מרוחק. מספר רכיבי השיטה בסביבת עבודה של ODBCDirect רב יותר מזה שבביבת העבודה של Jet. קרובה לוודאי שההבדל הגדל הגדול ביותר ביןיהם הוא שבעת הפעלת השיטה בסביבת עבודה של ODBCDirect ניתן לציין יותר ממשפט SQL אחד, וכך יכול משפט OpenRecordset

◀ .**Close**. שיטה זו סורגת חיבור פתוח.

◀ **CreateQueryDef**. גם שיטה זו כוללת רכיבים נוספים בסביבת העבודה של ODBCDirect. מה שברור הוא שאובייקט QueryDef איןם כוללים שdots. אם ברצונך להציג את השורות שהוחזרו באמצעות אובייקט QueryDef, עליך להפעיל את השיטה OpenRecordset של אובייקט זה. כל האובייקטים מסוג QueryDef בסביבות עבודה של ODBCDirect הם זמינים. במודול האובייקט לא ניתן ליצור שיגרה מאוחסנת במקור נתונים חיצוני. אובייקטי QueryDef יכולים להשתיך לאובייקט Connection בלבד. השיטה CreateQueryDef אינה קיימת עבור אובייקטים מסוג Database בסביבת עבודה של ODBCDirect, כפי שקיים כזו בסביבת עבודה של Jet. באפשרות לפתח אובייקט Recordset מתוך אובייקט Database בשתי סביבות העבודה.

◀ **Execute**. שיטה זו מפעילה שאליות פעולה, שאילתות פרטמר ושאליות בחירה. מגדרים את הקבוע dbRunAsync כדי לצין שעל האובייקט QueryDef לפעול בצורה אסינכרונית. בדיקן כמו במקורה השיטה Execute, משתמשים יכולים לבצע משימות במקביל לפעולות האובייקט QueryDef. המאפיין StillExecuting מאפשר לישום לבדוק את מצב ההשלמה של האובייקט QueryDef.

◀ **Cancel**. הפעלת השיטה ללא ציון פעולה אסינכרונית תסימן שאילתת אסינכרונית ותחזיר שגיאת זמן ריצה. שחרור משאים שצרך אובייקט QueryDef מתבצע על ידי שיטת Close, או הגדרה של הפניות האובייקט כ-Nothing.

עדכן באצווה

עדכן באצווה (batch updating) הוא אחד החידושים החזקים בסביבות העבודה של ODBCDirect. הוא מאפשר ליישום לטוען קבוצת רשומות, לעדכן אותן בזורה מקומית, ואחר כך לעדכן את רשומות המקור באצווה בודדת, במקום לעדכן כל אחת בנפרד.

לעדכו אצונות יתרון ברור על נעלית רשותה בודדת. עקב הסיכון להתנגשויות, מומלץ לפעול בדרך זו רק במקרים בהם קטינים הסיכון ממשמשים וربים יערכו شيئاוים בסיסד הנזונים. אך עיבוד באצונה כולל מספר רכיבים מוכלים לטיפול בהתנגשויות. לדוגמה, המאפיין BatchCollisions מחזיר סימניות שמצביעות על התנגשויות בערכת רשותה לאחר טעינתה. אפשר לכפות על מסד נתונים מרוחק להיות תואם לעדכו המבוצע, או לקבל את הערך שבמקור הנתונים המרוחק. שלושה מאפייני Fieldмарוצים לאפשר לבדוק ערך מקורי טרם טעינתו, ערך מעודכן בגרסת ערכת הרשותה המקומית ואת ערך השדה החדש במקור הנתונים המרוחק.

- לפניך חמוץ השלבים ליישום עדכו באצונה בסביבת עבודה של ODBC Direct:
1. הגדר את המאפיין DefaultCursorDriver של סביבת העבודה בתור .dbUserClientBatchCursor
 2. צור אובייקט Connection או אובייקט Database
 3. הפעיל את השיטה OpenRecordset עבור האובייקט שיוצרת בשלב 2 באמצעות הגדרת dbOptimisticBatch עבור הארגומנט LockEdits.
 4. ערכז את השדה בצורה מקומית, לפי הצורך.
 5. הפעיל את השיטה Update של ערכת הרשותה של שלב 3 באמצעות הגדרת dbUpdateBatch עבור ארגומנט הסוג. אם לאairaו התנגשויות כלשהן, סיימת בהצלחה. אם קיימות התנגשויות, תזדקק לוגיקה נוספת כדי לישב אותן.

סקירה כללית על ADO

ADO 2.1 תומך בגרסה 2.1 של ADO, הכוללת שלושה מודלים לגישה נתונים: הספרייה ADODB, הספרייה ADOX והספרייה JRO. חלוקת הגישה לשולש ספריות מאפשרת לתוכניות שאינן זוקקות לכל שלושת המודלים להשתמש רק בחלקם. מרכיב נוסף באסטרטגיית הגישה נתונים של Access הוא האמון בספקי DB. ספקים אלה פועלים בשיתוף עם ADO כדי לספק גישה למקורות נתונים מוקובלים ולחדשים אחד, כגון ספריות דואר אלקטטרוני. סידור זה מעיצם את התכונות של מסדי נתונים.

ספרייה ADODB היא ספריה קטנה ופשוטה, שמכילה אובייקטים חיווניים ומספקת את הרכיבים הבסיסיים לייצירת חיבורים, השרת פקודות ואחיזור ערכות רשותות. ספריה זו גם מאפשרת לנוט בערכת רשותות. ניתן להשתמש בה לביצוע מטלות אחזקה בסיסיות, כגון שינוי, הוספה ומחיקה של רשותות. העיצוב הלא היררכי של הספרייה מקל על משתמשים מתחילה.

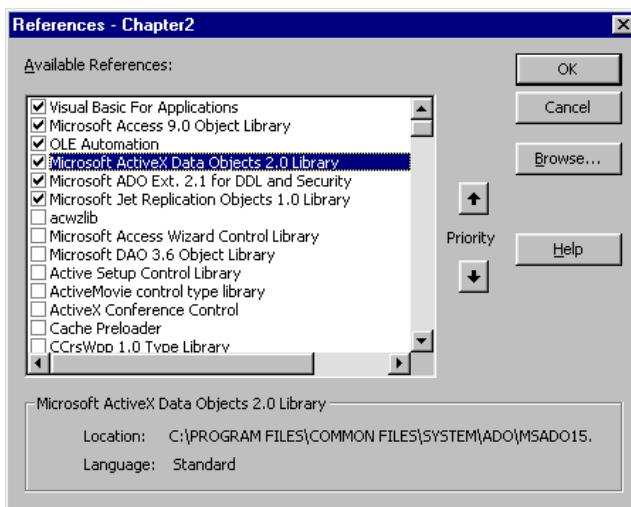
ספרייה ADOX תומכת בשפת הגדרת נתונים ובמושגים הקשורים באבטחה. היא כוללת אובייקטים המקשרים את המשמש עם הסכמה הכוללת של מסד נתונים. לדוגמה, היא מאפשרת ליצור טבלאות וקשרים. המודול כולל תמייה של מילויים קשרים (referential integrity), עדכו שדות קשרים ומחייבת רשותות קשריות, ומציע שגורות, תצוגות ואופנים של Groups ו-Users לאבטחת מסד נתונים ברמת-משתמש.

הספריה JRO מאפשרת יצירה שכפול של מסד נתונים Jet ו-SQL Server. פרק 11 עוסק בהרחבת שכפול מסד נתונים.

היתרון העיקרי של ADO הוא מודל האירוע. על ידי כך הוא משחרר את היישום מהצורך לבצע אסינכרוניות, אך ADO גם מספק אירועים. במקום זאת, ניתן ליצור מטפלים לbehör אובייקט ובדיקה המאפיין StillExecuting. בפרק הבא נסביר כיצד לבנות אירועים שיגיבו על אירועים בעת התרחשותם (בפרק הבא נסביר כיצד לבנות מטפלים באירועים).

ספק OLE DB מסוימים מפתח את העוצמה של ADO. הם מספקים דרך חדשה לגשת נתונים מרוחקים, דרך שמתבססת על ODBC ומרחיבה את יכולתה, ומציעים גישה למסדי נתונים יחסיים ולמקורות נתונים לא יחסיים באמצעות ממשק ODBC עקיби. Microsoft Access 2000 משוכן עם מילוי של ספק OLE, כולל כללה עבור OLE, SQL Server, Jet, Oracle, MySQL, מקורות נתונים ODBC כללים וגם מקורות לא מקובלים, כגון Microsoft Index Server ו-Microsoft Active Directory Service. בפואז הזמן יש לצפות לגידול במספר ספקים אלה.

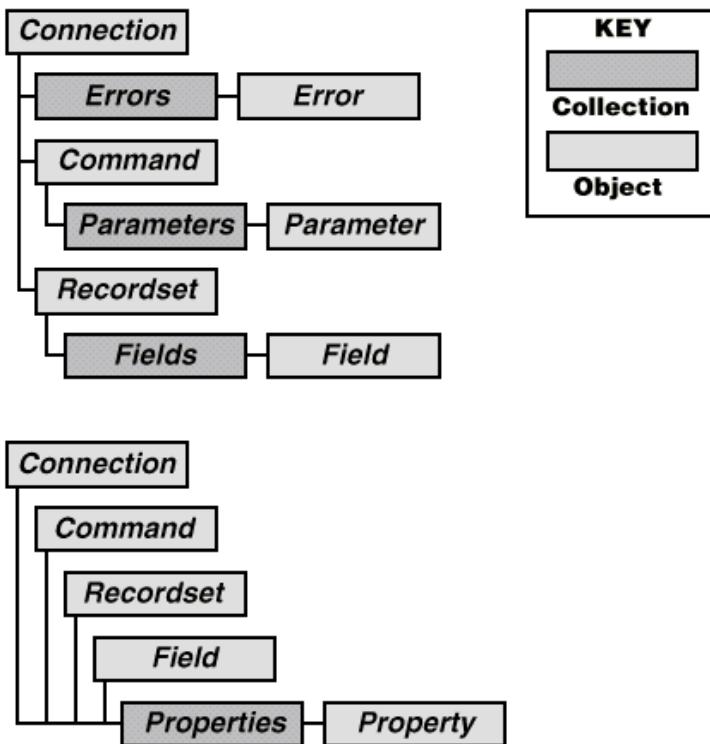
בטרם תוכל להשתמש בספרייה ADO כלשהי, عليك ליצור הפניה אל ספריה אחת לפחות. יוצרים הפניה מתוך החלון Visual Basic Editor (VBE) באמצעות הפקודה References (הפניות) בתפריט Tools (כלים). תרשימים 2.3 מציג את תיבת הדו-שיח References כשלוש הספריות נבחרו. נוח יותר לבחור את כל השלוש, אך בחירת הספרייה הדרישה בלבד מושמרת יותר מאשרים. בחר לניסיוני ספריות שונות בישומים שונים הפעילים במחשבים שונים, כדי לקבוע את הצורך המתאים ביותר לסייעת המחשב. אם ברשותך יישום המשמש לuebaודת שוטפת, שפועל בסוגי מחשבים שונים, عليك לשמור משאבים עבור דרישות אחרות של היישום.



תרשימים 2.3: תיבת הדו-שיח References משמשת להוספה ספריות ADO ליישום

הספרייה ADODB

ספריית האובייקטים ADODB כוללת שבעה אובייקטים עיקריים. ארבעה מתוכם כוללים אוסףים. האובייקט Connection מופיע בראש המבנה ההיררכי, אך ניתן ליצור חיבורים באמצעות (implicit) באמצעות אובייקטים אחרים. האובייקטים Properties, Recordset, Command ו-Field.



תרשים 2.4: ספריית האובייקטים ADODB

האובייקט Connection

אובייקט זה יוצר קישור אל מסד נתונים. משתמשים באובייקט Connection בצורה مستטטית או מפורשת (explicit) בעת טיפול במסד נתונים. בעת יירה מפורשת של קישור, ניתן להל בפעולות חיבור אחד או יותר ולהקצות מחדש את תפקידיהם בישום. בעת יירה مستטטית של חיבור ניתן לתקן את הקוד. כל אובייקט חדש שנוצר באמצעות חיבור مستטט, צורך יותר משאבים. אם היישום מכיל אובייקט אחד או שניים לכל היותר, שככל מהס דורש חיבור נפרד, חיבורים مستטטיים ייטיבו לענות על צורך. ADO מאפשר לבחור את דרך היצירה והניהול של חיבורים באופן שתמצא הנוח.

בניגוד ל-DAO, היא שפה כללית לגישה לנוטונים, ולכן לא כל השיטות והמאפיינים שלה מתאימים למנגנון Jet. לעומת זאת קיים ספק DB OLE Jet 4 עבור Access 2000以上. האובייקטים Connection Connection תלויים במכשירים נוספים במכשירים Jet. מטרו. שטויות ספק של Jet. בעת הפניה אל מסד נתונים שנמצא בקובץ אחר, יתכן שטריצה לשלול פרמטרו. Data Source שמצוין על המיקום הפיסי של מסד הנתונים שאינו נמצא בפרויקט הנוכחי.

הקוד שלפניך הוא דוגמה פשוטה לפתח מסד הנתונים המוכר Northwind. שים לב כי מושפט Dim מצהיר ויוצר הפניה אל cnnNorthwind Connection בטור אובייקט cnnNorthwind Open ב-Provider Provider (ספק) ו- (מקור נתונים) shigra. Shims לב כי הפרמטרים Data Source (ספק) ו- (מקור נתונים) מופיעים בין גרשאים. פרמטר Provider מציין על ספק של Jet 4, ופרמטר Data Source מציב על המיקום הפיסי של מסד הנתונים Northwind.

```
Sub OpenMyDB()
    Dim cnnNorthwind As New Connection
    Dim rsCustomers As Recordset
    ' Create the connection.
    cnnNorthwind.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Program Files\Microsoft Office\Office\" & _
        "Samples\Northwind.mdb;"
    ' Create recordset reference and set its properties.
    Set rsCustomers = New ADODB.Recordset
    rsCustomers.CursorType = adOpenKeyset
    rsCustomers.LockType = adLockOptimistic
    ' Open recordset and print a test record.
    rsCustomers.Open "Customers", cnnNorthwind, , , adCmdTable
    Debug.Print rsCustomers.Fields(0).Value, rsCustomers.Fields(1).Value
    rsCustomers.Close
    cnnNorthwind.Close
End Sub
```

לאחר יצירת הפניה אל החיבור, הקוד יוצר אובייקט Recordset. הקוד יוצר הפניה אל משתנה האובייקט המצביע את ערכת הרשומות, ולאחר מכן מצביע ערכיהם בכמה מאפיינים של ערכת הרשומות. קטע הקוד האחרון פותח את ערכת הרשומות ומדפיס שני שדות מתוך הרשומה הראשונה. השיטה Open של אובייקט Recordset יכולה להפנות חיבור אל מסד נתונים וגם אל מקור רשותות כלשהו במסד הנתונים. הקוד בוחר את כל הרשותות מהטבלה Customers שבמסד הנתונים Northwind. השיטה Open הופכת מלכתחילה את הרשותה הראשונה לזמן עברו היישום.

שתי השורות המסייעות את קטע הקוד האחרון, סוגרות את ערכת הרשומות ולאחר מכן גם את החיבור. סגורות חיבור הופכת את כל האובייקטים שמנפניהם אליו, כגון אובייקט Recordset, לבלי פעלילים. כל ניסיון להגדיר מאפיינים או להפעיל שיטות

עבור ערכת רשומות שמננה לחבר סגור, יוצר שגיאת זמן ריצה. יצירת חיבור בקרה משתמש עשויה להיות טובה יותר, מכיוון שהאובייקט יכול לשמש לחבר רק במקרה קיום האובייקט.

הקוד שלפניך פותח גם הוא ערכת רשומות המבוססת על הטבלה Customers במאסן נתוניים Northwind ומדפיס את הרשימה הראשונה. אך הפעולה מבוצעת באמצעות קוד פשוט יותר ומספר שורות קוד קטן יותר, כיון שהחיבור נותר בקרה משתמש תוך התבאסות על מספר הגדרות ברירת מחדל גדול יותר.

```
Sub OpenFast()
Dim rsCustomers As Recordset
    Set rsCustomers = New ADODB.Recordset
    ' Less code, but potentially greater resource consumption
    rsCustomers.Open "customers", "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Program Files\Microsoft Office\Office\" & _
        "Samples\Northwind.mdb;"
    Debug.Print rsCustomers.Fields(0), rsCustomers.Fields(1)
    rsCustomers.Close
End Sub
```

מהחר שלא קיים חיבור מפורש, השגירה OpenFast אינה צריכה להציגו על אובייקט חיבור (ולכן אינה חייבת לפתוח או לסתור אותו). כפי שניתן לראות, השיטה Open של אובייקט ערכות רשומות יכולה להכיל את נתוני החיבור החינויים של ספק ושל מקור נתונים. הקוד שהוצג מכל רק פרט אחד נוסף – מקור ערכות הרשומות הוא הטבלה Customers. השיטה Open מתבססת על הגדרות ברירת המחדל CursorType – LockType, שמייצגות את המאפיינים קדימה בלבד (forward-only) וקריאה בלבד (read-only). הגדרות אלו נועדו לפעולות מהירות מאוד, אך הן אינן מספקות תפקודיות רבה. אך אם ההגדירות הולמות את צורך ומאפשרות להפנות את תשומת ליבך להיבטים אחרים של פיתוח יישומים, יתכן שהן הבירה הטובה ביותר.

המאפיין Mode (מצב). כברירת מחדל, השיטה Open של האובייקט Connection יוצרת מסד נתונים עבור גישה מסוותפת. לעומת זאת, באפשרות להגדיר את המאפיין Mode של אובייקט זה בתורה אחת משבע ההגדירות שמשמעותן ורמות שונות של גישה מוגבלת למסד נתונים. הגדרות מצב אלו מתייחסות לכל ערכות הרשומות והפקודות שמקצתו לחבר למאפיין שלhn ActiveConnection מציג את השפעת הגדרת המצב קריאה- בלבד על היכולת לעדכן ערכות רשומות.

```
Sub OpenLookOnly()
Dim cnn1 As New Connection
Dim rsCustomers As Recordset
    'cnn1.Mode = adModeRead
    cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Program Files\Microsoft Office\Office\" & _
        "Samples\Northwind.mdb;"
```

```
'Use the next line to determine the default Mode setting.
```

```
Debug.Print cnn1.Mode  
Set rsCustomers = New ADODB.Recordset  
rsCustomers.ActiveConnection = cnn1  
rsCustomers.Open "Customers"  
  
' An adModeRead setting for cnn1.Mode causes an error in this procedure.  
' Remove the comment from the cnn1.Mode line to see an error here.  
rsCustomers.Fields("CustomerID") = "xxxxx"  
rsCustomers.Update  
Debug.Print rsCustomers.Fields("CustomerID")  
rsCustomers.Close  
End Sub
```

השлага OpenLockOnly מצהירה על אובייקט Connection חדש בשורת הקוד הראשונה שלה. השורה השלישית (אם תבטל את סימוננה בתור הערא) מגדירה את המאפיין Mode של החיבור בתור adModeRead כדי לאפשר גישה מסווג קריאה-בלבד. אם נזין בשתי שורות הקוד הבאות, נראה כי השיטה Open הופכת את ערכת הרשומות CustomerID(rsCustomer) לזוינה. זוג השורות הבא מנסה לעדכן את ערך השדה לשומה הראשונה. אם תסיר את הערא בשורה השלישית, עדכוני אלה יגרמו לשגיאה, מכיוון שלא ניתן לעדכן מסד נתונים לקריאה-בלבד.

הטבלה ש לפניה מתארת את שמות הקבועים שבאמצעותם ניתן להגדיר את המאפיין Mode של חיבור. באפשרותך לנצל קבועים אלה כדי לשЛОט בסוג העריכה שיכולים לבצע משתמש אחד או יותר באמצעות חיבור לערכת רשומות.

קבועים שגדירים את המאפיין Mode של אובייקט החיבור

תקף	ערך	קבוע
הרשאות לא הוגדרו או נקבעו	0	adModeUnknown
הרשאת קריאה-בלבד	1	adModeRead
הרשאת כתיבה-בלבד	2	adModeWrite
הרשאת קריאה/כתביה	3	adModeReadWrite
מנע אחרים לפתח מקור רשומות עם זכויות קריאה	4	adModeShareDenyRead
מנע אחרים לפתח מקור רשומות עם זכויות כתיבה	8	adModeShareDenyWrite
מנע אחרים לפתח את החיבור	12	adModeShareExclusive
גישה משותפת (ברירת המחדל)	16	adModeShareDenyNone

השיטה OpenSchema. השיטה OpenSchema של האובייקט Connection מאפשרת ליישום לעיין באובייקטים שבօספים הזמינים באמצעות חיבור מבלי למספר האלמנטים ברשימה. פلت השיטה עשוי להכיל מידע על טבלאות, תצוגות, שגורות,

איןדקטים עוד. הפרטים הספציפיים תלויים באופן בו ספק OLE DB נתון מימוש את יכולות השיטה הכלליות. הקוד שלבסוף מנצח את השיטה OpenSchema ייחד עם ספק Jet 4 כדי להציג את התוצאות הזרזנות באמצעות חיבור.

```
Public Sub OpenSchemaX()
Dim cnn1 As New ADODB.Connection
Dim rstSchema As ADODB.Recordset

cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office\Office\" & _
    "Samples\Northwind.mdb;"

Set rstSchema = cnn1.OpenSchema(adSchemaTables)

' Print just views; other selection criteria include
' TABLE, ACCESS TABLE, and SYSTEM TABLE.
Do Until rstSchema.EOF
    If rstSchema.Fields("TABLE_TYPE") = "VIEW" Then
        Debug.Print "View name: " & rstSchema.Fields("TABLE_NAME") & vbCrLf
    End If
    rstSchema.MoveNext
Loop
rstSchema.Close
cnn1.Close
End Sub
```

השיגרה פותחת בהצחה על חיבור ועל ערכת רשומות. ערכת הרשומות מכילה את הפלט שיוצרה השיטה OpenSchema. ארגומנט השיטה OpenSchema מצין כי אלמנטים של התחום Tables של סכימת נתונים ייצבו ערכיהם ברשומות. לעומת זאת, השיטה OpenSchema עוקבת אחר סוג טבלה אחדים, כולל תצוגות, טבלאות משתמש רגילות, טבלאות מערכות מיוחדות, טבלה נוספת של אובייקטי Access וטבלאות הקשורות. הקוד שהווג מדפיס את פלט השיטה עבור תצוגות בלבד.

האובייקט Recordset

ערכת רשומות היא מבנה תוכני (programmatic) לטיפול ברשומות. ניתן לבסס את הרשומות על טבלה או על תצוגה בפרויקט הנוכחי, או על קובץ אחר, משפט SQL או פקודה שמחזירה שורות. אפשרויות הטיפול בערכת רשומות תלויות בספק OLE DB שלה ובתכונות מקור הנתונים המקוריים.

באפשרות לשולף ערכות רשומות באמצעות אובייקטים אחרים, כגון חיבורים ופקודות, אך מיגוזן המאפיינים והשיטות העשר של האובייקט Recordset הופך אותו לבחירה הטבעית לביצוע חלק גדול מעיבוד הנתונים. ניתן להשתמש בערכות רשומות כדי לבצע פעולות רבות נגד קבוצת שורות: ניווט בין שורות; הדפסת תוכן השורות,

כולן או מקצתן ; הוספה, עדכון ומחיקה של רשומות ; חישוב רשומות ; וסינון רשומות כדי לבחור קבוצת-משנה כלשהי מתוך ערכת הרשומות המלאה. מבחינה היסטורית, ערכות רשומות היו ונותרו אובייקטים לא מתמידים – הן קיימות רק כל עוד הן פתוחות בתוכנית. גירסה 2.1 של ADO המשווקת יחד עם 2000 כוללת ערכות רשומות מתמידות, אותן אפשר לשמר בדיסק ולפתח במועד מאוחר יותר.

המאפיין ActiveConnection. מאפיין ערכת רשומות זה מאפשר לישום לנצל חיבור פתוח כדי לתמוך בערכת רשומות. ניתן לקבוע מאפיין זה בזמן כלשהו לאחר הגדרת האובייקט עבור ערכת הרשומות. השימוש במאפיין זה מפשט את משפט השיטה Open של ערכת הרשומות, כיון שהוא מבטל את הצורך לכלול את נתוני החיבור. בעת הגדרת המאפיין מראש, איןך צריך אפילו להתייחס לחיבור קיים באמצעות השיטה Open.

השיטה Open. שיטת ערכת רשומות זו מהווה נתיב מקובל להפעלת ערכת רשומות לזמןמה בשירה. ארגומנט המקור הוא הארגומנט הקרייטי ביוטר עבורה. הוא מצין את מקור הנתונים שעליו מבוססת השיטה את האובייקט שהיא פותחת. אפשרויות טיפוסיות של ארגומנט המקור כוללות טבלה, SQL, משבט Options של השיטה Open כדי לתאר את או שירה מאוחסנת. משתמשים בארגומנט Options של השיטה Open כדי לתאר את סוג המקור בעת פתיחת ערכת רשומות.

סוג הסמן. סוג הסמן הוא אחד הרכיבים הבסיסיים ביוטר של ערכת רשומות. סוג הסמן קובע את אופן הניוט בערכת הרשומות ואת סוג הנסיבות שנitin להחיל עליה.ADO תומכת באربעה סוגים סמן :

◀ **Dynamic**. סוג סמן זה מאפשר למשתמשים להציג שינויים במקור נתונים, שביצעו משתמשים אחרים. הוא מאפשר פונקציות אחזה של ערכת רשומות, כגון הוספה, שינוי ומחיקה של רשומות, וmarshaling נתונים דו-כיווני בסיס נתונים, מבלי להסתמך על סימניות.

◀ **Keyset**. סמן זה כולל את רוב המאפיינים של סמן דינמי, למעט גישה מיידית לשינויים שביצעו משתמשים אחרים במקור נתונים. דרך אפשרית להציג שינויים ככל היא על ידי הפעלת השיטה Requery של ערכת רשומות.

◀ **Static**. סמן זה הוא תמורה של ערכת רשומות בנקודת זמן מסוימת. הוא מאפשר ניוט דו-כיווני. לא ניתן להציג שינויים שביצעו משתמשים אחרים בסיס נתונים. גירסה 4 ואילך של Microsoft Internet Explorer תומכות בסוג זה בהיותו סמן צד-הלקוח הגמיש ביוטר של הדף.

◀ **Forward-only**. סמן זה, הנקרא לפעמים סמן ברוז-שריפה (hydrant), פועל בכיוון אחד ומסוגל להציג את ביצועי הסמן. זהו סמן ברירת המחדל של O. אם אתה זוקק לסוג סמן אחר, עלייך להגדיר את המאפיין CursorType, בטרם תפתח את ערכת הרשומות.



הערה:

הגדרת סוג הסמן מתחשרת עם הגדרות סוג הנעילה. אם תגדיר סמן מסווג forward-only בעל סוג נעילה שונה מ- (adLockReadOnly) read-only, ידרשו את הגדרת סוג הסמן שקבעת. לדוגמה, אם קבועים נעילה אופטימית, ADO ממיר אוטומטית סמן forward-only לסוג ketset.

המאפיין LockType. מאפיין זה מתקשר חלקית עם סוג הסמן, מכיוון שהוא יכול לכך יטפלו משתמשים בערכת רשותות. הגדרה אחת של סוג נעילה (adLockReadOnly) תואמת באופן ייחודי לסטנים מסווג forward-only. זהו סוג ברירת המחדל נעילה. הטבלה ש לפניך מתארת את ארבע ההגדרות האפשריות של המאפיין LockType. הגדירה שלפניהם מתחילה את ארבע ההגדרות במיוחד האפשרות של adLockBatchOptimistic עבור מסדי נתונים מרוחקים, כגון SQL Server או Oracle, ב比拟ו למסדי נתונים מקומי Jet. פרק 12 מרחיב בנושא זה.

קבועים שמגדירים את המאפיין LockType של אובייקט החיבור

קבוע	ערך	תפקיד
adLockReadOnly	1	גישה לקריאה בלבד (ברירת המחדל)
adLockPessimistic	2	נועל רשומה מיד לאחר שימושו מתחילה בערכתה
adLockOptimistic	3	נועל רשומה רק לאחר שימושו מוחיל את השינויים על מסד הנתונים
adLockBatchOptimistic	4	אפשר פעולות עריכה באזות רשותות טרם הניסיון לעדכן מסד נתונים מרוחק מתוך אזות רשותות מקומית



הערה:

תוכל לקבוע אם מערכת הרשותות בה אתה מטפל תוכל וርיבים תפקודיים מסווג מסוים, על ידי שימוש בשיטה Supports. כל שעריך לעשות הוא להציב בסוגרים את הקבוע שמייצג את הרכיב התפקודי הרצוי, בעת הפעלת השיטה. ערך החזרה True מציין כי מערכת הרשותות כוללת את הרכיב התפקודי הנדרש. התיעוד המופיע של Supports מתייחס שמות הקבועים. חפש את CursorOptionEnum בעזרת סורק האובייקטים (Object Browser), כדי להציג רשימת קבועים עבורם Supports מחזירה True או False.

ניווט בערכת רשותות. ארבע שיטות אפשרות לנאות בערכת רשותות על ידי שינוי מיקום הרשותה הנוכחית :

◀ **MoveFirst**. שיטה זו הופכת את הרשותה הראשונה בערכת הרשותות לרשותה הנוכחית. סדר הרשותות תלוי באינדקס הנוכחי, ואם לא קיים אינדקס, הסדר תלוי בסדר הרכנשה. השיטה פועלת בכל סוגים הסטנסים. השימוש בשיטה עם סמן forward-only עשוי לפחות ביצוע חוזר של הפקודה שיצרה את ערכת הרשותות.

◀ **MoveLast**. שיטה זו קובעת את הרשומה האخונה בערכת רשומות כרשומה הקיימת. היא דורשת סוג סמן התומך בתנועה לאחר, או לפחות צו שmbosst על סימניות. השימוש בשיטה עם סמן forward-only יוצר שגיאת זמן ריצה.

◀ **MoveNext**. שיטה זו ממקמת מחדש את הרשומה הקיימת מקום אחד קדימה ביחס למיקומה הנוכחי (בכוון הרשומה האخונה שבערכת הרשומות). אם הרשומה הקיימת היא גם הרשומה האخונה, המאפיין EOF של ערכת הרשומות יקבל את הערך True. אם השיטה נקראת בזמן שערך המאפיין EOF הוא True, תתקבל שגיאת זמן ריצה.

◀ **MovePrevious**. שיטה זו מזיהה את מקום הרשומה הקיימת רשומה אחת לאחר. אם הרשומה הקיימת היא הראשונה, המאפיין BOF של ערכת הרשומות יקבל את הערך True. אם השיטה נקראת בזמן שערך המאפיין BOF הוא True, יתקבל שגיאת זמן ריצה. השיטה יוצרת שגיאת זמן ריצה גם במקרה שפעילים אותה עם סמן forward-only.

השיטה Move (מעבר אל) פועלת בצורה שונה מארבע שיטות הניות בערכת הרשומות, מכיוון שהיא יכולה להעביר את מקום הרשומה הקיימת מספר רשומות משתנה בשני הכוונים. לցוון תנועה בכיוון הרשומה האخונה משתמשים בארגומנט חובי, ואילו לציוון תנועה בכיוון הרשומה הראשונה משתמשים בארגומנט שלילי. אם התנועה תעבור את הרשומה הראשונה או האخונה, Move תיתן למאפיין EOF או EOF את הערך True. אם ערך המאפיין הוא False, השיטה תיצור שגיאת זמן ריצה. התנועה היא יחסית לרשותה הקיימת, אלא אם מצינים פרמטר Start המאפשר לתנועה להתחילה ברשותה הראשונה או האخונה.

ניתן לשפר את ביצועי השיטה Move בכמה דרכים, על ידי הגדרת המאפיין CacheSize של ערכת הרשומות לערך גדול מ-1 שהוא ערך ברירת המחדל. שינוי הערך של CacheSize גורם ל-ADO לאחסן מספר רשומות קבוע בזיכרון המקומי של תחנת העבודה. אוחזור רשומות מהזיכרון מהיר בהרבה מאשר מאכזעי אחסון של הספק. וכן ניתן להציג את הניות בערכות Move על ידי הגדלת ערך CacheSize. CacheSize forward-only מוגדל, מאפשרים גלילה קדימה ואיפלו אחורה. סמן מסוג סמן מוגדל, מאפשרים גלילה קדימה ואיפלו אחורה. אם הגדרת המטען זהה למספר הרשומות בערכת הרשומות, תוכל לכלול את ערכת הרשומות למלא גודלה בשני הכוונים. המאפיין CacheSize אינו מאפשר גלילה לאחר שימושות השיטה MovePrevious (לשם כך משתמשים בשיטה Move עם ארגומנט שלילי).

◀ **Find**. שיטה זו מփשת אחר הרשומה הראשונה העונה על קריטריון בחירה מוגדר. השיטה דומה להפליא לאוסף השיטות Find של גרסאות מוקדמות של Access, אך הגירסה החדשה מאופיינית בתחביר ובתקוד שוניים. במקרים לנסות לפרט את קווי הדמיון והשוני בין הגרסאות, מוטב ללמידה את התחביר ואופן הפעולה של הגירסה החדשה.

הגירסה החדשה של השיטה Find כוללת עד ארבעה ארגומנטים. הארגומנט הראשון, WHERE, קרייטריון החיפוש, הוא ארגומנט נדרש. התהביר דומה לזה של פסוקיות WHERE במשפט SQL. אם אין מציינים ארגומנט נוסף כלשהו, השיטה עורכת חיפוש אחר רשומה העונה על הקרייטריון החל ברשומה הנוכחית ועד הרשומה האחרונה. לאחר שמנצאה רשומה תואמת, עלייך לעבור על פניה, כדי למצוא רשומה תואמת נוספת. אם לא נמצאה רשומה כזו, השיטה תונתן למאפיין EOF את הערך True. עיין בעזרה המוקוונת לקבלת תיאור של שלושת הארגומנטים האופציונליים הנדרשים.

המאפיין Sort. מאפיין זה יכול להשפיע על תוצאות פעולהן של השיטות Find ו-Move. מאפיין זה מייעד שדה אחד או יותר לקביעת סדר הצגת השירות. ערך המאפיין Sort מאפשר לקבוע סדר מיון עולה או יורדת בהתאם לשדה כלשהו. בירית המוחלט היא סדר מיון עולה. ערך המאפיין אינו משפיע בפועל על סדר השירותים, אלא על סדר הזמיןויות שלהם בערכת הרשומות.

המאפיין Filtered. מאפיין זה של ערכת הרשומות מגדר ערכת רשומות חדשה שהיא גירסה מסוננת של ערכת הרשומות המקורי. למאפיין זה יש יישומים מיוחדים עבור סינכרון מסד נתונים וערכו באצווה של מקור נתונים מרוחק, אך הוא יכול לשמש חלופה פשוטה להגדרת ערכת רשומות חדשה מבוססת על משפט SQL. אם אתה זוקק רק לקובץ-משנה למטרה כלשהי, מאפיין זה יענה היבט על צריך.

השיטה AddNew. שיטה זו מוסיפה רשומות חדשות לערכת רשומות. לאחר הפעלת השיטה, מגדרים את ערכי השדות בשורה החדשה. אחר כך עוזבים את הרשימה באמצעות השיטה Move, או קוראים לשיטה Update מבלי לצאת מהשורה החדשה (תוכל לשנות את הערכים בשדה באמצעות שתי טכניקות דומות: ערכו שדות על ידי מתן ערכים חדשים ואחר כך עזיבת הרשימה. לחילופין, קרייה לשיטה Delete. אפשר למחוק רשומה על ידי ניווט אליה וקרייה לשיטה Delete. הרשימה שנמחקה נשארת הרשימה הנוכחית עד המעבר לאחרת).

הדף ערכי שדה. השירה הפשוטה שלפניך פותחת מקור נתונים ולאחר מכן מדפיסה את שורות מסד הנתונים בצורה סדרתית. השירה כוללת לולאה שעוברת על כל הרשומות ומדפיסה את שני השדות הראשונים של כל רשומה.

```
Sub EasyLoop()

Dim rsCustomers As Recordset
Dim cnn1 As ADODB.Connection

Set rsCustomers = New ADODB.Recordset
rsCustomers.ActiveConnection = CurrentProject.Connection
rsCustomers.CursorType = adOpenKeyset
rsCustomers.LockType = adLockOptimistic
rsCustomers.Open "MyTable", , , adCmdTable
```

```

'rsCustomers.Open "customers", "Provider=Microsoft.Jet.OLEDB.4.0;" &_
"Data Source=C:\Program Files\Microsoft Office\Office\" & _
"Samples\Northwind.mdb;"

'rsCustomers.Sort = "customerid"

'Loop through recordset.
Do Until rsCustomers.EOF
    Debug.Print rsCustomers.Fields(0), rsCustomers.Fields(1)
    rsCustomers.MoveNext
Loop

rsCustomers.Close
End Sub

```

אחד החסרונות של השיגרה EasyLoop הוא שהיא מדפסה רק ערכי שדות מבוקשים. השיגרה EasyLoop2 שלහלן, מתגברת על הקושי. אין זה משנה מה מספר השדות במקור הנתונים של מערכת הרשומות, השיגרה מדפסת אותם במלואם ללא התurbות.

```

Sub EasyLoop2()

Dim rsCustomers As Recordset
Dim fldMyField As Field
Dim strForRow As String

Set rsCustomers = New ADODB.Recordset
rsCustomers.Open "customers", & _
"Provider=Microsoft.Jet.OLEDB.4.0;" & _
"Data Source=C:\Program Files\Microsoft Office\Office\" & _
"Samples\Northwind.mdb;"

' Loop through recordset and fields with rows.
Do Until rsCustomers.EOF
    strForRow = ""
    For Each fldMyField In rsCustomers.Fields
        strForRow = strForRow & fldMyField & "; "
    Next fldMyField
    Debug.Print strForRow
    rsCustomers.MoveNext
Loop

rsCustomers.Close
End Sub

```

קובצת השורות הראשונה והקובצת האחורונה בשתי השורות, פותחות ערכת רשומות וסגורות אותה. השגירה EasyLoop2 מknnt לולאה מסוג For בתוך לולאה מסוג So. הלולאה הפנימית For ממספרת את שדות השורה ויוצרת בכל שורה מהירות אחת שמורכבת מכל ערכי השדות (בתחילת הלולאה מנקים את המחרוזות מכל ערכיה כדי לבצע את התהליך פעם נוספת עבור השורה הבאה).

לולאה היא דרך קלה לביצוע פעולה על שורות ועמודות של ערכת רשומות. לעומת זאת, אין זו הדרך היעילה לאחזר ערכי שדה של ערכת רשומות. השגירה EasyLoop שלפניך מנצלת את השיטה GetString כדי לאחזר ולהדפיס את כל השדות בכל שורות ערכת הרשומות, וזאת בשלב אחד בלבד. השיטה GetString מחזירה ערכת רשומות בתווך מהירות. היא יכולה לקבל עד חמישה ארגומנטים; הקוד שלפניך מנצל שלושה מהם. מציבים את הקבוע adClipString בתווך ארגומנט ראשון – זהה לבחירה היחידה. הארגומנט השני מצין את תבנית הייצוג של ערכת הרשומות בתווך מהירות. הארגומנט השלישי מצין את מספר שורות ערכת הרשומות שיש להחזיר. הקוד מחזיר חמיש שורות. אם אין מגדירים ערך כלשהו לארגומנט, השיטה תחזיר את כל השורות של ערכת הרשומות.

הารוגומנט השלישי מציב נקודת-ופסיק (;) בתווךתו מפריד בין עמודות בשורה. מפריד בירית המחדל הוא טאב. הארגומנטים הרביעי והחמישי, שאינם מופיעים בדוגמה, מצינים מפריד עמודות וב吐וי לייצוג ערכי null. ערכי בירית המחדל של ארגומנטים אלה הם תו החזרת גורה (CR) ומחרוזת באורך אפס.

```
Sub NoEasyLoop()
Dim rsCustomers As Recordset
    Set rsCustomers = New ADODB.Recordset
    rsCustomers.Open "customers", _
        "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Program Files\Microsoft Office\Office\" & _
        "Samples\Northwind.mdb;"

'Print records without a loop.
    Debug.Print rsCustomers.GetString(adClipString, 5, ";")
    rsCustomers.Close
End Sub
```

השיטה GetString מחליפה שתי לולאות מקוננוות. אם ביריות המחדל מתאימות לצריכך, תוכל להשתמש בשיטה ללא ארגומנטים כלשהם. הדבר מאפשר לשלוּף ערכים מתוך ערכת רשומות. למרות שלולאות מקוננוות הוא דרך אינטואיטיבית לשלוּף ערכים מתוך ערכת רשומות, השיטה GetString יכולה להביא לתוצאה דומה בשורת קוד אחת בלבד.

הוספת רשומה. הקוד הבא מבצע משימה חדשה: הוספת רשומה חדשה למקור נתונים.

```

Sub AddARecord()

Dim rsMyTable As Recordset

' Set your cursor so that it is not read-only to delete.
    Set rsMyTable = New ADODB.Recordset
    rsMyTable.ActiveConnection = CurrentProject.Connection
    rsMyTable.Open "MyTable", , adOpenKeyset, adLockOptimistic, adCmdTable

' Invoke the AddNew method.
    rsMyTable.AddNew
        rsMyTable.Fields("Column1").Value = 16
        rsMyTable.Fields("Column2").Value = 17
        rsMyTable.Fields("Column3").Value = 18
    rsMyTable.Update

End Sub

```

השגרות EasyLoop1,EasyLoop2 ו-NoEasyLoop מקבלות כולן את הגדרות בירית המחדל של סוג הסמן וסוג הנעילה, אך השגירה AddARecord אינה נוהגת כך. כזכור, ביריות המחדל הם מסוג forward-only ונעילה מסוג read-only. הגדרות אלו מתאימות רק עבור הדפסת תוכן ערכות רשומות. אך יש צורך בסוג סמן ובסוג נעילה המאפשרים עדכון ערכות רשומות, כשהמשמעותה הנדרשת כוללת הוספה, עリכה או מחיקה של רשומות. הארגומנטים adLockOptimistic ו-adOpenKeyset של השיטה Open מאפשרים להוסיף שורות חדשות לערכות רשומות. שים לב שההגדרה ActiveConnection בקוד לעיל אינה מפנה לפROYיקט הדוגמה Northwind, אלא מפנה לחיבור הפROYיקט הנוכחי. שימושו להתייחס למקור נתונים בפROYיקט הנוכחי, פועלUPIי כליל התchapיר. משפט החיבור מציב מפורשות טבלה בפROYיקט הנוכחי כמקור הנתונים של ערכות הרשומות. קיימים מקורות חלופיים אחדים, כולל טקסט של משפט SQL, שיגרה מאוחסנת, קובץ CHIZONI שנשמר בתבנית מיוחדת ועוד.

כדי להוסיף רשומה באמצעות השיטה AddNew, קוראים לשיטה, מפעילים משפט Update. אין חובה מוחלטת לקרוא ל-Update ; ניתן פשוט לעזוב את הרשומה החדש. במקרה, תוכל להפעיל את MoveFirst או שיטה אחרת, כדי לנוט לרשומה חדשה.

עריכה או מחיקה של רשומה. הקוד הבא עורך או מוחק רשומה. הקוד אינו משתמש בשיטות Edit או Update כדי לשמור את הרשומות לאחר עריכתן, אלא עזוב את הרשומה. אם אין זה מעשי לעזוב את הרשומה, או אם היישום צריך לאמת את השינויים טרם עזיבת הרשומה, השתמש בשיטה Update .

```

Sub DeleteOrUpdateARecord()
Dim rsMyTable As Recordset

'Set your cursor so that it is not read-only to delete.
    Set rsMyTable = New ADODB.Recordset
        rsMyTable.ActiveConnection = CurrentProject.Connection
        rsMyTable.Open "MyTable", , adOpenKeyset, adLockOptimistic, adCmdTable
' Loop through recordset.
    Do Until rsMyTable.EOF
        If rsMyTable.Fields("Column1") = 16 Then
            rsMyTable.Fields("Column1") = 88
            rsMyTable.Delete
        End If
        rsMyTable.MoveNext
    Loop

    rsMyTable.Close
End Sub

```

lolalaah cdogmat zo shbshigraha DeleteOrUpdateARecord ykola la'siyyu l'k libchor rishomot lmachika avo le'revicha. shigraha boudkaat kl urk shl hashda Column1 b'revicha rishomot b'chafsha achor shda shurku 16. cshnatzia shda h'mboksh, h'siyta mohakka at shwora. shim lab shhalalah kollett shwora ha'ura. cdi le'ubor m'sherat machika le'sherat udcon, ha'ver at simon ha'ura m'sherat ha'ura le'sherat h'siyta .Delete.

chifush rishomot. shimosh npoz nosf b'revicha rishomot, hoa chifush rishoma achat avo yoter lanunot lkriytorionim mogderim. Access 2000 ma'apshar lb'atz m'siyma zo b'cmeha drachim. b'grasot motkdamot shel Access, rabbo le'hastmesh bo'oriyachia achat avo yoter shel h'siyta Find. cpi shebber hozerno, Access 2000 kolle girsa y'hida shel h'siyta Find ha'shona b'midat-mah m'grasototia h'kodmotot. g'm bgirsa zo tovel le'hastmesh b'revibim h'tefukodimim h'domim shel girsa ha'chadsha shel Find.

h'kod shel p'nei mazig yiyyos peshot shel shigraha Find, asher m'hafsh rishoma sh'mzahha h'lkot h'shla (CustomerID) mat'hil ba'ot D. casheh rishoma ma'otra, h'siyta kovut otot b'rishoma ha'nuchit. h'kod mdafis at h'shdot CustomerID - ContactName. cdi sha'afsh irih la'sher b'madivik at rishoma ha'una ul kriytorion chifush.

```

Sub FindAMatch()
Dim rsCustomers As Recordset
    Set rsCustomers = New ADODB.Recordset
        rsCustomers.ActiveConnection = _
            "Provider=Microsoft.Jet.OLEDB.4.0;" & _
            "Data Source=C:\Program Files\Microsoft Office\Office\" & _
            "Samples\Northwind.mdb;"

```

```

rsCustomers.Open "Customers", , adOpenKeyset, adLockPessimistic, _
adCmdTable
rsCustomers.Find ("CustomerID Like 'D*'")
Debug.Print rsCustomers.Fields("CustomerID"), _
rsCustomers.Fields("ContactName")
End Sub

```

אחד החסרונות של הגישה שתוארה הוא בכך שהיא מחייבת מופע תאימות בודד ותחליך החיפוש נוצר מיד לאחר שنمצא התאמה כזו. הקוד ש לפניך מגלה את כל הרשומות התואמות למשפט הקритריון. יישום פשוט זה מציג היבטים נוספים של גמישות השיטה `.Find`.

```

Sub FindAMatch2()
Dim rsCustomers As Recordset
Set rsCustomers = New ADODB.Recordset
rsCustomers.ActiveConnection = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office\Office\" & _
    "Samples\Northwind.mdb;"
rsCustomers.Open "Customers", , adOpenKeyset, adLockPessimistic,
adCmdTable
Do
    rsCustomers.Find ("CustomerID Like 'D*'")
    If rsCustomers.EOF Then
        Exit Sub
    End If
    Debug.Print rsCustomers.Fields("CustomerID")
    rsCustomers.MoveNext
Loop
End Sub

```

הפתרון לאייתור כל הרשומות העוננות על קритריון החיפוש הוא לשבץ את השיטה `Find` בולאה מסווג `Do`. כשהשיטה `Find` מגדרה את המאפיין `EOF` של ערכת הרשומות בתורו `True`, לא קיימות רשותות תואמות נוספות. במקרה זה, הקוד מבצע את המשפט `Exit Sub` ועל ידי כך מסיים את השיגורה. כל עוד `Find` תמשיך לגלוות מופע תאימות נוספים, השיגורה תדפיס את מזהה הל��ון בחלון `Immediate` (מיידי). לאחר הדפסת רשומה תואמת, השיגורה מתקדמת מהרשומה הנוכחית אל הבאה אחרת. ללא עשתה כן, השיטה `Find` הייתה חוזרת שוב ושוב לאוותה רשומה.

השיטה `Find` עוברת על ערכת רשותות בצורה סדרתית ובכל פעם מגלה התאמה אחת. השיטה אינה יוצרת גירסה נוספת של ערכת רשותות شاملת את כל הרשומות העוננות על הקритריון. אם יש צורך בערכת רשותות חדשה או חלופית המכילה את הרשומות התואמות בלבד, عليك לנוקוט גישה שונה מזו. המאפיין `Filter` (מסנן) של ערכת הרשומות עשוי להיות התשובה. מאפיין זה מאפשר להגדיר קритריון פשוט לשדה,

ומחזיר גירסה מסוננת של ערכת הרשומות המקורייה הכוללת רק את הרשומות העוננות על הקרייטריון שצוין. על ידי הגדרת המאפיין Filter קבוע כלשהו מתחו רישימת קבועים, תוכל להשיג תוכאות מיוחדות בתהיליכי שכפול מסד נתונים, או עדכון מקור נתונים מרוחק. קבוע המסנן adFilterNone מסיר את הגדרת המסנן מתוך ערכת הרשומות ומשחרר את ערכיה המקוריים.

סינון ורשומות. שתי השגורות שלפניך מסננות ערכת רשומות המבוססת על הטבלה Customers במסד הנתונים Northwind. השיגרה FilterRecordset מנהלת את השימוש הכלול במאפיין Filter, מדפסה את קבוצת התוצאות, מנקה את המסנן ולאחר מכן מדפסה את קבוצת התוצאות פעם נוספת. השיגרה FilterRecordset מtabסת על הפונקציה FilterLikeField המנהלת את הגדרת המאפיין Filter, בהתאם לפרמטרים שהעבירה אליה השיגרה .FilterRecordset.

```
Sub FilterRecordset()
Dim rsCustomers As Recordset
' Create recordset variable.
    Set rsCustomers = New ADODB.Recordset
    rsCustomers.ActiveConnection = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Program Files\Microsoft Office\Office\" & _
        "Samples\Northwind.mdb;"
' Open recordset.
    rsCustomers.Open "Customers", , , adCmdTable
'Filter recordset.
    Set rsCustomers = FilterLikeField(rsCustomers, "CustomerID", "D*")
    Debug.Print rsCustomers.GetString
' Restore recordset.
    rsCustomers.Filter = adFilterNone
    Debug.Print rsCustomers.GetString
    rsCustomers.Close
End Sub

Function FilterLikeField(rstTemp As ADODB.Recordset, _
    strField As String, strFilter As String) As ADODB.Recordset
'Set a filter on the specified Recordset object and then
'open a new Recordset object.
    rstTemp.Filter = strField & " LIKE '" & strFilter & "'"
    Set FilterLikeField = rstTemp
End Function
```

השיגרה FilterRecordset פותחת ביצירה ובפתחה של ערכת רשומות rsCustomer. לאחר מכן היא מיישמת מסנן על ידי קרייה לפונקציה FilterLikeField, אשר קולעת שלושה ארגומנטים ומחזירה ערכת רשומות המבוססת עליהם. FilterRecordset מציבה את ערך ההחזרה המסנון ב- rsCustomer ומדפסה אותה כדי לאמת את התוצאה.

הארוגומנטים שנשלחו אל FilterLikeField כוללים את rsCustomer (ערכות הרשומות), שם שדה שלפיו יש לערוך את הסינון (CustomerID), וערך קרייטריון סינון (*D). קרייטריון הסינון עשוי לכלול ביטוי חוקי עבור האופרטור Like של הפונקציה FilterRecordset.FilterLikeField מעבירה * D כדי לצמצם את החיפוש לרשותות שערך השדה CustomerID שלhn מתחילה באות D. המאפיין Filter אינו מב Gill את הסינון לאופרטור Like בלבד. ניתן גם לבצע סינון באמצעות האופרטורים <, >, <=, >= ו- =. אפשר לכלול את האופרטורים הלוגיים And ו-Or בביטויי הקרייטריון, כדי לחבר שני ביטויים או יותר בהתבסס על אופרטורים חוקיים נוספים.

מאפיין Filter מב Gill את הקרייטריון לביטויים בתצורת FieldName-Operator-Value קיימים קבועים של Filter המשמשים למיניהם. FilterRecordset משתמש במאפיין adFilterNone כדי לשחרר ערכת רשותות על ידי הסרת המנסנים.

שימוש ב-SQL ליצירת רשותות. עובדה אחרת שעילץ לדעת על ערכות רשותות: כיצד ליצור ערכת רשותות בהתבסס על משפט SQL. לעיתים קרובות משפט SQL אינם מורכבים יותר מהמשפט "SELECT * FROM TABLENAME", אך באפשרות נצל את מלאה הרכיבים התפקודים של SQL כדי ליצור ערכות רשותות. תוכל גם להשתמש במשפט SELECT מורכבים ורובי טבלאות עם שדות מחושבים המשמשים בצירופים (joins) פנימיים וחיצוניים אשר מב Gillים או מארגנים קבועות החזורה בעזרת הפסוקיות WHERE, GROUP BY ו- ORDER BY. דרך קלה ליצירת ערכת רשותות מותאמת אישית שmbוססת על משפט SQL, היא השימוש בפסוקיות WHERE. אפשרות נוספת לשלוף רשותות בקרה סלקטיבית מתוך מקור קיים באמצעות ביטויים מורכבים יותר מאשר ניתן להפעיל בעת שימוש במאפיין Filter.

הקוד שלפניך משתמש בשיטה Open ובמשפט SQL. כשבבסיסים ערכות רשותות על משפט SQL במקומות על טבלה קיימת, מעבירים את משפט SQL ומשתמשים בארוגומנט האופציונלי adCmdTable במקומות בארוגומנט adCmdText. זה כל מה שיש לעשות. לאחר מכן ניתן לנצל את ערכת הרשותות כדי לבנות ערכת רשותות פשוטה כלשהי שmbוססת על טבלה מסוימת. משפט SQL מורכבים יותר אינם משנה את ההצעה או את אופן הטיפול בערכת רשותות באמצעות ADO.

```
Sub SQLRecordset()
Dim rsCustomers As Recordset
' Create recordset variable.
    Set rsCustomers = New ADODB.Recordset
    rsCustomers.ActiveConnection = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Program Files\Microsoft Office\Office\" & _
        "Samples\Northwind.mdb;"

' Open the recordset.
    rsCustomers.Open "SELECT * FROM Customers", , adOpenForwardOnly, _
        adLockReadOnly, adCmdText
    Debug.Print rsCustomers.GetString

    rsCustomers.Close
End Sub
```

האובייקט Field

שדה הוא עמודת נתונים שמכילה ערכים מסווג נתונים זהה. בספריה ADODB, האוסף Fields שיביך בלהדיות לערוכות רשומות וחבריו הם האובייקטים Field. אובייקטים אלה כוללים מאפיינים ושיטות המשמשים לאחסון ושליפה של נתונים.

ערוכות רשומות מנצלות את המאפיין Value של האובייקט Field כדי להציג את תוכן העמודה של הרשימה הנווכחית. כמשמעותו, ערך זה יכול להשתנות כדי לייצג את תוכן הרשימה החדשה. מאפיינים רבים נוספים של Field מכילים metadata – נתונים אודות נתונים הרשימה. המאפיין Name הוא נקודת אחיזה (handle) שמאפשרת לישום להתייחס לשדה מסוים. המאפיין DefinedSize מציין את הגודל המרבי של שדה (בתווים, בשרות). המאפיין ActualSize הוא הגודל בפועל, בתווים, של ערך באובייקט Field. המאפיין Attributes מכיל מידע רכיבי מודיע על שדה. באפשרותו לציין אם ערך שדה הוא בר-עדכו או אם השדה יכול להכיל ערכי Null.

הערה:



המאפיינים DefinedSize ו-ActualSize משתמשים ביחידות מידת שונות עבור שדות DefinedSize .Text האו מספר התווים המרבי בשדה, ו-ActualSize הוא מספר הבטים שהשדה מכיל בפועל. שדות Text של Jet 4 מייצגים תווים באמצעות שני בתים לכל TWO, וכן הערך ActualSize שלהם עשוי להיות אפילו כפול מהערך הנוכחי. DefinedSize בשדות נומריים ובשדה Text במסדי נתונים המייצגיםתו באמצעות בית יחיד (כגון מסד נתונים Jet 3.51), הבדל זה אינו קיים.

השימוש של shiitot של AppendChunk ו-Field – אפשרות לעבד שדות גדולים של טקסט או של נתונים ביןaries בגושים קטנים שקל יותר לטפל בהם בזכרון המחשב. השיטה GetChunk משמשת להעיף לזכרון חלק מהשדה. הארגומנט Size מציין את מספר הבטים שיש לאחזר בהפעלה אחת של השיטה GetChunk. בכל הפעלה רציפה ללא הפרעות של השיטה, מתבצעת קריאת נתונים חדשים מהנקודה בה הסתיימה קריאת הנתונים הקודמים. השיטה AppendChunk מאפשרת לבנות מותוך הזיכרון שדה גדול מאוד המכיל טקסט או נתונים ביןaries, בורות גושים. בדומה לשיטה AppendChunk, גם שיטה זו כותבת בשדה נתונים חדשים החל מהמקום בו הסתיימה הכתיבה הקודמת של AppendChunk. כדי להפעיל את השיטות בצהורה נכונה, יש לתת סיבית adFieldLong של המאפיין Attribute באובייקט Field את הערך True.

המאפיינים Name ו-Value. השירה שלפניך מציגה את השימוש הנפוץ של מאפיינים אלה. השירה מציינה את כל שמות השדות לצד ערכיהם. היא יוצרת ערכת רשומות בת רשותה אחת בהתבסס על משפט SQL.

```

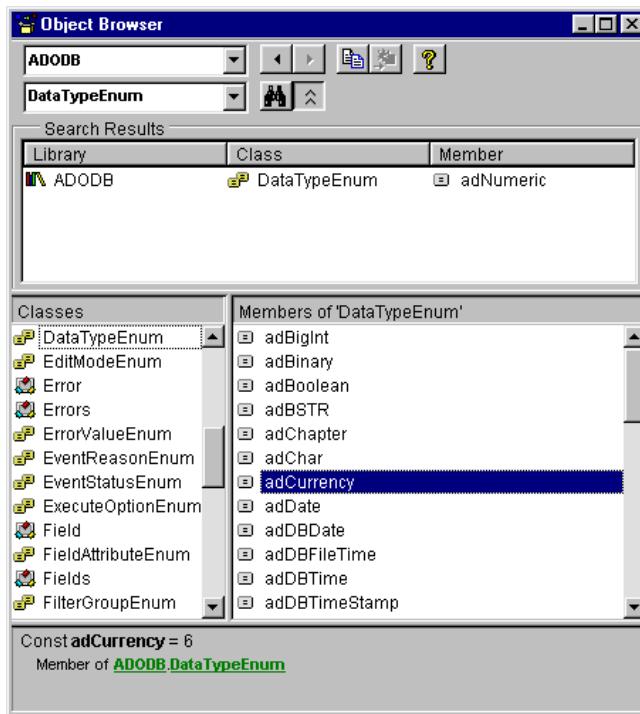
Sub FieldNameValue()
Dim cnn1 As ADODB.Connection
Dim rsCustomers As ADODB.Recordset
Dim fldLoop As ADODB.Field
' Open connection and recordset.
strCnn = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office" & _
    "\Office\Samples\Northwind.mdb;"
Set cnn1 = New ADODB.Connection
cnn1.Open strCnn
Set rsCustomers = New ADODB.Recordset
rsCustomers.ActiveConnection = cnn1
rsCustomers.Open "SELECT * FROM Customers" & _
    "WHERE CustomerID='BONAP'", , , adCmdText
' Report field names and values for record.
For Each fldLoop In rsCustomers.Fields
    Debug.Print fldLoop.Name, fldLoop.Value
Next fldLoop
End Sub

```

השיגרה מתחילה בפתיחת חיבור ולאחר מכן יוצרת ערכת רשומות בחיבור. משפט SQL שולף את רשומות הלקוח שערך השדה CustomerID שלה שווה ל-BONAP. לולאת So, הבאה לאחר יצירת ערכת הרשות, עוברת על שדות ערכת הרשות. הדפסת המאפיין Name יחד עם המאפיין Value מסיימת להבנת פועלות השיגרה.

המאפיין Type. המאפיין Type של האובייקט Field מצינית את סוג הנתונים שהשדה יכול להכיל. המאפיין מחזיר את אחד מקבוצי סוגי הנתונים שבתווח הערכים DataTypeEnum. תוכל להציג את הקבועים האפשריים שבמספריה ADODB באמצעות סורק האובייקטים. תרשימים 2.5 מציג את הקבועים הללו בחלון **Object Browser** בחרית סוג שדה מאפשרת לקבוע ערכים חוקיים עבור המאפיין Value של סוג השדה.

הdfsat סוגים נתונים של שדות. שתי השגרות הבאות פועלות יחד כדי לעבד קבועים של סוגים נתונים באמצעות ADO. השיגרה `FieldNameType` פותחת ערכת רשומות המבוססת על הטבלה Orders במסד הנתונים Northwind. טבלה זו מכילה מילון עשר למדים של סוגים נתונים, ולכן ניתן לנצל אותה כדוגמה לבחינת הנושא של סוגים נתונים. לאחר פתיחת ערכת רשומות, השיגרה עוברת בלולאה על כל השדות שבערכת הרשות ומדפיסה שם וסוג של כל אובייקט `Field`. הפקנץיה `FieldType` מתרגם את הערך המספרי של הקבוע למחרוזת המייצגת את הקבוע. הערך המספרי של הקבוע `adCurrency` הוא 6. הפקנץיה `FieldType` ממירה את הערך 6 למחרוזת `"adCurrency"`. בעת מדפסה השיגרה `FieldNameType` את שמות כל השדות ואת שמות כל הקבועים של סוגים נתונים.



תרשים 2.5: סורק האובייקטים מציג מבחן קבועים של סוגים נתונים

```

Sub FieldNameType()
Dim cnn1 As ADODB.Connection
Dim rsOrders As ADODB.Recordset
Dim fldLoop As ADODB.Field
' Open connection and recordset.
strCnn = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office" & _
    "\Office\Samples\Northwind.mdb;"
Set cnn1 = New ADODB.Connection
cnn1.Open strCnn
Set rsOrders = New ADODB.Recordset
rsOrders.ActiveConnection = cnn1
rsOrders.Open "orders", , , , adCmdTable

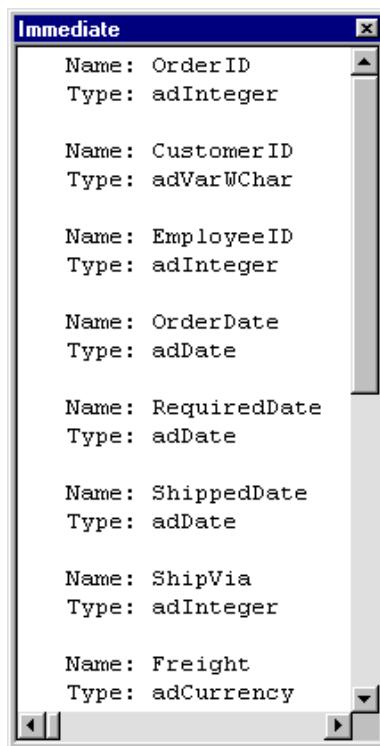
' Report field names and types for record.
For Each fldLoop In rsOrders.Fields
    Debug.Print " Name: " & fldLoop.Name & vbCrLf & _
        " Type: " & FieldType(fldLoop.Type) & vbCrLf
Next fldLoop
End Sub

```

```

Public Function FieldType(intType As Integer) As String
    Select Case intType
        Case adVarWChar
            FieldType = "adVarWChar"
        Case adCurrency
            FieldType = "adCurrency"
        Case adInteger
            FieldType = "adInteger"
        Case adDate
            FieldType = "adDate"
    End Select
End Function

```



תרשים 2.6: קטע מתוך הפלט של FieldNameType

תרשים 2.6 מציג קטע מתוך הפלט של FieldNameType. הקטע כולל לפחות שדה אחד מכל סוג נתוני שמשמעותו הפונקציה FieldType. תוכל להפעיל את FieldNameType ואת FieldType כנגד ערכות רשותות בהתבסס על מקורות נתונים שונים. ייתכן שתפגוש סוג נתונים נוספים לאربעה שברשימה. במקרה זה, שדה Type שבדויה יהיה ריק. תקן את הבעה על ידי קביעת ערך השדה. הכנס נקודת עצירה במשפט Debug.Print בלולאת Do

של השגירה FieldType. בדוק את ערך fldloop.Type של שדה שסוגו אינו מוצג, ולאחר מכן התאם את ערך הקבוע נגד הערךים בטוויח הערכים DataTypeEnum שמציג טורק האובייקטים (תרשים 2.5). לבסוף, הוסף את המשפט Select Case לשגירה כדי להמיר את הקבוע החדש.

חיפוש ערך השדה הגדול ביותר. השגירה FieldSizes שלפניך מיישמת את המאפיין ActualSize כדי למצוא את הערך הגדול ביותר בשדה CompanyName של הטבלה Shippers במאז הנתונים Northwind. השגירה מתחילה ביצירת חיבור אל מסד הנתונים ולאחר מכן פותחת ערכת רשותות המבוססת על הטבלה Shippers. חלקה השני של השגירה מאתר את שם המוביל הארוך ביותר ומציג תיבת הودעה ובה השם ומספר התווים שהוא מכיל.

```

Sub FieldSizes()
Dim cnn1 As ADODB.Connection
Dim rsShippers As ADODB.Recordset
Dim fldLoop As ADODB.Field
Dim intMaxChars As Integer, strMsg As String
Dim strName As String
' Open connection and recordset.
strCnn = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office" & _
    "\Office\Samples\Northwind.mdb;"
Set cnn1 = New ADODB.Connection
cnn1.Open strCnn
Set rsShippers = New ADODB.Recordset
rsShippers.ActiveConnection = cnn1
rsShippers.Open "SELECT * FROM Shippers", , , , adCmdText
' Find longest shipper's name.
intMaxChars = 0
Do Until rsShippers.EOF
    If rsShippers!CompanyName.ActualSize / 2 > intMaxChars Then
        intMaxChars = rsShippers!CompanyName.ActualSize / 2
        strName = rsShippers.Fields("CompanyName")
    End If
    rsShippers.MoveNext
Loop
strMsg = "The longest shipper's name is " & _
    strName & " (" & intMaxChars & " characters)."
MsgBox strMsg, vbInformation, "Programming Microsoft Access 2000"
rsShippers.Close
End Sub

```

המשתנה intMaxChars עוקב אחר אורך השדה הגדול ביותר. חלקה שני של השירה FieldSizes מאתחל את המשתנה ל-0 בטרם יחול ביצוע הלולה שעוברת על כל הרשומות שבערכת הרשומות. אין הכרח לאפס את intMaxChars, מכיוון ש-VBA ו-Access עושים זאת בעצמו. אך האתחול תורם למעין תיעוד עצמי של השירה. כל שם מוביל במספר התווים שהוא מכיל גודל יותר מהערך הנוכחי של intMaxChars הוא השם הארוך ביותר עד אותו שלב של הבדיקה. כשהשירה מוצאת שם העונה על קרייטריוון זה, היא מעדכנת את המשתנה intMaxChars ושומרת את השם. שים לב שהשירה משתמשת בשתי מערכות של כללים תחביריים כדי להפנות לשדה. משפטים אחדים משתמשים בתו bang (!) שהוא סימן שהוא מקובל בגרסאות ישנות של Access. בסימן הנוכחי, התו bang מפריד בין שם ערכת הרשומות לשם השדה. הדריך החדש והמקובל יותר היא ליצור הפניה אל השדה באמצעות האוסף Field. אפשרות להשתמש באינדקס המספרי, אם הוא ידוע לך, או לתחום את שם השדה בין גרשימים.

האובייקטים Parameter ו-Command

לאובייקטים Command של הספרייה ADODB שלושה יתרונות:

- ◀ הם מסוגלים לבצע שאילתת בחירה כדי להחזיר קבוצת שורות מקור נתונים.
- ◀ הם מבצעים שאילתת פרמטר, דבר שמאפשר לספק קרייטריוון חיפוש בזמן ריצה.
- ◀ הם תומכים בשאלות פעולה נגד מקור נתונים, כדי לבצע פעולות כגון עדכון, מחיקה והוספה רשותות.

כפי שנלמד בסעיפים הבאים, האובייקט **Command** יכול למלא תפקידים נוספים בנסיבות אחרות.

עליך ליעד אובייקט **Connection** שעליו תופעל פקודה. תוכל ליצור אובייקט Connection בצורה משתמעת בעת ציון פקודה, או לשיקך מפורשות אובייקט Connection קיים לפקודה. אפשרות אלו זהות לאפשרויות ערכות רשותות.

המאפיין CommandTimeout קובע את משך הזמן ש-ADO תמתין לסיום ביצוע פקודה. מאפיין זה קולט ערך מסוג Long שמציע את זמן המתנה המרבי בשניות. ערך ברירת המחדל שלו הוא 30. אם פסק הזמן חולף בטרם סיים האובייקט Command את ביצוע הפקודה, ADO תבטל את הפקודה ותחזר שגיאה. אובייקט Connection תומך אף הוא במאפיין CommandTimeout, אך איןו תלוי במאפיין CommandTimeout של אובייקט Command. מאפיין CommandTimeout של אובייקט Connection אינו יורש את הגדרת המאפיין CommandTimeout של אובייקט Connection.

המאפיין CommandType. קיימים למשהו סוגים אחדים ושוניים של האובייקט Command. המאפיין CommandType מגדיר את סוג אובייקט Command. תוכל לבסס את הפקודה על משפט SQL, על טבלה או על שירה מאוחסנת, כפי שמציגה הטבלה הבאה. הסיבה העיקרית לשימוש בהגדרת המאפיין CommandType היא לאפשר יצירת

אובייקט Command שבסיס על מشفט SQL. החלפת ערך בריית המחדל של הקבוע CommandType בערך אחר, עשויה להאיץ את ביצוע הפקודה. לכן, אם המקור ידוע לך, מומלך שתגדיר את הקבוע הזה.

המאפיין CommandText. כדי לכתוב מشفט SQL שייבצע את הפקודה, משתמשים בהגדרת CommandText של האובייקט Command. באפשרות גם להגדיר את המאפיין בתווך שם של שירה מאוחסנת. בעת הפעלת מشفט SQL, ניתן להשתמש במאפיין Prepared כדי לציין שהמשפט מיועד להיזור (קומפונלציה) ולאחר מכן בשרת מסד הנתונים. הדבר מאט את הביצוע הראשון של הפקודה, אך במקביל מאייז את ביצועיה בהמשך. תן למאפיין Prepared את הערך True כדי להדר מشفט SQL.

קבוע CommandType

קבוע	ערך	תפקיד
adCmdText	1	אפשר להפעיל פקודה המבוססת על מشفט SQL, שיגרה מאוחסנת או אפייל טבלה. בדרך כלל תשמור הגדרה זו עבור מشفט SQL
adCmdTable	2	מבוסס את קבוצת ערכי ההצהרה על טבלה שעוצבה קודם לכן. מחייב את כל עמודות הטבלה בהתאם למشفט SQL שנוצר באופן פנימי
adCmdStoredProc	4	מפעיל פקודה המבוססת על טקסט עבור שירה מאוחסנת
adCmdUnknown	8	לא קיים מפרט כלשהו לגבי סוג טקסט הפקודה. זו בריית המחדל
adCmdFile	256	מעיריך פקודה בהתאם לשם הקובץ עבור ערכת רשומות תמידית
adCmdTableDirect	512	מעיריך פקודה בהתאם לשם הטבלה. מחייב את כל העמודות בטבלה ללא SQL מתווך כלשהו

השיטה Execute. השיטה Execute של אובייקט Command מפעילה את קוד הרקע של האובייקט Command (שאילתה, מشفט SQL או שירה מאוחסנת). תוכל לציין עד שלושה ארגומנטים. הארגומנט הראשון מאפשר לאובייקט Command למסור לשירה המפעילה אותו את מספר הרשומות שעלייהן פעול. הארגומנט השני יכול להיות מערך Variant שמכיל פרמטרים להפעלת הפקודה. הארגומנט השלישי מודיע ל-ADO כיצד להעריך את המקור. ארגומנט זה יכול להיות אחד מתוך רשימת הקבועים המוגנים בטבלה "קבוע CommandType", לעיל.

השיטה CreateParameter. השיטה CreateParameter של אובייקט Command יוצרת פרמטר חדש של פקודה. לאחר יצירת הפרמטר, תוכל לנצל את השיטה Append כדי להוסיף את הפרמטר לאוסף Parameters של פקודה. בטרם תפעיל שאלחת פרמטר, عليك לספק לפרמטר ערך.

יצירת ערכת רשומות שאילתת בחירה. אחות המטלות המיידיות שניתן לבצע בעזרת אובייקט Command היא ליצור ערכת רשומות שבסיסת על שאילתת בחירה (select query). האובייקט Command מפעיל את שאילתת הבחירה ומיצג את הערךם המוחזרים שלו. בשלב זה יכול הקוד שכתבת לפתח אובייקט Recordset על הקבוצה המוחזרת מהאובייקט Command. זה בדוק מה שמבצעת השירה SelectCommand שלפניך. לשירה שני חלקים: האחד יוצר את האובייקט Command וחיבור שմחבר אותו למסד נתונים, והآخر מעבד ערכת רשומות הבסיסת על הקבוצה המוחזרת מהאובייקט.

```

Sub SelectCommand()
Dim cmd1 As Command
Dim rs1 As Recordset, str1 As String
Dim fldLoop As ADODB.Field
' Define and execute command.
    Set cmd1 = New ADODB.Command

    With cmd1
        .ActiveConnection = CurrentProject.Connection
        .CommandText = "SELECT MyTable.* FROM MyTable"
        . CommandType = adCmdText
        .Execute
    End With

' Open and print recordset.
    Set rs1 = New ADODB.Recordset
    rs1.Open cmd1

    Do Until rs1.EOF
        str1 = ""
        For Each fldLoop In rs1.Fields
            str1 = str1 & fldLoop.Value & Chr(9)
        Next fldLoop
        Debug.Print str1
        rs1.MoveNext
    Loop
End Sub

```

חלקה הראשון של השירה מצהיר על cmd1 בתור אובייקט Command ולאחר מכן מגדר שולחה מאפיינים קריטיים של האובייקט. כל פקודה חייבת לכלול מאפיין ActiveConnection כדי לפעול נגד מסד נתונים. האובייקט Command מסתמך על משפט SQL שייצג את שאילתת הבחירה שלו. תוכל להחליף את השאילתת SQL בשאליה, Execute משפט מפעיל את שאילתת הבחירה. בתום פעולה השיטה cmd1 מכיל הפניה אל ערכת רשומות.

חלוקת השני של השירה פותח אובייקט Recordset על cmd1 ומדפיס את קבוצת החזרה שחבריה מופרדים בתווי טאב ((chr(9)) בחלון Immediate (מיידי). יכולת הטיפול של השירה אינה מוגבלת למספר עמודות ושורות כלשהו.

יצירת ערכת רשותות באמצעות שאלת פרמטר. הקוד שפלני הוא דוגמה לשאלת פרמטר (Parameter Query). גם קוד זה מורכב משני חלקים. שאלת הפרמטר שבחלק הראשון כוללת קוד ADO אחדות ומשפט SQL בתחריר שונה משל שאלת הבחירה הקודמת. החלק השני שמדפיס את קבוצת החזרה זהה לזה של שאלת הבחירה הקודמת.

```
Sub ParameterQCommand()
Dim cmd1 As Command
Dim rs1 As Recordset, str1 As String
Dim fldLoop As ADODB.Field
Dim prm1 As ADODB.Parameter, int1 As Integer
' Create and define command.
    Set cmd1 = New ADODB.Command

    With cmd1
        .ActiveConnection = CurrentProject.Connection
        .CommandText = "Parameters [Lowest] Long;" & _
            "SELECT Column1, Column2, Column3 " & _
            "FROM MyTable " & _
            "WHERE Column1>=[Lowest]"
        . CommandType = adCmdText
    End With
' Create and define parameter.
    Set prm1 = cmd1.CreateParameter("[Lowest]", adInteger, adParamInput)
    cmd1.Parameters.Append prm1
    int1 = Trim(InputBox("Lowest value?", "Programming Microsoft Access 2000"))
    prm1.Value = int1

' Run parameter query.
    cmd1.Execute

' Open recordset on cmd1 and print it out.
    Set rs1 = New ADODB.Recordset
    rs1.Open cmd1

    Do Until rs1.EOF
        str1 = ""
        For Each fldLoop In rs1.Fields
            str1 = str1 & fldLoop.Value & Chr(9)
        Next fldLoop
        Debug.Print str1
        rs1.MoveNext
    Loop
End Sub
```

תחבר מشفט SQL כולל שורת הצהרה חדשה על Parameters, אשר מצינית את שם הפרמטר וסוג הנתונים שלו. הפסוקית WHERE אף היא צריכה להפנות אל פרמטר אחד או יותר, כדי שהפרמטרים יוכלו לפעול על קבוצת החזורה. לא די בתיקוני תחבר אלה במשפט SQL כדי לגרום לשאיילתת הפרמטר לפעול – עלייך להוסיף את הפרמטר ולצרכו לפקודה באמצעות קוד O.ADO.

מבצעים את השיטה CreateParameter כדי להוסיף את הפרמטר. הקוד שלעיל מבצע את השיטה באמצעות שלושה ארגומנטים. הארגומנט הראשון מזכה לפרמטר שם, השני מזכה לו סוג נתונים והארוגומנט השלישי מצהיר על כיוונו. הקבוע Input adParamInput הוא למעשה בריית המחדל שמצויר על פרמטר קלט לשאיילתת. קבועים נוספים מאפשרים להגדיר פרמטרים של פלט, קלט/פלט וערך החזרה. לאחר יצירת פרמטר, עלייך לצרף אותו לאוסף Parameters של הפקודה.

לאחר כתיבת הקוד שמוסיף את הפרמטר, עלייך להקצותו לפרטר ערך כדי ששאיילתת הפרמטר תפעל בצורה תקינה: הקוד שלעיל משתמש בפונקציה InputBox כדי לקבל קלט מהמשתמש. לאחר קבלת הקלט, השיגרה מפעילה את השיטה Execute של האובייקט Command כדי ליצור קבוצת החזרה.

מחיקת רשומות. באפשרות נצל את האובייקט Command כדי למחוק, לעדכן ולהוסיף רשומות למקור נתונים. האובייקטים Command מספקים אמצעי תכונתי לתחזוקת מקור נתונים. השגורות DeleteAllRecords ו-DeleteARecord שלහן אוספות רשומות מתוך מקור נתונים. מייעדים את מקור הנתונים ואת קרייטוריון בחירת הרשומות באמצעות המשבט DELETE של SQL. תוצאות SQL בחילון השאיילתת של Access אפשר לעצב שאילתה בצורה גרפית ולאחר מכן להעתיק את הקוד למאפיין CommandText של הפקודה. בוודאי תרצה לעורך את קוד SQL מתוך מעצב השאילותות של Access, כדי להסיר סוגרים מיותרים. אם השאילתת יצירת מתבצעת על טבלה בודדת, תוכל להסיר את תחילית הטבלה שלפני שמות השדות. כפי שניתן לראות, ההבדל בין שתי שאילותות המחייבת מתמקד בת לחבר משבט SQL.

```
Sub DeleteARecord()
Dim cmd1 As ADODB.Command

    Set cmd1 = New ADODB.Command

    With cmd1
        .ActiveConnection = CurrentProject.Connection
        .CommandText = "DELETE MyTable.Column1 FROM " & _
                      "MyTable WHERE (((MyTable.Column1)=13));"
        . CommandType = adCmdText
        .Execute
    End With

End Sub
```

```

Sub DeleteAllRecords()
Dim cmd1 As ADODB.Command

    Set cmd1 = New ADODB.Command

    With cmd1
        .ActiveConnection = CurrentProject.Connection
        .CommandText = "DELETE MyTable.* FROM MyTable"
        . CommandType = adCmdText
        .Execute
    End With

End Sub

```

הוספת רשומות. בעת פיתוח יישום, ניתן שתהייה מעוניין ביכולת למחוק את כל הרשומות מתוך טבלה ואחר כך לאפס את תוכנה. השירה InsertRecords משתמשת באובייקט Command כדי למלא טבלה בערכיהם. תוכל להפעיל את השירה יחד עם השירה DeleteAllRecords כדי לרענן את הטבלה בקבוצת רשומות בסיסית קטנה.

```

Sub InsertRecords()
Dim cmd1 As ADODB.Command

    Set cmd1 = New Command

    With cmd1
        .ActiveConnection = CurrentProject.Connection
        .CommandText = "INSERT INTO MyTable(Column1, " & _
                      "Column2, Column3) VALUES (1,2,'3')"
        . CommandType = adCmdText
        .Execute
        .CommandText = "INSERT INTO MyTable(Column1, " & _
                      "Column2, Column3) VALUES (4,5,'6')"
        . CommandType = adCmdText
        .Execute
        .CommandText = "INSERT INTO MyTable(Column1, " & _
                      "Column2, Column3) VALUES (7,8,'9')"
        . CommandType = adCmdText
        .Execute
        .CommandText = "INSERT INTO MyTable(Column1, " & _
                      "Column2, Column3) VALUES (10,11,'12')"
        . CommandType = adCmdText
        .Execute
    End With
End Sub

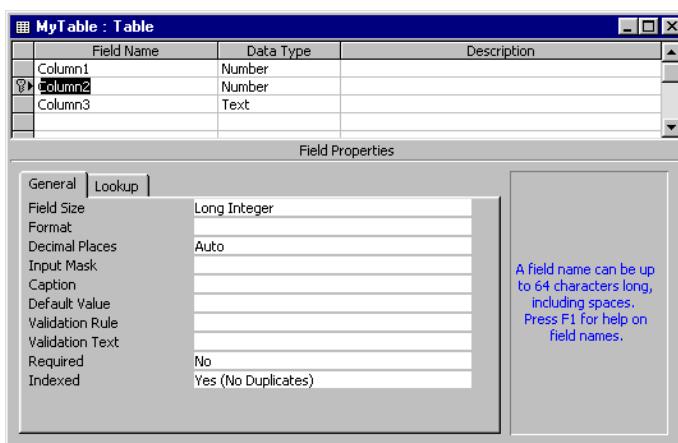
```

```

.CommandText = "INSERT INTO MyTable(Column1, " & _
    "Column2, Column3) VALUES (13,14,'15')"
.CommandType = adCmdText
.Execute
.CommandText = "INSERT INTO MyTable(Column1, " & _
    "Column2, Column3) VALUES (16,17,'18')"
.CommandType = adCmdText
.Execute
End With
End Sub

```

השירה InsertRecords כוללת אלמנטים כלליים וספציפיים אחד. האלמנטים הכלליים אינם תלויים בעיצוב טבלה מסוימת. בקוד ש לעיל, האלמנטים הספציפיים תואימים אלה הכלליים של הטבלה MyTable. תרשימים 2.7 מציג את MyTable בתצוגת **עיצוב** (Design). הטבלה כוללת שלוש עמודות: Column1, Column2 ו Column3. שתי העמודות הראשונות מכילות נתונים מסוג Long Integer, והעמודה השלישייה מכילה נתונים מסוג Text (בעת הוספת רשומות, יש להתחשב בסוג הנתונים של השדה).



תרשים 2.7: תצוגת **עיצוב** של הטבלה אליה מוסיפים רשומות באמצעות השירה InsertRecords

האלמנטים הכלליים של השירה InsertRecords משוטפים ליישומים נוספים של האובייקט Command. יוצרים הפניה אל אובייקט CommandType ומגדירים את מאפיין החיבור שלו. לכל שורה ברשומה שברצונך להוסיף, נחוצות שלוש שורות קוד: המאפיין CommandType, שמצוין את פעלות הפקודה; המאפיין CommandText, שמצוין את תבנית ההוראה; והשיטה Execute, שמבצעת את תהליכי הוספה של הרשומה החדשה. תוכל לחזור ולבצע את שלוש שורות הקוד לכל שורה שברצונך להוסיף למקור הנתונים. אם תציגו בתור מטרה קבוצת רשומות דינמית שניתן לעדכן אותה, שלבים אלה יכולים להוסיף רשומות לשתי טבלאות או יותר, בו-זמנית.

תחבר מפתח CommandText של SQL כולל שלושה מרכיבים (תחביר זה אינו זמין מתוך **צוגת SQL** של מעצב השאלות של Access). ראשית הוא משתמש במילת המפתח INSERT INTO. אחרי מילה זו רשום שם מקור הנתונים שאלוי ברכונך להוסיפה רשומות. שנית, המשפט מנצל את שלב האופציוני שמציג את שמות השדות שגולטים את הערכים החדשניים. אם לא תבצע שלב זה, הערכים שתוסיף בשלב השלישי יתווסף בצורה סדרתית. הדבר עלול להוות בעיה, אםutzt מקור הנתונים משתנה במשך הזמן. שלישיית, מילת המפתח VALUES מופיעה לפני ערכי השדות של הרשימה החדשה.

עדכון ערכי רשומה. השגרות OddToEven ו-EvenToOdd מעדכנות את ערכי מקור הנתונים של Column1 באמצעות האובייקט Command. תרשימים 2.8 מציג את הטללה מייד לאחר הפעלת השגרות DeleteAllRecords ו-InsertRecords.שים לב כי בתרשימים 2.8 הערכים בעמודה Column1 הם זוגיים ואי-זוגיים לחילופין: אם ערך כלשהו בעמודה Column1 הוא אי-זוגי, הערך המקביל בעמודה Column2 הוא זוגי. השגרות מנצלות מידע זה כדי לנהל את תוכן הטללה.

תרשימים 2.8: **צוגת גליאן נתונים** (Datasheet) של הטללה אותה מעדכנות השגרות EvenToOdd ו- OddToEven

```
Sub OddToEven()
Dim cmdO2E As ADODB.Command
Dim intRowsChanged As Integer
Set cmdO2E = New ADODB.Command
With cmdO2E
    .ActiveConnection = CurrentProject.Connection
    .CommandText = "UPDATE MyTable SET Column1 = " & _
        "Column1+1 WHERE ((-1*(Column1 Mod 2))=True)"
    . CommandType = adCmdText
    .Execute intRowsChanged
    Debug.Print intRowsChanged & " rows were affected."
End With
End Sub
```

```

Sub EvenToOdd()
Dim cmdE2O As ADODB.Command
Set cmdE2O = New ADODB.Command
With cmdE2O
    .ActiveConnection = CurrentProject.Connection
    .CommandText = "UPDATE MyTable SET Column1 = " & _
        "Column1-1 WHERE ((-1*(Column2 Mod 2))=False)"
    . CommandType = adCmdText
    .Execute
End With
End Sub

```

העיצוב הכלול של שגרות אלו בודאי מוכר לך. ההבדל המשמעותי ביותר בין הדוגמאות הנוחיות הללו שוחצנו קודם לכן טמון בתחביר משפט SQL של המאפיין CommandText. במקרה זה, אפשרותך לחלץ את תבנית התחביר הכללית מותך מעצב השאלות של Access. הפסוקית WHERE של השירה OddToEven בוחרת רשומות UPDATE בעמודה Column1 שהן היא-זוגי. חלק התחביר הכלול את המילה UPDATE מופיע 1 לערך כדי להפכו למספר זוגי. השיטה Execute מנצלת אחד מהארגומנטים המוכלים שלה כדי להחזיר את מספר השורות שהשתנו על ידי פקודה. השיטה הפושאה Print שולחת ערך זה לחילון Immediate (מיידי) לצורך הצגתנו במאז.

השירה EvenToOdd בוחנת את הערך שבעמודה2, Column2, כדי לקבוע אם יש לחסר 1 מהערך שבעמודה Column1. אם הערך שבעמודה Column2 אינו אי-זוגי, משפט SQL פועל על הערך בעמודה Column1. פעולה זו מושחרת את הערכים שבעמודה Column1 לעריכיהם ההתחלתיים, אם OddToEven פועלת מיד לאחר השירה EvenToOdd.

האוסף Errors

אוסף זה מאפשר לך מהשניות המופיעות בישום ADO. האוסף גם מחזיר שגיאות מותך ספק OLE DB. מצב שגיאה יחיד יכול להחזיר שגיאות רבות, ככל אחת מהן גורמת להצבת אובייקט Error חדש באוסף Errors. יש שגיאות הגורמות לסייע התוכנית, אך לא כוון. כל תקלת חדשה מנקה אוטומטית את האוסף Errors כדי שזה יוכל לקלוט את השגיאות הקשורות אליה. שגיאות ADO מסוימות נקלוטות באובייקט Err במקומות באוסף Errors, אך ניתן להשתמש גם באוסף למטרה זו. האוסף Errors מתאים ביותר לטיפול בשגיאות מבוססות-חיבור שהוחזרו מסדר נתונים מרוחק באמצאות ספק OLE DB שלו.

האובייקט Error של האוסף Errors כולל חמישה מאפיינים שימושיים לאוסף מידע נוספים המאפשר להגיב עליהם באמצעות לוגיקת התוכנית. המאפיינים Number (מספר) ו-Description (תיאור) מקבילים למאפיינים של האובייקט Err. מאפיינים אלה משלימים זה את זה. המאפיין Number מחזיר מספר ייחודי שמצויה שגיאה, והמאפיין Description מחזיר מחרוזת תיאור קצרה של השגיאה. המאפיין NativeError מספק קוד שגיאה מוגדר של הספק. אם אתה עובד לעתים קרובות עם ספק מסוים, מאפיין

זה עשוי לספק לך מידע שימושי לפתרון השגיאה. המאפיין Source (מקור) נוקב בשם האובייקט או היישום שחולל את השגיאה. המאפיין SQLState יכול להכיל הודעות שגיאות תחביר של משפט SQL, שמקורו בשרת מסד נתונים שאליו מפנים את הבקשה.

השיגרה OpenLookOnlyErrors שלහלן היא התאמת של שיגרה שהוצאה קודם לכך, אשר מציגה את השפעת המאפיין Mode של האובייקט Connection. הגדרת מאפיין זה כקריאה-בלבד גורמת לייצרת שגיאה בעת שימושים לעדכן מסד נתונים. מעניין לציין כי השגיאה אינה הופכת לחברון לאחר מכן Errors. Err מופיע גם הוא באובייקט Err. לוגיקת לכידת השגיאות שבסוף השיגרה מנעת מהדפסת שתי שורות בעלות מספר ותיאור זהים.

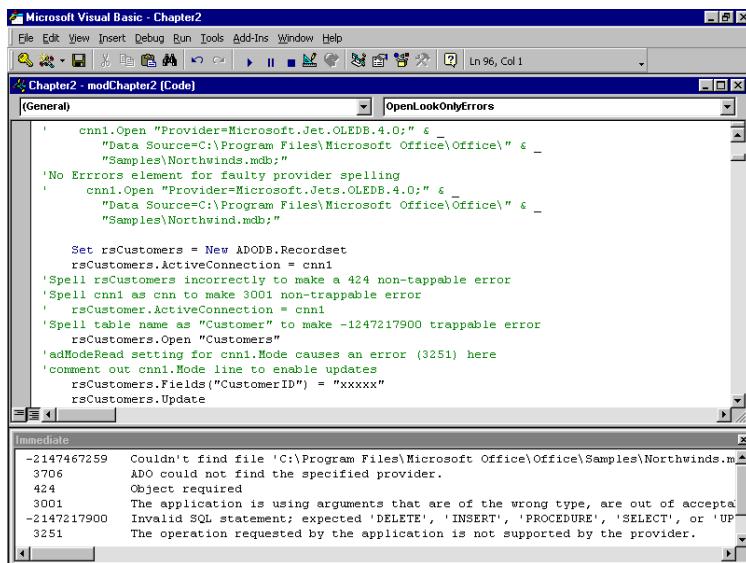
```
Sub OpenLookOnlyErrors()
Dim cnn1 As New Connection
Dim rsCustomers As Recordset
Dim errLoop As Error, intInErrors As Integer
On Error GoTo LookOnlyTrap
    cnn1.Mode = adModeRead
    cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Program Files\Microsoft Office\Office\" & _
        "Samples\Northwind.mdb;"
' Spell Northwind incorrectly to generate trappable error.
'   cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Program Files\Microsoft Office\Office\" & _
        "Samples\Northwinds.mdb;"
' No Errors element for faulty provider spelling
'   cnn1.Open "Provider=Microsoft.Jets.OLEDB.4.0;" & _
        "Data Source=C:\Program Files\Microsoft Office\Office\" & _
        "Samples\Northwind.mdb;"
    Set rsCustomers = New ADODB.Recordset
    rsCustomers.ActiveConnection = cnn1
' Spell rsCustomers incorrectly to make a 424 non-trappable error.
' Spell cnn1 as cnn to make 3001 non-trappable error.
'   rsCustomer.ActiveConnection = cnn1
' Spell table name as "Customer" to make -2147217900 trappable error.
'   rsCustomers.Open "Customers"
' adModeRead setting for cnn1.Mode causes an error (3251) here.
'Comment out cnn1.Mode line to enable updates.
    rsCustomers.Fields("CustomerID") = "xxxxx"
    rsCustomers.Update
    Debug.Print rsCustomers.Fields("CustomerID")
    rsCustomers.Close
```

```

LookOnlyTrap:
    intInErrors = 0
    For Each errLoop In cnn1.Errors
        Debug.Print errLoop.Number, errLoop.Description
        intInErrors = intInErrors + 1
    Next errLoop
    If intInErrors = 0 Then
        Debug.Print Err.Number, Err.Description
    End If
End Sub

```

השיגרה OpenLookOnlyErrors יוצרת מספר סוגי שגיאות שונים ומנסה לכתוב אל חיבור מסוג קרייה-בלבד. תרשימים 2.9 מציג את ההוראות ו-Code Immediate של VBE. מקור ההודעות בעלות קודו השגיאה הארכיים והשליליים הוא באוסף Errors. שאר השגיאות הן שגיאות ADO שמדווחות באמצעות האובייקט Err. רק לשתי שגיאות יש מספרים ארכיים ושליליים מהסוג המופיעין את השגיאות שבאוסף Errors. שאר השגיאות הן שגיאות ADO שניתן להגיע אליהן באמצעות האובייקט Err. שגיאה אחת מ奏וק מחרוזות החיבור (שגיאה מס' 3706) לא הוצאה עדין באמצעות האוסף Errors. זאת ביצירוף העובדה שהחבר האחורי באוסף Errors מופיע באובייקט Err, מהווים יתרון בכל הקשור לכידת שגיאות באמצעות האובייקט Err. מנגנון זה פועל גם בשגיאות VBA.



תרשים 2.9: ההוראות ו-Code Immediate של VBE מציגים קודו שגיאה של שגיאות טיפוסיות

**קויים.**

באפשרות להוסיף משפט Option Explicit באזור ההוראות הכלליות של מודול, כדי לבטל את האפשרות של שגיאות מסוימות, כגון הפניות לאובייקטים שאין

השירה LoopToUsingErrors שלפניך מספקת מספר גישות חלופיות לטיפול בשגיאות באמצעות האובייקט Err. השירה יוצרת שגיאה אחת וכוללת הערה המפרטת את שינויי הקוד שיש לבצע כדי ליצור שגיאה נוספת. השירה אף מגיבה באופן ייחודי לשתי שגיאות, וכוללת הליך כללי לטיפול בכל יתר השגיאות:

↳ במקרה של שגיאה 3251, השירה משנה את סוג הנעילה, כך שנitinן לעדכן את ערכת הרשומות. שגיאה זו מתרחשת מכיוון ששוגר הנעילה שגוי. לתקן הבעיה, קוד הטיפול בשגיאה סוגר את ערכת הרשומות הישנה, מאפס את המאפיין LockType ופותח מחדש את אובייקט ערכת הרשומות.

↳ במקרה של שגיאה 424, השירה אינה מנסה לתקן את השגיאה, אלא מתריעה בכך המשמש על סיבת הבעיה האפשרית. שגיאה זו מתרחשת כאשר מפעילים שיטה או מדיררים מאפיין נגדי משתנה שלא הוצהר כאובייקט. לדוגמה, שגיאת הקלדה עשויה לגרום לבעה כזו.

↳ אם מספר השגיאה שונה מ-3251 או 424, השירה מדפיסה את מאפייני המספר והຕיאור של האובייקט Err.

```
Sub LoopToUsingErrors()

On Error GoTo DErrorsTrap

Dim cnn1 As Connection
Dim rsMyTable As Recordset

Set cnn1 = New ADODB.Connection
cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office\Office\" & _
    "Samples\Northwind.mdb;"

Set rsMyTable = New ADODB.Recordset

' Open recordset with defaults.
OpenRSMyTable:
    rsMyTable.Open "MyTable", cnn1
```

```

' Loop through recordset.
Do Until rsMyTable.EOF

' Make 424 error by using next instead of preceding line.
' Do Until rsMyTables.EOF
If rsMyTable.Fields(0) = 4 Then

' This line makes 3251 error because recordset is read-only.
    rsMyTable.Fields(0) = 88
Else
    Debug.Print rsMyTable.Fields(0), rsMyTable.Fields(1)
End If

rsMyTable.MoveNext
Loop

rsMyTable.Close

ErrorsExit:
Exit Sub

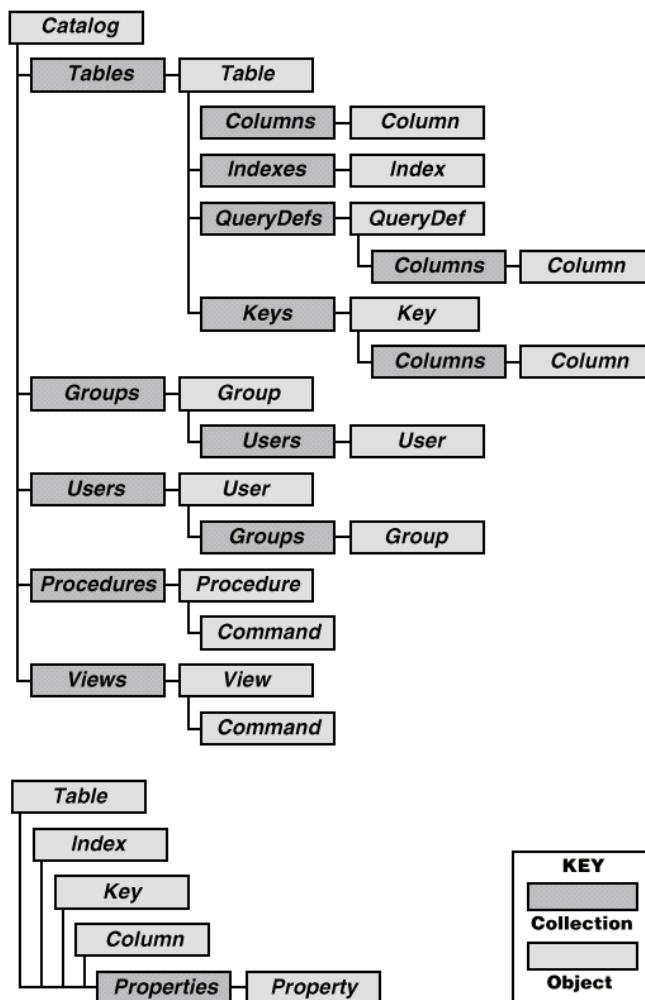
DErrorsTrap:
If Err.Number = 3251 Then
    MsgBox "OLEDB Provider does not support operation. " & _
        "Find another way to get the job done or get a new " & _
        "OLEDB Provider. Error happened in LoopToDeleteErrors."
    Debug.Print rsMyTable.LockType
    rsMyTable.Close
    rsMyTable.LockType = adLockOptimistic
    Resume OpenRSMyTable
ElseIf Err.Number = 424 Then
    MsgBox "The code tried to do something requiring an " & _
        "object, such as set a property or invoke a method, " & _
        "but the code did not have an object. Check spelling."
Else
    MsgBox "Check Immediate window for error # and desc."
    Debug.Print Err.Number, Err.Description
End If
Resume ErrorsExit

End Sub

```

הספריה ADOX

הספריה ADOX תומכת במסימות סכימה וابتחה. באפשרות לנצל ספריה זו כדי לשנות את הארכיטקטורה של עיצוב מסד הנתונים של היישום שלך. כל האובייקטים, למעט האובייקט Catalog, כוללים אוסףים תואמים. אוסףים אלה משמשים להוספה וארגון אובייקטיבים חדשים בקטלוג. אובייקטיבים נבחרים, כגון טבלאות, מפותחות אינדקס ועמודות, כוללים אוסףים מסווג (מאפיינים). אוסףים אלה משמשים לניהול פעולות האובייקטיבים ביישום. מנהלים את האוסףים (משתמשים) ו-groups (קבוצות) כדי לשנות בהרשאות של אובייקטיבים נוספים של ADOX, כגון טבלאות, תצוגות ושרורות. תרשים 2.10 מציג מבט כללי על הספריה ADOX.



תרשים 2.10: סדריית האובייקטיבים ADOX

הספריה ADOX היא שלוחה של הספריה ADODB. הספק Jet ADO供 ADO供 ב-AOX. הספק ADODB מספקים נוספים יקרים לישם את רכיביו בצורה סלקטיבית, ונitin גם להשתמש בשתי הספריות יחדיו כדי לבנות יישומים. לדוגמה, לאחר מכן, אפשרות להרכיב אובייקטים של פקודה בעזרת הספריה ADODB ולאחר לשמור אותם כשרונות בספריה ADOX. לחילופין, תוכל לבדוק זמינות טבלה באמצעות T-SQL עליה ערכות רשומות. אם הספריה אינה קיימת, תוכל להוסיפה ולאכלס אותה בערכיהם. יכולת הספריה ADOX להגדיר מבני נתונים חדשים הופכת אותה לחלופה של SQL DDL.

האובייקט Catalog

אובייקט זה הוא המכולה ברמה הגבוהה ביותר של הספריה ADO. חבירו מגדירים את הסכימה ואת מודל האבטחה של ממד נתונים. המאפיין ActiveConnection של האובייקט מגדיר את החיבור אליו שיקטולוג. האובייקט Connection הוא ממד הנתונים. האובייקט Catalog הוא מכלול הטבלאות, התצוגות, השגרות, המשתמשים והקבוצות בתוך חיבור או ממד נתונים. המאפיין Name (שם) של קטלוג הוא מסווג קרייה- בלבד. מגדירים אותו בעת ההצהרה על הקטלוג. משתמשים בשיטה Create של האובייקט Catalog כדי להקנות חיבור ומקור נתונים לקטלוג, כך שניתן יהיה לפתוח בו-זמן ממד נתונים חדש ולקבל גישה לקטלוג שלו.

דוגמאות הקוד שפנוי מציגות את האובייקט Catalog בשלושה הקשרים טיפוסיים. השירה CatCon פותחת את ממד הנתונים Northwind ומספקת ליישום הנוכחי גישה באמצעות קוד לבנייה מסד הנתונים. שם לב שיש צורך באובייקט Connection עבור הקטלוג, כדי ש-ADO תדע איפה קטלוג עליו להפוך לזמן. מזמנים את האובייקט Connection לקטלוג באמצעות המאפיין ActiveConnection. כ-ADO יודעת לאיזה מסד נתונים עליה להפנות בעזרת הקטלוג, יש לך גישה באמצעות קוד לתוכן הקטלוג. שולטים בגישה באמצעות טכניקות מסד נתונים ואבטחה ברמת-משתמש.

```
Sub CatCon()
Dim cnn1 As New Connection
Dim cat1 As New Catalog
Dim proc1 As Procedure

cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office\Office\" & _
    "Samples\Northwind.mdb;"
Set cat1.ActiveConnection = cnn1
Debug.Print cat1.Tables(0).Name

End Sub
```

```

Sub CatCon2()
Dim cnn1 As New Connection
Dim cat1 As New Catalog
Dim proc1 As Procedure

' Open the catalog of this database.
' Print a range of selected collection information.
    Set cnn1 = CurrentProject.Connection
    Set cat1.ActiveConnection = cnn1
    Debug.Print cat1.Tables(1).Name
    Debug.Print cat1.Views(0).Name
    Debug.Print cat1.Procedures(0).Name
    Debug.Print cat1.Users(0).Name
    Debug.Print cat1.Groups(0).Name

End Sub

Sub CatCon3()
Dim cat1 As New ADOX.Catalog

' Open the catalog to a new database.
    cat1.Create "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=c:\My Documents\NewDB.mdb"
    Debug.Print cat1.Tables(0).Name

End Sub

```

השיגרה CatCon2 מציגה חבר מכל אוסף של האובייקט Catalog ב景德 הנתונים הקיימים. תוכל להפנות לתוך טבלה באמצעות הפניה למספרה, אך עליך להשתמש בהפנייה ל-ADOX כדי לעבור בלולאה על האוסף Columns (עמודות). האוספים Tables ו-Columns (טבלאות) קיימים רק בספריה ADOX.

הערה:



עליך לכפות דרישת כניסה (logon) לפני שניתן להציג את חברי האוספים Users ו-Groups. כל ניסיון להציג אותם ללא כניסה עלול ליצור שגיאת זמן ריצה. תוכל לכפות דרישת כניסה על ידי קביעת סיסמה עבור המשתמש Admin.

השיגרה CatCon3 יוצרת מסד נתונים חדש ומציג את הקטלוג שלו באותו שלב. שים לב כי אתה מפעיל את השיטה Create על האובייקט Catalog. השיטה מקבלת מחרוזת חיבור כארוגומנט. אם הקובץ המצוין במחuzeות החיבור כבר קיים, CatCon3 תיצור שגיאת זמן ריצה. כשהקובץ פועל בהצלחה, הוא מדפיס את שם אחת מטבלאות המערכת, מכיוון שלא קיימים קבצי משתמש זמינים כלשהם מייד לאחר יצירת מסד נתונים.

באפשרות נצל את האובייקט Catalog כדי למספר חברים של אוסף כלשהו מהוספים שהוא כולל. השגירה שלפניהם ממספרת את חברי האוסף Views (התצוגות), תחילה המקביל להגדרת קבוצת כל שאלות החזרת-שורות שאין מבוססת על פרמטרים בעלי מסדי נתונים Jet. בעת שימוש במסד נתונים SQL Server, התצוגות זמינים בפורש מתוך מכולת מסד הנתונים של Access 2000.

```
Sub ListViews()
Dim cnn1 As New Connection
Dim cat1 As New Catalog
Dim view1 As View
cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office\Office\" & _
    "Samples\Northwind.mdb;"
Set cat1.ActiveConnection = cnn1
For Each view1 In cat1.Views
    Debug.Print view1.Name
Next view1
End Sub
```

האוסף Views זמין מתוך הקטלוג עבור החיבור. כדי למספר את התצוגות, עליך להציג על תצוגה אחת שהקוד שלבן ינצל כדי להפנות אל כל חבר באוסף Views במהלך מעבר הלוואה על האוסף. הדפסת המאפיין Name (שם) של כל תצוגה מהווה רישום מצאי של אובייקט View שבקטלוג.

האובייקט Table

אובייקט זה הוא חבר באוסף Tables, כאשר זה הוא חבר באובייקט Catalog. כל אובייקט Table כולל מאפיין Name ומאפיין Type. אובייקט Table יכול להיות טבלה ורילה בתוך מסד הנתונים הנוכחי, או טבלה מקושרת ממוקורות נתונים ODBC וכolumbia שאים ODBC. האובייקט יכול להיות אפליו תצוגה. ערך המאפיין Type כולל גם שני סוגי של טבלאות מערכת: טבלאות מערכת Jet וטבלת מערכת Access.

תיאור	ערך מאפיין Type של אובייקט Table
טבלת מערכת של Access	ACCESS TABLE
טבלה מקושרת מתוך מקור נתונים שאין ODBC	LINK
טבלה מקושרת דרך מקור נתונים ODBC	PASS-THROUGH
טבלת מערכת Jet	SYSTEM TABLE
טבלה שפותחה באמצעות יישום או עבורו	TABLE
טבלה מתווך שאילתת החזרת שורות לא-פרמטרית	VIEW

אובייקט Table בספריה ADO יכול להכיל עד שלושה אוספים: Columns, Indexes, Keys.

האובייקט Column

האוסף Columns מיועד לטבלאות, מפתחות ואינדקסים. אובייקט מסוג Column דומה במידת-מה לאובייקט מסוג Field בספריה ADODB. עמודה בטבלה היא קבוצת כל הנתונים המתיחסים לתוכה מוגדרת של היישות שמייצגת באמצעות הטבלה.

לאובייקט Column יש כמה מאפיינים :

- ◀ **Name**. מאפיין זה הוא שם העמודה.
 - ◀ **Type**. המאפיין מצין את סוג הנתונים שבעמודה. כל הנתונים מסוג אחד.
 - ◀ **Attribute**. מאפיין זה מתאר את האפיונים של עמודה. לעמודה שני אפיונים : היכולת להכיל ערכי Null והיוותה בעלת אורך קבוע.
 - ◀ **DefineSize**. מאפיין זה מצין את האורך המירבי של ערך בעמודה.
 - ◀ **NumericScale** ו- **Precision**. מאפיינים אלה משמשים באופן בלעדי עבור שדות מספריים, כגון שדות מסוג integer, מטריך וערך נקודת צפה (floating-point). המאפיין Precision מייצג את מספר הספרות הכולל שימושים להעתרת ערך העמודה. NumericScale קובע את מספר הספרות העשויות שיבתו את הערך.
- כאשר אובייקט מסוג Column הוא אינדקס, מאפיינים אחרים כגון SortOrder ו-**RelatedColumn** יהיו זמינים אף הם.

הערה:



המאפיין NumericScale עלול לגרום לבלבול. לדוגמה, ערכים מסוג Currency מבוטאים באמצעות ארבע ספרות מימין לנקודת העשרונית, אך המאפיין scaled Currency שליהם הוא 0, כיוון ש-**Access** מתחסן נתונים בתווך Decimal integers. בעת שינוי הגדרת Scale של עמודה שמשתמשת בסוג הנתונים Decimal, המאפיין NumericScale של NumericScale של העמודה מתאים את עצמו לפי הנسبות.

האובייקט Index

אובייקט זה מגדר אינדקסים עבור טבלה. לאובייקט חמשה מאפיינים : Clustered, PrimaryKey, Name, Unique ו- IndexNulls. לאחר הוספת האינדקס, כל המאפיינים למעט Name, הם מסוג קרייה- בלבד. המאפיין Name הוא שם האינדקס. יש שלושה מאפיינים נוספים מסוג Boolean, ותפקידיהם (בהתאם) : לציין אם האינדקס **מקובץ-באשכולות** (clustered) (אינדקס נקרא מקובץ-באשכולות, כאשר הסדר הפיזי של השורות תואם לסדר ה'מאונדקס' של השורות) מפתח ראשי (PrimaryKey) או ייחודי (Unique).

המאפיין IndexNulls יכול לקבל ערך אחד מתוך שלושה ערכים. במקרה שהאינדקס מכיל ערך Null עבור عمودה, הגדרתו בתור adIndexNullsDisallow גורם לכשלון בבניה. IndexNullsIgnore את הקבוע, adIndexNullsIgnore, איפשר לבנות את האינדקס גם אם הוא מכיל ערך Null, אך המאפיין **התעלם מ-Nulls** (Ignore Nulls) שבסמך המשמש). גם הערך יקבל ערך כן (Yes) (בחולון **איןדיקסים** (Indexes) שבסמך המשמש). גם במקרה זה המאפיין **התעלם מ-Nulls** מקבל ערך לא (No).

האובייקט Key

אובייקט זה מגלה את התייחסות זרים (foreign keys) באמצעות מאפייניו. כמובן, המאפיין Name הוא שם המפתח. המאפיין McGinn את RelatedTable UpdateRule ו-DeleteRule קובעים מה הטבלה שעלייה מצביע מפתח זר. המאפיינים McGinn מסוג Type הוא סוג המפתח, והוא סוג Type McGinn. המאפיין McGinn Über מפתחות ייחודיים אפשריות: adKeyForeign עבור מפתחות זרים, adKeyUnique עבור מפתחות ייחודיים ו-adKeyPrimary עבור מפתחות הראשיים.

מספר טבלאות. אחת הדרכים הקלות ביותר להתחילה לטפל בטבלאות היא למספור אותן. השירה ListTables שלפניך מציגה כיצד משבצים את הטבלאות במסד הנתונים Northwind. ההוראות הופכות את cat1 לモפע של האובייקט Catalog, ואת tbl1 לモפע של אובייקט Table. לאחר מכן, השירה מציבה את מסד הנתונים Northwind ואת ספק Jet במאפיין ActiveConnection של הקטלוג. הלולה שלפניך מזיהה את שם הטבלה הארוך ביותר בקטלוג. הקטע האחרון מדפיס את שמות הטבלאות בחולון Immediate. נסחfat מחרוזת מסוימת תווית רוחה בסוף שמות הטבלאות כך שמספר התווים בכל השמות יהיה קבוע. הקוד מצורף את סוג הטבלה.

```
Sub ListTables()
Dim cat1 As New ADOX.Catalog
Dim tbl1 As ADOX.Table
Dim intMaxLength As Integer

' Specify active connection for the Catalog .
cat1.ActiveConnection = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office\Office\" & _
    "Samples\Northwind.mdb;"

' Find longest table name.
intMaxLength = 0
For Each tbl1 In cat1.Tables
    If Len(tbl1.Name) > intMaxLength Then intMaxLength = Len(tbl1.Name)
Next tbl1
```

```

' Print table names to Immediate window.
intMaxLength = intMaxLength + 2
For Each tbl1 In cat1.Tables
    strName = tbl1.Name
    strFiller = String(intMaxLength - Len(tbl1.Name), " ")
    Debug.Print strName & strFiller & tbl1.Type
Next tbl1
End Sub

```

Immediate	
Categories	TABLE
Category Sales for 1995	VIEW
Current Product List	VIEW
Customers	TABLE
Employees	TABLE
MSysAccessObjects	ACCESS TABLE
MSysACEs	SYSTEM TABLE
MSysObjects	SYSTEM TABLE
MSysQueries	SYSTEM TABLE
MSysRelationships	SYSTEM TABLE
MyTable	TABLE
Order Details	TABLE
Order Details Extended	VIEW
Order Subtotals	VIEW
Orders	TABLE
Product Sales for 1995	VIEW
Products	TABLE
Products Above Average Price	VIEW
qryForSource	VIEW
Quarterly Orders	VIEW
Sales by Category	VIEW
Shippers	TABLE
Suppliers	TABLE
Ten Most Expensive Products	VIEW

תרשים 2.11: פלט השירה ListTables

תרשים 2.11 מציג את פלט השירה ListTables. שים לב שהפלט מופיע בשתי עמודות, וכל אחת מציגה שם טבלה. ליד השם מופיע המאפיין Type של הטבלה. כזכור, האובייקט Table כולל שישה סוגים. ארבעה מהם מופיעים בתרשים.

שדות מספור. השגורות ListTableTypeColumns ו-ColumnType שלפניך הן דוגמאות מסוכילות יותר, שמתפלות במאפייני הטבלה ובהיררכיה שלה. כ-sh-ListTableTypeColumns מארתת אובייקט מסווג Table שהמאפיין Table שלו מוגדר בטור TABLE, היא מציגה את שמות העמודות ואת סוגי כל עמודות הטבלה. אלה מופיעים תחת שם הטבלה ומספר העמודות שלה. כל שם עמודה מופיע לצד הקבוע ColumnType. תרשים 2.12 מציג קטע מהתדפיס. זו דוגמהعشירה, כיון שהיא מטפלת בכמה סוגים אובייקטים מסווג Table ובמאפיינים רבים, ומשום שהאובייקטים והמאפיינים נלקחו מנוקודות שונות בהיררכיה מודול האובייקט של ADOX.

Immediate	
Categories	4
CategoryID	adInteger
CategoryName	adWChar
Description	adLongVarWChar
Picture	adLongVarBinary
Customers	11
Address	adWChar
City	adWChar
CompanyName	adWChar
ContactName	adWChar
ContactTitle	adWChar
Country	adWChar
CustomerID	adWChar
Fax	adWChar
Phone	adWChar
PostalCode	adWChar
Region	adWChar
Employees	17
Address	adWChar
BirthDate	adDate
City	adWChar
Country	adWChar
EmployeeID	adInteger
Extension	adWChar
FirstName	adWChar
HireDate	adDate
HomePhone	adWChar
LastName	adWChar
Notes	adLongVarWChar

תרשים 2.12: פלט השגרות ColumnType-ו ListTableTypeColumns

```

Sub ListTableTypeColumns()
Dim cat1 As New ADOX.Catalog
Dim tbl1 As ADOX.Table
Dim col1 As ADOX.Column
    cat1.ActiveConnection = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Program Files\Microsoft Office\" & _
        "Office\Samples\Northwind.mdb;"
For Each tbl1 In cat1.Tables
    If tbl1.Type = "TABLE" Then
        strName = tbl1.Name
        strFiller = String(30 - Len(tbl1.Name), " ")
        Debug.Print strName & strFiller & tbl1.Columns.Count
        For Each col1 In tbl1.Columns
            strFiller = String(20 - Len(col1.Name), " ")
            Debug.Print String(5, " ") & col1.Name & _
                strFiller & ColumnType(col1.Type)
        Next col1
    End If
    Next tbl1
End Sub

```

```
Public Function ColumnType(intType As Integer) As String
```

```
    Select Case intType
        Case adVarWChar
            ColumnType = "adVarWChar"
        Case adCurrency
            ColumnType = "adCurrency"
        Case adInteger
            ColumnType = "adInteger"
        Case adDate
            ColumnType = "adDate"
        Case adWChar
            ColumnType = "adWChar"
        Case adLongVarWChar
            ColumnType = "adLongVarWChar"
        Case adLongVarBinary
            ColumnType = "adLongVarBinary"
        Case adBoolean
            ColumnType = "adBoolean"
        Case adSmallInt
            ColumnType = "adSmallInt"
        Case Else
            ColumnType = CStr(intType)
    End Select
End Function
```

הצהרות כוללות קטלוג, טבלה ועומדה מותך הספריה ADOX. הצהרת הקטלוג יוצרת מופיע של האובייקט שלו, אך השתיים הנוטרות משמשות כהפניות לחבר בתוך אוסף. ActiveConnection מייד לאחר הצהרות, השגירה מקצת מחוזות חיבור אל המאפיין Catalog של האובייקט .

שתי לולאות For מבצעות חיפוש בטבלאות ובעמודות שלhn. הלולאה החיצונית ממחשת טבלאות שהמאפיינו Table שלהם הוגדר בתור TABLE. ברגע שנמצא כזה, היא מדפיסה את השם ומספר העמודה של אותה טבלה. לולאת For הפנימית מספרת את עמודות הטבלה ומטפלת בעיצוב הפלט. הלולאה קוראת לפונקציה ColumnType שמחזירה מחוזות תווים. המחרוזות מייצגת את הקבוע שמנדרט את המאפיין Type. הפונקציה מצטיינת בחסינותה, ולכן תוכל לאמץ אותה לביצוע מטלות מרירה. אם הפונקציה נתקלת בסוג בלתי מזוהה, היא מmirrah את ערך הסוג למחוזות. הדבר מסיע לחזור לפונקציה ולהוסify לה מפרט פענוח חדש.

יצירת טבלאות. לספריה ADOX חשיבות מיוחדת, כיוון שהיא מאפשרת ליצור טבלאות באמצעות קוד, יחד עם האינדקסים והഫתחות שלhn.

אם נוח לך להוסיף טבלאות בתצוגת **עיצוב** (Design), عليك להשיקע מעט יותר כדי ללמוד לעשות זאת באמצעות קוד. מגדירים את הטבלה ולאחר מכן מוסיפים אותה לאוסף Tables של הקטלוג. הגדרת טבלה דורשת לוגיקה מסווג דומה. מצהירים תחילת על משתנה אובייקט טבלה ולאחר מכן מצרפים עמודות למשתנה האובייקט. בעת הוספת העמודות, הקוד יכול להזכיר סוגים נתונים ומאפייני עמודה נוספים. בסיום הגדרת הטבלה, ניתן למלא אותה נתונים. באחד השימושים הקודמים בפרק, הצינו שתי גישות לטיפול בנושא (שימוש בקוד SQL ובשיטה AddNew של ערכת הרשומות).

בפרק 3 נבחנו את הנושא בהרחבה.

הוספת אינדקס. השירה AddIndex שלහן מדגימה כיצד מוסיפים אינדקס לטבלה באמצעות קוד. כמו בתחום יצירת טבלה, משתמשים בשיטה Append פעמיים. ראשית, הוסף עמודת נתונים אחת או יותר לאינדקס שיצרת. לאחר מכן הגדר את המאפיין Name של האובייקט Index. Index. Append גם להגדיר מאפיינים נוספים. שורות העשרה בשירה שלפניך מציגות הוראות להגדיר מאפיינים אופציונליים אחדים ומתראות את ההשפעה של מתן ערכים למאפיינים. לאחר שתסייעים להגדיר את מפרט האינדקס, הפעיל את השיטה Append פעם נוספת, אך הפעם השתמש בה כדי להוסיף את האינדקס החדש לאוסף Indexes של הטבלה.

```
Sub AddIndex()
Dim cat1 As New ADOX.Catalog
Dim tbl1 As New ADOX.Table
Dim idx1 As New ADOX.Index
    cat1.ActiveConnection = CurrentProject.Connection

    Set tbl1 = cat1.Tables("MyTable")
    idx1.Name = "MyFirstIndex"
    idx1.Columns.Append ("Column1")

' Rules and syntax for setting IndexNulls property
' Does not create index if field contains Nulls; yields error.
'   idx1.IndexNulls = adIndexNullsDisallow
' Sets Ignore Nulls index property to Yes -- creates index.
'   idx1.IndexNulls = adIndexNullsIgnore
' Sets Ignore Nulls index property to No -- creates index.
'   idx1.IndexNulls = adIndexNullsIgnoreAny

' If you want to set the PrimaryKey property
'   idx1.PrimaryKey = True
' you must also set the Unique property for the
' primaryKey property to take effect and avoid an error.
'   idx1.Unique = True
    tbl1.Indexes.Append idx1
End Sub
```

השיגרה PKErrorCatcher שלפניך מבצעת לכידת שגיאות בעת יצירת מפתח ראשי חדש בטבלה. הרעיון הכללי של שיטת לכידת השגיאות מיושם גם באינדקסים נוספים ובמפתחות זרים. מנגנון לכידת השגיאות מטפל/mporushot בשני סוגים של שגיאה ומסוגל לכלוד גם שגיאות מסווגים אחרים.

```

Sub PKErrorCatcher()
On Error GoTo PKErrorCatcherTrap
Dim catMyCat As New ADOX.Catalog
Dim tblMyTable As New ADOX.Table
Dim idxMyIndex As New ADOX.Index
Dim colMyCol As New ADOX.Column
Dim iNumber As Integer
catMyCat.ActiveConnection = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office\Office\" & _
    "Samples\Northwind.mdb;"
Set tblMyTable = catMyCat.Tables("MyTable")
PKErrorCatcherTry:
With idxMyIndex
    .Name = "MyPrimaryKey"
    .PrimaryKey = True
    .Unique = True
    .IndexNulls = adIndexNullsDisallow
End With
idxMyIndex.Columns.Append "Column1"
tblMyTable.Indexes.Append idxMyIndex
Set catMyCat = Nothing
PKErrorCatcherExit:
    Exit Sub
PKErrorCatcherTrap:
If Err.Number = -2147217856 Then
    MsgBox "Table currently in use."
ElseIf Err.Number = -2147467259 Then
    For Each idxMyIndex In tblMyTable.Indexes
        If idxMyIndex.PrimaryKey = True Then
            tblMyTable.Indexes.Delete (iNumber)
            Resume PKErrorCatcherTry
        End If
        iNumber = iNumber + 1
    Next idxMyIndex
Else
    MsgBox "Error #" & Err.Number & ": " & Err.Description
End If
Resume PKErrorCatcherExit
End Sub

```

השיגרה פותחת בהצחה על האובייקטים החדשים Catalog ו-Index (מפתח primaryKey הוא אינדקס שהמאפיינים Unique ו-True). חלק הארי של PKErrowerCatcher מגדיר את המאפיינים הדורשים, מצרך עמודה לאינדקס ולאחר מכן מצרך את האינדקס לטבלה. קיימים לפחות שני מצבים בהם הוראות פשוטות אלו ייכשלו בזמן ריצה. האחד, הטבלה עלולה להיות פתוחה. ADO מנסה למצב זה את מספר השגיאה **2147217856**. כשמנגנון לכידת השגיאות מזהה מספר זה, הוא מציג הודעה למשתמש, בה נאמר כי הטבלה נמצאת כרגע בשימוש. מצב נוסף, הניסיון לצרף מפתח ראשי חדש עלול להיכשל, אם מוגדר כבר אחד כזה עבור הטבלה. השיגרה מזזה את המפתח הראשי היישן ומנסה ליצור מפתח ראשי חדש. כיוון שלא קיים יותר מפתח ראשי כלשהו, השיגרה אינה יכולה להיכשל שנית מאותה סיבה.

מטען ערכים לשדה autoincrement. היחסום Access 2000 הוא הגירסה הראשונה של Access שמאפשרת למפתחים להגדיר את ערכי התחלת והפסקה של עמודות autoincrement (מספר אווטומטי בהפרשים קבועים). באפשרות להגדיר ערכים אלה באמצעות קוד, על ידי יצירת טבלה באמצעות משפטי SQL של Jet 4.0. הפקודה CREATE TABLE משמשת ליצור המבנה הכללי של הטבלה, וסוג הנתונים ID של Jet SQL משמש לטיפול בשדה autoincrement. סוג הנתונים ID יכול ערך התחלתי וערך פסיעה שמאפשרים לתת ערך התחלתי לשדה autoincrement, ואת ערך ההגדלה שלו בכל רשותה חדשה. השיגרה SetStartAndStep שלפניך מנצלת את הטכנולוגיה החדשה.

```
Sub SetStartAndStep()
Dim cnn1 As Connection
Dim cmd1 As Command
Dim tbl1 As New Table
    Set cnn1 = CurrentProject.Connection
    Set cmd1 = New ADODB.Command

    With cmd1
        .ActiveConnection = cnn1
    End With

    ' First create a table with two columns.
    ' Make one column an Identity column.
    ' Set its start value first, and its step value second.
    .CommandType = adCmdText
    .CommandText = "CREATE TABLE Contacts (ContactID " & _
                  "IDENTITY(2,4),ContactName Char)"
    .Execute

    ' After creating the table with the autoincrement/identity
    ' column you should add data.
    .CommandText = "INSERT INTO Contacts(ContactName) " & _
                  "Values ('Kevin Mineweaser')"
    .CommandType = adCmdText
    .Execute
End Sub
```

```

.CommandText = "INSERT INTO Contacts(ContactName) " &
    "Values ('Mike Gilbert')"
.CommandType = adCmdText
.Execute
.CommandText = "INSERT INTO Contacts(ContactName) " &
    "Values ('Neil Charney')"
.CommandType = adCmdText
.Execute
End With

End Sub

```

הקוד פותח ביצירת טבלה. למרות שהשיגרה אינה כוללת קוד לכלכית שגיאות, עליך להבטיח כי הטבלה שאתה יוצר אינה קיימת כבר, או כוללת מנגנון לכלכית שגיאות Smoother את הטבלה במידה שהוא אכן קיימת. מגדירים את סוג הנתונים את CREATE TABLE המשפט (contacts). הערך initial של ארגומנטו הראשון (ContactID) יתחלו בערך 2 ויגדל ב-4 עבור כל ערך חדש. תרשימים 2.13 מציג את פלט השיגרה 4. אלו הן ההגדרות של סוג הנתונים IDENTITY המצוין בשיגרה.

	ContactID	ContactName
▶	2	Kevin Mineweaser
	6	Mike Gilbert
	10	Neil Charney
*		(AutoNumber)

Record: [◀◀] [◀] [▶] [▶▶] [▶*] of 3

תרשים 2.13: פלט השיגרה SetStartAndStep

האובייקט View

אובייקט זה הוא שאילתת החזרת שורות שאינה מחייבת פרמטרים כלשהם. כשהיישום שומר את סוגי השאילותות האחרים, הוא מאחסן אותם כחברים באוסף Procedures. באפשרות לטפל בחברי האוסף Views באמצעות האובייקט Tables-Views. האובייקט View כולל שני מאפיינים קריטיים לנושא בו אנו עוסקים: המאפיין Name הוא שם התצוגה, והמאפיין Command מאפשר הגיעו למשפט SQL שعليו מבוססת התצוגה.

בסעיף זה ניצור תצוגה באמצעות השגירה MakeAView וגם נערוך את משפט SQL שלו בו מוצעת תצוגה קיימת באמצעות השגירה ChangeAView. השגורות Immediate ו- ViewAView נעזרות שתיהן בשגירה ChangeAView. הקוד מדפיס בחלון ChangeAView שלושה שדות מתוך כל תצוגה.

יצירת תצוגה. השגירה MakeAView שלפניך מוסיפה תצוגה חדשה על ידי יצירת אובייקט Command שמייצג את התצוגה, ומצרפת את האובייקט Command לאוסף cnn1 Connection האובייקט cnn1 Connection פותח קישור אל מסד נתונים Northwind.Views. לאחר מכן מציין את ערך המאפיין ActiveConnection של האובייקט Command שבוגמה, cmd1.cmd1. בעת מגע השלב CommandText: הקוד מגדר את המאפיין cmd1.CommandText של CommandText. המאפיין cmd1.magdir את משפט SQL שמנדר את התצוגה. הקוד מבנה את MakeAView במבנה את התצוגה כדי להציג שם ושם משפחה ולאחר מכן מספר שולחת הטלפון. הקוד מגדר את ערך המאפיין ActiveConnection של cmd1 (שהוא אובייקט מסוג Catalog), cmd1.cat1. במסד נתונים Northwind Append נותנת את השם AllEmployees ומוסיפה אותו לкатלוג MakeAView. לבסוף, קוראת Append ל视图 ViewAView כדי להדפיס את תוכן התצוגה בחלון **Immediate**.

```
Sub MakeAView()
Dim cnn1 As New ADODB.Connection
Dim cmd1 As New ADODB.Command
Dim cat1 As New ADOX.Catalog
'Open the connection.
    cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Program Files\Microsoft Office\Office\" & _
        "Samples\Northwind.mdb;"
' Create the command representing the view.
' Remember to delete it first.
    Set cmd1.ActiveConnection = cnn1
    cmd1.CommandText = "SELECT FirstName, LastName, " & _
        "Extension FROM Employees"
' Open the catalog.
    Set cat1.ActiveConnection = cnn1
' Create the new view.
    cat1.Views.Append "AllEmployees", cmd1
' Show view.
    ViewAView
End Sub
```

הדפסת תצוגה. השגירה ViewAView שלהן מדפסת את התצוגה AllEmployees. השגירה פותחת ביצירת חיבור אל מסד נתונים Northwind ולאחר מכן פותחת את האובייקט rst1 Recordset על המבוסס AllEmployees. כעורך הרשומות פותחה ומצביעה על מקור הנתונים המתאים, השגירה עוברת בלולאה על הרשומות

כדי למצוא את השם הארוך ביותר. ערכת הרשומות קטנה יחסית, ולכן אין לנו משלמים מחיר גבוה על ביצוע הלולאה הנוסף. נוח להתבסס על השם הארוך ביותר, מכיוון שכן מקבלים נקודת התחלה להדפסת השלווה לאחר השם. שלב זה מאפשר לישר את מספרי השלוות לשמאל ללא תלות באורכי השמות. בטרם מתחלים לעבור על השמות בלולאה נוספת, השיגרה ViewAView חוזרת לתחלת ערכת הרשומות. לאחר מכן, היא עוברת על השמות פעמיים נוספים, והפעם היא בונה מחוזות בעזרת המידע שנאסף בלולאה הראשונה.

```

Sub ViewAView()
Dim cnn1 As New ADODB.Connection
Dim rst1 As New ADODB.Recordset
Dim intMaxLength As Integer, Length As Integer

' Open the connection.
cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office\" & _
    "Office\Samples\Northwind.mdb;"

' Find the longest name.
rst1.Open "AllEmployees", cnn1
Do Until rst1.EOF
    Length = Len(rst1.Fields("FirstName")) + Len(rst1.Fields("LastName"))
    If Length > intMaxLength Then intMaxLength = Length
    rst1.MoveNext
Loop

' Print first name, last name, and third field.
rst1.MoveFirst
Do Until rst1.EOF
    strFiller = (intMaxLength + 2) - _
        (Len(rst1.Fields("FirstName")) + _
        Len(rst1.Fields("LastName")))
    Debug.Print rst1.Fields("FirstName") & " " & _
        rst1.Fields("LastName") & String(strFiller, " ") & rst1.Fields(2)
    rst1.MoveNext
Loop
End Sub

```

שינויי תצוגה. השיגרה ChangeAView שלhalten משנה את משפט SQL שעליו מבוססת התצוגה. באפשרות נצל גישה פשוטה זו כדי להוסיף שדות חדשים, להחליף שדות קיימים או אפילו לשנות את עיצוב התצוגה בשלהמו, על ידי הוספת רכיבי מיון וסינון. השיגרה משנה את התצוגה כך שתראה את השדה HomePhone במקום את השדה Extension.

```

Sub ChangeAView()
Dim cnn1 As New ADODB.Connection
Dim cat1 As New ADOX.Catalog
Dim cmd1 As New ADODB.Command
' Open the connection.
    cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Program Files\Microsoft Office\" & _
        "Office\Samples\Northwind.mdb;"
' Open the catalog.
    Set cat1.ActiveConnection = cnn1
' Update the view.
    cmd1.CommandText = "SELECT FirstName, LastName, " & _
        "HomePhone FROM Employees"
    cat1.Views("AllEmployees").Command = cmd1
' Show view.
    ViewAView
End Sub

```

כדי לעורך את משפט SQL שלו מבסיסת התצוגה, יש לפתוח את האובייקט הקשור Command ולחזור את המאפיין CommandText שלו. לשם כך, עליך להפנות את המאפיין ActiveConnection של האובייקט Catalog (ראה בקוד שלאיל) למסד הנתונים שמכיל את התצוגה שברצונך לשנות – במקרה זה, מסד הנתונים Northwind. הקצה את משפט SQL של התצוגה המבוקשת אל מאפיין CommandText של האובייקט Command החדש. אחר כך הציב את האובייקט Command החדש בתצוגה שברצונך לשנות. ההצבה תשמור אוטומטית את משפט SQL החדש במקום הקודם לו.

מחיקת תצוגה. השגירה DeleteAView שלפניך מוחckaת תצוגה אחת. היא פשוטה ביותר, אך כדי שתהיה זמינים, אם ברצונך להפעיל את MakeAView יותר מפעם אחת. הסיבה לכך היא Sh-ADO אינה מאפשרת לצרף תצוגה אחת על גבי תצוגה אחרת בעלת שם זהה, ולכן עליך למחוק תצוגה אחרת כדי להפעיל מחדש את Catalog. MakeAView כדי למחוק תצוגה, הגדר את ActiveConnection של אובייקט Catalog כך שיצבע על מסד נתונים הכלל את התצוגה שברצונך למחוק. לאחר מכן הפעיל את השיטה Delete של האוסף Views תוך הפניה לחברים הספציפיים שברצונך למחוק מתוך הסכמה.

```

Sub DeleteAView()
Dim cat1 As New ADOX.Catalog
' Open the catalog.
    cat1.ActiveConnection = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Program Files\Microsoft Office\Office\" & _
        "Samples\Northwind.mdb;"
' Delete the procedure.
    cat1.Views.Delete ("AllEmployees")
End Sub

```

האובייקט Procedure

שיגרה היא שאלת החזרת שורות מבוססת-פרמטרים שמחזירה שורות, או שאילתת פעולה שמוסיפה, מוחקת או מעדכנת רשומות. בסעיף זה עוסוק בשאלות מבוססות-פרמטרים ; עיין בפרק 4 לקבלת מידע מפורט יותר על שאלות, כולל שגורות שמחייבות שימוש בשאלות פולה.

קיימים דמיון בין שגורות ותצוגות : שתיהן יכולות להשתמש באובייקטי Command. תצוגות מייצגות שאלות החזרת שורות שאין מבוססות-פרמטרים , ואילו שגורות מייצגות את שאר סוגי השאלות. אם תמספר את האובייקטים מסוג Procedure בקטלוג, המספר יכלול את קבוצת כל התצוגות. לעומת זאת, שגורות אין מופיעות במספר אובייקטים מסוג View. תחביר המספר של שני סוגי השאלות זהה .

אובייקטים מסוג Procedure כוללים שני מאפיינים קריטיים : Name (שם) ו-CommandText (פקודה). מאפיין Name הוא שם השיגרה. מאפיין CommandText מאפשר גישה למאפייני אובייקט Command שעליו מבוססת השיגרה. המאפיין CommandText שימושי במיוחד, מכיוון שהוא מגדר או מחזיר את משפט SQL עבור האובייקט Command. תוכל לנצל מאפיין זה כדי לשנות את משפט SQL באובייקט Command שעליו מבוססת השיגרה, או כדי להציג את משפט SQL של האובייקט. בחינת משפט SQL של האובייקט יכולה לידע אותך על הפרמטרים שעлик להגדיר כדי להפעיל את השיגרה.

מספר תצוגות ושגורות. השיגרה LinkMyProcs שלහן מציגה בפועל את האוספים Views (תצוגות) ו-Procedures (שגורות). לוLATat הרשותה ממספרת את חברי האוסף Views בלבד. הלולהה השנייה ממספרת את האוסף Procedures ומציגת את השגורות והתצוגות ייחדו. הלולהה השנייה גם את משפט SQL של כל שאלתה שהיא ממספרת.

```
Sub ListMyProcs()
Dim cnn1 As New Connection
Dim cat1 As New Catalog
Dim proc1 As Procedure
Dim view1 As View
' Set database connection for catalog.
cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office" & _
    "\Office\Samples\Northwind.mdb;"
    Set cat1.ActiveConnection = cnn1
' Enumerate views -- notice this returns just
' nonparameterized row-returning queries.
    For Each view1 In cat1.Views
        Debug.Print "View name: " & view1.Name
    Next view1
```

```

' Enumerate views -- this returns views and procedures.
For Each proc1 In cat1.Procedures
    Debug.Print "Procedure name: " & proc1.Name
    If proc1.Name <> "Invoices" Then
        Debug.Print "SQL: " & proc1.Command.CommandText
    End If
Next proc1
End Sub

```

הצוגות ושגרות מאפשרות להשתמש באובייקטי Command. כל שעליך לעשות הוא לזרף את האובייקט Command לאוסף המתאים. הקוד שלפניך שומר שאילתת פרמטר כסוגה מאוחסנת. השאלתה משתמשת בטבלה MyTable (שנופיעה בחלק מדוגמאות השגורות שהוצעו בפרק), וניתן להוריד מהתקליטור הנלווה לספר, יחד עם דוגמאות לטיפול בה. התרשיםים 2.7 ו-2.8 מציגים את התצוגות **עיצוב** (Design) ו**גיליון נתונים** (.MyTable) של הטבלה (DataSheet)

```

Sub SaveParameterQuery()
On Error GoTo SavePQTrap
Dim cmd1 As Command
Dim rs1 As Recordset, str1 As String
Dim fldLoop As ADODB.Field
Dim prm1 As ADODB.Parameter, int1 As Integer
Dim cat1 As New ADOX.Catalog
Dim cnn1 As ADODB.Connection

Set cnn1 = CurrentProject.Connection

'Create and define command.
    Set cmd1 = New ADODB.Command

With cmd1
    .ActiveConnection = cnn1
    .CommandText = "Parameters [Lowest] Long;" & _
        "SELECT Column1, Column2, Column3 " & _
        "FROM MyTable " & _
        "WHERE Column1>=[Lowest]"
    . CommandType = adCmdText
End With

'Open the catalog.
    Set cat1.ActiveConnection = cnn1

' Create the new procedure based on parameter query.
    cat1.Procedures.Append "spLowestRow", cmd1

SavePQExit:
    Exit Sub

```

```

SavePQTrap:
If Err.Number = -2147217816 Then
'If err.number = -214... query already exists
    deleteProcedure("spLowestRow")
    Resume
Else
    Debug.Print Err.Number, Err.Description
End If

End Sub

Sub deleteProcedure(procName as String)
Dim cnn1 As New Connection
Dim cat1 As New adox.Catalog

'Open the catalog.
Set cnn1 = CurrentProject.Connection
Set cat1.ActiveConnection = cnn1

'Delete existing procedure.
cat1.procedures.Delete (procName)

End Sub

```

השיגרה בונה שאלתת פרמטר ושומרת אותה כSIGERA מאוחסנת. השאלתת מבקשת שהערך הנמוך ביותר יופיע בעמודה הראשונה. שים לב כי באפשרות להגדיר את המאפיינים CommandType ו CommandText של השאלתת. משפט SQL כולל הצהרה על הפרטט. לאחר יצירת האובייקט Command של השיגרה המאוחסנת, הפעל את השיטה Append של האוסף Procedures כדי למש את האובייקט כSIGERA מאוחסנת. אם תפעיל את SaveParameterQuery יותר מפעם אחת, הניסיון השני לשומר את השיגרה המאוחסנת spLowestRow יגרום לשגיאת זמן ריצה.LOCATED שגיאה זו ומוחקת את העותק היישן אם הוא קיים כבר. השיגרה שמוחקת את השיגרה המאוחסנת קולטת ארגומנט עבור שם השיגרה המיועדת למחיקה. השיגרה ParameterQCommand שהוצגה קודם לכן, מבצעת משימה זהה לו של השיגרה המאוחסנת ADO .spLowestRow מפעילה שגורות מאוחסנות כאובייקטים מהודרים, ולכן הן פועלות מהר יותר מאשר אובייקט Command שיש ליצור כל פעם מחדש לשם ביצוע משפט SQL.

יצירה והפעלה של שגורות מאוחסנות. זוג השגורות הבא מכין ומפעיל SIGERA מאוחסנת. הראשונה יוצרת SIGERA מאוחסנת שמחפשת את מסטר שלוחות הטלפון של עובדים במסד הנתונים Northwind. השיגרה השנייה מפעילה את השיגרה המאוחסנת.

```

Sub procLookupNumber()
Dim cnn1 As New ADODB.Connection
Dim cmd1 As New ADODB.Command
Dim prm1 As ADODB.Parameter
Dim cat1 As New ADOX.Catalog

' Open the connection.
cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office" & _
    "\Office\Samples\Northwind.mdb;"

' Create the parameterized command.
Set cmd1.ActiveConnection = cnn1
cmd1.CommandText = "SELECT FirstName, LastName, Extension " & _
    "FROM Employees WHERE LastName = [LName]"
Set prm1 = cmd1.CreateParameter("[LName]", adWChar, adParamInput, 20)
cmd1.Parameters.Append prm1

' Open the catalog.
Set cat1.ActiveConnection = cnn1

' Create the new procedure based on parameter query.
cat1.Procedures.Append "spEmployeeExtension", cmd1
End Sub

Sub RunLookUpProc()
Dim cnn1 As New Connection
Dim cat1 As New Catalog
Dim rst1 As New Recordset
Dim cmd1 As New Command
Dim prm1 As Parameter
Dim typedName As String

' Create and assign a connection for the catalog.
cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office" & _
    "\Office\Samples\Northwind.mdb;"
Set cat1.ActiveConnection = cnn1

' Set the Command and parameter object references.
Set cmd1 = cat1.Procedures("spEmployeeExtension").Command
Set prm1 = cmd1.CreateParameter("[LName]", adWChar, adParamInput, 20)
cmd1.Parameters.Append prm1

```

```

' Gather the parameter value from the user and assign it.
typedName = InputBox("Last name for extension?", _
    "Programming Microsoft Access 2000")
prm1.Value = typedName

'If prm1.Value <> "" Then
'Execute the parameter query and show first match.
    cmd1.Execute
    rst1.Open cmd1
    MsgBox "The extension for " & rst1.Fields(0) & _
        " " & rst1("LastName") & " is " & rst1.Fields(2), _
        vbInformation, "Programming Microsoft Access 2000"
End If
End Sub

```

השיגרה ProcLookupNumber יוצרת שאליתת פרמטר לשגירה מאוחסנת. שאליתה זו מחפשת אחר מספר שלוחות טלפון של עובד, לפי שם המשפחה שלו. השיגרה מדירה את האובייקט Command והפרמטר הקשור אליו [LName]. הקבוע AdWChar מדיר את הפרמטר כמחרוזת טקסט ברוחב קבוע. הקבוע adParamInput מדיר את הפרמטר בלבד בלבד, והמספר הנגיד מציין שהפרמטר יכול להכיל עד 20 תוים. לאחר הגדרת האובייקט Command והפרמטר הקשור אליו, השיגרה מדירה את חיבור הקטלוג אל מסד הנתונים Northwind ומצורפת את האובייקט Command לאוסף Procedures תחת השם .spEmployeeExtension.

אם תפעיל את procLookupNumber יותר מפעם אחת, היא תיכשל בעת ניסיון לכתוב עותק חדש של spEmployeeExtension על גבי זה המקורי. הוסף לשיגרה קוד ל釐ידת שגיאות לטיפול במצב מעין זה. לאחר מכן צור שגרת מחיקה. יהיה عليك לעורך את שגרת המחיקה שבדוגמה הקודמת, מכיוון שהוא מוחקת שגרות בפרויקט הנוכחי, בעוד שבדוגמה זו עליה למחוק שיגרה במסד הנתונים Northwind.

השיגרה RunLookUpProc מפעילה את השיגרה המאוחסנת שנוצרה על ידי השיגרה procLookupNumber. שיגרה קצרה זו מבצעת ארבע מטלות:

- ﴿ יצירת חיבור אל מסד הנתונים באמצעות השיגרה המאוחסנת.
- ﴿ הצבת משתני אובייקט שמצוינים על השיגרה המאוחסנת והפרמטרים שלה. הקוד מנצל אובייקט פקודה כדי להפנות אליה.
- ﴿ הצגת תיבת דו-שיח בה משתמש חיב לציין את שם המשפחה של העובד. הקוד מציב את ערך החזרה מתוך תיבת הדו-שיח בפרמטר הפוקודה.
- ﴿ ביצוע הפוקודה ופתיחת ערכת רשומות המבוססת על קבוצת החזרה של הפוקודה. השיגרה מציגה את השורה הראשונה מקבוצת החזרה בתיבת דו-שיח. הקוד משתמש בשתי מערכות כלליות מנתן השמות כדי לבחור חבר מתוך האוסף Fields בערכת רשומות. תוכל לבחור את המקום הסידורי או את המאפיין Name שלו.

עיצוב טבלאות

עסקים רבים המשמשים במסדי נתונים, אינם מיעחדים את החשיבות הראوية לעיצוב טבלאות. במקומות זאת הם נוטים להתמקד בעיצוב טפסים ודווחות, אלמנטים הגלויים למשתמש. חשוב להציג כי העיצוב והתוכן של טבלאות וקשרי הגומלין ביניהם, יכולים להשפיע משמעותית על השירותים שניתן לספק באמצעות פתרונות מסד נתונים מותאמים אישית. עיצוב טבלאות יכול להשפיע על עיצוב מסדי הנתונים האלה וגם על קלות הוספה שירותים חדשים למערכת.

פרק זה פותח בסקירה קצרה של שיקולים כלליים בעיצוב טבלאות, כגון פיצול המידע בקבוצותalogies כדי שניתן יהיה לאחסנו בטבלאות שונות. אחר זו הפרק ביצירת טבלאות באמצעות Access 2000, בוצרה ידנית ובעזרת האשף של Access. נלמד גם אודוות אינדקסים, מפתחות ראשיים ועל אופן יצירת קשרי גומלין בין טבלאות.

הפרק מלמד כיצד ליצור טבלאות באופן דינמי ולملא אותן בתנאים, יכולת שמאפשרת ליצור פתרונות מסדי נתונים רבי-עוצמה ועתירני וככיבים. התקליטור המצורף מכיל דוגמאות ובות של אובייקטי נתונים ActiveX (ADO - ActiveX Data Objects) - שיוצרים טבלאות Access באופן דינמי וממלאים אותן בתנוני Access. דוגמאות אלו מבוססות על המבואה ל-ADO שהוזג בפרק 2. קבוצת דוגמאות אחת אף מציגה כיצד לשלב את ADO ואת האובייקט הידידותי DoCmd כדי לתקשר עם מקורות נתונים ISAM (Indexed Sequential Access Method) ו- ODBC (קישוריות מסד נתונים פתוחה).

טבלאות ובסיס נתונים יחסיים

טבלאות נמנעות עם יחידות המבנה הבסיסיות ביותר של מסד נתונים יחסי. דוגמת מסד הנתונים Northwind המשווקת עם Access, היא דוגמה למסד נתונים המנהל עסק דמיוני. מסד נתונים זה כולל טבלאות הקשורות בעקבות אחר מידע חשוב. הטבלה Customers עוקבת אחר מידע על לקוחות, כגון שם החברה ושם איש קשר. הטבלה Products עוקבת אחר נתונים הקשורים למוצר, כגון שם המוצר, מספר ייחודי במאלי ומחריך היחידה.

כל טבלה במסד נתונים חייבת להכיל מידע המתאים ליישות מסווג מוגדר אחד בלבד. לדוגמה, בית-ספר טיפוסי כולל בין היתר, תלמידים וכיתות. תלמידים רשומים בכיתות; מורים מלמדים בכיתות. מסד הנתונים של בית הספר עשוי להכיל טבלת תלמידים (Students), טבלת כיתות (Classes) וטבלת מורים (Teachers). הטבלה **מורים** צריכה להכיל מידע על מורים, אך לא צריך להיות בה מידע על תלמידים או כיתות. באופן דומה, הטבלה **כיתות** אינה אמורה להכיל פרטי מורים, כגון כתובות, מספרי תעודה זהות ומספרי טלפון.

טבלאות מסד נתונים דומות במבנה שלחן לגילוונות אלקטרוניים. **שורות** (Rows) או **רשומות** (Records) בטבלה מייצגות מופעים ייחודיים של היחסות המאוחסנת בטבלה. לדוגמה, הטבלה Customers כוללת שורה אחת לכל לקוח, ואילו הטבלה Products כוללת שורה אחת עבור כל מוצר (אין חשיבות לכלי השורות; תוכל לסדר אותן כרצונך מבלי לשנות את משמעות המידע שבטבלה).

כל **שדה** (Field) בטבלה מכיל מידע מסווג מוגדר. לדוגמה, לכל אחד מהלקוחות במסד הנתונים Northwind יש שם, כתובות, מספר טלפון וכן הלאה - מידע זה מאוחסן בשדות הטבלה **Customers**.

טבלאות מסד נתונים רבות כוללות שדה אחד או יותר שמצויה באופן ייחודי כל שורה בטבלה. זיהוי ייחודי זה נקרא **מפתח ראשי** (Primary Key). המפתח הראשי של הטבלה Customers שבמסד הנתונים Northwind הוא השדה CustomerID (מצויה לקוח).

נרטול

נרטול (normalization) הוא חילה של קבוצת כללי עיצוב על טבלאות מסד נתונים. נרטול ארבעה יתרונות עיקריים:

↳ **ביטול מידע מיותר**. מסדי נתונים רבים שלא עברו נרטול מחיברים להכנס אוטם נתונים של אנשי קשר בנסיבות רבות. ביטול הנסיבות מצמצם סיכון קליטה של נתונים שגויים, דבר שולול לשבש את מסד הנתונים. הנרטול יכול גם לפשט את תחזוקת מסד הנתונים, כיון שקל יותר למחוק או לעדכן ערך המאוחסן במקום אחד בלבד.

↳ **מצומצם גודל מסך הנתונים.** כל סוג מידע מאוחסן במקומות אחד בלבד, ולכן מסך הנתונים אינו צריך לשמר עوتקים רבים של פרטי מידע זרים. הנרמול אף ממצמצם את מספר העמודות בטבלה, מה שמקטין את מסך הנתונים כולו.

↳ **פישוט תהליכי חיפוש.** מקצועני מסדי נתונים המתמחאים בכללי הנרמול, יידעו כיצד לננות בטבלאות מסד הנתונים לשם איתור מידע. משתמשי מסדי נתונים מזדמנים יכולים להבין טוב יותר את עיצוב הטבלה, מכיוון שכל טבלה מייצגת יישות יחידה שככל מאפייניה הן עמודות בטבלה זו.

↳ **פישוט ביצוע שאלות.** עמודת טבלה מאחסנת סוג נתונים יחיד, כגון שם פרטי או שם משפחה, אך לא את שני סוגי השם. אחסון שם המשפחה בעמודה נפרדת מאפשר להפיק בקלות רשימה של כל שורות מסד הנתונים המכילות שם המשפחה זהה. במסד נתונים לא מנורמל שמות פרטיים ושמות משפחה באותה עמודה, יש לחוץ תחילתה את שם המשפחה לפני ביצוע שאלתה המטפלת בשמות.

קיימים שלושה כללי נורמל מוקובלים : **First Normal Form** (כלל הנרמול הראשון), **Second Normal Form** (כלל הנרמול השני) ו- **Third Normal Form** (כלל הנרמול השלישי). קיימים גם כלליים מיוחדים רבים נוספים.

כלל הנרמול הראשון - First Normal Form

כלל הנרמול הראשון קובע כי כל שדה בטבלה חייב להכיל פרטי יחיד שאינו ניתן לחלוקה. הטבלה Order Details של מסד הנתונים Northwind ממחישה חוק זה. תרשימים 3.1 מציג קטע מהטבלה ובו שלוש הזמנות. כל עמודה בטבלה מכילה סוג נתונים יחיד. העמודה השנייה מכילה שמות מוצרים, אך לא שמות ומחרירים גם יחד. בנוסף, כל שורה בטבלה כוללת מופע אחד בלבד של מוצר.

Order Details : Table					
Order ID	Product	Unit Price	Quantity	Discount	
10248	Queso Cabrales	\$14.00	12	0%	
10248	Singaporean Hokkien Fried Mee	\$9.80	10	0%	
10248	Mozzarella di Giovanni	\$34.80	5	0%	
10249	Tofu	\$18.60	9	0%	
10249	Manjimup Dried Apples	\$42.40	40	0%	
10250	Jack's New England Clam Chowder	\$7.70	10	0%	
10250	Manjimup Dried Apples	\$42.40	35	15%	
10250	Louisiana Fiery Hot Pepper Sauce	\$16.80	15	15%	

תרשים 3.1: קטע מהטבלה Order Details הממחיש את כלל הנרמול הראשון. האלמנטים בכל עמודה (או שדה) נמנים עם סוג נתונים יחיד; האלמנטים שבעמודות השורה אינם ניתנים לפירוק

טבלאות מסד נתונים שאינן עומדות בדרישות כלל הנרמול הראשון, מתאפיינות בדרך כלל בשתי תכונות: ראשית, הן מאחסנות יותר פרטי אחד באותו שדה (שים לב שהטבלה בתרשים 3.1 מכילה פרטי בוודד בכל שדה). דוגמה לאי-עמידה בדרישות הכלל הראשון היא טבלה ששדה השם מכיל שם פרטי ושם משפחה. צירוף סוגי

מידע בשדה מסווג מסבך את תהליך של ליפת המידע. הפרה שכיחה נוספת נסافت של כל הנרמול הראשון היא טבלה המכילה שדה לכל מוצר בהזמנה, דבר שחייב ליצור טבלה בעלת שדות רבים של מחיר, כמות והנחה, ובמהמשך גורם להפרת הכלל הראשון. עיצוב כזה "מנפח" את הטבלה, וחלק גודל מערכיו עם דוחות יהיה ריקם.

טבלאות העומדות בדרישות כלל הנרמול הראשון מבוססות בדרך כלל על יישיותalogies שאפשרות לאת מידע בקהלות. לעיתים קרובות יש להן מפתח ראשי שמבטיח כי כל שורה בטבלה היא ייחודית. כזכור, טבלאות מסדי נתונים מייצגות מופעי יישיות. בטבלה שבתרשים 3.1, כל שורה מייצגת פריט בהזמנה ללקוח. המפתח הראשי של הטבלה הוא מפתח מרכיב (Compound Key) שבסיסו על השדות OrderID ו-ProductID, מכיוון שנitinן זהות כל פריט בהזמנה הלוקה באמצעות צירוף ייחודי של מספר הזמנת הלוקה ומזהה המוצר.

– כלל הנרמול השני – Second Normal Form

כלל הנרמול השני מצין קשרי גומלין נדרשים בין העמודות בשורה. כדי לעמוד בדרישת הכלל השני, טבלה חייבת לעמוד קודם כל בתנאי כלל הנרמול הראשון. בנוסף, כל הערכים בשורה חייבים להכיל מידע על היעילות הספציפית שמייצגת באמצעות השורה. דרישת נוספת: אסור שתהייה תלות הדידית בין שני שדות כלשהם. אחת ההפרות הנפוצות של כלל זה היא הצבת שדות לשתי יישיות ייחודיות בשורה אחת.

הטבלה Order Details של Northwind עומדת בדרישות חוק הנרמול השני. שדה הכמות מתאים בצורה ברורה לכל מוצר. מחיר היחידה והנחה יכולים להשנות בנסיבות עצמאיות לכל פריט, כאשר הצבת מחיר ייחידה והנחה בטבלה מאפשרת ליעשות לעקבם בקהלות אחר פריטים אלה לכל הזמנה.

במסד הנתונים Northwind קיים שדה מחיר ייחידה הנקרא בטבלה Order Details והוא בטבלה Products. זה נראה כמו הפרה של כלל הנרמול השני, אולם עיצוב זה מאפשר למנהל לעדכן מחירי פריטים מבלי להשפיע על מחירי פריטים שכבר הוזמנו. ההזדמנויות יכולות לנצל את המחיר והנחה שבטבלה Products בתור הנחיה ולאו דווקא בכלל ברזל לכל העסקאות.

– כלל הנרמול השלישי – Third Normal Form

כלל הנירמול השלישי קובע כי כל השדות בכל שורה חייבים להיות ייחודיים ובלתי תלויים זה בזה. לדוגמה, החוק מאפשר רק שדה תאריך אחד בשורה. אם הטבלה Orders מכילה שדה תאריך הזמנה, אסור שתכיל שדות נוספים נספחים של יום, שבוע או חודש של הזמנה (הישום יכול לחלז בעצמו את החודש מזמן תאריך הזמנה). הטבלה Orders מכילה שדות אחדים שמקילים ערכי שעיה/תאריך (Date/Time), אך אלה מתייחסים למועד הזמנה המקורי, התאריך בו הגיעו דרישת הלוקה, יום האספקה וכן הלאה.

עמידה בכללי הנרמול

נובן יהיה למלא את דרישות הכללים, אך בנסיבות מסוימות ייתכן שתרצה לעשות זאת בצורה סלקטיבית - למשל, אם ברצונך לצמצם את מספר הtablאות. השודת מיקוד, קוד אзор וקוד מדינה (בארה"ב) יכולים להתקיים בטבלה נפרדת. אך אם ידוע לך המיקוד של כתובות מסוימת, ידועים לך גם העיר, האזור, או המדינה, וכן אין צורך לשמור את המיקוד ושם העיר באותה טבלה.

בפירושים מסוימים של מסדי נתונים קטנים ובינוניים יצירט טבלאות השומרות מידע על מיקוד, שם עיר, אזור ומדינה עלולה להיות בלתי מעשית. גם בפירוש מסד נתונים גדול מאוד, כשנדרשים פרטי אישי קשר, תכונן מסווג זה מחייב לצרף את הטבלה הראשית Contacts עם טבלאות רבות בכל מקרה הדורש מידע אישי קשר. הדבר עלול לפגוע קשות בביוצרי המערכת, במיוחד אם צירוף הטבלאות נדרש פעמים רבות.

קשרי גומלין בין טבלאות

בעת יצירט טבלאות, عليك להביא בחשבון גם את קשרי הגומלין ביניהן. קשרים אלה מעניקים למסד נתונים **יחס** (relational) את רוב עצמתו. קיימים שלושה סוגים קשרי גומלין: **יחיד-ליךידה** (One-to-One), **יחיד-לרביבים** (One-to-Many) ו**רביבים-לרביבים** (Many-to-Many).

קשרי גומלין מסווג יחיד-ליךידה

במערכת קשרי גומלין מסווג יחיד-ליךידה, כל רשומה בטבלה אחראית מותאמת לרשותה יחידה בטבלה אחרת. סוג קשר זה אינו נפוץ במיוחד, אך ביכולתו להציג מספר יתרונות. ראשית, תוכל להציג שדות משתי הטבלאות בטבלה משולבת. אחת הסיבות לשימוש בשתי טבלאות היא שכלה הוא מאפיין של יישות נפרדת, כגון נהגים ומשאיות. כל נהג יכול לנוהג משאית אחת בלבד בזמן נתון, אך שדות הטבלאות של הנהגים והמשאיות מתיחסים ליישויות שונות.

קשרי גומלין מסווג יחיד-ליךידה יכולים גם לצמצם את הזמן הדרוש לפיתוח טבלה גדולה על ידי הצבת חלק מעמודות הטבלה בטבלה נפרדת. גישה זו הгинונית במילוי שטבלה מכילה שדות אחדים שהשימוש בהם אינו תקין. לבסוף, קשרי גומלין מסווג יחיד-ליךידה יכולים לתמוך באבטחה. Access מיישם אבטחה ברמת-משתמש ברמת הטבלה. כך, אם קבוצת-משנה של שדות בטבלה מחייבת הגנה באמצעות אבטחה, הצבת שדות אלה בטבלה נפרדת מאפשרת ליישום להגביל את הגישה לשדות מסוימים. היישום יכול לחשור את הטבלה המוגבלת בחזרה לטבלה הראשית באמצעות קשרי גומלין מסווג יחיד-ליךידה, ועל ידי כך לאפשר למי שמחזיק בהרשאות מתאימות לעורך, למחוק ולהוסיף רשומות חדשות לשדות אלה.

קשרי גומליין מסווג ייחיד לרבים

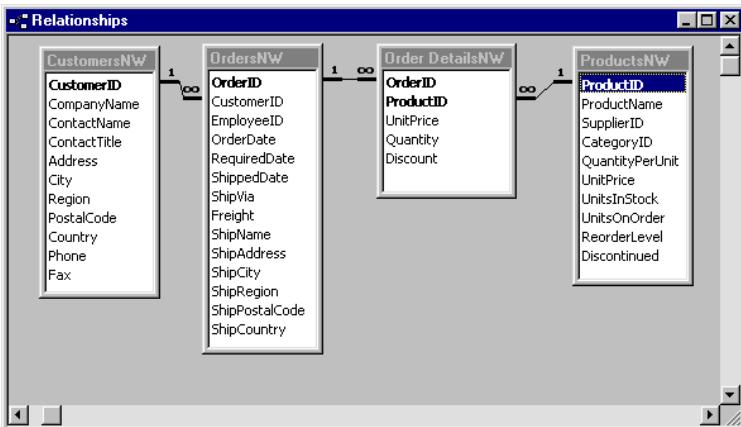
נפוץ יותר הוא הקשר ייחיד-ל רבים, שבו שורה אחת מתאימה לשורה אחת או יותר מהטבלה השנייה. סוג קשר זה יכול ליצור גם בסיס לקשרי גומליין מסווג רבים-ל רבים. הטבלאות Customers (לקוחות) ו- Orders (הזמנות) של מסד הנתונים Northwind מקיימות קשרי גומליין מסווג ייחיד-רבים. לדוגמה כלשהוא יכולה להיות הזמנה אחת או יותר, אך כל הזמנה שייכת רק ללקוח אחד. תרשימים 3.2 מציג את תצוגת **עיצוב** (Design) של הטבלה Orders. המפתח הראשי הוא OrderID. כל אחד מערכי ID OrderID יכול להופיע בשורה אחת בלבד. הטבלה מכילה גם שדה שנקרא CustomerID, שמקורו לטבלה Customers. השדה CustomerID הוא המפתח הראשי של הטבלה Customers. שדה המשמש כ קישור לטבלה אחרת, נקרא **מפתח זר**. שדה יכול להופיע פעמיים רבעות בטבלה Orders. (Foreign Key). ערכי CustomerID יכולים להופיע פעמיים רבעות בטבלה Orders. מפתח זר זה של דבר, קוד זיהוי הלוקוט מופיע פעמיים רבעות בכל הזמנה שהלוקוט מופיע. מפתח זר זה מקשר את צד ה"רבים" של מערכת קשרי הגומליין חוזרת לצד ה"יחיד".

תרשים 3.2: תצוגת **עיצוב** של הטבלה Orders; השדה CustomerID משמש כמפתח זר בקשרי הגומליין מסווג ייחיד-רבים בין הטבלה Orders לטבלה Customers.

קשרי גומליין מסווג רבים-רבים

קשרי גומליין מסווג רבים-רבים קיימים בצורה עקיפה בלבד ומבוססים על שתי מערכות קשרי גומליין מסווג ייחיד-רבים. הטבלאות Orders ו- Products מקיימות ביניהן קשרי גומליין מסווג ייחיד-רבים. הזמנה כלשהיא יכולה להכיל מוצרים רבים. באופן דומה, מוצר ייחיד יכול להופיע בהזמנות רבות ושונות.

ממסד הנתונים Northwind יובאו ארבע טבלאות לדוגמת מסד הנתונים של פרק זה: ProductsNW ,Orders DetailsNW ,CustomersNW ,CustomersNW Relationships (Relationships). תרשימים 3.3 מציג בחלון CustomersNW Relationships את קשרי הלקוחות בין הטבלאות. הטבלה OrdersNW מקיימת קשרי גומלין מסוג יחיד-לרבבים עם הטבלה Order DetailsNW ,הטבלה Order DetailsNW מקיימת קשרי גומלין מסוג יחיד-לרבבים עם הטבלה ProductsNW .Order DetailsNW מקיימת קשרי גומלין מסוג יחיד-לרבבים עם הטבלה ProductsNW . הטבלה Order DetailsNW משמשת כ"תיבת הסתעפויות" שמקשרת את הטבלאות ProductsNW ו-OrdersNW בקשרי גומלין מסוג רבים-לרבבים.



תרשים 3.3: החילון קשרי גומלין מציג את קשרי הלקוחות מסוג רבים-לרבבים בין הטבלאות ProductsNW ו-OrdersNW

קשרי גומלין מסוג רבים-לרבבים מעוצבים באמצעות "תיבת הסתעפויות", כשתבלה שלישית, תיבת הסתעפויות, משמשת כ קישור בין שתי הטבלאות האחרות במערכת קשרי הלקוחות. עיצוב זה מסייע לשימור על ייעילות מסד הנתונים, כיוון שתיבת הסתעפויות יכולה להכיל מספר שורות גדול, אך מספר עמודות קטן. בדומה לכל הטבלאות שעברו נרמול, עליך להגביל את העמודות לאלו שמאפייניות את היישוויות. כלל זה חשוב במיוחד לגבי טבלאות מסוג תיבת הסתעפויות, מכיוון שהן עלולות להכיל מספר שורות גדול יחסית.

שלמות הקשרים

אפשר להקנות חוסן למסד הנתונים באמצעות **שלמות הקשרים** (Referential integrity) ו**עדכוניות ומחיקות על פי היררכית הקשרים** (Cascading updates and deletes). כללי שלמות הקשרים מבטיחים את חוקיות הקשרים הקיימים בין טבלאות. כלליים אלה גם מונעים שינוי בשוגג של נתונים קשורים (לדוגמה, לא כדאי למחוק לקוח, אם יש לו הזמנה שלא שולמה או מספר הזמן כלאו).

שלמות הקשרים אינה מאפשרת הוספת שורה לצד ה"רבים" בקשרי הגומלין באמצעות ערך מפתח זר שאינו תואם לערכים שבצד ה"יחיד" של הקשר. אך אפשר להכניס ערך מפתח זר חדש או ריק (Null) שאינו תואם מפתח ראשי כלשהו לצד ה"יחיד" של קישורים הגומلين.

שלמות הקשרים מסייעת גם להימנע מצב של רשומות יתומות - רשומות שנמצאות הצד ה"רבים" של קישורים הגומلين, שאין להן בנות-זוג בצד ה"יחיד". הדבר מאפשר הודות לחסימת אפשרות מחיקת רשומות הצד ה"יחיד", אם יש להן עדין בנות-זוג הצד ה"רבים" (לדוגמה, מן הסתם לא יעלה על דעתך להסיר את המכירות הקשורות לעובד מסוים רק משום שהוא עזב את החברה). רשומות יתומות מנמקות את מסד הנתונים ברשומות מיותרות ולא שימושיות; במקרים קיצוניים יותר, הן עלולות לעורר את חוקיות התוצאות המשוגות מממד הנתונים.

לעתים תרצה שהשינויים יתבצעו בשני צדי קשר הגומلين. במקרים אלה תכונות המחיקה והעדכון על פי היררכית קישורים של שלמות הקשרים עשויים להיות שימושיים. תכונות אלו אינן מופעלות אוטומטית; עליך להפעיל במפורש. אם תפעיל מחיקה על פי היררכית קישורים, מחיקת רשומה לשתי הצדדים הצד ה"יחיד" של קשר הגומلين תסיר את כל הרשומות התואמות הצד ה"רבים". במקרה של שינויים בפתח ראשי עדכונים על פי היררכית קישורים פועלם בצורה דומה. אם תשנה ערך מפתח ראשי מצד ה"יחיד" של קשר הגומלי, הוא יתbeta גם הצד ה"רבים". כל ערכי המפתח הראשי התואמים מתעדכנים אוטומטית.

יצירת טבלאות באמצעות אשפי

אשפי Access מפשטים מטלות רבות, כגון יצירה ותחזוקה של טבלאות. ל-Access 2000 אשפי מסד נתונים, למשל: Time and Billing, Contact Management, Collection Management Video ו- Call Management. כל אלה יוצרים יישומים שלמים, כולל עיצוב הטבלאות. אשף הטבלאות ייחדoot ואף מוסיף לכל אחת מהן מפתחות ראשיים ומקשר ביןיהן. האשף **בונה שדות** (Field Builder) מסייע לתחזוק טבלאות על ידי פישוט הוספת השדות.

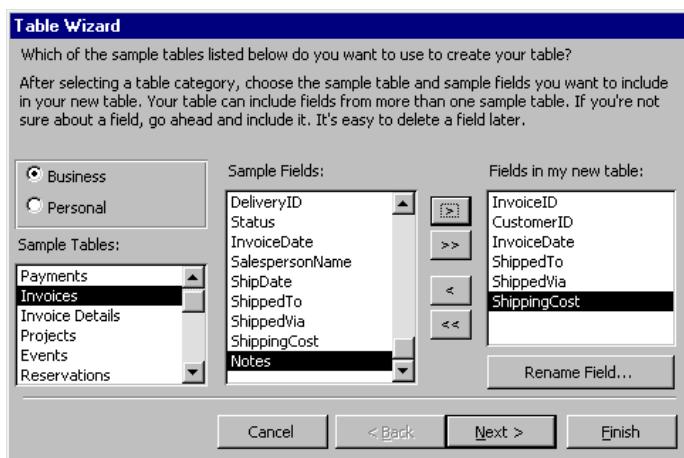
אשפי מסד הנתונים

בעת הפעלת Access 2000 תוכל להציג את אשפי מסד הנתונים, על ידי בחירה באפשרות **אשפי מסד נתונים, דפים ופרויקטים של Access Database** (Microsoft Access Wizards, Pages And Projects) בתיבת הדוח-שים (Wizards) בתיבת הדוח-שים (Databases) ולהיצה על אישור. בהפעלת תיבת הדוח-שים **חדש** (New),בחר בקרטיסיה **מסדי נתונים** (DataBases) ולאחר כך לחץ להיצה כפולה על סמל האשף שברצונך להפעיל (לחילופין, ניתן לפתוח את תיבת הדוח-שים מתפריט **קובץ** (File), האפשרות **חדש**).

מסד נתונים שנוצר על ידי אשף מושלם מבחינה תפקודית למשתמש לא מנוסה, מכיוון שהוא כולל טפסים, טבלאות, דוחות ואנ' נתונים לדוגמה. תוכל לבחון את עיצוב מסד הנתונים ולהשתמש בו כמודל ליישומים מותאמים אישית. מפתחים מנוסים יותר יכולים לעורוך את העיצוב הבסיסי על ידי הוספה טבלאות, שאלות, טפסים, דוחות ומודולים.

אשף הטבלאות

אשף הטבלאות מתמקד בלהיות בעיצוב טבלה. תוכל לפתחו אותו מתוך הפריט **טבלאות** (Tables) באוסף **אובייקטים** (Objects) שבחלון **מסדי נתונים** (Database). לחץ>Create Table By Using Wizard אשף **טבלה באמצעות אשף** (Table Wizard). אשף זה כדי לפתח את תיבת הדו-שיח התחלתי **אשף הטבלאות** (Table Wizard). מציע מגוון יעיצובים מוכנים מראש בסגנון עסקי או אישי. יעיצובי הטבלאות העסקיים כוללים בין היתר את **אנשים קשר** (Contacts), **לקוחות** (Customers), **עובדים** (Employees), **מוצרים** (Products), **זמןנות** (Orders), **חשבוניות** (Invoices) ו**אירועים** (Events). יעיצובי הטבלאות האישיים עונים על כל הצרכים ויכולים לשרת גם מגוון צרכים של עסקים קטנים. בין העיצובים המוצעים: **מתכונים** (Recipes), **צמחיים** (Plants), **יומן תרגילים** (Exercise Log), **ספרים** (Books), **אוסף וידאו** (Video Collection) והקלטות (Recordings). האשף אינו מלא את הטבלה בתנאים ראשוניים, אך הוא מספק קישורים לטבלאות אחרות שכבר כוללות בעיצוב מסד הנתונים. הוא איפלו בנה קשרי גומלין אלה בעצמו.

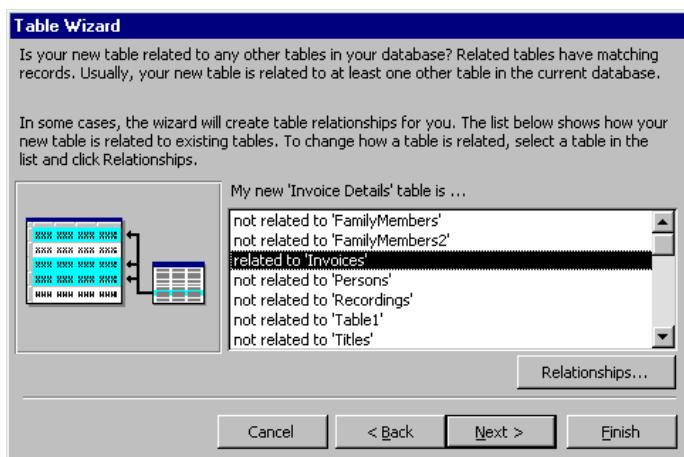


תרשים 3.4: ייצרת הטבלה **חשבוניות** בתיבת הדו-שיח הפתוחת של **אשף הטבלאות**

לאחר בחירת עיצוב טבלה מוכן מראש, תוכל למלא את הגירסה המותאמת אישית שיצרת בשדות מותוך תיבת רשיימה. תיבת הדו-שיח התחלתי **אשף הטבלאות** כוללת קבוצת לחיצים להעברת שמות שדה בין תיבת הרשימה **שדות הדוגמה** (Sample Fields) לתיבת הרשימה **שדות בטבלה החדשה** (Fields in my new table). תרשים 3.4 מציג שלב ביצירת טבלת **חשבוניות**. שדות מוגדרים אחדים הועברו לתיבת הרשימה **שדות**

בטבלה החדשה. בשלב זה, תוכל ללחוץ שלוש פעמים על לחץ הבא (Next) כדי ליצור את הטבלה **חשבוניות** (Invoices). הגדירות ברירת המחדל יוצרות מפתח ראשי, משירות בצדקה טבעית את הטבלה החדשה לטבלה כלשהי במסד הנתונים, ומאפשרת להכנס נתונים. לאחר קבלת כל ברירות המחדל ולחיצה על לחץ סיום (Finish), הטבלה תיפתח בתצוגת **גילוון נתונים** (Datasheet) ותאפשר למלא אותה נתונים. בעת המעבר בין תיבות הדיו-שים של האשף תוכל לדرس כל ברירת המחדל לא רצויה.

אם תחלת זה עתה את צעדיך הראשונים בעבודה עם מסדי נתונים יחסיים, זיהוי הקישורים האוטומטי לטבלאות נוספות שימושי במיוחד. תרשימים 3.5 מציג את תיבת הדיו-שים השילשית של יצירת הטבלה **פרטי חשבוניות** (Invoice Details). הטבלה יכולה לכלול את השדה ID הפעול כمفחץ זר של הטבלה **חשבוניות** (Invoices). אם האשף מזזה את הטבלה **חשבוניות**, הוא מקשר אוטומטי בין שתי הטבלאות. תוכל לדرس את הקישור, או ליצור קשרים שונים מלבד שמציע האשף, אך אם אין לך ניסיון מוטב שתקבל את ברירות המחדל המוצעת.



תרשים 3.5: תיבת דיו-שים של אשף הטבלאות המציג זיהוי של קישור אוטומטי בין הטבלה **פרטי חשבוניות** והטבלה **חשבוניות** שנוצרה לפניה

בונה השדות

לאחר יצירת טבלה באמצעות אשף, יתכן שתרצה להוסיף לה שדה נוסף. המשימה עלולה להתגלות כקשה יותר מאשר היא נראהית מלבת הילה, במיחוד אם השדה צריך להיות מקשר אל שדות נוספים בטבלאות אחרות. הדבר נובע בכך שכלל יותר ליצור שדה חדש בעל סוג נתונים שונה מאשר שדה בטבלה אחרת. כמשמעותם לקשר בין שדות שונים בטבלאות שונות, שמותיהם דומים וסוגי נתונים שונים, הפעולה תיכשל מבלי שתקבל הסבר כלשהו. פתרון אפשרי לביעית ההוספה של שדות חדשים לטבלה הוא **בונה שדות** (Field Builder).

פותחים את **בונה שדות** על ידי לחיצה ימנית על שורה ריקה בתצוגת **עיצוב** (Design) של טבלה ובחירה **בבנייה** (Build) מתוך תפריט הקיצור. תיבת הדוחשיך בונה שדות כוללת שתי תיבות רשימה: אחת לבחירת סוג טבלה, כגון **חובניות** או **פרטי חשבוניות**, ותיבה נוספת לבחירת שדה מתוך סוג הטבלה שנבחר. בעת הארת סוגים טבלה שונים, משתנים הערכים בתיבת הרשימה **שדות דוגמה** (Sample Fields). חפש את השדה שברצונך להוסיף וליחס על אישור (OK) כדי להוסיפו לטבלה. פעולה זו מבטיחה שהשדות החדשים שהוספה ייכללו סוגים נתוניים תואמים לשדות שהוספו באמצעות אשף הטבלאות.

יצירת טבלה באופן ידני

ליצירת טבלה באופן ידני, לוחצים לחיצה כפולה על הסמל **יצירת טבלה בתצוגת עיצוב** (Create Table In Design View), כדי לפתח חלון **טבלה** (Table) ריק בתצוגת **עיצוב** (Design). מכאן ניתן להוסיף שדות, מפתח ראשי ואינדקס (מפתחות הראשיים הם אינדקסים בעלי הגדרות מאפיינים מיוחדים).

להוספה שדה, הקלד את שמו בעמודה הירקה **שם שדה** (Field Name) שבחלון תצוגת **עיצוב**. שמות שדות עומדים בכללי מתן השם הרגילים של Visual Basic for Applications (VBA). אורכם המקסימלי יכול להגיע ל-64 תוויים – אותיות, מספרים, רווחים ותוויים מיוחדים, למעט התווים נקודה, סימן קריאה, סוגרים מרובעים סימון הטעמה מסוג grave (''). כמו כן, אסור לפתח שם שדה בטו רוח או בטו בקרה (ערך ASCII בטוח 31-127). תווי רוח יכולים להופיע באמצע שמות שדה, אך בביטויים ובשאלות יש להציגם בין סוגרים רגילים.

סוגי נתונים

העמודה **סוג נתונים** מאפשרת לציין את סוג הנתונים של השדה. תיבת רשימה נפתחת מצעה 10 אפשרויות: **טקסט** (Text), **תזכיר** (Memo), **מספר** (Number), **תאריך/שעה** (Date/Time), **מטבע** (Currency), **מספר אוטומטי** (AutoNumber), **כן/לא** (Yes/No), **אובייקט OLE** (OLE Object), **yperlink** (Hyperlink) ואשף **בדיקה מידע** (Lookup Wizard) (סוגי נתונים ממשיים בדרך כלל לזרחי המידע שבסדה). שדה **טקסט** מכיל נתונים מסוג **טקסט**, שדה **מספר אוטומטי** מכיל נתונים מסוג **מספר אוטומטי** וכן סוג הנתונים. לדוגמה, סוג הנתונים **מספר** כולל שבעה סוגים משנה. באפשרות להגדיר אחד מהם על ידי בחירת **מספר** בתווך סוג הנתונים ולאחר מכן לבחור באפשרות הרצiosa מຕוך תיבת הרשימה הנפתחת **גודל שדה** (Field Size) בכרטיסיה **כללי** (General) בחלקו התיכון של החלון **טבלה**.

שדות מספור אוטומטי

שדות מסווג **מספר אוטומטי** (AutoNumber) משמשים לעיתים קרובות בתוך מפתח ראי של טבלה. Access מציב בעצמו ערך חדש בשדה בעת הוספה רשומה לטבלה. שדה זה אינו בר-עדכו בצורה ידנית, ולכן ערכיו אידיאליים לשימוש ייחודי של שורה בטבלה.

Access מגדר בעצמו את הערך של שדה מסווג **מספר אוטומטי**. כדי לגרום לשדה כזו לגדול בהפרש קבוע בצורה סדרתית,בחר באפשרות **תוספת קבועה** (Increment) (שהיא ערך בירית המוחלט) מתוך תיבת הרשימה **ערבים חדשים** (New Values) בכרטיסיה **כללי** בחלקו התחתון של החלון **טבלה**. כדי לצוין שדה מסווג **מספר אוטומטי** צריך לקבל ערך אקראי, בחר באפשרות **אקראי** (Random) מתוך תיבת הרשימה **ערבים חדשים**.

באפשרות להשתמש בכרטיסיות **כללי ובדיקה מידע** (Lookup) כדי לבחור אפשרות המשפיעות על השדה, כגון הנחיה למשתמש להכניס ערך בשדה, או ציון שהשדה מכיל ערך בירית מוחלט. מאפייני שדה שהוגדרו בرمת הטבלה חלים דרכה על טפסים ודוחות. מאפייני שדה טבלה יכולים גם לפחות את הקוד שנכתב עבור טפסים ודוחות. המשמעות הנוספת של תחזוקת מאפייני שדה בرمת הטבלה היא שהמאפיינים משתנים במקומות אחד ולא בכל טופס ודוח שמשתמשים בשדה.

Access 2000 הוא הגירסה הראשונה של Access שמאפשרת שליטה באמצעות קוד על הערך ההתחלתי וגודל הצעד של סוג השדה **מספר אוטומטי** (פרק 2 עוסקת בהיבטים הבסיסיים של יכולת זו). באפשרות להשתמש במילוי המפתח ALTER TABLE ו- ALTER COLUMN ב-*SQL* כדי לעדכן את ערכי ההתחלת והצעד של שדה **מספר אוטומטי**. כזכור פרק 2, שדה מספור אוטומטי של *SQL* כולל את סוג הנתונים **IDENTITY**. המאפיינים **start** ו-**step** של סוג נתוניים זה מאפשרים לשנות באמצעות קוד את ערך המספר האוטומטי הבא ואת גודל הצעד של הערכים הבאים.

שדות טקסט

שדות טקסט משמשים לשימירת ערכי מחروفות באורך של עד 255 תווים. סוג הנתוניים **טקסט** (Text) יכול לאחסן פרטיים כגון נתונים אנשי קשר וערכים מספריים שאינם מיועדים לחישוב (לדוגמה, מספר זהות, מספר טלפון ומספר קטלוגי). באפשרות להשתמש בשדות **טקסט** בטבלה כדי ליצור ערכי מחروفות חשובים. ניתן לאינדקס מפתחות ראשיים לצורך מיון ואחיזור מהירם המבוססים על שם משפחה, או על סוג שדה אחר מסווג **טקסט**.

שדות בדיקת מידע

שדה **בדיקה מידע** (Lookup) מציג ערך בעל משמעויות המקביל לערך שמאחסן בשדה (לדוגמה, מסד נתונים עשוי להשתמש במספרים ייחודיים כדי לייצג מוצר. שדה **בדיקה מידע** יכול להציג את שם המוצר במקומות המספר הקטלובי שמייצג אותו). סוג שדה זה (המוכר גם כ**עמודת מפתח**, Key Column) מאפשר לאחסן ערך אינדקס

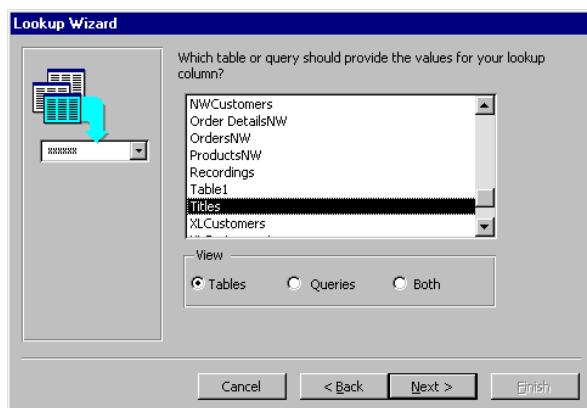
בטבלה, אך מציג ערך טקסט בעל משמעות כשהטבלה מוצגת בתצוגת **גילוון נתונים** (Datasheet). שדה **בדיקות מידע** יכול להיקשר לערכים בטבלה או בשאלתה אחרת, או לרשימת ערכים מותאמת אישית שהטבלה מתחזקת עבורה שדה בדיקת המידע. לאחר הוספת שדה בדיקת מידע המפנה לערכים הנמצאים בטבלה אחרת, לא ניתן למחוק את השדה בטרם מוחקים את קשרי הגומלין לשדה האחר, באמצעות החלון **קשרי גומליין** (Relationships). בעת מיציקת קשרי הגומליון, Access יזכיר לך שהשדה מהוועדיין חלק הקשרים הגומליין ובקשת אישור לפעולות המהיקה. בשלב זה בחר את האפשרות **כן** (Yes) כדי להסיר את שדה בדיקת המידע מטבלת.

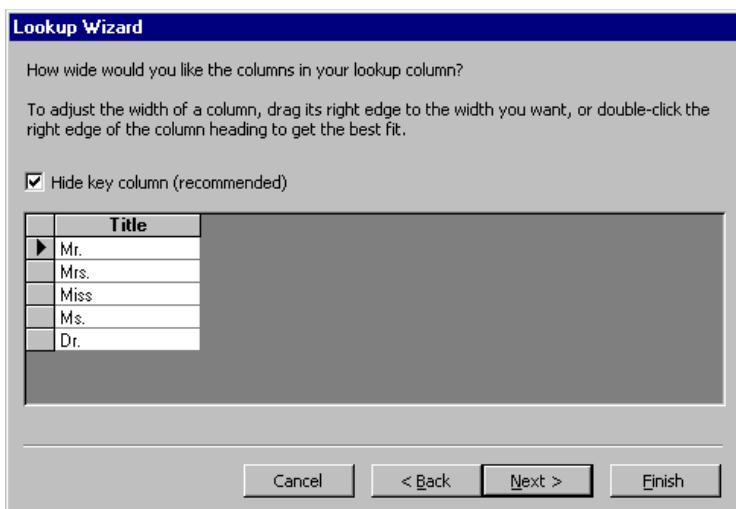
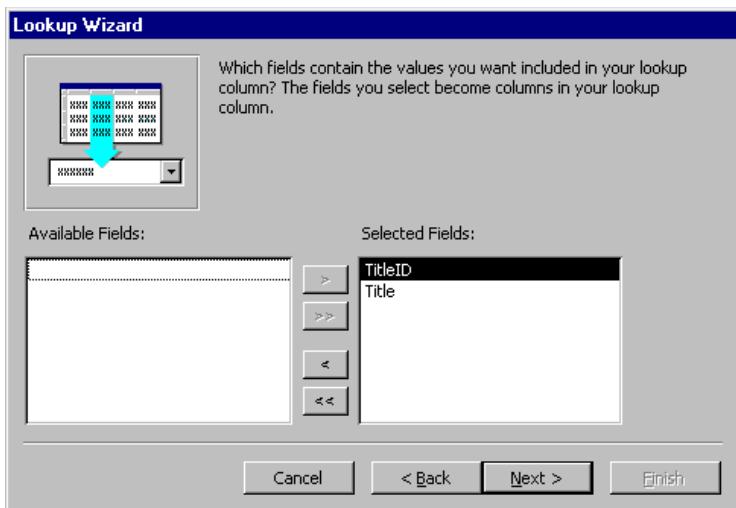
שדה **בדיקות מידע** יוצרים באמצעות **ASF בדיקת מידע** (Lookup Wizard). בחר באפשרות מתיבת הרשימה הנפתחת **סוג נתונים** (Data Type) כדי לקבל טבלה בתצוגת **עיצוב** (Design). תיבת הדו-שיה ההתחלה של האשף שואלת אם ערכיו בבדיקה המידע מכוונים בטבלה אחרת או ברשימה של ערכים מותאמת אישית. בדרך כלל תלחץ על **הבא** (Next) כדי להשתמש בערכי בדיקת המידע מטבלה אחרת.

אם ברצונך להשתמש בערכי בדיקת מידע מטבלה אחרת, באפשרות**לבחור מtoo** תיבת הדו-שיה השנייה בטבלה, או בשאלתה שמקיפים את הערכים המוצגים באמצעות שדה בדיקת המידע. בתיבת הדו-שיה השלישי, בחר בשדה או בשדות שמקיפים את הערכים המוצגים באמצעות שדה בדיקת המידע שיצרת (בדרך כלל תבחר שדה טקסט שימושי כמפתח ראשי).

תיבת הדו-שיה הרביעית מאפשרת לקבוע את רוחב העמודה (או העמודות) להציגת רשימת בדיקת המידע, בתצוגת **גילוון נתונים** (Datasheet). רשימת בדיקת המידע יכולה להציג את ערך שדה בדיקת המידע ופריט אחד מהרשימה לכל שדה שבחרת בטבלה הקשורה (בדרך כלל תרצה להסתיר את עמודות המפתח). תיבת הדו-שיה الأخيرة מאפשרת להגדיר כיצד עבור שדה בדיקת המידע.

תרשים 3.6 מציג את תיבות הדו-שיה השנייה, השלישייה והרביעית שבאפשרותן יצרו את השדה TitleID של הטבלה Persons בקובץ מסד הנתונים של פרק 3 המופיע בתקליטור המצורף לספר.

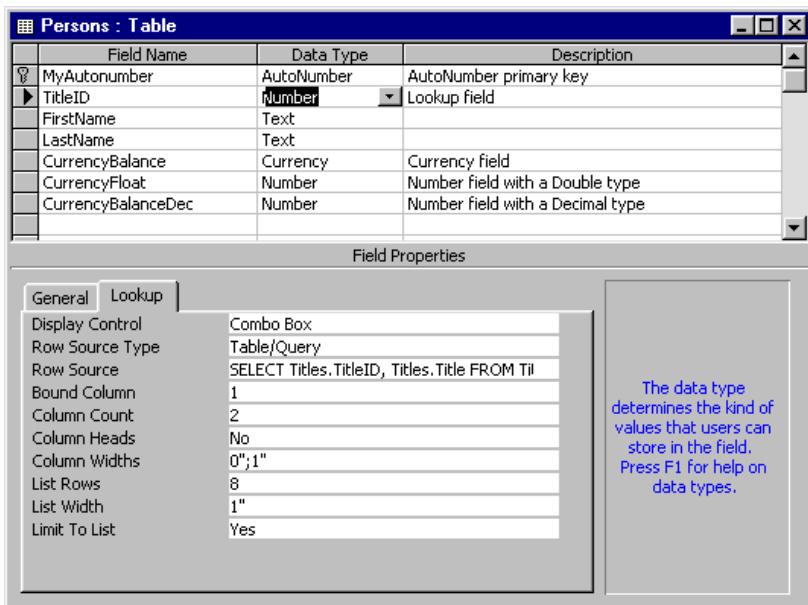




תרשים 3.6: תיבות הדו-שיך השנייה, השלישית והרביעית של אשף בדיקת מידע (Lookup Wizard)

הטבלה Persons בתצוגת **עיצוב** מציגה את השדה TitleID בתווך שדה מספר, ולא בתווך שדה **בדיקה מידע** או שדה טקסט (תרשים 3.7). סוג הנתונים של שדה **בדיקה מידע** תלוי בסוג הנתונים של עמודות המפתח. גם אם תסתר את עמודת המפתח, סוג הנתונים שלא יקבע את סוג הנתונים של שדה בדיקת המידע.

הכרטistica **בדיקה מידע** בתצוגת **עיצוב** של הטבלה מכילה את משפט SQL המשמש את שדה בדיקת המידע. Access מכין אותו בעצמו בעת שאתה בוחר את האפשרויות הרצויות באמצעות אשף בדיקת המידע.



תרשים 3.7: הtáבלה Persons בתצוגת **עיצוב** ובה שדה **בדיקה מידע** שיצירתו הוצגה בתרשים 3.6

שדות מספר

שדות **מספר** (Number) שונים משדות **טקסט** (Text) מכיוון שהם יכולים לקבל מיגון של סוגים משנה, החל בבית יחיד (הסוג **בית** - Byte) וכלה ב-16 בתים (סוג המשנה **קוד העתק משוכפל** - Replication ID). סוגים משנה נוספים בטוחה הסוגים הם **שלם** (Integer), **מספר שלם אורך ארוך** (Long Integer), **יחיד** (Single), **כפול** (Double) ו**עשרוני** (Decimal). למעט סוגים המשנה **בית** ו**קוד העתק משוכפל**, כל השאר מתוארים בפרק 2, בסעיף "סוגי נתונים".

סוג המשנה **בית** דומה למשתנה מסוג Boolean. שני הסוגים יכולים לאחסן ערכיים בוליאניים, אך הסוג **בית** דורש זיכרון בגודל בית אחד, בעוד שהסוג Boolean דורש שני בתים. סוג הנתונים **קוד העתק משוכפל** אינו זמין בתור סוג נתונים של משתנה. ייעודו העיקרי הוא לשכפול, אך הוא משמש בתור מזהה ייחודי. אורכו ושיטת יצירתו הופכים אותו לבטוח יותר לשמירה על ייחוזיות לעמודת שדה מספור אוטומטי.

סוג המשנה **עשרוני** מאפשר את ביטולן של שגיאות עיגול ובה בעת מאפשר לאחסן מספרים גדולים באמצעות המאפיינים **דיוק** (Precision) ו**סרגל** (Scale). מאפיינים אלה שלוטים במספר הספרות ממשני צידי הנקודה העשרונית. ערך המאפיין **דיוק** המייצג את מספר הספרות הכולל שנitin לאחסן בשדה, נع בין 1 ל-28. סרגל, המציין את מספר הספרות מימין לנקודה העשרונית שנitin לאחסן בשדה, נע בין 0 לערך המאפיין **דיוק**. הנקודות למאפיין **סרגל**, סוג המשנה **עשרוני** יכול להכיל ספרות אחרי הנקודה מסוגי משנה אחרים של **מספר** מבלי לגרום לשגיאות עיגול.

תרשים 3.8 מציג את הטבלה Persons בתצוגת **גילין נתוניים**. השדה CurrencyBalance המשמש בסוג הנתונים **מספר**, CurrencyFloat משמש בסוג הנתונים **מספר** ובסוג המשנה **כפול** ו-**CurrencyBalanceDec** משמש בסוג הנתונים **מספר** עם סוג המשנה **עשרות**. המאפיין **סרגל** של השדה CurrencyBalanceDec מקבל את הערך 6, שפירושו שהשדה יכול לאחסן שיש ספרות מימין לנקודה העשרונית. ערך זה גובה מספרות הספרות שיכول סוג הנתונים **מספר** לייצג בצורה מדויקת – הוא מוגבל לארבע ספרות מימין לנקודה. סוג המשנה **כפול** יכול לייצג מספר בעל עד שיש ספרות עשרוניות, אך הוא אינו מבצע את משימתו בדיקוק של מספר שלם (Integer). השורה הראשונה של הטבלה Persons מציגה את הערך 1.0001 בתבניות הנתונים **מספר**, **כפול** ו-**עשרות**. השורה השנייה מבטא את 1.00001 בתשלוש התבניות הללו. שים לב כי התבנית **מספר** בתצוגת **גילין נתוניים** מציגה את הערך 1.00001 בתור 1, מכיוון שהיא מוגבלת לארבעה מקומות עשרוניים. התבניות **כפול** ו-**עשרות** מציגות ערכים זהה.

Persons : Table							
	MyIndex	TitleID	First Name	Last Name	CurrencyBalance	CurrencyFloat	CurrencyBalanceDec
▶	1	Miss	Shelly	Leghorne	\$1.0001	1.0001	1.0001
◀	2	Mr.	Glen	Hill	\$1.0000	1.00001	1.00001
* [toNumber]							
Record: [First] [Previous] [Next] [Last] 1 [First] [Last] * of 2							

תרשים 3.8: הטבלה Persons בתצוגת **גילין נתוניים**, כשהיא מכילה ערכים מספריים בתבניות הנתונים **מספר**, **כפול** ו-**עשרות**.

הSIGERA DecimalArithmetic שלפניך מציגה הבדלים נספפים בין תבניות הנתונים שתוארו. SIGERA מחשבת את הפרשיים בין המספרים המאוחסנים בשדות המתאים, בין ערך 1. SIGERA פותחת ערך שדומה רשותה המבוססת על הטבלה Persons, מפחיתה 1 מכל אחד משש שדות המספר שבשורה הראשונה ומדפיסה את התוצאות בחלון **Immediate** (מיידי). לאחר מכן היא עוברת לשורה השנייה ומבצעת את התהליך פעם נוספת.

```
Sub DecimalArithmetic()

Dim cnn1 As New ADODB.Connection
Dim rst1 As Recordset
Dim intCounter As Long, sumD As Variant
Dim sumC As Variant, sumF As Variant

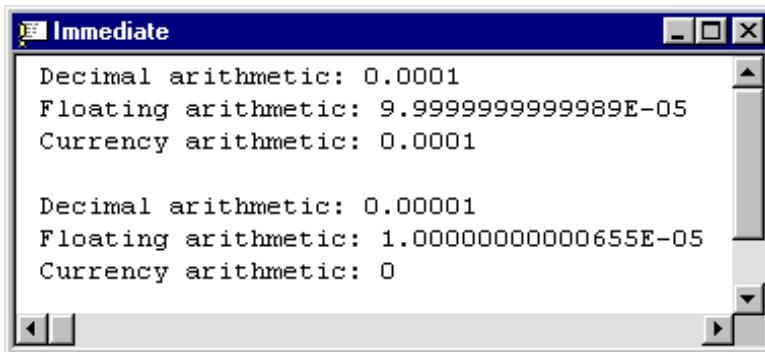
' Open and set recordset.
Set rst1 = New ADODB.Recordset
rst1.ActiveConnection = CurrentProject.Connection
rst1.CursorType = adOpenKeyset
rst1.LockType = adLockOptimistic
rst1.Open "Persons", , , adCmdTable
```

```

Debug.Print "Decimal arithmetic: " & rst1.Fields(6) - 1
Debug.Print "Floating arithmetic: " & rst1.Fields(5) - 1
Debug.Print "Currency arithmetic: " & rst1.Fields(4) - 1
rst1.MoveNext
Debug.Print
Debug.Print "Decimal arithmetic: " & rst1.Fields(6) - 1
Debug.Print "Floating arithmetic: " & rst1.Fields(5) - 1
Debug.Print "Currency arithmetic: " & rst1.Fields(4) - 1
End Sub

```

תרשים 3.9 מציג את תוצאות הפעלת השירה. התבניות **מطبع** ו**עשרוני** נותנות תוצאה זהה, 0.0001. התבנית **כפול** אינה מסוגלת להציג דיוק כזה; היא נותנת 0.00009999999999989. למטרות רבות תוצאות אלו של התבניות **מطبع** ו**עשרוני** זהות. לעומת זאת, כMISSISSIPPI בתוכה מדוקית, אין DI בCLK. כמשמעותם פועלה זו על השורה השנייה שמכילה את הערך 1.00001, התבנית **עשרוני** היא היחידה שתפיק תוצאה מדוקית. חישובים אלה ממחישים את תפקידו המוחדר של סוג המשנה של סוג הנתונים **עשרוני**. עליך להשתמש בו כסדרוש דיקוק בرمה שלא ניתן להשיג בתבניות **מطبع** וכ**כפול**.



תרשים 3.9: פלט השירה DecimalArithmetic

השדות: **תזכיר**, **אובייקט OLE**, **תאריך/שעה**, **כן/לא**

סוגי נתונים נוספים כוללים את הסוג **תזכיר** (Memo) שמאחסן מחזוזות טקסט ארוכות מאוד, אפילו יותר מ-255 תווים (המגבלה של סוג הנתונים **טקסט**). סוג הנתונים **תזכיר** יכול להגיע עד 64 ק"ב. באפשרות לגשת אליו ולכתוב את תוכנו בקטעים של 64 ק"ב באמצעות השיטות AppendChunk ו-GetChunk.

4 Jet תומך באינדוקס של 255 התווים הראשונים של שדה **תזכיר**. הדבר שימושי במיוחד לסוגי הנתונים **היפר- קישור** (Hyperlink) התלויים בסוג הנתונים **תזכיר**.

אובייקט OLE (OLE Object) הוא סוג נתונים גדול נוסף. הוא מטפל באובייקטים בתבנית ביןארית, כגון חוברת עבודה של Microsoft Excel או מסמך של Word.

סוגי הנתונים **תאריך/שעה** (Date/Time) יכולים לייצג גם תאריכים וגם שעות. ערכי תאריך מאוחסנים משמאלי לנוכח העשויות; ערכי שעה מאוחסנים לימינה (עיין בדוגמאות הקוד של פרק 2 המטפלות בערכי שדה **תאריך/שעה**).

סוג הנתונים **כן/לא** (Yes/No) הוא הקטן ביותר. הוא מופיע באחד משני המLAYERS: כן/לא,אמת/שקר (True/False) או **מופעל/מכובה** (On/Off). סוג זה תופס זיכרון בגודל בית אחד בלבד.

אימות נתונים

לעיצוב טבלאות עבור מסד נתונים חסין לתקלות, עליך להבטיח כי רק נתונים חוקיים יוכלו להיכנס למסד הנתונים. Access כולל כמה רכיבים שישיעו לך להשיג זאת.

המאפיינים נדרש ואפשר אורך אפס

לעתים רשותה אינה חוקית, אלא אם היא כוללת ערך שדה מסוים, כגון שדה מפתח ראשי או שדה מפתח זר. קביעת המאפיין **נדרש** (Required) של מפתח זר בתור **כן** (Yes) מבטיח כי המשתמשים לא יוכלו להיכנס לרשומה בצד ה"רבבים" של קשר הגומלין מבלי שתהיה התאמה לרשותה אחת לפחות בצד הנגדי של קשר הגומלין.

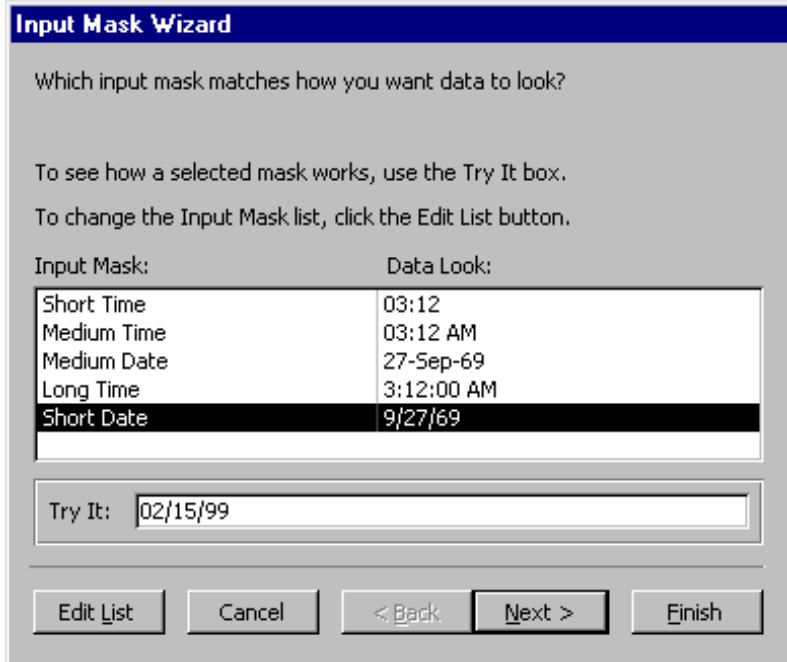
בשנותנים למאפיין **נדרש** של שדה את הערך **כן**, Access אינו מקבל רשומה שעריך השדה שללה null. הגדרת המאפיין **אפשר אורך אפס** (Allow Zero Length) מאפשרת (או לא) להציב בשדה מחרוזת באורך אפס (""). Access מבידיל בין שדה שלא קיבל ערך כלשהו (Null) לבין שדה שאינו מכיל ערך (מחרוזות באורך אפס בשדות טקסט).

המאפיין מסכת קלט

מסכת קלט (Input Mask) היא תבנית שמנחה את השימוש בנושא סוג ותבנית הנתונים הדורישים. בדומה למאפייני שדה אחרים, מאפיין זה חל על טפסים ודוחות באמצעות השימוש בשדה.

באפשרות להשתמש במסכות קלט תקניות, או ליצור בעצמך מסכות קלט מותאמות אישית. תרשימים 3.10 מציג את **אשר מסכות הקלט** המציג מבחן מסכות קלט של השדה **תאריך/שעה** (Date/Time). באפשרות להכניס ערכים בתיבת הטקסט **ניסיוני** (Try It) כדי לראות כיצד תפקד המסכה במצבים של נתונים אמת.

האשר קובע את קלט השדה ואת תצוגת נתונים השדה, אלא אם תציין גם את המאפיין **תבנית-עיצוב** (Format) של שדה. כשיישום מגדר את המאפיין **תבנית-עיצוב** של שדה, המאפיין קובע את תצוגת נתונים השדה. המאפיין **תבנית-עיצוב** אינו משפיע על התצוגה של ערך, אלא רק לאחר שהיישום שומר את הערך במסד הנתונים.



תרשים 3.10: אשף מסכות הקלט מציג מבחר מסכות אפשריות עבור שדות מהסוג תאריך/שעה

המאפיינים חוק אימות ותקスト אימות

מאפיינים אלה נועדו להבטיח כי המשתמש מקליד מידע חוקי. המאפיין **חוק אימות** (Validation Rule) נועד להגדיר קритריוני חוקיות לערבים בשדה (אם שדה אמרור לקלוט נתונים גדולים מ-1, למשל, עליך להגדיר את חוק אימות בטור `<1>`). באפשרותך להשתמש במאפיין **תקסט אימות** (Validation Text) כדי להגדיר הودעות תגובה שתוצגנה למשתמשים בעת שניסו להקליד ערך מחוץ לטווח הערכים החוקיים שנקבעו במאפיין חוק אימות.

באפשרותך לציין גם כללי אימות עבור הטבלה כיחידה אחת. לשם כך, פתח את הטבלה בתצוגת **עיצוב ולחץ על לחצן מאפיינים** (Properties) בסדרת הכלים **עיצוב טבלה** (Table Design). לאחר מכן השתמש במאפיינים **חוק אימות ותקסט אימות** כדי להגדיר כלל אימות עבור הטבלה בשלהמתה והודעה שתוצג בתגובה להפרת הכלל. כלל האימות של טבלה מאפשר להגדיר קритריונים שהיקף ההחלה שלהם משתרע מעבר לשדה ייחיד.

אם דרישים לך מערכי כללים רבים, תוכל לכלול אותם בбиיטוי כלל האימות באמצעות פסוקיות **And** (וגם) (אם שדה אמרור לקלוט רק מספרים גדולים מ-1 וקטנים מ-10, באפשרותך להגדיר את חוק אימות בטור `<1> And <10>`). לחץ על לחצן **בנייה** (Build) הסמוך לתיבת המאפיין **חוק אימות** כדי לפתוח את **מבנה בייטויים**

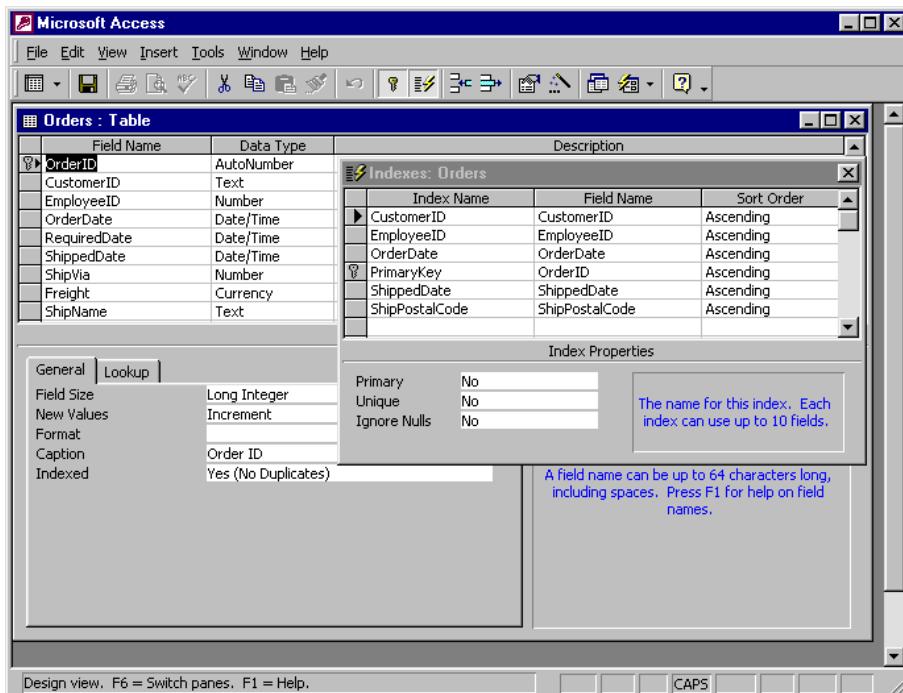
(Expression Builder). באפשרות לחשוף בפונקציות המוכללות של Access לאימוט שודות טבלה, אך פונקציות מותאמות אישית אסורות לשימוש בביטויי אימוט של טבלה או של שדה. **בונה ביטויים** מתאים ליצור אימוט טבלה כולל בדיקת כמי שהו מתאים לביטויי אימוט של שדה.

יצירת אינדקסים

אינדקסיםקובעים את אופן הפקוד של טבלאות ואת היחסים ביניהן. אינדקסים מייצגים פעולות מיון, חישוב ובחירה ברמת השדה. מרבית השימוש באינדקסים פוגע ביצועם יישומיים בעט קליטת נתונים (מכיוון שאינדקסים דורשים הצבת ערכים בכל רשומה חדשה), יתרוניותו עולה על חסרונותיו.

אינדקסים תומכים גם בשלמות הקשרים. לפחות אחד מהשדות המקשרים בקשר גומלין חייב להיות מפתח ראשי או אינדקס ייחודי.

תרשים 3.11 מציג את הטבלה Orders בתצוגת **עיצוב**, כשהחלון הראשי וחלון האינדקסים פתוחים. החלון **אינדקסים** (Indexes) שבסרגל הכלים **עיצוב טבלה** (Table) (אפשר להציג או להסתיר את החלון האינדקסים) (Design



תרשים 3.11: החלון הראשי והחלון **אינדקסים** של הטבלה Orders בתצוגת **עיצוב** החלון **אינדקסים** שבתרשים 3.11 כולל גם שורה ובה מפתח. בדיקת כמו בחalon הראשי, שורה זו מסמנת את המפתח הראשי. בחalon האינדקסים נבחר האינדקס CustomerID.

אינדקס זה תלוי בשדה CustomerID שהוא מפתח זר בטבלה Orders. האינדקס CustomerID מקשר בין הטבלאות Customers ו-Orders בקשר גומלין מסוג יחיד לרבים. הוא אינו המפתח הראשי של הטבלה Orders, ולכן ניתן לשכפל אותו בין רשומות. שים לב גם כי כמה אינדקסים, כגון OrderDate ו-ShipPostalCode, אינם מפותחות של טבלאות אחרות. אחת המטרות לציררת אינדקס היא האצת הפעולות המתבצעות נתונים הטבלה, כגון בחירה לפי תאריך, או מיון בהתאם לקוד דיוור.

הערה:



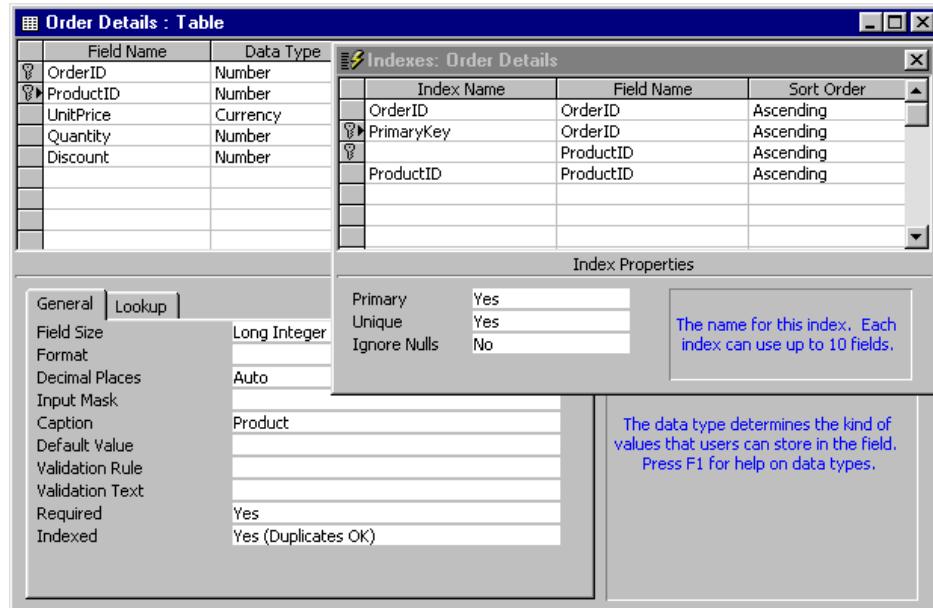
את הדרכים להבטיח את תקינות הגדרות של מאפיין מפתח זר היא להעתיק את השדה אל הלוח (Clipboard) של Windows בצד "יחיד" של קשר הגומلين. פועלה זו מאפשרת לבדוק את השדה בתצוגת **עיצוב** של הטבלה בצד ה"רבים" של קשר הגומلين. כך ניתן להימנע מה הצורךagog לאיתר מאפייני המפתח הזר.

אינדקס יוצרים באמצעות הקלדת שם בעמודה **שם אינדקס** (Index Name) בחלון **אינדקסים**. לאחר מכן בוחרים שדה עבור האינדקס מתוך תיבת הרשימה הנפתחת, ובוחרים את סדר המיון של השדה: עולה או יורד. אם לאינדקס שייך שדה נוסף, בוחרים את שמו ואת סדר המיון שבוrah הבאיה מייד לאחר מכן. באפשרותן להוסיף שדות נוספות לאינדקס באופן זהה. כל שדה שתוסיף לאינדקס חייב להופיע מייד מתחת לשדה הקודם. רק השדה הראשון של האינדקס צריך להיות ערך עבור העמודה **שם אינדקס**. כל השורות הבאות לאחר מכן שומות שם האינדקס שלוhn ריקה, שייכות לאותו אינדקס. כל ערך בעמודה **שם אינדקס** פותח אינדקס חדש.

באפשרותן להגדיר את שולשת מאפייני האינדקס בנפרד. יחד עם זאת, אם תגדיר את **ראשי** (Primary) בתווך **כן** (Yes), המאפיינים ייחודי (Unique) והתעלם מ-**Nulls** (Ignore Nulls) יקבלו את הערכיהם **כן** ולא, בהתאם. מפתח ראשי חייב להיות ייחודי לכל רשומה. לא ניתן ליצור מפתח ראשי בשדה או בשדות שמכילים ערכי Null. לטבלה נתונה יכול להיות מפתח ראשי אחד בלבד. טבלה יכולה להכיל הרבה רבים בעלי אינדקסים ייחודיים שמתעלמים מערכי Null. אינדקס כלשהו, למעט מפתח ראשי, יכול להתעלם מערכי Null. הדבר מאפשר להגדיר אינדקס שմבוסס על ערכי שדה שונים מ-Null. הבחירה להתעלם מערכי Null יכולה לצמצם את דרישות האחסון של אינדקס.

תרשים 3.12 מציג את החלון הראשי וחולון האינדקסים של הטבלה Order Details בתצוגת **עיצוב**. שים לב שהחלון הראשי נבחר השדה ProductID. הכיתוב שלו הוא Product, והוא אחד משני השדות המגדירים את המפתח הראשי. השדה השני הוא OrderID. שני השדות הם שדות **מספר** (Number) שסוג הנתונים שלהם הוא **מספר שלם ארוך** (Long Integer); אף אחד מהם אינו שדה **מספר אוטומטי** (AutoNumber), אך לעומת זאת שניהם משמשים בתווך מפתחות זרים המבוססים על שדות מספר אוטומטי בטבלאות Orders ו-Products. שניהם ייחודיים באופן ייחודי כל שורה בטבלה Orders. זהה התנהגות אופיינית לtablאות תיבת הסתעפויות.

חלון האינדקסים בתרשים 3.12 מוסיף ומצהה את המפתח הראשי. שים לב כי הטבלה Order Details מקיימת גם אינדקסים נפרדים שmbוססים על השדות OrderID ו-ProductID. אינדקסים אלה מקיימים קישורים עם הטבלאות Orders ו-Products. שלמות הקשר בין Products ו-Order Details מבוססת על אינדקסים אלה.

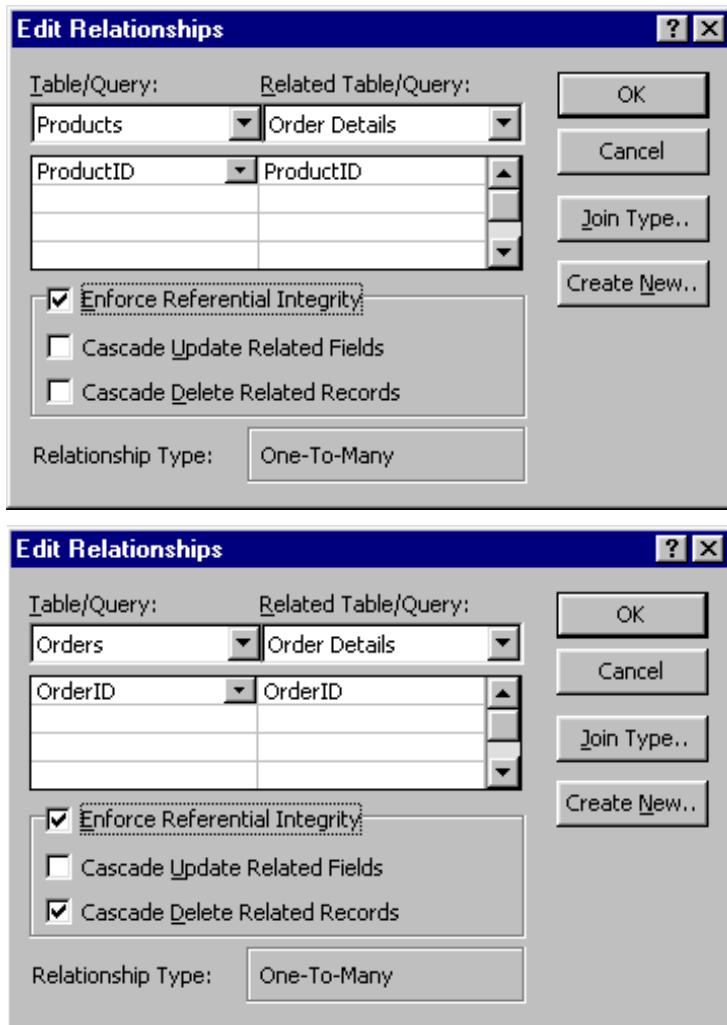


תרשים 3.12: החלון הראשי והחלון אינדקסים של הטבלה Order Details של מסד הנתונים Northwind

באפשרותך ליצור ולנהל ידנית יחסיות שלמות הקשורות בין טבלאות מתוך החלון **קשרי גומליין** (Relationships).

1. הוסף את הטבלאות לחלון, במידה שלא עשית זאת עד כה. לחץ לחיצה ימנית לחלון, בחר באפשרות **הציג טבלה** (Show Table) והוסף לחלון טבלאות לפי הצורך.
2. צרף את הטבלאות באמצעות קישור השדות המשותפים של הטבלאות. עשה זאת על ידי גירurat שדה אחד או יותר מטבלה לטבלה, מצד ה"יחיד" לצד ה"רביבים" של קשר הגומליון.
3. בחר את שורת הצירוף.
4. לחץ לחיצה ימנית על הקו הקשר ובחר את **עריכת קשר גומליון** (Edit Relationships).

.5 סמן את תיבת הסימון **אכוף שלמות הקשרים בין הטבלאות** (Enforce Referential Integrity). בצע את שינויי העיצוב הדורשים בקשר הגומלין, כגון בחירת תיבות שלם לעדכנים ומחיקות בהתאם להיררכיות הקשרים. באפשרות גם ללחוץ על **סוג צירוף** (Join Types) כדי לבחור אחד משני סוגי הצירוף נוסף על הצירוף הרגיל הכלול שורות מושתפי הטבלאות רק כשותות הצירוף בשתי הטבלאות זהים. שתי האפשרויות האחרות כוללות את כל הרשומות מצד ה"יחיד", או כל הרשומות מצד ה"רבים" של קשר הגומלין.



תרשים 3.13: שתי תיבות הדו-שייך **עריכת קשר גומלין** המתארות את קשרי הגומלין של שלמות הקשרים בין הטבלאות Products ו- Order Details (העליונה) ובין הטבלאות Orders ו- Orders Details (התחתונה) של מסד הנתונים Northwind.

תרשים 3.13 מציג את תיבوت הדו-שיח **ערכות קשר גומליין** שמנדרירות שלמות קשרים בין הטבלה Order Details לבין הטבלה Orders והטבלה Products. שים לב ש-Products מפרש את שתי מערכות קשרי הגומליין בטור יחיד לרבים. קשר הגומליין בין הטבלאות Orders ו- Order Details מגדר מחייב מחייב בהתאם להיררכית הקשרים. הדבר מאפשר למחוק הזמן ואת כל שורות הפריטים שלו בפעולה אחת. קשר הגומליין בין הטבלאות הישום לבצע שתי שאלות מחיקה, אחת לכל טבלה. קשר הגומליין בין הטבלאות Products ו- Order Details אינו כולל מחייב בהתאם להיררכית הקשרים. במקרה זה, לא תבוצע מחיקה אוטומטית של שורות פריט בזמןנות כשיתעורר צורך להסרה מוצר מהטבלה Products. המודול העסקי עשוי לחייב מאמץ מיוחד לרכישת המוצר עבור הזמןנות הפתוחות שטרם סופקו.

יצירה וניהול של טבלאות באמצעות קוד

לפעמים – ואולי אפילו ברוב הפעמים – תעדר ליצור טבלאות באופן ידני. לעומת זאת, לעיתים תמצא כי יותר יוצר טבלה באופן דינמי, כמו במקרים של שימור תוצאות ביןיהם לשימוש חוזר בפעולה הבאה. Microsoft Access 2000 מאפשר לעשות זאת באמצעות טכניקות של ADO ו גם בטכניקות מקובלות יותר של DAO (Data Access Objects). בסעיף זה נתאר יצירה וניהול של טבלאות באמצעות טכניקות DAO. גישה זו תאפשר לך לנצל את הייתרונות של חידושי הטכנולוגיה העתידית של Microsoft לגישה נתונים.

לאחר שיוצרים טבלה באמצעות קוד, לעיתים קרובות יש צורך למלא אותה בתונינים. אף כאשר יוצרים טבלה בדרך אחרת, מילוייה בתונינים באמצעות קוד הוא רעיון מצודד. ניתן לנצל מקורות נתונים רבים כדי למלא טבלה של Access בתונינים. בסעיף זה נסקר בהרחבה את השימוש בטבלה נוספת של Access, טבלה של Excel (בutor דוגמה למקור נתונים מסוג ISAM), ומקור נתונים ODBC. נלמד גם כיצד למלא טבלה של Access בתונינים באמצעות ספקי מסד נתונים OLE DB (OLE DB) והאובייקט הפנימי DoCmd של Access, המאפשר ליצור קישור אל מקורות נתונים מרוחקים בצורה קלה ו פשוטה. ברגע לשימוש ב- OLE DB, DoCmd הופך את מקור הנתונים הקשור זמני מתוך החלון **מסד נתונים** (Database).

יצירת טבלה

כדי להוסיף טבלה באופן דינמי, כל שדרוש לך הוא ADO – הספרייה ADOX, ליתר דיוק. בדיקות כפוי שיש לתת שם לטבלה, בין אם יוצרים אותה באופן ידני או דינמי, יש גם להוסיף לה עמודות ולצרף אותה לאוסף Tables.

השיגרה MakeLocalTable שלפניך, יוצרת טבלה באופן דינמי. השיגרה פותחת בהכרזה על האובייקט Catalog והאובייקט Table. Catalog הוא מכולה של סכמת מסד הנתונים, כולל האוסף Tables. לאחר מכן, השיגרה יוצרת מופיע של האובייקטים Table-Catalog ו-FamilyMembers2. השיגרה נותרת לטבלה החדשה את השם FamilyMembers2 ומוסיפה

לה ארבע עמודות באמצעות ארבע שיטות Append מקוננות בתוך משפט With...End. כל אחת מהשיטות Append כוללת שם עמודה, קבוע שמנדרט סוג נתוניים וארגוני אוורך, במידת הצורך. כל אחד משלושה שדות הטקסט (אליה שנוצרו על ידי הגדרת סוג הנתונים) כולל פרטיהם על גודל השדה. השגירה מסתיימת בצירוף הטבלה המושלמת לאוסף Tables של הקטלוג ובສיגרת האובייקט Catalog.

```

Sub MakeLocalTable()
Dim cat1 As ADOX.Catalog
Dim tbl1 As ADOX.Table

'Reference objects for table
Set cat1 = New Catalog
cat1.ActiveConnection = CurrentProject.Connection
Set tbl1 = New Table

' Name table and append columns.
With tbl1
    .Name = "FamilyMembers2"
    .Columns.Append "FamID", adInteger
    .Columns.Append "Fname", adVarWChar, 20
    .Columns.Append "Lname", adVarWChar, 25
    .Columns.Append "Relation", adVarWChar, 30
End With

' Append new table to Tables collection
'and free catalog resource.
cat1.Tables.Append tbl1
Set cat1 = Nothing
End Sub

```

הטבלה שלפניך מציגה את סוג הנתוניים ואת סוגי המשנה המיוצגים באמצעות קבועי המחלקה DataTypeEnum של הספרייה ADOX.

קבועי סוג עמודה וסוגי הנתוניים הידניים השקולים להם

קבוע	ערך	סוג נתונים בקלט יידי
adBoolean	11	כן/לא (Yes/No)
adCurrency	6	מטבע (Currency)
adDate	7	תאריך/שעה (Date/Time)
adDecimal	14	מספר – עשרוני (Decimal)
adDouble	5	מספר – כפול (Double)
adGuid	72	מספר – קוד העתק משוכפל (Replication ID)

קבוע	ערך	סוג נתונים בקלט ידני
adInteger	3	מספר אוטומטי (AutoNumber)
adInteger	3	מספר – מספר שלם ארוך (Long Integer)
adLongVarBinary	205	אובייקט OLE (OLE Object)
adLongVarWChar	203	היפר- קישור (Hyperlink)
adLongVarWChar	203	תזכיר (Memo)
adSingle	4	מספר – יחיד (Single)
adSmallInt	2	מספר – שלם (Integer)
adUnSignedTinyInt	17	מספר – בית (Byte)
adWChar	130	טקסט (Text)

הטבלה מציגה כמה נקודות מעניינות. ראשית, סוג הנתונים **היפר- קישור** שקל לссוג הנתונים **תזכיר** מנקודת מבט של תכונות. שנית, לא קיים סוג נתונים נפרד לשדות מספור אוטומטי בספריה ADOX. אם הקוד בודק את המאפיין **Type** (סוג נתונים) של שדה מספור אוטומטי, קיבל את הערך **adInteger**. ערך זה אינו משקף את טבעו הדינמי של סוג נתונים מספור אוטומטי. 4 Jet, לעומת זאת, כולל סוג נתונים נפרד שנקרא **Identity**, שמקביל לסוג הנתונים מספור אוטומטי (לקבלת מידע נוסף אודוות השימוש בסוג נתונים זה, עיין פרק 2). הסעיף "יצירת טבלה באופן ידני" שהופיע קודם לכן בפרק, מתאר מילוט מפתח נוספת לניהול שדות מספור אוטומטי באמצעות קוד.

הימנעות מהחלפת טבלה

בעת הוספה ידנית של טבלה למסד נתונים, קל לבדוק אם שם הטבלה קיים כבר. אם תנסה בשוגג לשומר במסד הנתונים טבלה בשם של טבלה שכבר קיימת, Access יזהיר אותך וישאל אם ברצונך לדرس את הטבלה הקיימת. לעומת זאת, אם שיגרה מנסה ליצור באמצעות קוד, טבלה בשם של טבלה קיימת, Access נוצר ומציג שגיאת זמן ריצה. עליך לצורך מנגנון לכידת שגיאות במצב כזה. קיימות כמה גישות לטיפול במצב זה. השיטה הולמת ביותר תלויות בתדרירות שבה תיצור טבלאות חדשות.

השיגרה `MakeLocalTableErrCatcher` שלפניך מבוססת על גישה קלאסית לכידת שגיאות. ראשית, היא מאפשרת את השימוש בשיגרה מותאמת אישית לטיפול בשגיאות, כך שהתוכנית יכולה לנצל שגיאות. המשפט `On Error` בתחילת השיגרה אחריו על כך. לאחר מכן, השיגרה מנסה ליצור ולצרף את הטבלה `FamilyMembers2` (בדומה לשיגרה `MakeLocalTable`). אם הטבלה קיימת כבר בקטלוג, Access VBA יוצר שגיאה (-2147217857) ומעביר את השליטה לROUTINE הטיפול בשגיאות. לוגיקת לכידת השגיאות מוחפשת את השגיאה "כבר קיים". אם היא

مزזה אותה, היא מוחקמת את הטבלה הקיימת ומחזירה את השליטה לשורה שגרמה לשגיאה. הדבר מאפשר לתוכנית לשמור את הטבלה החדשה ולצאת מהשיגרה בצוות מסודרת. אם שגיאת גרמה לתוכנית לעבור דרך רוטינת הטיפול בשגיאות, השיגרה מדפיסה את מספר השגיאה בחלון **Immediate** (מיידי) בטרם تستティים בצוות מסודרת. לא יקרה מצב שהשיגרה تستティים בהציג הودעת מערכת.

```
Sub MakeLocalTableErrCatcher()
On Error GoTo TableErrCatcher
Dim cat1 As ADOX.Catalog
Dim tbl1 As ADOX.Table

'Reference objects for table
    Set cat1 = New Catalog
    cat1.ActiveConnection = CurrentProject.Connection
    Set tbl1 = New Table

' Name table and append columns.
    With tbl1
        .Name = "FamilyMembers2"
        .Columns.Append "FamID", adInteger
        .Columns.Append "Fname", adVarWChar, 20
        .Columns.Append "Lname", adVarWChar, 25
        .Columns.Append "Relation", adVarWChar, 30
    End With

' Append new table to Tables collection and free catalog resource.
    cat1.Tables.Append tbl1
    Set cat1 = Nothing

' Exit the procedure.
TableErrExit:
    Exit Sub

    TableErrCatcher:
' Trap "table already exists" error.
' Delete table and resume.
    If Err.Number = -2147217857 Then
        cat1.Tables.Delete "FamilyMembers2"
        Resume
    End If

' Print details for other errors and exit.
    Debug.Print Err.Number, Err.Description
    Resume TableErrExit
End Sub
```

החלפת טבלה

אם יישום מסד הנתונים יוצר באופן קבוע את הטבלה FamilyMembers2, השיגרה עלולה לגרום לשגיאה מהסוג "כבר קיים" בכל פעם שמבצעים אותה. במצב זה, השיגרה תפעל מהר יותר אם תנסה למחוק את הטבלה הקיימת לפני שתצרף את הטבלה החדשה. כך ניתן להימנע מהצורך לטפל בשגיאה. עדיין יש צורך בשגרת טיפול בשגיאות למצבים שטבלה אינה קיימת, או למצבים שגיאה אחרים. השיגרה שלפניך כותבת את מנגנון הטיפול בשגיאות כশ邏חים ללא תנאי טבלה בעלת שם זהה לווען נסיסים לצרף למסד הנתונים.

```
Sub MakeLocalTableErrCatcher2()
On Error GoTo TableErrCatcher
Dim cat1 As ADOX.Catalog
Dim tbl1 As ADOX.Table
' Reference objects for table
    Set cat1 = New Catalog
    cat1.ActiveConnection = CurrentProject.Connection
    Set tbl1 = New Table
' Name table and append columns.
    With tbl1
        .Name = "FamilyMembers2"
        .Columns.Append "FamID", adInteger
        .Columns.Append "Fname", adVarWChar, 20
        .Columns.Append "Lname", adVarWChar, 25
        .Columns.Append "Relation", adVarWChar, 30
    End With
' Delete the old table (if it is there).
' Append the new one, and free the resource.
    cat1.Tables.Delete "FamilyMembers2"
    cat1.Tables.Append tbl1
    Set cat1 = Nothing
' Exit the procedure.
TableErrExit:
    Exit Sub
TableErrCatcher:
'Trap "object not in collection" error.
' Resume at next line.
    If Err.Number = 3265 Then
        Resume Next
    End If
' Print details for other errors and exit.
    Debug.Print Err.Number, Err.Description
    Resume TableErrExit
End Sub
```

שגיאה 3265 הנזכרת בשגרת הטיפול בשגיאות של MakeLocalTableErrCatcher2, היא תוצאה של הניסיון למחוק אובייקט שלא נמצא באוסף. השגירה MakeLocalTableErrCatcher2 פשוט לוכדת את השגיאה ומשיכה להתבצע לאחר השורה שגרמה לה. כל שגיאה אחרת גורמת לסיום מסודר של התוכנית, כשהחסימן **Immediate**.

עבודה עם אינדקסים

באפשרות להוסיף גם מפתחות ראשיות, אינדקסים וקשרי גומלין באמצעות קוד. תוכל להגדיר מפתחות ראשיות ואינדקסים בשדה ייחיד או בשדות רבים.

יצירת מפתח ראשי

הוספת מפתח ראשי או אינדקס לטבלה כמו כהוספה טבלה חדשה לקטלוג. ראשית, עליך ליזור הקשר שלו יתוסף האינדקס, כולל קטלוג וטבלה. שנית, עליך להגדיר את מאפייני האינדקס. אלה עשויים להשנות בין אינדקסים ומפתחות הראשיות. שלישי, עליך להוסיף עמודה לאינדקס ולאחר מכן לצרף את האינדקס החדש לטבלה. אם חלה שגיאה, כגון אם האינדקס כבר קיים, עליך לחת מענה הולם לכך. השגירה AddPK שלפניך יוצרת מפתח ראשי באופן דינמי.

```
Sub AddPK()
Dim cat1 As New ADOX.Catalog
Dim tbl1 As New ADOX.Table
Dim pk1 As New ADOX.Index

' Create a context for the new primary key.
cat1.ActiveConnection = CurrentProject.Connection
Set tbl1 = cat1.Tables("FamilyMembers2")

' Set the primary key properties.
With pk1
    .Name = "MyPrimaryKey"
    .PrimaryKey = True
    .Unique = True
    .IndexNulls = adIndexNullsDisallow
End With

' Append column to index and index to table.
pk1.Columns.Append "FamID"
tbl1.Indexes.Append pk1

' Free resources.
Set cat1 = Nothing
End Sub
```

השיגרה פותחת בהכרזה ויצירת מופיע של האובייקטים Catalog ו-Table (ליצירת מפתח ראשי יש צורך בכל שלושת האובייקטים).שים לב שלא קיים אובייקט ממורש עבור מפתח ראשי. בשלב הבא, השיגרה מגדרה את ההקשר להגדרת מפתח ראשי חדש. השיגרה מגדרה את המאפיין ActiveConnection של האובייקט Catalog כך שהקטלוג יציביע על מסד נתונים מסוים. לאחר מכן היא מגדרה את הפניות הטבלה שנמצאת במסד נתונים זה. הפניה זו היא הטבלה שאליה תוסיף השיגרה שכבתת את המפתח הראשי החדש.

כעת, השיגרה מגדרה ארבעה מאפייני אינדקס. הראשון הוא שם המפתח הראשי. הוא מופיע בתור ערך בעמודה **שם אינדקס** (Index Name) בחולון **אינדקסים** (Indexes) של הטבלה. שלושת המאפיינים הנוטרים מייחדים את המפתח הראשי מאינדקס פשוט כלשהו. עליך להקפיד ולהגדיר מאפיינים אלה כפי שהם מופיעים בשיגרה AddPK בעת יצירת מפתח ראשי.

בשלב זה, מפעילה השיגרה שתי שיטות Append. הראשונה מצרפת את העמודה ID מתוך הטבלה FamilyMembers2 אל האינדקס. השיטה השנייה מצרפת את האינדקס לטבלה. לבסוף, השיגרה מגדרה את האובייקט Catalog בתור Nothing, ועל ידי כך משחררת את המשאים ששימשו לייצירת המפתח הראשי.

לעתים קרובות יש צורך בשדה שסוג הנתונים שלו הוא מספר אוטומטי, מספר שלם אורך או טקסט, שיישמש כמפתח ראשי. דבר זה מושך את ביצועי מסד הנתונים בהשוויה למפתח ראשי המבוסס על שדות רבים. לעומת זאת, יש מקרים שהשימוש באינדקס המבוסס על שדות רבים יותר במנוחים של אףו השימוש בנתוני הטבלה. במקרה שאינדקס מבוסס-שדות רבים הולם את הרכבים, כל שעיליך לעשות הוא לצרף יותר מעמודה אחת לאינדקס בטרם תצרא את האינדקס לטבלה.

בדוגמה לעיל, ניתן להחילף את השורה :

```
pk1.Columns.Append "FamID"
```

בשורות הבאות :

```
pk1.Columns.Append "Lname"
pk1.Columns.Append "Fname"
pk1.Columns.Append "Relation"
```

שורות אלו מגדרות מפתח ראשי בשלושה שדות במקום בשדה אחד. השיגרה AddPK יוצרת מפתח ראשי המבוסס על שדה יחיד ; החולון **אינדקסים** הראשון בתרשים 3.14 מציג את תוצאת פעולה השיגרה AddPK. החולון השני בתרשים מציג את תוצאת פעולה השיגרה AddPK3 – שיגירה זהה ל-AddPK, למעט השינוי בשורות הקוד (שתי השגרות מופיעות בתקליטור המצורף בספר, בחלק של פרק 3). לא ניתן לקיים אלא מפתח ראשי יחיד בזמן נתון, ולכן ניתן לשער ידנית את המפתח הראשי בין הקריאה לשגרות AddPK ו-AddPK3.

The screenshot shows two instances of the 'Indexes' dialog box for the 'FamilyMembers2' table.

Top Dialog (Left):

Index Name	Field Name	Sort Order
MyPrimaryKey	FamID	Ascending

Bottom Dialog (Right):

Index Name	Field Name	Sort Order
MyPrimaryKey	Lname	Ascending
	Fname	Ascending
	Relation	Ascending

Common Properties:

- Primary: Yes
- Unique: Yes
- Ignore Nulls: No

A tooltip in the bottom right corner states: "The name for this index. Each index can use up to 10 fields."

תרשים 3.14: חילונות אינדקסים לאחר הפעלת השגרות AddPK3 ו-AddPK

השגרות AddPK ו-AddPK3 יכולות להיכשל מסיבות אחדות. השגירה AddPKErr שלפנינו לוכזת שתי שגיאות: מפתח ראשי שכבר קיים (קוד שגיאה 2147467259-) וטבלה שכבר נפתחה (קוד שגיאה 2147217856-). כפי שנאמר, לא ניתן להוסיף מפתח ראשי חדש אם קיים כבר מפתח ראשי. בנוסף, לא ניתן לשנות את מבנה האינדקס אם הטבלה פתוחה.

```
Sub AddPKErr()
On Error GoTo PKErr
Dim cat1 As New ADOX.Catalog
Dim tbl1 As New ADOX.Table
Dim pk1 As New ADOX.Index
Dim iNumber As Integer
' Create a context for the new primary key.
cat1.ActiveConnection = CurrentProject.Connection
Set tbl1 = cat1.Tables("FamilyMembers2")
```

```

' Set the primary key properties.
' The label (SetPKvariable) gives the procedure a recovery point
' from a previously existing primary key.
SetPKvariable:
    With pk1
        .Name = "MyPrimaryKey"
        .PrimaryKey = True
        .Unique = True
        .IndexNulls = adIndexNullsDisallow
    End With
' Append column to index and index to table.
    pk1.Columns.Append "FamID"
    tbl1.Indexes.Append pk1
' Exit procedure.
PKErrExit:
    Set cat1 = Nothing
    Exit Sub
PKErr:
' Checks for table already in use
    If Err.Number = -2147217856 Then
        MsgBox "FamilyMembers2 currently in use. This" & _
            " operation requires the table to be closed."
' Checks for primary key already exists
    ElseIf Err.Number = -2147467259 Then
        For Each pk1 In tbl1.Indexes
            If pk1.PrimaryKey = True Then
                tbl1.Indexes.Delete (iNumber)
                Resume SetPKvariable
            End If
            iNumber = iNumber + 1
        Next pk1
' Traps for other errors
    Else
        MsgBox "Open Immediate window for Bug report"
        Debug.Print Err.Number; Err.Description
        Resume PKErrExit
    End If
End Sub

```

.AddPK מכך חריגות, הלוגיקה של השגירה AddPKErr זהה ברובה לו של On. AddPKErr מאפשרת להפעיל תחילת רוטינה לטיפול בשגיאות באמצעות המשפט On. אזור ההכרזה כולל משפט Dim חדש למשתנה מסוג Integer. Error מנגןו לכידת השגיאות מנצח משתנה זה בתור אינדקס בלולאה שעוברת על האוסף Indexes של הטבלה.

מנגנון לכידת השגיאות של AddPKErr מתחילה את פועלתו בתוויות If...Then If...ElseIf. בזאת בשלב ראשון אם הטבלה פתוחה. אם הטבלה פתוחה, השגירה מציגה תיבת הودעה שמסבירה את הבעיה ומציעה פתרון לפני שתסתטיים בזורה מסודרת. הפסוקית IfElseIf מזהה את קיומו של מפתח ראשי. במקרה זה, הרוטינה מספרת את האינדקסים בטבלה, עד שmagieva את האינדקס שהמאפיין PrimaryKey שלו מוגדר בטור True. בעת היא מוחקמת את האינדקס ומחזירה את השיליטה לשלב ההתחלה של הגדרת מפתח ראשי חדש. פעולה זו הכרחית, מכיוון שהשגיאה מבטלת את ההגדירות הקודומות. אם הטבלה אינה פתוחה וגם לא קיים מפתח ראשי כלשהו, השגירה כותבת את מספר השגיאה ואת תיאורה לחלוּן **Immediate** לפני שמסתטימת. מוצגת למשתמש תיבת הודעה שמורה לו להציג את החלוּן **Immediate** לקבלת פרטי סיבת השגיאה.

יצירת אינדקס

הוספת אינדקס פשוט לטבלה אינה שונה בהרבה מהוספת מפתח ראשי. ההבדל העיקרי הוא שאין מגדרים את המאפיין PrimaryKey בתור True (כברירת מחדל, ערכו False). השגירה AddIdx שלפניך דומה במבנה שלה ל-AddPK. מלבד זה שאיןנה מגדרה את מאפיין המפתח הראשי, ההבדל העיקרי ביןיהו הוא שהיא קובעת מפורשות סדר מיון לאינדקס. השגירה ממיינת את הטבלה לפי העמודה FamID בסדר יורד, ועל ידי כך גורמת לה להציג קודם את הרשומות המאוחרות (בנהנחת שרשומת בעלות ערך FamID גדול יותר מטופסות לאחר רשומות קודמות). כאשרה קובע את סדר המיון, עליך לציין את מספר העמודה שעלייה הוא חל. מספר העמודות הוא מבוסס-אפס (מספר העמודה הראשונה הוא 0).

```
Sub AddIdx()

Dim cat1 As New ADOX.Catalog
Dim tbl1 As New ADOX.Table
Dim idx1 As New ADOX.Index

' Create a context for the new index.
cat1.ActiveConnection = CurrentProject.Connection
Set tbl1 = cat1.Tables("FamilyMembers2")

' Set the index properties.
With idx1
    .Name = "LastIsFirst"
    .Unique = True
    .IndexNulls = adIndexNullsDisallow
End With
```

```

' Append column to index and set its sort order.
' Append new index to table.
    idx1.Columns.Append "FamID"
    idx1.Columns(0).SortOrder = adSortDescending
    tbl1.Indexes.Append idx1

' Free resources.
    Set cat1 = Nothing
End Sub

```

אכלוס טבלה באופן דינמי

לאחר שעצבת טבלה והגדרת את האינדקסים שלה, عليك לאחסן משתי גישות כדי למלא אותה בערכים. כשתווני הטבלה המיעודת לאכלוס נמצאים בטבלה אחרת, באפשרותך להגדיר ערכת רשומות בטבלת מקור הנתונים ובטבלה החדשה. לאחר מכן عليك לנוט בשתי ערכות הרשומות בצוරה מסונכרנת תוך העתקת רשומות לטבלה החדשה באמצעות השיטה AddNew. גישה זו עוברת בלולאה על ערכת רשומות, וכך אינה פתרון מעשי לטבלאות גדולות מאוד. הגישה השנייה היא שימוש ב-SQL להוספת ערכים לטבלה החדשה, תוך התבוססות על ערכי הטבלה המקורית. גישה זו אינה מבוססת על ערכות רשומות או השיטה New, אך היא דורשת אובייקט Command.

אחד לפחות.

שימוש בערכות רשומות

הגישה הראשונה מומחשת באמצעות השגירה הבאה, ודורשת אובייקט Connection ואובייקט Catalog נוספים לזוג ערכות רשומות (Recordsets). האובייקט Catalog ושתי ערכות הרשומות חולקים אובייקט Connection משותף. אובייקטים רבים מאוד יכולים ביניהם חיבור משותף, וכך יש טעם להציג על אובייקט Connection ולהפעילו, לכל אחד מהאובייקטים האחרים הזוקקים לו.

```

Sub AddValues()
Dim cnn1 As ADODB.Connection
Dim cat1 As New ADOX.Catalog
Dim rst1 As New ADODB.Recordset
Dim rst2 As New ADODB.Recordset

' Set context for populating new table (FamilyMembers2).
    Set cnn1 = CurrentProject.Connection
    Set cat1.ActiveConnection = cnn1
    Set rst1.ActiveConnection = cnn1
    Set rst2.ActiveConnection = cnn1

```

```

' Open recordsets based on new and original tables.
rst1.Open "FamilyMembers2", , adOpenKeyset, _
adLockOptimistic, adCmdTable
rst2.Open "FamilyMembers", cnn1, adOpenForwardOnly, _
adLockReadOnly, adCmdTable

' Loop through recordsets to copy from original to new table.
With rst1
    Do Until rst2.EOF
        .AddNew
            .Fields(0) = rst2.Fields(0)
            .Fields(1) = rst2.Fields(1)
            .Fields(2) = rst2.Fields(2)
            .Fields(3) = rst2.Fields(3)
        .Update
        .MoveNext
        rst2.MoveNext
    Loop
End With
End Sub

```

לאחר ההצהרה על אובייקטים והקצתה החיבור לאובייקטים שישתמשו בו, השיגר פותחת ערכות רשומות בטבלה המכילה את הרשומות המקוריות, FamilyMembers, ובטבלה החדשה, FamilyMembers2.

הקוד שלעיל מציג שתי גישות להקצת חיבור לערכתי רשומות. באפשרותן להקצות חיבור למאפיין ActiveConnection של ערכתי רשומות – AddValues – משתמש בטכניקה Open Connection בshiיטה Open של rst2. בגישה השנייה, תוכל להפנות אל אובייקט rst1-ב- rst2 מושתמשת בטכניקה של ערכתי רשומות – השיטה Open הפעלת על rst2 מושתמשת בטכניקה זו. ניתן לפעול בכל אחת משתי הגישות בעובדה עם שתי ערכות הרשומות. השירה AddValues כוללת שורת קוד שמראה כיצד להגדיר את המאפיין ActiveConnection של rst2 באמצעות פעולות הקצאה. שורה זו מסומנת בתור הערבה. אם תחליט לנוקוט גישה זו בטיפול ב- rst2, תוכל להסיר את הפניה אל 1 לחץ בשיטה Open הפעלת על rst2.

המשפט Do...Loop העobar על הרשומות של rst2 מוקון בתוך המשפט With...End המתייחס אל rst1 (עקב הפניות הרבות אל rst1 בלבדה, משפט זה מאייך את פעולת הקוד ומשפר את קרייאותו). בתוך הלולאה, השיטה AddNew מאפשרת את הקצת הרשותה הנוכחית ב- rst2 אל rst1. השימוש בשיטה Update להשלמת פעולה השיטה AddNew הוא אופציונלי (הדוגמה מציגה אותה בצורה הערבה, ולא כחלק פעיל בקוד). לאחר הוספת הרשותה הנוכחית מ- rst2 לrst1, השירה מתקדמת לרשותה הבאה ב- rst2. אין צורך להתקדם גם ב- rst1, מכיוון שמעבר הלולאה הבא יוסיף רשותה חדשה בסוף ערכתי הרשומות rst1. הלולאה Do ממשיכת להתבצע עד הגיעו לסוף הקובץ rst2 (EOF).

שמירת ערכת רשומות

לאחר שאלילסת בערכים את ערכת הרשומות והטבלה שעליה היא מבוססת (ואולי אף עדכנת אותה), אפשרוות לשמר עותק של ערכת הרשומות. עושים זאת באמצעות שיטת **ערכת הרשומות החדש** Save בהתאם לסוג הקובץ שלו, המבוסס על תבניות בינהריות קומפקטיות. Access 2000 מציע כמה דרכים לשימירה ואחזר של ערכות רשומות באמצעות תבנית חדשה זו והפק המתאים לה. השגרות שלפניך שומרות ערכות רשומות על ידי שימוש בסמן לקוח (client cursor), ולאחר מכן פותחות אותה.

```
Sub SaveRST()
On Error GoTo SaveRSTErr
Dim rst1 As New ADODB.Recordset
' Open client cursor and recordset.
    rst1.CursorLocation = adUseClient
    rst1.Open "FamilyMembers2", CurrentProject.Connection, _
        adOpenStatic, adLockBatchOptimistic
    rst1.Save "c:\FamilyMembers3.adtg", adPersistADTG
SaveRSTExit:
    Exit Sub
SaveRSTErr:
    ' Trap file already exists error
    If Err.Number = -2147286960 Then
        Kill "c:\FamilyMembers3.adtg"
        Resume
    End If
    ' Exit for other errors.
    MsgBox "View Immediate window for error diagnostics.", _
        vbCritical, "Programming Microsoft Access 2000"
    Resume SaveRSTExit
End Sub

Sub OpenSavedRST()
Dim rst1 As New ADODB.Recordset
' Open saved recordset file.
    rst1.Open "c:\FamilyMembers3.adtg", "Provider=MSPersist"
' Print selected info to confirm retrieval.
    Debug.Print rst1.Fields(0).Name & " = " & rst1.Fields(0).Value
End Sub
```

חלקה הראשי של השירה SaveRST פותח סמן לקוח ומציב בו ערכת רשומות. לאחר מכן הוא מפעיל שיטה Save של ערכת רשומות. שימרת ערכת רשומות לקובץ אינו תהלייך מורכב יותר מכז. הקובץ ADTG (Advanced Data TableGram) הוא מיניימי בגודלו (1KB בלבד בגרסה של מחבר הספר). יש בכך חיסכון משמעותי לעומת דוח אחסון ערכות רשומות בצורה טבלה בקובץ מסד נתונים.

אם אתה רגיל לשומר ערכות רשומות לאחר עדכונה, השיטה Save תיכשל אם לא תמחק את שם הקובץ הישן, או תיתן שם חדש. לחילופין, באפשרותך לລמוד את השגיאה, למוחק את גרסת הקובץ הישנה ולאחר מכן להפעיל שנית את השיטה Save. שגורת הטיפול בשגיאות של SaveRST מדגימה טכניקה זו.

באפשרותך לפתח מערכת רשומות שמורה באמצעות שתי שורות קוד בלבד, כפי שדגימה OpenSaveRST. השורה הראשונה מצהירה על אובייקט Recordset ויוצרת מופיע שלו. השורה השנייה מפעילה את השיטה Open של מערכת הרשומות. המ庫ר שלו הוא הקובץ ששמרת השירה SaveRST, וארוגומנט החיבור שלו מפנה לספק MSPersist הכלול בחבילת המוצר Access 2000. שורה שלישיית מדיפסה את תוכן השדה מתוך מערכת הרשומות כדי לאשר שהמידע שבקובץ תקין.

שימוש ב-SQL

רעיון המעבר בלולאה על מערכת רשומות, רשומה אחר רשומה בזמן נתון קוסט למפתחים רבים עקב מוחשיות הפעולה. אכן, קל לדמות את המעבר מרשותם לרשותם ב-rst2 ו-rst1 ואת ההוספה של רשומות ייחדות לrst1. יחד עם זאת, ייעיל יותר לבצע את המשימה בעזרת קוד SQL. בנוסף, הגישה המבוססת על SQL זקופה לאובייקט ADODB Command אחד בלבד מתוך הספרייה ADODB. גישת המעבר בלולאה מסתמכת על אובייקטים אחידים הנקווים מתוך הספריות ADODB ו-ADOX ייחדיו. אם קשה לך לכתוב קוד SQL, אפשרות נצל את מעצב השאילותות החזוית של Access ל渴בלת גירסה ראשונית ותפקידית של תחביר SQL תקין לביצוע משימה, כגון הוספת רשומות מתוך טבלה אחת לטבלה אחרת. השירה AddValuesSQL ממחישה את יתרונותיו החסכוניים של קוד SQL (לאMITO של דבר, אם היישום שלך אינו חייב לדעת את מספר הרשומות שהוא מוסיף, ניתן ל��ר עוד יותר את הקוד).

```
Sub AddValuesSQL()
Dim cmd1 As ADODB.Command
Dim intRowsAdded As Integer
Dim str1
' Instantiate cmd1.
  Set cmd1 = New ADODB.Command
' Set the connection and text for cmd1
' before executing it.
  With cmd1
    .ActiveConnection = CurrentProject.Connection
    .CommandText = "INSERT INTO FamilyMembers2 " & _
      "SELECT FamilyMembers.* FROM FamilyMembers;"
    . CommandType = adCmdText
    .Execute intRowsAdded
  End With
'Report how many records cmd1 added.
  str1 = intRowsAdded & " rows were added to the table."
  MsgBox str1, vbInformation, "Programming Microsoft Access 2000"
End Sub
```

השיגרה פותחת בהצהרה ויצירת מופע של אובייקט Command. היא משתמשת במשפט With...End ActiveConnection במשפט With...End CommandType SQL. קטע הקוד הכלול במשפט CommandText של האובייקט Command (בגישה זו, טקסט הפקודה יהיה תמייד משפט SQL). קטע הקוד מגדר את המאפיין CommandType בטור adCmdText כדי לטעות את הביצועים (אם לא תגדיר את המאפיין CommandType, CommandText יטרגם לקובע את הסוג של האובייקט Command לפני לטפל בפקודה). לבסוף, קטע הקוד קורא לשיטה Execute עם הארגומנט האופציוני intRowsAdded. בסיום השיגרה, מופיע intRowsAdded מכיל את מספר הרשומות שמשפט SQL טיפול בהן. מופיע תיבת הودעה נוספת מנצל את intRowsAdded כדי לדוח למשתמש את מספר הרשומות שנוסףו לטבלה החדשה.

עבודה עם נתונים בתבניות אחרות

פרק 2 לימדנו כיצד להשתמש בספק DB Jet 4 OLE מתוכה יישום. באפשרות לעשוט זאת כדי לגשת נתונים שנמצאים במסד נתונים מסוימים של Access מtoc כל מסד נתונים אחר של Access. שיטות עובודה זו ישימה אפילו נתונים המאוחסנים בתבניות אחרות שעבורן קיים ספק DB OLE. גם במקרים שיש פסק DB OLE אין זמין, באפשרות ליצור קישור אל מסד נתונים אחר באמצעות DAO, לגשת בקלות למידע שהוא מכיל ולנצל את יתרונותיו של מקור נתונים מקוישר כלשהו.

באפשרות נצל את ספק Jet 4 כדי לפתח מקור נתונים מסווג Excel. תרשימים 3.15 מציג גיליון עבודה פשוט בחוברת עבודה של Excel, שנקראת Customers.xls. הגיליוון Sheet1 מכיל טוחם בשם Customers (ראה **תיבת שם** - Name Box) ובו מידע אודות קוד לקוחות, שם החברה, שם איש קשר ותנאי תשלום. השיגרה ש לפניה פותחת טוחם בחוברת עבודה של Excel 2000 באמצעות Access 2000

```
Sub OpenPrintXLDataSource()
Dim cnn1 As New ADODB.Connection
Dim rst1 As Recordset
' Open and set recordset.
cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Programming Access\Chap03\Customers.xls;" & _
    "Extended Properties=Excel 8.0;"
Set rst1 = New ADODB.Recordset
rst1.CursorType = adOpenKeyset
rst1.LockType = adLockOptimistic
rst1.Open "customers", cnn1, , , adCmdTable
' Open recordset and print a test record.
Do Until rst1.EOF
    Debug.Print rst1.Fields(0).Value, rst1.Fields(1).Value
    rst1.MoveNext
Loop
End Sub
```

	A	B	C	D	E	F
1	CustomerID	CompanyName	ContactFirstName	ContactLastName	PaymentTerms	
2	VBD	Visual Basic Developer	Lori	Coffey	Day of publication	
3	MOD	Microsoft Office & VBA Developer	Kaye	Waterhouse	30 days after publication	
4	IW	Internet World	Bob	Danners	Day of seminar	
5	SA	Smart Access	Paul	Augustus	Upon approval	
6	BYTE	Byte	Tom	Davis	Upon approval	
7	WT	Web Techniques	Dave	DeMott	When commissioned	
8	MODJ	Microsoft Office Developer's Journal	Tami	Dreyfuss	Day of publication	
9	VPBJ	Visual Basic Programmer's Journal	Jim	Donaldson	End of Month for publication	
10	AOVBA	Access Office VB Advisor	Mike	Giddy	30 days after publication	
11	MWB	Microsoft Web Builder	Sunil	Mehrotra	Day of publication	
		Active Server Developer's			Day of	

תרשים 3.15: יכול לטפל במידע שבגילוון עבודה של Excel

השיגרה OpenPrintXLDataSource משתמשת באובייקט Connection כדי לפתח חוברת עבודה של Excel. השיטה Open של האובייקט cnn1 יוצרת חיבור אל קובץ חוברת העבודה. הספק בקריה זה הוא אותו ספק בו השתמשנו בעבר קבצי Jet, או Jet 4. מנגנון Extended Properties מצין תבנית תואמת לקבצי Excel 2000 וארוגומנט מקור הנתונים מצביע על חוברת עבודה של Excel בספריה Programming Access שבסיכון C. ארבע השורות הבאות בשיגרה יוצרות ופותחות הפניה לערצת רשומות שמוגדרת בטוחה הראשוניים שבערצת הרשומות לחולן Immediate. השיגרה Do שבסוף השיגרה, מדפיסה את שני השדות Customers.xls וCustomers.Customers.

לאחר פתיחת ערצת רשומות עבור מקור נתונים ב-Excel, באפשרות לבצע כל פעולה Access-ית אפשר לבצע על מקור רשומות, כגון עדכון ערכי שדות. ההבדל במקרה זה, הוא שהיישום מעדכן ערכיהם בטוחה בחוברת העבודה של Excel, ולא ערכיהם בטבלה של Access. שגרת הדוגמה OpenPrintXLDataSource ממחישה כיצד ברורה את עצמתם של ספקי ADO ו-OLEDB כחלק ממערכות יישום Access.

פרק 3: עיצוב טבלאות 193

שימוש בספק MSDASQL

Access 2000 עובד עם ספק OLEDB נוסףים, כגון MSDASQL, הפעיל על כל מקורות הנתונים מסווג ODBC. למרות שקיים ספק מיוחד עבור Microsoft SQL Server, השיגרה שלפניך מונצלת את הספק הכללי ODBC כדי להמיחש את השימוש בספק בעבודה עם מקור נתונים ODBC כלשהו. ספק זה מתיחס תמיד למקור נתונים ODBC, ולכן באפשרות להשתמש בו בתיאום עם DSN. השיגרה מציגה ספק MSDASQL המפנה למסד נתונים SQL, כדי להציג את פרטי המחבר: מס'utzer, שם פרטי ושם משפחה.

```
Sub GetODBCThroughOLEDB()
Dim cnn1 As New ADODB.Connection
Dim rst1 As ADODB.Recordset
' Open ODBC sources with msdaSQL provider and DSN reference.
cnn1.Open "Provider=MSDASQL;DSN=Pubs;"
Set rst1 = New ADODB.Recordset
rst1.CursorType = adOpenKeyset
rst1.LockType = adLockOptimistic
rst1.Open "authors", cnn1, , , adCmdTable
' Open recordset and print a test record.
Do Until rst1.EOF
    Debug.Print rst1.Fields(0).Value, rst1.Fields(2), rst1.Fields(1).Value
    rst1.MoveNext
Loop
End Sub
```

השיגרה GetODBCThroughOLEDB מתחילה בפתיחת חיבור אל מסד הנתונים SQL Server Pubs ופעילה את השיטה Open של האובייקט. ארגומנט הספק מצינו את ספק MSDASQL, ואילו הארגומנט DSN מכיל את נתוני החיבור ופועל בהתאם לארגוומנט Data Source של ספק Jet 4. השיטה Open יכולה לציין את נתוני מחוזות החיבור, מה שambil את הצורך להתייחס ל-DSN. המשפט שלפניך מראה כיצד פותחים מסד נתונים : DSN מבלי לציין SQL Server Pubs

```
cnn1.Open "Provider=MSDASQL;DRIVER=SQL Server;" & _
"SERVER=CAB2200;DATABASE=Pubs;uid=sa;pwd=";
```

שםו של SQL Server הוא CAB2200. שים לב שמחוזות החיבור מכילה גם את הארגומנטים של זיהוי המשתמש והסיסמה. מחוזות החיבור ייחודיים למסד הנתונים שלו אתה מתחבר.

ארבע השורות הבאות של השיגרה פותחות ערכות רשומות בטבלה Authors במסד הנתונים SQL Server Pubs. תחביר השיטה Open זהה לזה ששימוש לפתיחה טווח בחוברת העבודה של Excel. היחס הסימטרי בין שני מקורות שונים אלה מחייב את עצמותו של ADO עם ספקי OLE DB. לו לא תאפשר עוברת על הרשומות כדי

להדפיס את מספר הזיהות, השם הפרטוי ושם המשפחה של כל מחבר. המאפיין Value הוא בירית המהידל של האוסף Fields של האובייקט Urcretת רשותmot, ולכן אין צורך לציין אותו (שים לב שהארוגמנט השני של השיטה Debug.Print אינו מפנה למופרשת למאפיין Value של האוסף Fields). העיצוב הכלול של השירה GetODBCThroughOLEDB מראה ש- Access 2000 יכול ליצור קישור אל מסד נתונים SQL Server באותה קלota שהוא יוצר קישור אל חוברת עבודה של Excel.

השיטות TransferDatabase ו-TransferSpreadsheet של האובייקט DoCmd הן אמצעי קל וחסין תקלות לkishor לנתונים שנמצאים ביישום אחר; הן אף אינן תלויות בהפניה למספרה DAO. שיטות אלו חושפות את מקורותיהן באמצעות החלון **מסד נתונים**, בעוד שהספק OLE DB מספק ממשק תכנותיו לעדדי למקורות הנתונים שלו. יכולך להפיק תועלת מחשיפת המקור באמצעות החלון **מסד נתונים** תליה ביישום שלך ובהעדפות של המשתמשים בו.

שימוש בשיטה TransferDatabase או בשיטה TransferSpreadsheet

השיטות TransferDatabase ו-TransferSpreadsheet הן תוצאה של פקודות מאקרו המאפשרות ייבוא, קישור וייצוא. השיטה TransferDatabase תומכת במיגוןו רחב של תבניות Lotus ו-Excel. השיטה TransferDatabase תומכת במקרים נתונים ODBC, כמו SQL Server ו-Oracle, וגם במקרים נתונים ISAM, Paradox, dBBase, ועוד. השיטות TransferSpreadsheet ו-TransferDatabase מצלות Microsoft FoxPro ו-Jet. השיטות ISAM ו-ODBC המשווקים עם Access. מידע רב-ערך אודות את מנהלי ההתקנים של ISAM ו-ODBC המשווקים עם Access. ובחילון **מאקרו הפתיחה בתצוגת עיצוב**, הכולל תיבות רשימה נפתחת שמהן ניתן לבחור ארגומנטים אפשריים.

השירה linkXLCustomers מציגה את תחביר השיטה TransferSpreadsheet לאפשרות להשווות טכניקה זו עם התפקודיות של ספק DB OLE. שגרת הדוגמה יוצרת קישור לטוחה customers.xls (הקובץ customers.xls) בחרבתה העובדה customers.xls בתיקיotor המצורף לspreadsheet. חוברת העובדה מבוססת על תבנית 97, Excel 97, Excel 2003 ו-WK3 ו-WK4. האובייקט DoCmd יוצר טבלה מקושרת שנקראת customers (-1) TransferSpreadsheet במאסד הנתונים. הארגומנט האחד לפניו אחרון של השיטה linkXLCustomers מצין כי הטווח של Excel כולל שמות שדות בשורה הראשונה.

```
Sub linkXLCustomers()
Dim cnn1 As New ADODB.Connection
Dim rst1 As ADODB.Recordset
' Use DoCmd to programmatically make the link.
DoCmd.TransferSpreadsheet acLink, acSpreadsheetTypeExcel97, _
    "XLCustomers", "C:\Programming Access\Chap03\Customers.xls", _
    -1, "Customers"
```

```

' Open and set recordset.
Set rst1 = New ADODB.Recordset
rst1.ActiveConnection = CurrentProject.Connection
rst1.CursorType = adOpenKeyset
rst1.LockType = adLockOptimistic
rst1.Open "XLCustomers", , , adCmdTable

' Open recordset and print a test record.
Do Until rst1.EOF
    Debug.Print rst1.Fields(0).Value, rst1.Fields(2)
    rst1.MoveNext
Loop
End Sub

```

לאחר שהשיטה TransferSpreadsheet מתבצעת, היחסום יכול לנצל את קבוצת החזרה (return set) שלו בזורה דומה לו שמתאפשרת מספק OLE BD. ראשית,فتح את ערכת הרשומות בטבלה המקורית. לאחר מכן טפל בערכת הרשומות בעוזרת קוד כדי למלא את דרישות היחסום (LinkXLCustomers מדפסה שורות אחדים לכל שורה).

השיטה TransferDatabase כוללת מיגון גדול יותר של אפשרויות גישה לנוטונים לעומת הפקודה TransferSpreadsheet. האפשרויות כוללות את ניהול ההתקנים היחסוטריים של SAM ו-ODBC. הואיל ורוב מסדי הנתונים תומכים ב-ODBC, שיטה זו יכולה לגשר כמעט אל כל מקור נתונים של מערכת ניהול מסדי נתונים יחסיים (RDBMS). יתרה מזאת, התחבריר של השיטה TransferDatabase דומה לזה של TransferSpreadsheet מה שמקל את השימוש בה.

השיגרה LinkODBCAuthors מנצלת את השיטה TransferSpreadsheet כדי ליצור קישור אל הטבלה Authors במאסד הנתונים Pubs של SQL Server. יוצרת LinkODBCAuthor ויצרת קישור אל הטבלה Authors תחת השם adoAuthors במאסד הנתונים הנוכחי. אם adoAuthors כבר קיימת במאסד הנתונים הנוכחי, השיטה שומרת את העותק המקורי מבלי להתריע על כך, יוצרת עותק נוסף ונוננת לו את השם adoAuthors1.

```

Sub linkODBCAuthors()
Dim cnn1 As New ADODB.Connection
Dim rst1 As ADODB.Recordset
' Use DoCmd to programmatically make the link.
DoCmd.TransferDatabase acLink, "ODBC Database", _
    "ODBC;DSN=Pub;UID=sa;PWD=;DATABASE=pubs", _
    acTable, "Authors", "dboAuthors"
' Open and set recordset.
Set rst1 = New ADODB.Recordset
rst1.ActiveConnection = CurrentProject.Connection
rst1.CursorType = adOpenKeyset
rst1.LockType = adLockOptimistic
rst1.Open "dboAuthors", , , adCmdTable

```

```
' Open recordset and print a test record.  
Do Until rst1.EOF  
    Debug.Print rst1.Fields(0).Value, rst1.Fields(2)  
    rst1.MoveNext  
Loop  
End Sub
```

הารגוומנט הראשון המועבר אל TransferDatabase מצין את סוג ההעברה אל מקור הנתונים ; הארגוומנט acLink יוצר טבלה מוקורת במאז הנתונים. שני הארגוומנטים הבאים מצינים את סוג מסד הנתונים ואת שמו. (השיגרה LinkODBCAuthors מדירה מקור נתונים מסוג ODBC). לאמיתו של דבר, דוגמה זו משתמשת במאז הנתונים המקובל Pubs של SQL Server (Authors). הארגוומנטים הנותרים מצינים את סוג אובייקט מסד הנתונים שעליו ברצונך לבסס את המקור, את שמו במקור המרוחק (Authors) ואת שמו ביחסום שלך (dboAuthors). שאר חלקי הקוד של השיגרה מבצעים פעולות זהות לאלו של שאר דוגמאות הקוד בסעיף זה.

טיפול בנתונים באמצעות שאלות

שאלות הן סוס העבודה של יישומי מסד נתונים. תוכל ליחסם אותן כדי להשיג מטרות רבות. שאלות מאפשרות לטפל נתונים שנמצאים בטבלאות מסד נתונים. תוכל לנצלן כדי להגדיר תכנים של טפסים ודוחות. שאלות יכולות גם לציין את מקור הנתונים של דף Web.

אחד הסיבות העיקריות לכך של שאלות היא המיגון העצום שלהן. פרק 2 למדנו כיצד לעצב שאלות החזרת שורות (row-returning) ושאלות bulk operation (bulk). שאלות פועלה בתבנית המוגדרת על ידי המשמש), באמצעות אובייקטי נתונים ActiveX (ADO). ניתן גם לבצע פעולה אלה ואחרות באמצעות שפת שאלתה מובנית (SQL). מפתחים משתמשים לעיתים קרובות ב- SQL כדי לטפל בטבלה על ידי חילוץ שורות ועמודות נבחרות, אך אלו אינן האפשרויות היחידות לטיפול במקור נתונים. שאלות יכולות כוללות צירוף ומחיקה של רשומות וערכו שדות. ניתן גם ליצור שאלות חדשות באמצעות SQL.

- פרק זה נדון בנושא השאלות של Access 2000 במילוי רמות.
- ◀ סקירה כללית על סוגי השאלות השונות. הפרק עוסק בטיבן של שאלות, بما שהן אפשרות לעשות ובאופן בו הן משלימות זו את זו.
- ◀ תיאור טכניקות ידניות לבניית שאלות. סקירה זו משרתת מטרות רבות. היא מרחיבה את הסקירה הכללית בהציג שלבים בייצרת כל אחד מסוגי השאלות.
- ◀ הפיכת משימה לאוטומטית באמצעות שאלות החזרת-שורות ושאלות פשוטות. ניתן להשתמש קוד שאלות החזרת-שורות ושאלות פשוטות פועלה פשוטות.
- ◀ התבססות על SQL באמצעות קוד. כאן נשלים את הסקירה מפרק 2 בנושא ADO ונדון בתוצאות ושורות מאוחסנות.
- ◀ טיפול במסדי נתונים מרוחקים. גרסת 2000 מטפלת במסדי נתונים מרוחקים טוב יותר מגרסאות קודמות. שימוש בשאלות למטרה זו ישפר את היישומים שתכתב ויעלה את רמת הביצועים שלך בתור מפתח בסביבת Access.

סקירת סוגים של שאלות

ניתן להתייחס לשאלות ככלים לחילוץ מידע מסודר נתונים אחד או יותר, אך קיים מגוון עצום של דרכים בהן שאלות מבוצעות משימה זו. בנוסף, שאלות יכולות לבצע פונקציות מסוימות שונים, במיוחד אם כוללים את תכונות הגדרת הנתונים של SQL וקשרם למקורות נתונים מרוחקים. בסעיף זה נסקר אפשרויות שאלתית רבות - 2000 Access תומך בהן. סקירה זו תשפק את המידע הדרוש כדי לבחור בסוג השאלתית הנכון למצב נתון.

שאלות בחירה (Select)

שאלות בחירה מאפשרות ליצור קבוצת משנה של פרטי מידע בטבלה מסד נתונים אחד או יותר. לרשות המפתחים עומד מערך אפשרויות עשיר שמאפשר בחירה ועיבוד נתונים בדרכים שונות. אפשרויות אלו יסייעו לך לקבוע את ערכי העמודה והשדה בקבוצת המוחזרת (Return Set) מותך שאלתית. שאלות בחירה הן מקור רשותות טוב עבר טפסים, דוחות ודף Web, וכן שיליטה בעקרונות הבסיסיים שלן תספר מאוד את עובdotך עם רכיבי פיתוח נוספים.

שאלת החזרת קבוצת משנה של שדות ברשותה היא אחת השאלות הבסיסיות ביותר שיש.noch מואוד להשתמש בה כשורצים להציג עמודות אחידות מתוך טבלה הכוללת עמודות רבות. חילוץ מספר עמודות חלקי יכול להציג את פועלות השאלתית. Access מחשב שאלתית בחירה בכל פעם שהוא פועל, מובהך כי המשמש יקבל תמיד את המידע העדכני ביותר. שימושים אינו אמור לחושף את המידע העדכני ביותר מהטבלה, יש להשתמש בקריטריונים שונים כדי להגביל את ערכי השורה שהשאלה מזקירה. לחילופין, תוכל להפעיל סוג אחר של שאלתית כדי ליצור טבלה זמנית חדשה שמכילה קבוצת משנה של עמודות ברגע מוגדר בציר הזמן.

קביעת קריטריונים

בין אם שאלתית בחירה בוחרת קבוצת משנה מתוך מקור נתונים ובין אם לאו, היא יכולה להגביל את השורות בקבוצת המשנה. באפשרות לציין את השורות שתוחזרנה על ידי קביעת קריטריונים מתאימים. השתמש בשורה **קריטריונים** (Criteria) בראשית עיצוב שאלתית כדי לקבוע קריטריונים. קביעת קריטריונים באמצעות קוד כולל את הפסוקיות WHERE ו-HAVING במשפט SQL. הפסוקית WHERE נועגת לרשומות בודדות במקור הרשומות עלייו מבצעים את השאלתית, אך הפסוקית HAVING יכולה להגביל את קבוצות ההחזרה בהתאם לערכי הפסוקית Y GROUP BY.

שאלות בחירה יכולות להחזיר קבוצות שורות שמתאימות לערך או לטווח ערכים מסוימים. בעת טיפול בטבלה ערכים גיאוגרפיים, ניתן להגיד שאלתית בחירה שתחזיר את שם העיר והמדינה (בארה"ב) בהתאם למיקוד. אם הרשומות במקור נתונים מכילות שדה מסוג **תاريיך/שעה**, כגון תאריך הזמנה או חשבונית, תוכל לבחור הזמינות או חשבונית מתאריך מוגדר. ניתן להשתמש גם באופרטורי השוואת, כגון גדול (>),

קטן (<) או שונה (>), כדי לבחור טווח שורות. אפשר להתאים את הפריט המבוקש לערך מחרוזת מוגדר, או לטווח ערכים באמצעות תווים כלליים (wild card). תווים אלה משמשים לקביעת טווח ערכי מחרוזות. לדוגמה, הкрיטריון S^* מתאים למחרוזות Sal ו-Saul, מכיוון שהתו * מתאים למספר תווים כלשהו. לעומת זאת, התו ? מתאים לתו אחד בלבד, ולכן הкрיטריון ?S מותאים למחרוזות Sal, אך לא למחרוזות Saul.

פונקציות צבירה (aggregate)

פתחחים יכולים להשתמש בפונקציות צבירה (Aggregate Functions) בשילוב SQL, כדי לסקם את הרשומות המקוריים. פונקציות צבירה טיפוסיות מונוטון, מסכימות או ממוצעות את רשותות מקור הנתונים בתגובה לשאלילת בחירה. ניתן להפעיל פונקציות צבירה כדי לחשב את מידת ההשתנות (Variability) של מקור רשומה (record source) באמיצאות פונקציות כגון VarP או stDev או Var. אפשר אף לחשב את המינימום או את המקסימום של קבוצת ערכים בשדה בעזרת פונקציות צבירה.

שאלילות בחירה יכולות לישם פונקציות צבירה בשילוב עם הפסוקית BY. הדבר מאפשר לשאלילה לחשב נתוני צבירה עבור רשותות מקובצות המקוריים. שימוש בפונקציות צבירה יחד עם פסוקית GROUP BY מאפשר לישום לחזיר את מספר שורות הפריט עבור כל הזמנה בשאלילת בחירה.

צירופים (joins)

אפשר לבסס שאלילות בחירה על טבלה אחת או יותר, על שאלילת אחרת או על שילוב כלשהו של טבלאות ושאלילות בחירה. בעת בחירה מתוך טבלה אחת או יותר, הישום יקשר את הטבלאות באמצעות שדות מסוימים אחדים. אם מסד הנתונים מקשר טבלאות באמצעות הגדרות שלמות הקשרים, Access יזהה זאת בעצמו ויצראן אותן כשתשלב אותן יחד בתורן מקור רשומה של שאלילה. Access מזהה גם שדות מסוימים במצביים נוספים, ועל ידי כך מאפשר לקיים, לשנות או להחליף קשרי גומלין שזוהו אוטומטית, בין טבלאות.

ברירת המחדל לצירוף טבלאות היא שילוב כל הרשותות בין שתי טבלאות שאחד משdotihen מכיל ערך תואם. צירוף הטבלאות Orders - Orders Details של מסד הנתונים Northwind בשאלילת בחירה פשוטה, מסוגל לחזיר את כל שורות הפריטים לכל הזמנה. במצב זה אנו מעוניינים רק בפריטים שתואמים להזמנת. טבלאות הקשורות זו לזו יכולות להכיל בטבלה אחת ערכי שדות ללא ערכים תואמים בטבלה המצורפת (לדוגמה, תוכל לאחרר נתונים מסד נתונים ישן שאינו מלאץ שלמות קשרים). הישום יכול לzechot רשותות שאין להן רשותות תואמות, על ידי הכללה מואצת של כל רשותות טבלה מצורפת אחת או יותר לקובצת החזרה. במצב זה, כל הרשותות המשולבות המכילות ערך בטבלה אחת, והחסרות שדה צירוף מהטבלה אחרת, תצביענה על רשותות לא תואמות.

تוצאה זוג כלשהו של ערכות רשומות מצורפות יכולה להctrar לטבלה או לשאלתה אחרת. אם אתה מנוסה ב-SQL תוכל לפתח קבוצת החזרה על ידי טיפול במספר טבלאות ושאלות באטען קשר גומלי המבוססים על צירוף מרכיב, זאת במשפט SQL אחד בלבד. מתלמיד ב-SQL עשוי למצוא שבתוכו יותר להעדיף בניית שאלות בקרה ורפית בראש עיצוב השאלתה כשםדובר בזוגות טבלאות או שאלות אחרות. בשמתכני מסדי נתונים נוקטים צעדים מקימי זה, הם יכולים לאמת את תוצאה צירוף שתי הטבלאות לפני צירוף קבוצת החזרה לטבלה או לשאלתה אחרת.

עדכו נתונים מקור

אחת התוצאות הנפוצות של שאלת בחירה היא **קבוצת רשומות דינמית** (dynaset). קבוצת רשומות דינמית היא ארכט רשות שומרת מפותחות ראשיים במקום נתונים קבוע. שאלת בחירה מאפשרת ליישום לשנות את השדות ברשותם שלעליהם מבוצעת השאלתה (לשם הנוחות, השתמש במילה רקע). לעיתים, השדות שברקע שאלת בחירה אינם ניתנים לעדכו. בשאלות בחירה המבוססות על טבלה בודדת או שתים בקשר יחיד-ליך, קיימת תמיד אפשרות לעדכו את רשומות הרקע (אלא אם משתמש אחר נועל אותה). בדרך כלל ניתן לעדכו טבלאות הקשורות הגומלי שלחן הוא מסווג יחיד-לרבבים.

הציג את השאלת בתוצאות **גיליון נתונים** והבט בשורת המצב כדי לקבוע אם שדה מסוים הוא בר-עדכו. אם השדה אינו כזה ויש לך צורך לשנותו, שולג ישירה לטבלה שבה הוא נמצא, או עיצוב מחדש של השאלת.

שמירת שאלות

כששאלות הן מורכבות. כדי לשמר אותן ואחר כך להפנות אליהן בשאלות אחרות. לדוגמה, שאלת בחירה המכרפת שתי שאלות נוספות (או יותר), ולאחר מכן מפעילה עליהם פונקציות צבירה של SQL כדי ליצור סיכון נתונים. שומר שאלת מורכבת והפנה אליה בשאלות אחרות כדי להימנע משגיאות תחביר. לעיתים, שימושים שאלת אחת או יותר כדי לקבל שאלת חדשה, לוגיקת השאלת עצמה יכולה להיות מורכבת מדי לכוח החישוב של מגנון מסד הנתונים. במצב זה, שומר שאלת קלט אחת או יותר לטבלה זמנית. לאחר מכן, השתמש בטבלאות אלו כתחליפים עבור השאלות. חשב מחדש את הטבלאות הזמניות בכל מקרה של שינוי בתוכני הקלט של הטבלאות המקוריות.

שאלות פעולה (Action)

במקום להחזיר קבוצת שורות כמו בשאלת בחירה, **שאלות פעולה** (Action Queries) מבצעות את משימותן מול טבלה אחת או יותר. שאלות פעולה מאפשרות להפעיל אוטומציה של יישום. אפשר לעצב שאלות פעולה באופן גרפי, והן תמשכנו לבצע את הפעולות החיים: הוספה, מהיקה ועדכו ורשומות. שאלות פעולה משתמשות ב-SQL לביצוע משימותיהם, ולכן ניתן לנצל אותן כפקודות טבלה

מתקচומות לטיפול ברשומות. מפתחי Access מנוסים תוהים מן הסתם על שאלתת הפעולה של Access שנקראת שאילתת יצרת טבלה (make table). שאילתת פועלה זו מאפשרת הגדרת נתונים וגם טיפול בהם. שאילתת יצרת טבלה היא הנושא המרכזי בסעיף שאילותות יצרת טבלה שבמהמשך הפרק.

שאילותות נוספת

שאילותות הוספה (Append Queries) מאפשרות להווסף רשומות לטבלה אחת המבוססות על אלו שבטבלה אחרת. הפעל שאילתת הוספה כשהיישום מקבל רשומות מקור חיצוני, כגון טבלה שנמצאת במחשב אחר. טבלת המקור חייבת להיות במסד הנתונים הנוכחי, או לפחות מקושרת אליו. טבלת היעד יכולה להיות במסד הנתונים הנוכחי או בקובץ מסד נתונים אחר. שמות השדות בשתי הטבלאות יכולים להיות שונות זה מזה, אך חייבת להיות תאימות בין סוגים הנתונים כדי לצרף ערכיהם משדה בטבלה אחת לשדה תאום בטבלה אחרת. טבלת המקור אינה מחייבת שכל השדות יהיו בטבלת היעד.

קיים מetriums העולמים ליצור שגיאות בשאילותות הוספה. בעיקרונו, עלייך להימנע מהוספת ערכים לשדה מספור אוטומטי, מכיוון ששדה מסווג זה מתאcls מעצמו. אם תנסה להווסף רשומה שגורמת לכפיפות לשדה המפתח הראשי, היישום ייצור שגיאה של הפרת מפתח. לא ניתן להווסף רשומות, אם הוספטן מפירה את תקופות הטבלה או את כללי שלמות הקשרים.

שאילותות מחיקה

שאילותות מחיקה (Delete queries) מאפשרות לישום להפוך את תהליך הסרת הרשומות מטבלה לתהליכי אוטומטי. ניתן לקבוע את הרשומות המיועדות למחיקה בעזרת קרייטריוניים מוגדרים. אפשר למחוק שתי רשומות או יותר באמצעות שאילתת מחיקה בודדת. כאשר סכימת מסד הנתונים מצינית מחיקות על פי היררכיה הקשרים, שאילתת מחיקה תמחק רשומות תאומות הן בלבד והן בצד רבים, בקשר גומליין מסווג יחיד-לרבבים.

אם עלייך למחוק רשומות מתוך צד היחיד בלבד, בטל את המחיקות על פי היררכיה הקשרים מתוך הגדרות שלמות הקשרים. אם עלייך לבטל את כל הרשומות בצד רבים של קשר גומליין מסווג יחיד-לרבבים בעל שלמות קשרים, כבה את קשר הגומליון, בצע את שאילתת המחיקה ולאחר מכן, שחרור את הגדרות שלמות הקשרים.

שאילותות עדכון

בניגוד לשאילותות הוספה ומחיקה, **שאילותות עדכון** (Update queries) אינן פועלת על רשומות שלמות. במקומות זאת הן מושנות ערכים בשדות שנבחרו. הקצה שורות מוגדרות באמצעות קרייטריוני הגדרה. אפשר לנצל שאילתת מסווג זה לחישוב מחירי פריטים חדשים שעולים בשיעור קבוע. אם העליה במחקרים חלה על קטגוריות מוצר אחת בלבד, ציין את הקטgorיה בהגדרת הקרייטריוניים.

סוגי שאלתה אחרים

קיימת קבוצת שאלות מיוחדת נוספת שמרחיבה את שימושיות השאלות. בדרך כלל, אם כי לא תמיד, שאלות אלו מרחיבות או משלימות את יכולת של שאלות בחירה והוספה.

שאלות פרמטר

שאלות אלו הם סוג מיוחד של שאלות שיכלולות להחזיר שורות או לבצע פעולות. בזמן ריצה, שאלת מסוימת יכולה להנחות את המשמש לספק קלט שקובע את אופן פעולה. באפשרות לבקש מהמשמש קלט אחד או יותר על ידי שימוש בסוגי נתונים שונים. מורים לשאלת הפרמטר מה עלייה לבצע על ידי הכנסת ערכים בתשובה לביקורתו שלה, או על ידי הגדרת הפרמטרים שלה באמצעות קוד VBA טרם ביצוע השאלת, כדי לשנות בקבוצת החזרה שלה או בפועל שהיא מבצעת. הדבר מאפשר לספק קוד זיהוי שלLKוקוח בזמן ריצה, כדי לקבוע את הלוקוח שהאלת**הבחירה תחזיר את פרטיו**.

בחלופה לשאלת פרמטר, היחסום יכול להפנות למחוזות SQL של שאלת בחירה או פעולה, באמצעות משתני המחרוזת. טרם תבצע את משפט SQL בפקודת ADO, הקצה ערכים מוגדרים למשתני המחרוזת. על ידי כך ניתן לקבל תוכאות גמישות יותר מאשר מקבלים בשאלת פרמטר, מכיוון באפשרותך לשנות, למעשה, פסקיות שלמותーム במשפט SQL של שאלת. במקרים מסוימים, שאלות פרמטר מפצות על כך באמצעות הקלדת נתונים ובקשות מוכילות לקבלת ערכים. בנוסף, שאלות פרמטר מבטלות את הצורך במnipולציות של מחוזות SQL, כל שעליינו לעשות הוא לעדכן את ערך הפרמטר.

שאלת איחוד

שאלת האיחוד (query union) היא סוג של שאלת SQL. זהה לשאלת מעניינת במיוחד, עקב התנהוגה החדשנית (היא מציבה טבלאות זו לאחרי זו במקומות זו לצד זו). שאלות איחוד מצרפות יחד את השדות מתוך שתי טבלאות או יותר. במקומותocrף את רשומות הטבלאות זו לצד זו, שאלות איחוד מוסיפת את הרשומות מטבלה אחת ישרות אחרי אלו של הטבלה האחרת. עליך לבנות שאלות איחוד על ידי שימוש במשמעותי SQL. לא ניתן לעצב אותן בצורה גרפית בראש עיצוב השאלת. הקלד את קוד SQL של שאלת האיחוד ישרות בתצוגת SQL של השאלת. תוכנה זו מייחדת את שאלות האיחוד מכל סוג של שאלות בהן עסקנו עד כה, מכיוון שבאפשרותך ליצור את כל שאר השאלות בצורה גרפית. שאלות איחוד מעניקות לך את מלאה העוצמה של שפת SQL להגדלת קרייטריונים ולקביעת סדרי מילון.

שאלות הצלבות

שאליתת הצלבות (crosstab query) קולוטות טבלה או שאליתה בתור מקור נתונים ומחזירה סכום, ממוצע או מונה של שדה אחד המבוסס על שני שדות קטגוריה נוספים. שאליתה מסוג זה זקופה לשדה שהיישום יכול למן, לסכם או למצע. נוסף לכך שדה המיועד להציג תוצאות צבירה, מפתחים חיברים לספק שדות קטגוריות שורה ועמודה. שדות אלה צריכים להכיל ערכים בדים שימושו בתור קטגוריות לדיווח על תוצאות צבירה. Access תומך ביצירת שאלות הצלבות באמצעות אשף ו-Jet SQL. בנוסף, אפשרות לטפל ידנית בעיצוב שאלות הצלבות בתוצאות **עיצוב** (Design) ו**גיליון נתונים** (Datasheet).

שאלות משנה

שאליתת משנה SQL היא משפט SELECT בשפת SQL שמקונן בתוך שאליתת פעולה אחרת. השתמש במשפט SELECT המקונן בתור ביטוי בקריטריון של שדה. המשפט המקונן מחזיר ערך שבבואה הזמן יכול לתקן בתור קרייטריון נוסף. למרות שעיליך כתוב את שאליתת המשנה בשפת SQL, תוכל濂ן אותה בראש עיצוב השאליטה. יתרה מזאת – באפשרות להשתמש בשאליתה נוספת נספח כדי לאשר שעיצבת את המשפט המקורי בדומה תקינה. השתמש במשפט SQL המקורי כפי שהוא בקריטריון כלשהו אחר, כדי לציין ערך או תחום ערכים עבור קבוצת החזרה.

פועלות להגדרת נתונים

שפת SQL Data Definition Language (SQL DDL) מאפשרת למפתחים ליצור טבלאות המכילות משפטים SQL. השאליטה לייצרת טבלה מנצלת את DDL כדי להפוך יצירת טבלה לתהליך אוטומטי. אחד ההיבטים המצדדים של שאליתת ייצרת טבלה הוא האפשרות לעצב אותה בדורת גרפית. מקובל מאוד להפעיל את עיצוב שאליתת ייצרת טבלה בתור שאליתת בחירה סטנדרטית. לאחר עיצוב שאליתת בחירה כך שתציג בדיקות מה שאתה מחפש, הפוך אותה לשאליתת ייצרת טבלה.

השתמש ב- DDL לעיצוב ידני של טבלאות באמצעות משפטים SQL. אםnoch לך לעבוד עם משפטים SQL, זהה הדרך המהירה והטובה ביותר לעצב טבלאות. SQL DDL מהו זה חלופה ל-ADO. קיימות Möglichkeiten נבחרות, כגון קביעת ערך הנוכחי והערך ההתחלתי של סוג הנתונים **מספר אוטומטי** (AutoNumber), או הפעלה וביטול דרישת Unicode, שנitinן לקבל בלבדית ב- Jet SQL. כמו כן, מפתחים לטביבות Access נהגים להשתמש במשפטים SQL העיקריים CommandType של אובייקט Command של ADO. אם המאפיין CommandType בתוכנית שלך קיבל את הערך adCmdText, عليك להשתמש במשפט SQL כדי להגדיר את המאפיין CommandText של Command.Execute.sql

עבודה עם מקורות נתונים מרוחקים

כשעובדים עם מקורות נתונים מרוחקים, כגון Microsoft SQL Server, כמו Oracle, MySQL, PostgreSQL ועוד, הם יכולים לשלב נתונים מרוחקים במקור נתונים ולמיוטם ביצועי השאילתת. ארבעת נתבי הגישה הבסיסיים לנתונים מרוחקים, כוללים טבלאות צמודות, שאלות מעבר (pass-through queries) OLE DB ו- ODBC Direct.

בעבודה עם Access 2000, קרוב לוודאי שתעדיף להימנע מגישה לנתונים באמצעות ADO/OLE DB. עדיף ללמוד להשתמש בספקי Microsoft Universal Data (ADO DB, מכיוון ש- ODBC Direct מהוות חלק מגישת הנתונים האוניברסלית של Microsoft). הטכנולוגיה ODBC Direct הולכת והופכת מיושנת, ו- OLE DB עתיד להחליף אותה.

שאלות מעבר SQL הן שאלות המבוססות על תחביר SQL טבעי של מנגנון מסד נתונים מרוחק. סוג שאלתה זה מאפשר לעבוד ישירות עם טבלאות בשרת המרוחק במקומות קשר אותן תחילת. העבודה שניתן להשיג תוצאה זהה בעזרת ADO עם תחביר בעל אפשרות העברת טובה יותר הופכת גם את שאלות המעבר לאמצעי מיושן.

בעודה עם טבלאות מקומיות יכולה לאפשר לישום גישה מבוססת-מטרמן לרשומות במקור נתונים מרוחק. הדבר עשוי להיות מחייב אמינותה של יישום, אך בו-זמן גם להציג שאלות שיש להן מקורות נתונים מקומיים, במיוחד כשהנתונים אינם משתנים לעיתים תכופות.

במהלך עיצוב שאלות למקורות נתונים מרוחקים, חשוב לתכנן אותן כך שתבצענה את רוב השאילתת המחשב המפעיל את מנגנון מסד הנתונים המרוחק (השרת), שביצועיו בעליים לאין שיעור על אלה של מחשב שולחני טיפוסי. בנוסף, הפעלת השאילתת בשרת מאפשרת להקטין את תעבורת הרשות ולהעביר רק חלק מהנתונים ברשות. לפניך עצות אחדות למיוטם שאלות המבוצעות על מקור נתונים מרוחק:

↳ הגבלת פונקציות צבירה SQL לאלו שמנגנון מסד הנתונים המרוחק תומך בהן (פונקציות AVG ,MIN ,SUM ,COUNT)

↳ הגדרת סוג הנעילה adLockBatchOptimistic והפעלת השיטה UpdateBatch כדי למטב את העבודה עם מקור נתונים מרוחק בצורה לא מחוברת

↳ הימנעות מהפעלת פונקציות מותאמות אישית שדורשות משאבי עיבוד מקומיים שימוש בקריטריונים המגדירים טווח קבוע, כגון בין 100 ל-1,000, לעומת זאת קריטריונים "פתוחים", כגון >100

↳ מילוי חוקים כליליים של מיוטם שאלות, מילוי, חיפוש וצירוף בשדות סדרי אינדקס, ושימוש בפרמטרים של תווים כליליים (wildcard) עם האופרטורים LIKE רק בסוף קריטריון חיפוש מחוץ



כלים עם האופרטור LIKE SQL Server תומר בטו % במקומם בתו * לחיפושים המבוססים על תוים Server היכול את Microsoft Data Engine החדש, הקפד לשלב בשאלות את התו הכללי החדש.

עיצוב שאלות באופן ידני

ישומים מותאמים אישית יכולים לבנות שאלות באמצעות קוד, או לנצל שאלות בינויו מראש בתור מקורות של רשומות. בדיק כמו בטבלאות, סביר מאוד שלעיותם תבנה שאלות באופן ידני ולעיותם – באמצעות קוד. ידע מבוסס בעקרונות עיצוב ידני של שאלתה יהיה לך לעוזר רב בבזק ללמידה עיצוב והפעלה של שאלות באמצעות קוד.

בסעיף זה נבחן לעומק בניה של שאלות באמצעות אשפים ונסקרו את התוצאה **עיצוב** (Design) של שאלתה. קל מאוד להשתמש באשפים, אך הם מגבלים במידה רבה את אפשרויות העיצוב בהשוואה לבניית שאלות באופן ידני בראש עיצוב השאלתת שבתוכות **עיצוב**. סקירת טכניקות השימוש ברשות עיצוב השאלתת כוללת: הוספה טבלאות, בחירת שורות וציוון קרייטוריונים. כאן תגלה גם כיצד להרכיב שאלות פרמטר, שאלות איחוד ושאלות משנה.

שימוש באשפים

אשפים לשאלות של Access מאפשרים ליצור שאלות בצורה קלה ומהירה. סעיף זה מתאר את ארבעת האשפים.

ASF השאלתת פשוטה

החולון **בסיס נתונים** (Database) של Access 2000 עוצב מחדש באופן שהופך את הגישה **ASF שאלתת פשוטה** (Simple Queries Wizard) לפעללה ייחודית. בחר את **שאלות** (Queries) בחולון **בסיס נתונים** ולחץ להחיצה כפולה על הסמל **יצירת שאלתת באמצעות ASF** (Create Query By Using Wizard) כדי לפתוח את האשפ. באפשרות גם להציג אל ASF זה בדרך המקובלת יותר, על ידי בחירה באפשרות **שאלות** ולהחיצה על לחץ חדש (New) בסרגל הכלים של החולון **בסיס נתונים**.

ASF השאלתת פשוטה מאפשר שלושה סוגים עיצוב שאלתת. כל רכיביו מיועדים למפתחים מתחילה. יחד עם זאת, גם מפתחים מנוסים ייוכחו שהוא מאפשר ליצור שאלות במהירות המשתווות לו של עבודה עם רשות עיצוב השאלתת. חסרונו

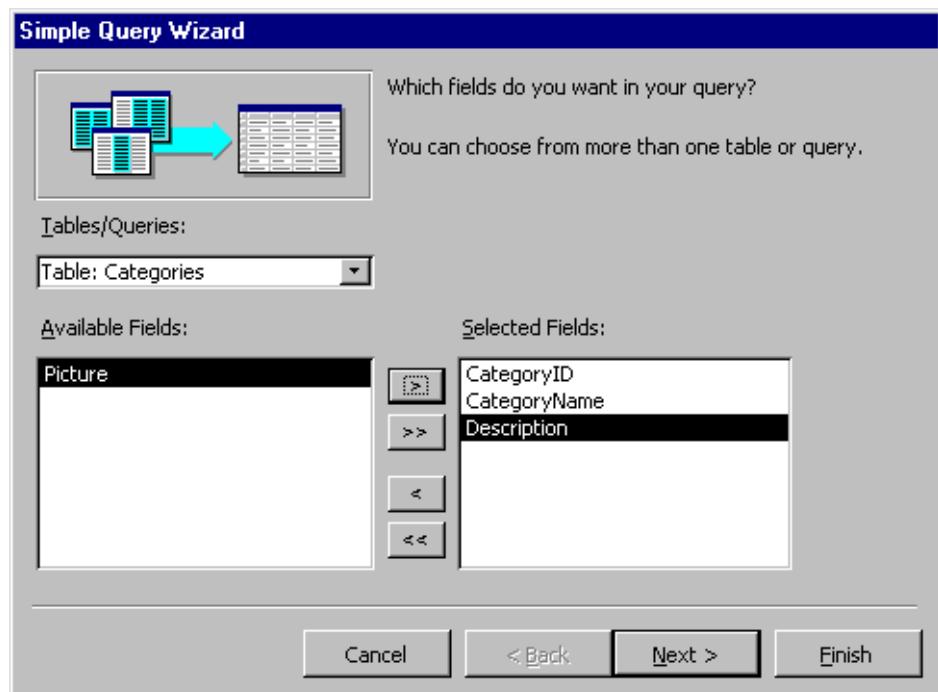
העיקרי של אשף השאלתה פשוטה הוא שאינו מאפשר את טווח עיצובי השאלה המלא שניתן להשיג באמצעות רשות עיצוב השאלה או SQL. מפתחים השולטים ב-SQL ובכלי היצור (joining rules), יחברו את הממשק הגרפי שלו.

הערה:



הYSISOM החדש Microsoft Access Project אינו כולל אשפי שאלתה כלשהם (וגם פרויקט זה יוצר קבצים מסווג *.adp.*.mdw). ברגע לקובץ מסד הנתונים *.mdb. בנוסף, עיצוב התצוגה עיצוב עבור קבץ *.adp.*. שונה מזה של קבצי *.mdb.*. עיין בסעיף האחרון בפרק זה לקבלת הסבר ראשון בעיצוב שאלתה בעזרת Access Projects. פרק 12 מרחיב בנושא העבודה עם מסדי נתונים מרוחקים ו-.Microsoft Data Engine

הדרך הבסיסית ביותר לבנות שאלתה בעזרת אשף השאלתה פשוטה (ראה תרשימים 4.1) היא לבסס אותה על טבלה או על שאלתה יחידה. הפעל את האשף באופן זה כדי להפריד בין השדות שהשאלתה מחזירה. למרות פשטות השימוש באשף, תפיק ממנו תועלת רבה, מכיוון שהגבלת קבוצת החזרה משפרת את ביצועי השאלתה.

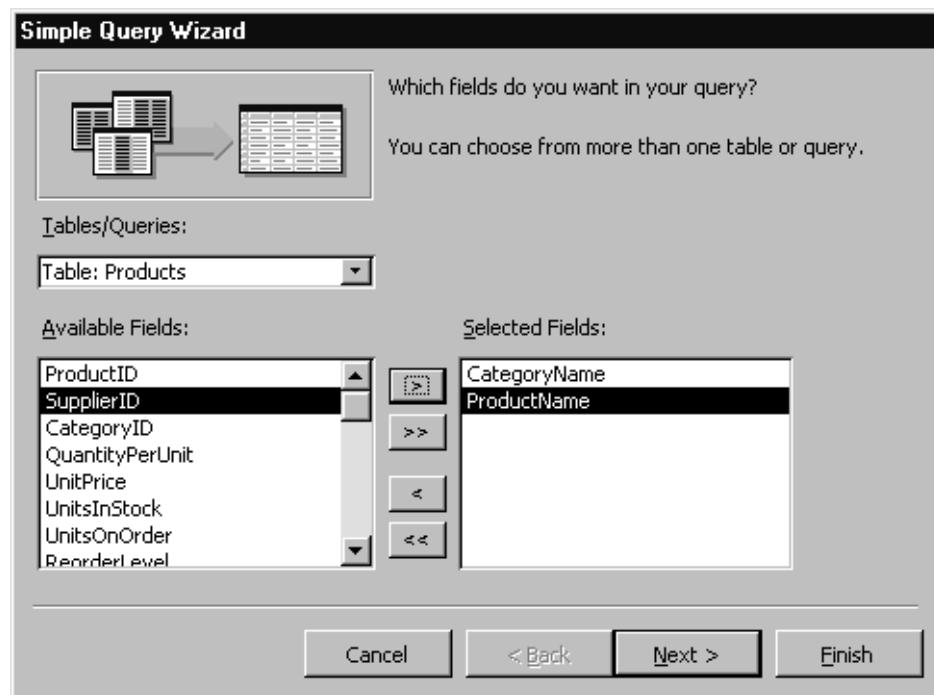


תרשים 4.1: אשף שאלתה פשוטה במהלך יצירת שאלתה המבוססת על קבוצת משנה של שדות בטבלה Categories במסד הנתונים Northwind

עוצמתו האמיתית של ash' השאלתה פשוטה באה לידי ביטוי ביכולתו לצרף שתי טבלאות או יותר, מבלתי חייב את מעצב השאלות ליצור צירופים (joins) בפועל בין הטעלאות. הדבר מאפשר למשתמש להתמקד בשדות הדרישים בשאלתה ובטבלאות שמן יילקחו השדות, מבלתי להזכיר זמן לעיצוב הנאות של שאלתה. ash' נבון במידה הדורשיה ליצור תוכאות תקין.

תרשים 4.2 מציג את ash' השאלתה פשוטה בשלבי הפיתוח של שאלתה המיצרת שתי טבלאות בסיס הנתונים Northwind. כל שוליך לעשות הוא לבחור טבלה ולאחר מכן לבחור שדות בטבלה זו ולהעבירם מתיבת הרשימה **שדות זמינים** (Available Fields) אל התיבה **שדות שנבחרו** (Selected Fields). החלון התיכון מציג קטע מקובצת **ProductName** השדה **CategoryName** מהטבלה **Categories** והשדה **ProductName** מהטבלה **Products**.

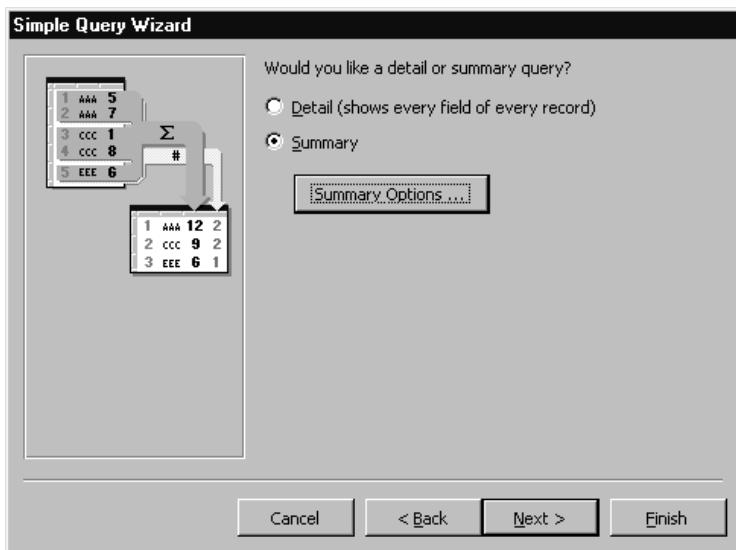
השאלתה שבתרשים 4.2 מיצרת את הטבלאות **Products**-**Categories** בשדה המשותף **CategoryID**, ואולם, בשום שלב אין צורך להגדיר זאת. ash' השאלות שלhn, **CategoryID**, וואולם, בשום שלב אין צורך להגדיר זאת. ash' השאלות המשותפות מזוהה בעצמו את השדה המשותף ומיצרת באמצעותם של השדה **CategoryID** בשתי הטבלאות. אם רשומה באחת הטבלאות מכילה ערך **CategoryID** שלא קיים ברשומה בטבלה המקביל, ash' השאלתה פשוטה יתעלם ממנו. אם היישום שולץ אמור להכיל רשומות שקיימות רק בטבלה אחרת ללא הקבלה בטבלה השנייה, יוכל לעזרך את עיצוב השאלתה מתוך ash', לעצב את השאלתה בתוצאות **עיצוב** או להרכיב אותה באמצעות משפטי SQL.



Category Name	Product Name
Beverages	Chai
Beverages	Chang
Beverages	Guaraná Fantástica
Beverages	Sasquatch Ale
Beverages	Steeleye Stout
Beverages	Côte de Blaye
Beverages	Chartreuse verte
Beverages	Ipoh Coffee
Beverages	Laughing Lumberjack Lager
Beverages	Outback Lager
Beverages	Rhönbräu Klosterbier
Beverages	Lakkaliköori
Condiments	Aniseed Syrup
Condiments	Chef Anton's Cajun Seasoning

תרשים 4.2: תיבת הדו-שיך של אשי השאלתה פשוטה ותציג גילון הנתונים המתקבלת ממנו

באפשרות גם לעצב שאלות המבצעות פעולות צבירה מסוימות בעזרת אשף השאלתה פשוטה. האשף מזוהה בעצמו אם ניתן לבצע צבירה ועושה ככל האפשר לסייע לעצב השאלתה. האשף מציג תיבת דו-שיך כמו זו שבחלקו העליון של תרשימים 4.3 רק כשניתן ליצור צבירות מסוימות. ניתן עדיין לצמצם את נתיב הצבירה על ידי בחירה באפשרות **פירוט** (Detail) בשלב השני של האשף. אם תבחר באפשרות **סיכום** (Summary), יהיה عليك גם ללחוץ על לחץ **אפשרויות סיכום** (Summary Options) כדי לציין את השדות שייצברו ואת הפונקציות שבמוצעו תבצע הצבירה. האשף מכין פעמים שדה סיכום נוסף. אם הוא עוזה זאת, מחק את השדה ברשות עיצוב השאלתה.



The image shows two windows from Microsoft Access. The top window is titled "Categories Products Count Query : Select Query". It displays two tables: "Categories" and "Products". The "Categories" table has fields: CategoryID, CategoryName, Description, and Picture. The "Products" table has fields: ProductID, ProductName, SupplierID, and CategoryID. A relationship line connects CategoryID in the Categories table to CategoryID in the Products table. Below the tables is a query grid with the following data:

Field:	CategoryName	Count Of Products: Count(*)
Table:	Categories	
Total:	Group By	Expression
Sort:		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		
or:		

The bottom window is titled "Categories Products Count Quer...". It shows the results of the query in a table format:

Category Name	Count Of Products
Beverages	12
Condiments	12
Confections	13
Dairy Products	10
Grains/Cereals	7
Meat/Poultry	6
Produce	5
Seafood	12

Record: 1 of 8

תרשים 4.3: שלב 2 של תיבת הדו-שיך אשי**פשו**טה שאילת**ה**א**ש**י**ב**ר**ה** ב**ש**א**ו**ת**ה**

שני חלונות התוצאות של תרשימים 4.3 מציגים את תוכנות העיצוב וගילוון הנתונים של השאילת**ה**. שים לב שהשאילת**ה** מונה את המוצרים בכל קטgorיה. החלון האמצעי מציג את רשות עיצוב השאילת**ה** שמחשבת את התוצאה. לעומת זאת, אשי**פשו**טה שאילת**ה** הפשו**ה**ת**ה** מבטל את הצורך לדעת היכן להציג את פונקציית הצבר**ה** ואת הבחירה שיש לבצע בשורה **סך הכל** (Total) (עליך ללחוץ על הלחצ**ה**ן **סיכום** (Totals) בסרגל הכלים כדי להציג את השורה **סך הכל** ברשות עיצוב השאילת**ה** בעת עיצוב ידני של השאילת**ה**). האשי**פשו**טה שחרר אותך מכל הפעולות האלה.

ashf_shailtut_chifush_cfiliot

צמד אשפי שאלות מסיע ליצור שאלות המבוצעות משימות שכיחות. **ashf_shailtut_chifush_cfiliot** (Find Duplicates Query) מփש בערך רשותים כדי לקבוע אם קיימות שתי רשומות או יותר המכילות ערכים זהים. האשף מוחזר את הערכים הçıולים של כל הרשותים בעלות ערכים מסוימים בשדות צוינו בקריטריוני החיפוש.

אחד הסיבות השכיחות לחיפוש כפליות היא ביטול עותקי רשומות נוספים במקור רשומה. יחד עם זאת, לא ניתן להפוך את השאלת שיצר האשף לשאלת מחלוקת, מכיוון שהיא תסיר את הרשותים המקוריים יחד עם בניית זוגן. מערכת העזרה המקורי של Access 2000 מתארת שיגרה להסרה אוטומטית של כפליות בהתאם לתוצאה שהתקבלה מאשף שאלת חיפוש כפליות. כדי להציג את השגירה, הקלד "מחק רשומות כפולות" (automatically delete duplicate) בתורו קרטירוני החיפוש בקרטיסיה חיפוש של העזרה (שם המאמר: **מחק באופן אוטומטי רשומות כפולות מטבלה**).

פתח את ההוראות הספרטניות על ידי בחירה בנושא **מחק באופן אוטומטי רשומות כפולות מטבלה**.

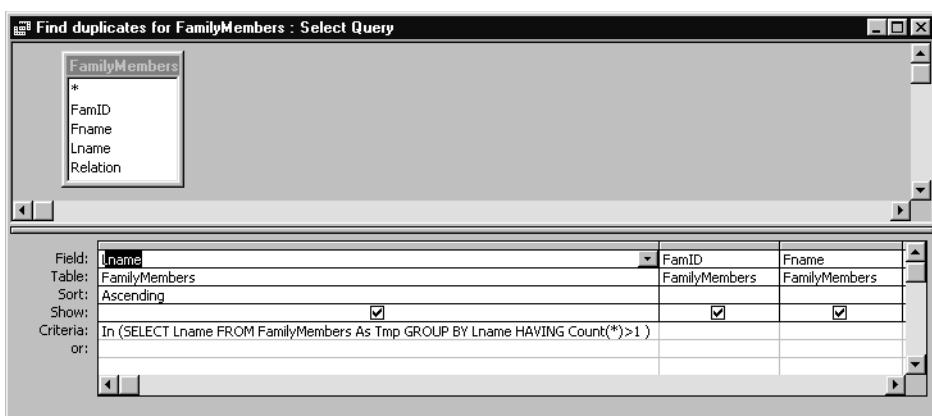
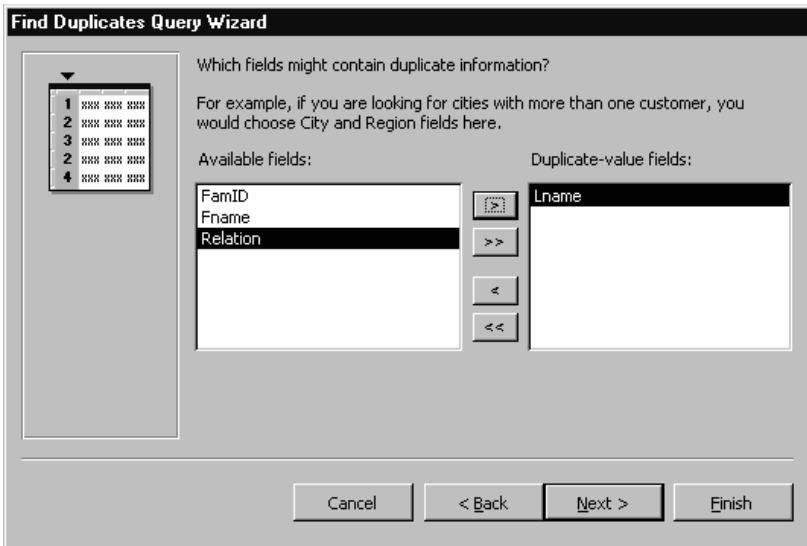
הדוגמה שלנו לאשף זה מבוססת על הטבלה FamilyMembers (עיין בחלון העליון בתרשים 4.4). תיבת הדו-שייך **ashf_shailtut_chifush_cfiliot** במרכז התרשימים, מכילה זוג תיבות רשימה לבחירת שדות בהם יש לחפש ערכים כפליים. כמו במקרה של אשף השאלת הפשוטה, מעבירים שדות מטיבת הרשימה **שדות זמינים אל התיבה שדות בעלי ערכים כפולים** (Duplicate-Value Fields). בדוגמה, נבחר השדה Lname שבאמצעותו יתבצע חיפוש אחר כפליות. כפי שנראה בחלון העליון בתרשים 4.4, כל הרשותים שבטבלה FamilyMembers ההן כפליות של ערך אחד או יותר של שדה זה. שלב 3 של האשף מאפשר ליעיד שדות נוספים שיוצגו בקבוצת ההחזרה. הדוגמה בוחרת בכל השדות הנוטרים.

החלון התיכון בתרשים 4.4 מציג את אחד הקיימים החזקים ביותר של אשפי Access: פישוט משימות מורכבות. החלון התיכון מציג גם את עיצוב רשות עיצוב השאלת שיצר האשף. שים לב שהተצוגה כוללת שאלת משנה-SQL. משפט שאלת המשנה כולל פונקציית צבירה וגם את הפסוקיות BY GROUP HAVING. זו שאלת המשנה בשורת הקרטירון של רשות העיצוב שמשזהה ערכים כפליים בערך הרשותים של המקור. באפשרות לכוון את עיצוב השאלה שנוצרת על ידי הוספה או הסרה של שדות, כולל שדות מקור חדשים או חליפיים, או אפילו על ידי שימוש בקריטריונים נוספים להסרת כפליות מסוימות.

FamilyMembers : Table

	FamID	Fname	Lname	Relation
▶	1	Rick	Dobson	Me
	2	Virginia	Dobson	wife
	3	Glen	Hill	son
	4	Tony	Hill	son
*	5	Shelly	Hill	daughter-in-law

Record: [◀] [◀] 1 [▶] [▶] [▶*] of 5

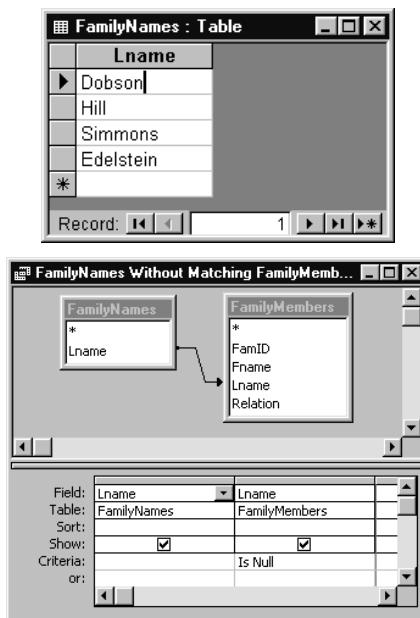


תרשים 4.4: אוסף שאלות חיפוש כפליות. החלון העליון מציג את מקור רשומות הקלט. תיבת הדיו-שיכון המרכזית מציגה בחירת שדה שלפיו יבוצע חיפוש הcpfליות. החלון התחתון מציג את השאלה שנוצרה בתצוגת עיצוב

ashf_shailat_chiposh_shel_la_towamim

ashf_shailat_chiposh_shel_la_towamim (Find Unmatched Query Wizard) הוא רכיב נוסף התומך במתלה ניהול מקובלת של מסדי נתונים. האשף מחויר רשומות בטבלה אחת שאין להן רשומות תואמות בטבלה מקבילה. נוח במיוחד לנחל באמצעות טבלאות שאין להן שלמות קשרים, אך קיים ביןיהם קשר גומלי מסווג ייחיד לרבים. במצב זה, באפשרותך להסיר, בשוגג או במתכוון, את צד היחיד או את צד הרבים של קשר הגומלי. כאשר לאחד מצידי הקשר, היחיד או הרבים, אין בן-זוג בטבלה המקבילה, ashf_shailat_chiposh_shel_la_towamim יגלה זאת. באופן טיפוסי תרצה להסיר רשומות אלו ממקור הרשומות. אין כל בעיה – כל שעליק לעשות הוא להפוך את השאלתה שנוצרה לשאלת חיקוי.

השאלתה שבתרשים 4.5 מחפשת רשומות לא תואמות בטבלאות FamilyNames ו-FamilyMembers. ההשוואה מתבצעת על השדה Lname בשתי הטבלאות. השאלה מצפה את שתי הטבלאות על ידי אילוץ כל הרשומות מהטבלה FamilyNames לתוך FamilyMembers, כמו צוין בחז ההכוונה מהטבלה FamilyNames אל הטבלה FamilyMembers. האשאלת, כמו צוין בחז ההכוונה מהטבלה FamilyNames אל הטבלה FamilyMembers, מחייב ששלש ערכי null בשדה Lname בטבלה FamilyMembers. אם ערך הרשומה של Lname בצד FamilyMembers של קבוצת החזרה הוא null, לצד הנאות של האשאלת זו מחייב הבנה של הקשרים ושל ערכי null. התעמקות בעיצוב שאלות כגון זו ובשאלות שמחזירות ערכים כפולים עשויה לחדר את מיזוגיותם בעיצוב שאלות, במקרה שאינך רגיל לכתוב שאלות כאלה דרך שיגרת.



תרשים 4.5: ashf_shailat_chiposh_shel_la_towamim. מקור הרשומות השני הוא הטבלה .FamilyMembers. קריטריון ההתאמה הוא השדה Lname.

ashf_shailta_mozelbat

מנת hei תמייה בהחלטות מגליים לפעמים ששאלות הצלבות (crosstab queries) מסייעות להגעה לתובנות חדשות אודוט אובייקטיבים ותהליכיים. תרשימים 4.6 מציג טבלת קלט של ashf_shailta_mozelbat ואת קבוצת ההחזרה שהוא יוצר ממנה. תצוגת קבוצת ההחזרה טופלה בתוצאות עיצוב (Datasheet) וגיילוֹן נתוניָם (Design).

שאלות הצלבות שבגיילוֹן הנתונים התחתון מוניה את FamID לפי השדות Relation ו-Uname של גיילוֹן הנתונים העליון. ה"טייפול" שהענקי את תצוגת העיצוב מכובן את מיון השורות לפי Relation, מסדר עולה לירוד. בתצוגה גיילוֹן נתוניָם, גראתי את העמודה המסומנת בתור ID Total of FamID כך שבמקום שתהיה העמודה השנייה, היא מעתה העמודה האחורונה. אחרת, העיצוב היה אוטומטי. לא היה צורך להכניס שמות שדות ומילוט מפתח בראש עיצוב השאלתת, וגם לא היה צורך כתוב קוד SQL.

FamilyMembers : Table				
	FamID	Fname	Lname	Relation
▶	1	Rick	Dobson	Me
	2	Virginia	Dobson	wife
	3	Glen	Hill	son
	4	Tony	Hill	son
*	5	Shelly	Hill	daughter-in-law

FamilyMembers_Crosstab : Crosstab Query			
Relation	Dobson	Hill	Total Of FamID
▶ wife	1		1
son		2	2
Me	1		1
daughter-in-law		1	1

תרשימים 4.6: שאלתת שנוצרה באמצעות ashf_shailta_mozelbat של Access 2000. גיילוֹן הנתונים העליון מציג את הקלט של קבוצת ההחזרה של השאלתת שמופיעה בגיילוֹן הנתונים התחתון

שתי טכנולוגיות ניתוח חדשנות נוטלות את הבכורה משאלות הצלבות. הראשונה, טבלת ציר (pivot table) יכולה לחשב תוצאות הצלבה פשוטות מותווך מקור ורשותה, אך באפשרותה גם להפעיל טיפול דינמי נתונים לאחר חישוב הצלבה. הדבר מאפשר למנת hei החלטות לתקשר באופן דינמי עם תוצאות הניתנות שלהם. הופך טבלת ציר לזרימה באמצעות הטפסים ודף הגישה לנתונים שלו (Data Access Pages), אך ציר הוא הכליל הראשי של לביצוע ניתוח טבלת ציר. הטכנולוגיה Microsoft Excel

השניה, מסד נתונים עיבוד ניתוחי מקוון – OLAP (Online Analytical Processing) מאפשר ניתוח נתונים דינמי מול מקרים נתונים גדולים מאוד. טכנולוגיה זו מחייבת שימוש בשרת מסד נתונים מרוחק (אחד או יותר), כגון SQL Server וזכרן OLAP עבור הנתונים. בקרב הרכיבים המקבילים של Excel 2000, Office 2000 הוא המשאב המועד לתפקיד זה.

שימוש בתצוגה עיצוב

אשי השאליות מאפשרים לעצב שאלות בלי תלות בرمת השילטה בתחריר או בסוגיות העיצוב, אך הם מוצאים את היקף האפשרויות שניתן להציג באמצעות שאלות. כפי שקרה בעת השימוש באפשרים, לעתים הם יוצרים עיצוב מורכב יותר מזה הדרוש בפועל כדי למלא משימה מסוימת. הדבר עלול לסבך את תהליך ערכית שינויים קטנים בשאלתה. אם עצב את כל השאלה בעצמך, קרוב לוודאי שתיזכר בלוגיקה הנכונה ותצליח לעורך בה שינויים בהמשך ללא קושי מיוחד.

הוספת טבלאות ושאלות

באפשרות ליצור עצמאלי שאלות מותאמות אישית על ידי הוספת טבלאות או שאלות אחרות לרשות עיצוב השאלה. אם תבחר בדף זו, בדרך כלל לא תऋ לכתוב קוד SQL כדי ליצור שאלות ולערוך אותן. פתח את רשות עיצוב השאלה על ידי לחיצה כפולה על **יצירת שאלה בתצוגה עיצוב** (Create Query In Design View). לאחר בקבוק האובייקטים **שאלות** (Queries) שבחלון **מסד נתונים** (Database). לאחר מכן, הוסף טבלה או שאלה אחת או יותר, או צירוף כלשהו של טבלאות ושאלות בראש רשות העיצוב. בחר טבלה או שאלה ולחץ על **הוסף** (Add) כדי להעביר אותה מຕיבת הדו-שים **מצגת טבלה** (Show Table) לראש רשות העיצוב.

לאחר שהוספה טבלה לראש התצוגה עיצוב, عليك לבחור שדות שיהו חלק מה שאלה בראש התchapונה. קיימות לפחות שלוש דרכים להוספה שדות לשאלתך. דרך אחת – גירירה וחרור של שדה מתיבת הרשימה שמייצגת את הטבלה בחלק העליון אל השורה **שדה** (Field) בראש התchapונה. אם תגרור ותשחרר שדה בקצת הימני העליון של עמודה שכבר מאוכלסת נתונים, יעברו אוטומטית את העמודות הנותרות לצד שמאל (בממשק עברי). דרך שנייה – באפשרותך להזין לחיצה כפולה על ערך בראשית השדות. על ידי כך תעתק את השדה לעמודה הפנوية הראשונה בראשת. דרך שלישית – ניתן לבחור שדה מຕוק תיבה נפתחת בשורה **שדה** של כל אחת מהעמודות. אם חלקה העליון של הרשות מכליף יותר מקור רשות אחד, הרשימה הנפתחת של השורה **שדה** תמספר שדות שקובצו לפי מקור רשותות.

באפשרותך לפעול בכל אחת משלוש הדרכים בטיפול בטבלה בודדת כדי לציין עמודות אחדות בטבלה המכילה עמודות רבות. תרשימים 4.7 מציג את התצוגות **עיצוב** ו**גיליון נתונים** (Datasheet) של שאלה המבוססת על הטבלה Products בדוגמה. השאלה מצינית שלושה שדות מתוך הטבלה.

השורה **sort** בעמודה **ProductName** בתרשים 4.7 נקבעה בהתאם **סדר עולה** (Ascending). השורה זו מכילה רשימה נפתחת ובה שולש אפשרויות: **סדר עולה**, **סדר יורד** (Descending) ו**לא מופיע** (Not Sorted). אם תבחר באפשרות האחורונה, התא יופיע כשהוא ריק בראשת נתונים העמודה לא יהיה ממוקמים בעת הצגת השאלה בתצוגת **גיליון נתונים**. בחירה באפשרות **סדר עולה** בעמודה **ProductName** גורמת לרשומות **ProductName** להופיע בסדר אלפבטי בהתאם ל-**ProductName**. אם שורת המילוי של ריקה, גליון הנתונים בחלון התחתון של תרשימים 4.7 יופיע ממוקן לפי המפתח הראשי של הטבלה, **ProductID**.

The screenshot shows two windows related to a query named "qryProducts3Cols".

qryProducts3Cols : Select Query (Design View):

- Table:** Products
- Fields:** ProductID, ProductName, Discontinued
- Sort:** Ascending
- Show:** checked for all three fields

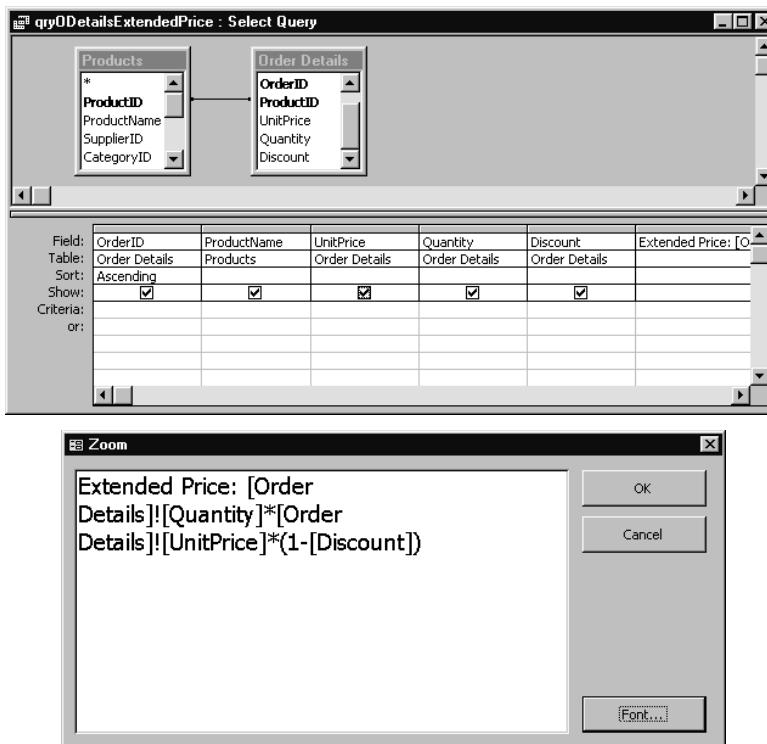
qryProducts3Cols : Select Query (Preview View):

Product ID	Product Name	Discontinued
17	Alice Mutton	<input checked="" type="checkbox"/>
3	Aniseed Syrup	<input type="checkbox"/>
40	Boston Crab Meat	<input type="checkbox"/>
60	Camembert Pierrot	<input type="checkbox"/>
18	Carnarvon Tigers	<input type="checkbox"/>
1	Chai	<input type="checkbox"/>
2	Chang	<input type="checkbox"/>
39	Chartreuse verte	<input type="checkbox"/>
4	Chef Anton's Cajun Seasoning	<input type="checkbox"/>
5	Chef Anton's Gumbo Mix	<input checked="" type="checkbox"/>
48	Chocolade	<input type="checkbox"/>
38	Côte de Blaye	<input type="checkbox"/>
58	Escargots de Bourgogne	<input type="checkbox"/>
52	Filo Mix	<input type="checkbox"/>

תרשים 4.7: החילון העליון מציג את תצוגת השאלה של שאלתה שנוצרה באופן ידני וمبוססת על הטבלה Products. החילון התחתון הוא גליון הנתונים שהתקבל כתוצאה מהשאלתה

קל ופיטו להוסיף מוקור רשותות אחד ל视窗 שאלתה. באפשרותך לבחור מוקור רשותות קלט מתוך טבלאות ושאלות אחרות. מומלץ מאוד לצרף מוקורות רשותות (אם Access אין מצרך אותן בעצמו בהתאם לקשרי הגומלין המצוינים בחולן **קשרי גומליין**). כשלון צירוף הטבלאות יגרום לכך שגיליוון הנתונים שיתקבל יציג את התוצאה הרצויה של שתי הטבלאות (כל השירותים הסדריים האפשריים של שורות מהטבלה הראשונה עם כל השירותים מהטבלה השנייה). זו תהיה לעתים קרובות טבלה בעלת מספר שורות גדול במידה מופרזת.

תרשים 4.8 מציג את הטבלאות Order Details ו-Products מצורפות באמצעות השדה ProductID. באפשרותך ליצור צירוף בריית מחדל כמו זה על ידי גיררת שדה מטבלה אחת ובחירהו בשדה התואם בטבלה השנייה. כך תקבל גיליוון נתוניים עם שורה אחת לכל שדה תאום בשתי הטבלאות. אם רשותה כלשהי מופיעה בטבלה אחת בלבד, היא לא תיראה בגיליוון הנתוניים של השאלה. קיימים שני סוגי צירוף נוספים. סוג אחד מצירף את כל השירותים מהטבלה השמאלית, Products, והאחר מצירף את כל השירותים מהטבלה הימנית Order Details. באפשרותך לבחור אחד מסוגי הצירוף הללו על ידי לחיצה ימנית על קו הצירוף ובחירה סוג הצירוף המבוקש לאחר מכן.



תרשים 4.8: החלון העליון מציג את עיצוב השאלה של שאלתה שנוצרה באופן ידני בהתאם לשתי הטבלאות, Products ו- Order Details. החלון התיכון מציג את תיבת הדוח-שיה שנה מרחק מהתצוגה (Zoom) המציגת את הנוסחה של שדה מחושב בשאלתה

השורה **טבלה** (Table) בתרשים 4.8 מציגה ארבע עמודות ששייכות לatable. productName, Order Details, שיכת לטבלה productName. השדה האחרון של עיצוב השאלתה אינו נראה בשלמותו בחלון העליון של התרשים.

התייחשות לגילוונות נתוניים משניים

אחד החידושים שהונגו ביישום Access 2000 הוא יכולת להתייחס בצורה היררכית לטבלה או לשאלתה אחת מתוך שאילתתה אחרת. במובן מסוים אפשר לומר שלושאלתה אם יש טבלה או שאילתתה בת.

תרשים 4.9 מציג גילוון נתוניים משני בתוך גילוון נתוניים של שאילתתה. השאילתת האם נקראת qryCategories. סימן לב סימן החיבור (+) בעמודה הראשונה של השאילתת. לחיצה על אחד הסימנים הללו, למשל הקטגוריה **Meat/Poultry**, פותחת גילוון נתוניים בן וחותמת את סימן החיבור לסימן חיסור (-). הסימן המתחולף מתפרק כמתג דו-מצבי לפתיחה וסגירה של גילוון נתוניים שני הקשור לשורה כלשהי בשאלתת האם. גילוון הנתוניים המשני (הבן) שבתרשים מציג את שדות המוצר עבור שורה אחת של גילוון הנתוניים האב שלו. ככלית ניתן לומר שgilioon הנתוניים הבן מפנה לשאלתת או לטבלה נוספת שמקושרת לשורה הנוכחית. באפשרות לפתח בו-זמנית גילוונות נתוניים בנימר רבים. למעשה של דבר, מאפיין של שאילתת מאפשר להשאיר את כל גילוונות הנתוניים הבנים פתוחים כברירת מחדל.

qryCategories : Select Query

Category ID	Category Name	Description	Picture
1	Beverages	Soft drinks, coffees, teas, beers, and ales	Bitmap Image
2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	Bitmap Image
3	Confections	Desserts, candies, and sweet breads	Bitmap Image
4	Dairy Products	Cheeses	Bitmap Image
5	Grains/Cereals	Breads, crackers, pasta, and cereal	Bitmap Image
6	Meat/Poultry	Prepared meats	Bitmap Image

Product ID	Product Name	Supplier	Category	Quantity Per Unit	Unit Price
9	Mishi Kobe Niku	4	Meat/Poultry	18 - 500 g pkgs.	\$97.00
17	Alice Mutton	7	Meat/Poultry	20 - 1 kg tins	\$39.00
29	Thüringer Rostbratwurst	12	Meat/Poultry	50 bags x 30 sausgs.	\$123.79
53	Perth Pasties	24	Meat/Poultry	48 pieces	\$32.80
54	Tourtière	25	Meat/Poultry	16 pies	\$7.45
55	Pâté chinois	25	Meat/Poultry	24 boxes x 2 pies	\$24.00
*	(AutoNumber)		Meat/Poultry		\$0.00

Record: [Navigation Buttons] 1 [Navigation Buttons] of 6 [Navigation Buttons]

Product ID	Product Name	Supplier	Category	Quantity Per Unit	Unit Price
7	Produce		Dried fruit and bean curd		Bitmap Image
8	Seafood		Seaweed and fish		Bitmap Image

Record: [Navigation Buttons] 1 [Navigation Buttons] of 8 [Navigation Buttons]

תרשים 4.9: שאילתת אם, qryCategories, מציגה את גילוון הנתוניים המשני שלה עבור הקטgorיה **Meat/Poultry**

חמשה מאפייני שאילתת חדשים שלטטים במראה ובתקופוד של גילוונות נתוניים משניים. המאפיין **שם גילוון נתוניים שני** (Subdatasheet Name) מצביע על טבלה או שאילתת המכילות את נתוני הבן. כדי לגשת למאפיין זה או לארכעות האחרים,فتح את תיבת הדוח-שיכון **מאפיינים** (Properties) בתצוגת **עיצוב** של השאילתת. חמשת

המאפיינים החדשניים מופיעים בתחתית תיבת הדו-שיח. תיבת המאפיין **שם גיליון נתוני משני** מכילה תיבה נפתחת שmaps את תחילך הבחירה של שאלתה או טבלה אחרת בתורן מקור רשותם עבור גיליון נתוני משני. במקרה של השאלתה שבתרשים 4.9, שם גיליון נתוני משני הוא הטבלה MyProducts. הכנס שמות שדות עבור המאפיינים **קישור שדות צאצא (Link Child Fields)** ו **קישור שדות מראשי Link** (Master Fields) כדי לסכמו את שאלתת האם עם שאלתת הבן או טבלת הבן. כמו במקרה של טופס עיקרי עם טופס שני, שמות השדות אינם חייבים להיות זהים, אך סוג הנתונים שלהם צריך להיות זהה. המאפיין גובה גיליון נתוני משני (Subdatasheet Height) שולט בגובה גיליון הנתונים המשני בתצוגת גיליון נתוני משן (Subdatasheet) גיליון האם. ואחרון חביב, המאפיין **gilioen נתוני משני מורחב Expanded** מקבל את הערכיהם כן (Yes) ולא (No). תיבת הדו-שיח **מאפייני שאלתה** (Query Properties) כוללת רשימה נפתחת להכנת אחד משני ערכים אלה.

Access 2000 תומך בגלגולות נתונים לצאים עבור גליונות הורים המבוססים על טבלאות כמו גם על שאלות. לעומת זאת, מגנוון התמיכה בגלגולות נתונים משניים עבור שאלות הורים שונה במקצת עבור טבלאות הורים. בטבלה אם ניתן לקבוע את קשר הגומלין בין טבלה אם והטבלה או השאלה הבית שלה בחلون **קשרי גומליין Relationships**. אם תבחר דока להשתמש בתיבת הדו-שיח **מאפייני טבלה Table Properties**, عليك לפתוח את הטבלה בתצוגת **עיצוב**. לאחר מכן, عليك לפתח בפורש את תיבת הדו-שיח **מאפייני טבלה** על ידי לחיצה ימנית בחלון העליון של הרשת ובחירה **במאפיינים Properties**). פעולה זו מאפשרת להשתמש באופןם חמשת המאפיינים כמו עבור שאלתך כדי לקבוע את קשרי הגומליין בין טבלה אם לגיליון הנתונים המשני שלה.

שימוש בשדות מחושבים

באפשרותך ליצור שדה מחושב הגוזר את ערכו מתוך שדה אחד או יותר של שאלתך. חשב שדות אלו באמצעות אופרטורים חישוביים או אופרטורי מחרוזות, פונקציות מוכללות או פונקציות מותאמות אישית.

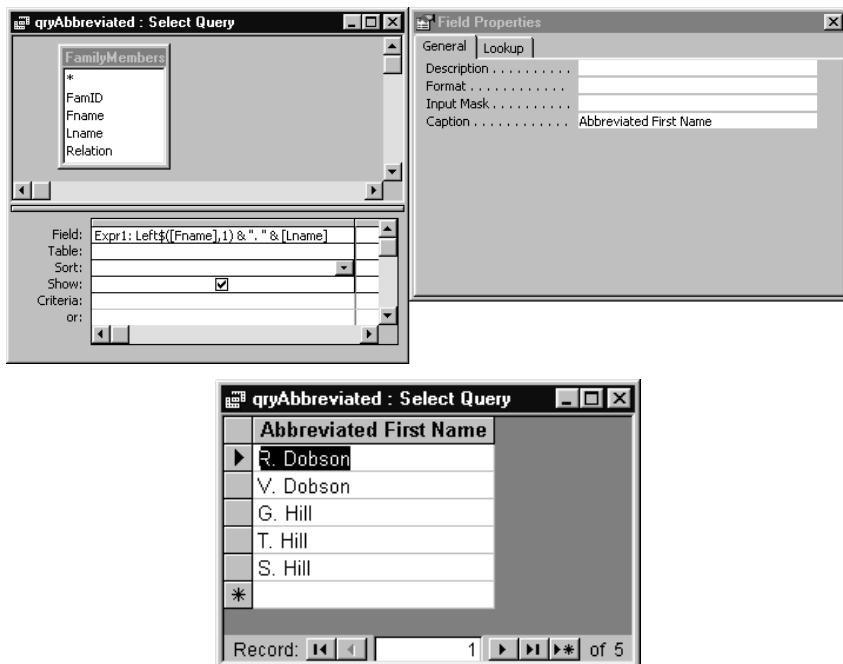
שדות מחושבים אינם ניתנים לעדכו באופן ישיר, מכיוון ש-Access מאחסן רק את הביטוי של משפט SQL ולא את תוכנת החישוב. אפשרות לשנות שדות מחושבים על ידי שימוש הקלטים לביטוי החישוב. למروת שדות מחושבים אינם ניתנים לעדכו, היישום יכול לצבור אותם לאורך הרשומות.

הגדרת שדות בתור ביטויים בשאלתך טומנת כמה יתרונות. ראשית, לאחר שביוטו יכול להציג נתונים מבלי לאחסן אותם בפועל, שימוש בביטוי עבור שדה מאפשר לשאלתך להציג תוצאה ללא צריכת שטח אחסון. שנית, ביטויים מחושבים מחדש באופן אוטומטי. כך הם יכולים לשחק את הנתונים העדכנים ביותר ללא התערבות כלשהי של מנהל מסד הנתונים. שלישית, קיים מגוון אפשרויות להכללת ביטויים בתור שדות. לימודי האפשרויות הללו יגדיל את הבנתך הכלכלת בעבודה עם Access.

אופרטורים חשבוניים תרשימים 4.8 מציג שדה מחושב שמחסב מחיר סיכון לשורה. ביטוי מחיר הסיכון לשורה מtabבוס על אופרטור חשבוני פשוט, כגון כפל וחילוק. שים לב שהשורה **טבלה** (Table) של השדה המוחשב בחולון העליון של תרשימים 4.8 ריקה.משמעות הדבר היא שהשדה אינו תופס שטח אחסון כלשהו. השדה המוחשב שבתרשים 4.8 משתמש בתוויות שדה לייצוג תוויות בעל משמעות של השדה בתצוגת **גיליון נתונים**.

פונקציות מוכללות תרשימים 4.10 מציג שדה מחושב שמתבסס על פונקציות מוכללות. הביטוי מורכב מחרוזת, ולכן באפשרותו להשתמש בפונקציה `Left$`. השימוש בפונקציה זו חוסך זיכרונו בהשוואה לפונקציה הכללית `Left` שמחזירה מחרוזת מסווג הנתונים שלה משתנה.

הביטוי בתרשימים 4.10 משרשר שלוש מחרוזות נפרדות. ראשית, הוא מażזר את האות הראשונה בשמו הפרטיא של בן המשפחה באמצעות הפונקציה `Left$`. שנית, הוא מצרף את המחרוזת ". ". כדי להוסיף נקודה ורווח לאחר האות הראשונה של שם האיש. הביטוי מסתiensים בצירוף שם המשפחה. לשדה מחושב זה יש עדין תוויות ברירת מחדל, אך תצוגת **גיליון נתונים** מציגה את העמודה עם כתורת שונה, תיאורית יותר. תיבת הדיו-שיך הימנית העליונה בתרשימים 4.10 מלמדת שניתן להשיג תוצאה זו על ידי מתן ערך לכיתוב של השדה. ערך כלשהו בתיבת מאפיין הכתיבה יחליף בתצוגה את שם ברירת המחדל של העמודה.



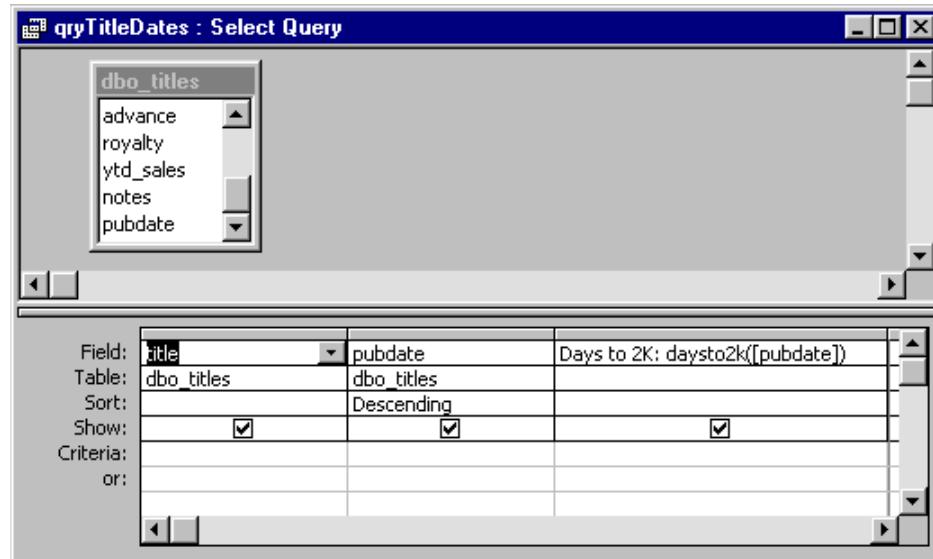
תרשים 4.10: שלושת החלונות שבתרשים חושפים את כללי הת לחבר והתוויות החלים על הוספה ביטוי מחרוזת לשאלתה, במטרה להגדיר את השדה היחיד שלה. החלון התיכון מציג את קבוצת ההחזרה מתוך השאלה.

ניתן לפשט את תהליך השימוש בפונקציות מוכללות ולהפנות לערכי טבלה, שאילתה וטופס, באמצעות תיבת הדו-שיך **בונה ביטויים** (Expression Builder). אם נמצא לנוח שיפור את הביטוי לאחר הוספתו באמצעות בונה הביטויים או באופן ידני, השימוש בתיבת הדו-שיך **שנה מרחק ממצואה** (Zoom). באפשרות לעשות זאת על ידי לחיצה ימנית בשורה **שזה** (Field) ובחרה באפשרות **מרחק ממצואה**. תיפתח תיבת הדו-שיך **שנה מרחק ממצואה** ובה הביטוי שכתבת. שנה את הביטוי לפי הצורך בטרם תלץ על אישור כדי לסגור את תיבת הדו-שיך ולהכנס את הביטוי שערכת לשאילתת. החלון התיכון שבתרשים 4.8 הוא תיבת שינוי המרחק ממצואה שמצויה שדה מחושב מתוך שאילתת.

פונקציות מותאמות אישית נוסף לשימוש באופרטורים ובפונקציות מוכללות, באפשרותך לפתוח בעצמך פונקציות מותאמות אישית שיחזרו ערכי שדה.TRS 4.11 מציג שאילתת המקורית לטבלה dbo_titles במדד הנתונים Pubs של SQL Server

השאילתת מציגה את השדות title ו-pubdate של כל אחת מהרשומות. בנוסף, היא כוללת שדה מחושב שקורא לפונקציה המותאמת אישית daysto2K. הביטוי מעביר את ערך השדה pubdate אל הפונקציה. הפונקציה daysto2K מופיעה מעל התרשים. אם תשווה את שם הפונקציה שלפניך עם עיצוב השאילתת שבתרשים 4.11, תיווכח כי הפניות לפונקציה בשאילתת אינן תלויות ברישיות (case sensitive).

```
Public Function DaysTo2K(InDate As Date) As Integer
    DaysTo2K = DateDiff("d", InDate, #1/1/2000#)
End Function
```

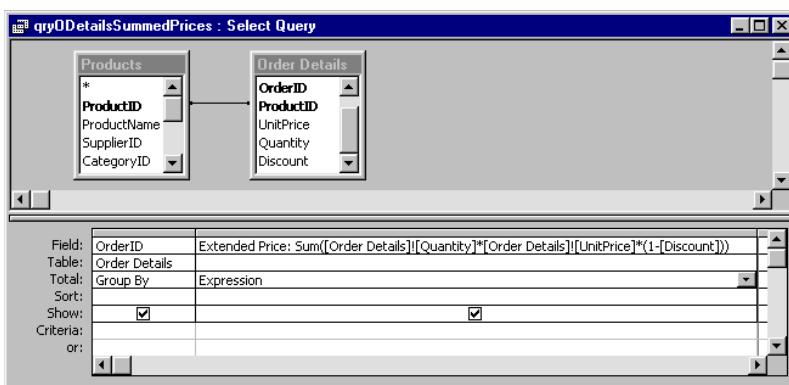


תרשים 4.11: השאילתת שלפניך משתמשת בפונקציה מותאמת אישית לחישוב ההפרש בין ערך שדה בשאילתת לבין התאריך 1 בינואר, 2000

הfonקציה קולעת את ערך השדה update בתור ארגומנט ומחזירה ערך מסווג Integer אל השאלתה. פועלה זו מאפשרת יישור אוטומטי לימיין (אם לא מצינים את סוג הנתון המוחזר בתור Integer), הfonקציה תחזיר משתנה (variant) לשאלתה שיגורם לישור לשמאל). הfonקציה המותאמת אישית מכילה שורה בודדת. היא מיישמת את הfonקציה המוכללת DateDiff כדי לחשב את ההפרש בימים בין השדה update לבין התאריך 1 בינוואר, 2000.

יש בעיה ביחסים פונקציות מותאמות אישית בBITS וBITS של שדות שאילתת, בעיקר עם מסדי נתונים גדולים ומרוחקים. על היחסים להשתנות בזמן שטבלה גדולה ווברת בKOYOI התקשרות מהשרת המרוחק אל תחנת העבודה מקומית. בנוסף, לא ניתן לנצל את עצמת החישוב של השאילתת במחשב השירות המרוחק. במצבים בהם הכרחי להפעיל פונקציה מותאמת אישית, הרכיב השאילתת שמחלצת את מספר הרשומות המינימלי מהשרת המרוחק ומבצעת את כל שאר הפונקציות בשרת זה. לאחר מכן, החל את הפונקציה המותאמת אישית על קבוצת החזרה המוצמצמת שהתקבלה מהשרת. הסטייגוויות אלו מפונקציות מותאמות אישית אין נוגעות לדוגמה שבתרשים 4.11, מכיוון שהטבלה dbo_titles קטנה מאוד. פונקציות מותאמות אישית פועלות היטב עם מסדי נתונים מבוססי-Jet בכל גודל.

פונקציות צבירה SQL באפשרות לחשב ערך ללא שימוש BITS כלשהו. אחת הדרכים לעשות זאת היא על ידי לחיצה על חצן **סיכום** (Totals) – הלחצן הנושא את האות סיגמא (Σ) – בסרגל הכלים **עיצוב שאילתת** (Query Design). לאחר מכן גורר שדה אחד או יותר אל רשות עיצוב השאילתת ובחר באפשרות **קיבוץ לפי** (Group By) בשורה **סכום הכל** (Total). חביב להיות לפחות שדה אחד בעל ערך מספרי שהשורה **סכום הכל** שלו אינה מוגדרת בתור **קיבוץ לפי**. באפשרות ליעיד שדה מחושב בתור שדה הערך המספרי. בשורה **סכום הכל** בחר ב-Expression. הכנס פונקציית צבירה SQL לשדה המוחשב כדי למנות, לסקם או לחשב ממוצע בשדה המוחשב עבור כל שילוב ייחודי של שדות שהשורה **סכום הכל** שלחן מוגדרת בתור **קיבוץ לפי** במקור הרשומות.



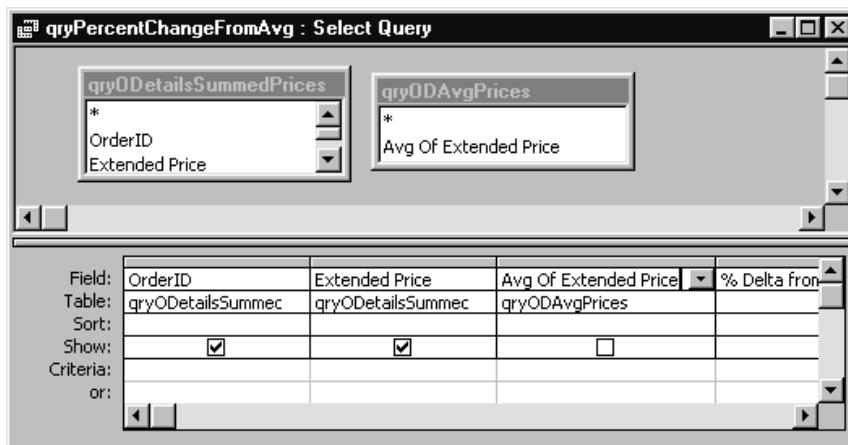
תרשים 4.12: הביטוי בשדה המוחשב השאילתת זו מחשב את מחיר הסיכום לכל שורת פריט בכל הזמן. מילת המפתח **קיבוץ לפי** בשורה **סכום הכל** מאלצת את פונקציית הצבירה שבBITS לחשב את סכום הערכיהם על פני שורות הפריטים בכל הזמן בנפרד

תרשים 4.12 מציג תצוגת עיצוב שמסכמת את מחיר הסיכום (סה"כ) לפי השדה OrderID. השאלה מתחשבת את מחיר הסיכום לכל שורה פריט לפני סיכום וקובץ של השורות לפי OrderID. מילת המפתח **קיובץ** לפי מופיעה בשורה **סך הכל** של העמודה OrderID. הפונקציה **Sum** בביטוי Extended Price מצינית שהשאילתת מסכם את מחיר הסיכום עבור כל הזמן.

חישוב ערך מתוק כמה שאלות יש עניין רב בתוצאות סטטיסטיות שימושות שני פרטי מידע או יותר. חשוב לדעת את שוויה הכספי של הזמן, אך לעתים מעוניינים גם להשווות מחיר הזמן למחיר ממוצע ולדעת באיזו מידת הוא גבוה או נמוך ממנו. Access אינו מאפשר לחשב תוצאה כזו בשאליתהבודדת. הדוגמה שביקשה זה דרשת שלוש שאלות, כאשר מהן מכילה תוצאה קרוטזית של שתי השאלות האחרות. השאלתה הראשונה היא זו המוצגת בתרשים 4.12. השאלתה השנייה מחשבת את המחיר הממוצע של כל הזמן. שאלתה זו כוללת פונקציית צבירה פשוטה **MSG** AVG הפועלת על מחיר הסיכום עבור השאליטה שתרשים 4.12. החלון העליון בתרשים 4.13 מדגים תוצאה קרוטזית של השאליטה שמסכמת את מחירי הסיכום לפי הזמן (**qryODetailsSummedPrice**) ושל השאליטה שמחשבת את מחיר הסיכום הממוצע בכל הזמן (**qryODAvgPrices**). כזכור, הדבר לחשב תוצאה קרוטזית היא לכלול את שתי השאלות בתוצאות העיצוב מבלי לצרף אותן. החלון התיכון בתרשים 4.13 מציג את השינוי באחוזים בין הסה"כ של כל הזמן לבין הממוצע, בכל הזמן. אם הסה"כ גדול מהממוצע, ערך האחוז חיובי, אחרת – ערכו שלילי.

הנוסחה שמחשבת את שינוי האחוז מהממוצע אינה מוצגת בתרשים 4.13. הביטוי הוא לא יותר מיחס בין שני מספרים. המונה הוא הפרש בין מחיר הסיכום בהזמנה לבין המחיר הממוצע, של כל הזמן. המכנה הוא מחיר הסיכום הממוצע של כל הזמן. מאפיין הבנית העיצוב של שדה מחושב זה מוגדר בתור **אחוז** (Percent) עם 0 מקומות אחרי הקודה העשרונית.

```
% Delta from Avg: ([Extended Price]-[Avg of Extended Price])/  
[Avg of Extended Price]
```



qryPercentChangeFromAvg : Select Query

Order ID	Extended Price	% Delta from Avg
10248	\$440.00	-71%
10249	\$1,863.40	22%
10250	\$1,552.60	2%
10251	\$654.06	-57%
10252	\$3,597.90	136%
10253	\$1,444.80	-5%
10254	\$556.62	-64%
10255	\$2,490.50	63%
10256	\$517.80	-66%
10257	\$1,119.90	-27%
10258	\$1,614.88	6%
10259	\$100.80	-93%
10260	\$1,504.65	-1%

Record: 1 of 830

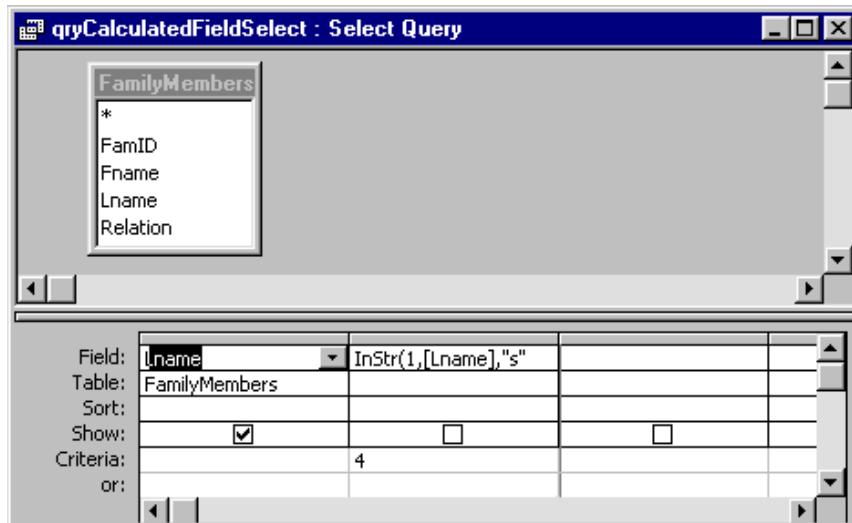
תרשים 4.13: החלון הראשון מציג שאלתה שמחשבת תוצאה קרטזית של שתי שאלות. הביטוי שבשדה המחשב מחשב את השינוי באחוזים בין מחיר הסיכון להזמנה לבין מחיר הממוצע של כל הזמנות

עבודה עם קרייטריונים

התנהגות שאלות מושפעת בצורה מכרעת על ידי קרייטריונים. ציון קרייטריונים הכוללים ביטויים וקבועים מאפשר לקבוע את סוג הרשותה שהשאילתת תחזיר. קרייטריונים גורמים להכללה או השטחה של רשותות בקבוצת החזרה של שאלת בחירה. הדוגמאות המוצגות בסעיף זה מתארות שאלות פשוטות הכוללות תנאי אחד. ברגע שתלמיד יציג אותו, יוכל לנצל יכולתו באופן מיידי ליצירת תנאים מורכבים המשלבים שני קרייטריונים ואף יותר באמצעות האופרטורים הלוגיים And ו-Or.

שימוש בקרייטריון בודד תרשים 4.5 מדגים קרייטריון פשוט העושה שימוש במשפט Null Is. משפט זה משמש לזיהוי שדה ריק בשאלתה. עיצוב השאילתת מצרף את הטבלאות FamilyMembers ו-FamilyNames בשדה Lname. סוג הצירוף כולל את כל השמות מתוך הטבלה FamilyNames בין אם הם توאמים לערך Lname בטבלה FamilyMembers ובין אם לאו. בטבלה FamilyNames, השדה Lname מקבל ערך אחד מתוך ארבעהערכים אפשריים: Simmons, Dobson, Hill ו-Edelstein. השדה Lname בטבלה FamilyMembers מכיל את הערכים Dobson ו-Hill. השאילתת מחזירה את השמות Simmons ו-Edelstein, כיון שני השמות נמצאים בטבלה FamilyNames אך לא בטבלה FamilyMembers Is Not Null -L-. FamilyMembers Is Null -L-. Dobson ו-Hill להשתנות ל-.

תרשים 4.14 מציג עיצוב שאללהה הכללת רק את רשומות הטעלה FamilyMembers שהשדה Lname שלhn מכיל את האות הקטנה s במקומות הרבייעי במחירות השם. הפונקציה InStr היא אחת הפונקציות החזקות של Access. הפרמטר הרבייעי מגדר את סוג החיפוש ; 0 מציין מחירות באחת מכמה דרכים. הפרמטר הרבייעי מציין את המיקום של המופע הראשוני InStr ,4.14 מחפש את המחרוזת s במקומות 4 בשם Dobson. מכאן שקביעת הקритריון 4 תגרום לבחירת רשומות שהשדה Lname שלhn מכיל מחירות שבמקום הרבייעי שלhn נמצאת האות s.



תרשים 4.14: שאללהה זו מחזירה רשומות מהטבלה FamilyMembers שהשדה Lname שלhn מכיל s במקומות הרבייעי במחירות
אם תוסיף לטבלה FamilyMembers רשומות חדשות שהשדה Lname שלhn מכיל מחירות שבמקום הרבייעי שלhn נמצאת האות s, השאללהה תחזיר גם אותן. לדוגמה, הוספה רשומה חדשה שהשדה Lname שלhn מכיל את השם Samson, תגרום לשאללהה להחזיר את Samson יחד עם Dobson.

שים לב שתוצאת הביטוי שבתרשים 4.14 אינה מוצגת בקובצת ההחזרה של השאללהה, מכיוון שהתיבה הציג (Show) אינה מסומנת.

שימוש בקריטריונים רבים השאללהה שבתרשים 4.15 יוצרת סדרת משפטים. השאללהה מרכיבה את המשפטים באופן דינמי מערכיה השודות, אך באותו זמן גם משמשת את הרשותה שערך השדה Relation שלhn הוא me (משפטי קבוצת ההחזרה של השאללהה מתארים את היחס בין כל בן משפחתי אליו (המחבר), ולכן הרשותה שערך השדה Relation שלhn הוא me (אני) לא אמורה להיות כוללה בשאללהה. השאללהה משתמשת בשני ביטויים: האחד מציין קריטריון להכללת רשומות, והשני הוא ביטוי מחירות שמרכיב את המשפטים באמצעות ערכי שדה נבחרים.

תרשים 4.15: שאלתה שמרכיבה משפטים בצורה דינמית

שימוש בקריטריון מורכב לאחר יצירה ובדיקה של קריטריונים לשדות מסוימים, ייתכן שתמצא לנוחץ ליצור קריטריונים מורכבים שימושיים שני קритריונים ויתר לגבי שדות מסוימים. ניתן לעשות זאת בשתי גישות בסיסיות. האחת, לשלב קריטריונים באמצעות האופרטור And (וגם). טכניקה זו גורמת לשאלתה להחזיר שורות של רשומות העומדות בו-זמנית בכל הקריטריונים המצוינים. הגישה השנייה היא לשלב קריטריונים באמצעות האופרטור Or (או). הדבר גורם לשאלתה להחזיר שורות של רשומות העומדות לפחות באחד הקריטריונים מלאה שהוגדרו.

תשמה בודאי לדעת כי רשות עיצוב השאלתה של Access פוטרת אותה מעיסוק בפרטי התchapir של משפטי SQL. בעת שימוש באופרטור And, הצב את כל הקריטריונים בשורה בודדת של הרשות. אם ברצונך להציג את כל הלקוחות מעיר מסוימת שהזמנתם עליה על סכום הזמן המינימלי, אפשרותך להתחיל בשאלתה שמסכמת מכירות לפי ליקוח מעיר כלשהו. לאחר מכן, בשאלתה אחרת שפנה לו הראונה, תוכל להשתמש בקריטריון מורכב בעל התנינה מסוג And: קריטריון אחד לקביעת העיר

שאודוטיה אתה מתענין, וקריטריון נוסף לקביעת גבול תחתון כך שהשאילתת תחזיר פרטים על לקוחות מסוימים הקיימים מזמן מסוים. הפעולה מחיבבת שימוש באופרטור And, וכן שני הקריטריונים יופיעו באותה שורה של רשות עיצוב השאילתת.

לאחר מציאת כל הלקוחות העומדים בקריטריונים בעיר המבוקשת, תוכל להחילת להתמקד בדףו הכספי הקנייה של הלקוחות באותה עיר. כדי לאסוף נתונים בנושא, תוכל להתחיל בשאילתת המכרפת את הטבלאות Orders ו- Order Details. רשות עיצוב השאילתתך יכולה להכיל זוגות קריטריונים נפרדים בשורות רבות. הצבת קרייטריונים בשורות נפרדות ברשות עיצוב השאילתתך משלבת אותך באמצעות הולוגית Or. הקרייטריונים בכל אחת מהשורות יתייחסו לעיר וללקוחות. העיר תהיה משופתת לכל השורות, אך לכל שורה יהיה לך ייחודי משלה. כדי לקבץ מכירות בצוות אלפביתית לפי לקוח, לחץ על השורה מיין (Sort) של השדה Customer ובחירה עולה (Ascending).

שאילותות בחירה מיוחדות

רובה הדיוון על עיצוב שאילותות עד כה התמקד במרכיבי העיצוב הבסיסיים של שאילותות בחירה (select queries). בסעיף זה נציג שלוש הרחבות של שאילותות בחירה, אשר משרות שימושיות את הגמישות וכושר ההתאמאה של היישומים שתפתח. תחילת נלמד על אוזות שאילותות פרט. מפתחים וBITS אוחבים לעובוד עם שאילותות מסווג זה, מכיוון שהוא מעניקות מימד אינטראקטיבי ליישומים. אחרים מחייבים אותו עקב כלות השימוש. שאילותות איחוד (union queries) מאפשרות לעצב נתוניים בצורה שלא ניתן להשיג באמצעות צירוף. כדי>Create שטי ערכות רשומות או יותר שיש להן שדות מסווגים, שאילותות איחוד ידע מעשי בשפת SQL. הסעיף האחרון בפרק זה מציג סקירה קצרה על התחריר של SQL ודוגמאות רבות להמחשת השימוש ביטור. אליה וקוץ בה – שאילותות איחוד מחייבות ידע מעשי בשפת SQL. השימוש האחרון בפרק זה מציג סקירה קצרה על הכלים הנבחר כשייש צורך בקריטריונים נבונים שמתחאים את עצם לערכיהם מסדר הנתונים. לעיתים קרובות, שימוש ביןונית או מתקדמת ב-SQL עשויה להקל על ייצור קרייטריונים דינמיים שימושיים לפי הערכים במסדר הנתונים.

שאילותות פרט

שםן של שאילותות פרט נובע מה הצורך לספק להן פרט בזמן ריצה כדי שתוכננה להתבצע. בעת יצירת שאילתת פרט בתצוגת עיצוב (Design), תוכל לקבוע את ההנחה להכנסת קלט בשורה **קרייטריונים** (Criteria). קבוע את ההנחה במשפט הקרייטריון בתור משתנה שערכו יקבע על ידי המשמש בזמן ריצה. משק המשמש אפשר גם להזכיר על סוג נתונים לכל פרט קלט. פרק 2 ממחיש כיצד לקודד שאילתת פרט באמצעות ADO. בסעיף זה נתאר כיצד לבצע שימוש זהה באמצעות משק המשמש.

תרשים 4.6 מציג עיצוב של שאליתת פרמטר ייחד עם תיבת הדיו-שיכון המכלולת של ההנחה.שים לב שההנחה במשפט הקרייטריונים של השדה UnitePrice בתחום בסוגרים מרובעים. הסימן קטן מ(<) שלפני הסוגר הפותח מגידר את הגבול העליון של ערכי השדה UnitePrice של קבוצת החזרה. אם המשתמש יקליד 35 כאשר יונחה להכניס ערך באפשרות שאליתת הפרמטר וילחץ על אישור, השאלתה תציג לו את רשימת כל המוצרים שמחירם נמוך מ-35 דולר. החלון התהוו בתרשימים 14.6 מציג את המוצרים הראשוניים ברשימה וגם את מספר הרשומות הכלל, העוניים על הקרייטריונים).(56)

The screenshot shows two windows from the Microsoft Query application.

Top Window: Enter Parameter Value

- Title: qprProductsLessThan : Select Query
- Content: A text input field containing "35". Below it are "OK" and "Cancel" buttons.

Bottom Window: qprProductsLessThan : Select Query

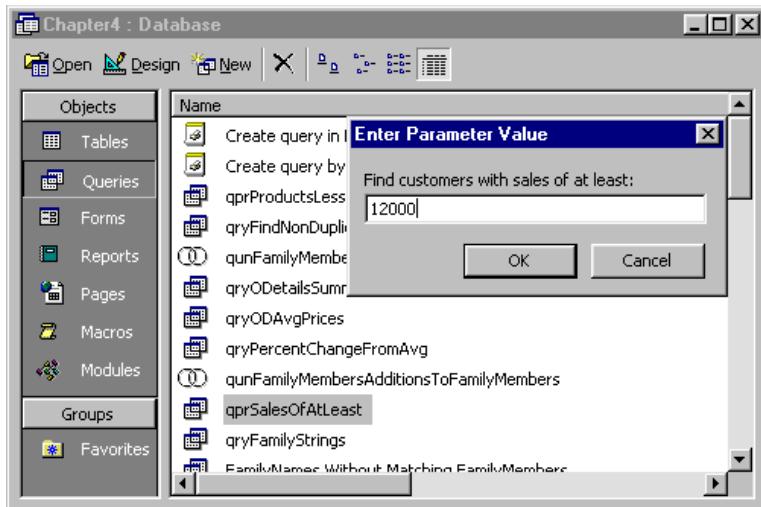
- Title: qprProductsLessThan : Select Query
- Content:

Field:	ProductName	UnitPrice	Discontinued
Table:	Products	Products	Products
Sort:	Ascending		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:	<[Return all products less than:]		False
or:			

Product Name	Unit Price
Geitost	\$2.50
Konbu	\$6.00
Filo Mix	\$7.00
Tourtière	\$7.45
Rhönbräu Klosterbier	\$7.75
Tunnbröd	\$9.00
Teatime Chocolate Biscuits	\$9.20
Røgede sild	\$9.50
Zaanse koeken	\$9.50
Jack's New England Clam Chowder	\$9.65
Sir Rodney's Scones	\$10.00
Aniseed Syrup	\$10.00
Longlife Tofu	\$10.00
Spegesild	\$12.00
Scottish Longbreads	\$12.50

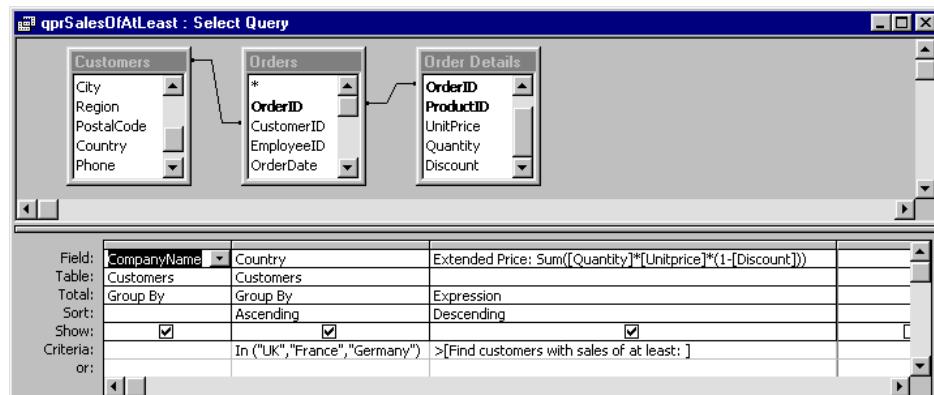
תרשים 4.16: שאליתת פרמטר וקבוצת החזרה שלה

משתמשי היישום לא יפעילו בדרך כלל שאלות פרטנר מתוך תצוגת העיצוב, כפי שעשינו בתרשים 4.16. תרשימים 4.17 מציג את השאלה כפי שהיא נראה בuest הפעלה מתוך מכולת מסד הנתונים. השאלה בדוגמה שלפניך מחייבת את הליקות בריטניה, גרמניה וצרפת, שהזמנויותיהם עלות על הערך שמצוין המשמש. כפי שנוכח לדעת, אפשרות המשמשים לקבוע את קבוצת החזרה הופכת את שאלות הפרטנר לדינמיות ו互動יות.



תרשים 4.17: שאלת פרטנר מופעלת מתוך חלון מסד הנתונים

תרשים 4.18 מציג את השאלה מאIOR 4.17 בתצוגת **עיצוב**. לשאלתה זו שני קriterיוונים: האחד מנצח את מילת המפתח In כדי לבחור ל��וחות שמשרדיים הראשיים נמצאים בבריטניה, צרפת או גרמניה. כך מתבטל הצורך להגדיר שלוש שורות נפרדות שככל אחת מכילה שם מדינה אחר. ההנחה בעמודה Extended Price מוקמת אחרי הסימן גדוֹל מ (>), וכך השאלה מחזירה את כל הרשומות שמכילות מחיר סיכון גדול מזה שמצוין המשמש.



תרשים 4.18: תצוגת העיצוב של השאלה מתרשים 4.17

השאילתה שבתרשים 4.18 מבצעת פעולות נוספת. היא מחשבת ומסכמת את מחיר הסיכום לפי חברה ומ민ית את קבועת ההחזרה שלה בסדר אלפבטי עולה לפי מדינה ובסדר יורד לפי מחיר סיכום של החזמנאות. שיטת העיצוב גורמת להציג הלקחות הגדולים בראש רשימת הלקחות בכל מדינה.

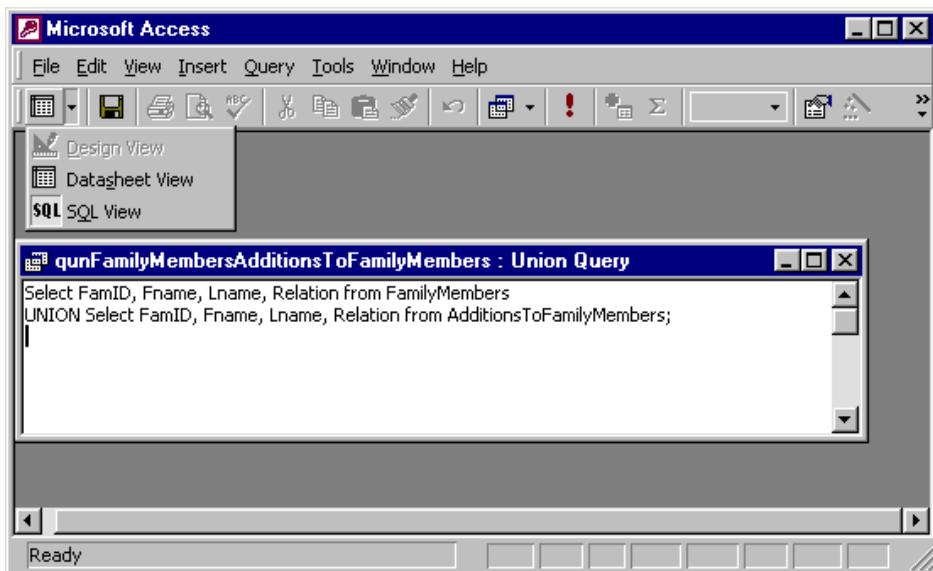
שאלות איחוד

שאלות איחוד הן ייחודיות מכמה היבטים. ראשית, הן מפשטות בצורה דרמטית את פעולות השרשור של שתי רשומות או יותר (כזכור, צירוף של שתי ערכות רשומות מציב אותן זו לצד זו, ולא זו אחר זו). שנית, ניתן להגדיר שאילתות איחוד בעורת SQL בלבד. שתי התוצאות היחידות של שאילתות איחוד הן **גיליון נתוניות** (Datasheet) ו-**SQL**; בשום אופן לא ניתן להציג בתצוגת **עיצוב** (Design). שלישית, לא ניתן לעדכן ישירות את ערכי השדה בתצוגת **גיליון נתוניות** של שאילתת איחוד. שאילתות בחרה רבות שנראות זהות לשאילתות איחוד בתצוגת **גיליון נתוניות**, הן קבועות רשומות דינמיות (dynaset), אשר מאפשרות למשתמש לעדכן את הטבלה שעליה מבוססת מערכת הרשות. אם עלייך לעדכן את הנתונים שהם תוצאה של שאילתת איחוד, عليك ליצור עותק של הנתונים באמצעות שאילתת יצרת טבלה (make table) ולאחר מכן לאחר מכן. פעולה העERICA לא תנסה את נטוני הקלט המקוריים של שאילתת האיחוד, אך תאפשר לפחות לשנות את העותק של הנתונים שהתקבלו.

תרשים 4.19 מציג שאילתת איחוד פשוטה ביותר שימושת שטלה את הטבלה FamilyMembers והטבלה AdditionsToFamilyMembers. אם אתה מתaskaה בעבודה עם SQL, תוכל להמשיך לשמר על פשטות שאילתות העיצוב. כתוב את המשפט הראשון בתור משפט SELECT פשוט שמציג את שמות השדות ומקור הרשות של שדות בפסקית FROM. ההפניות לכל שאר מקורות הרשות צrüכות להתחילה במילimits UNION, אך הרשימה עדין צריכה לכלול את שמות השדות ואת מקור הרשות. ניתן לבצע פעולות מתקדמות יותר, אך אין הדבר הכרחי. תוכל תמיד לשלב את מקורות הרשות בצורה פשוטה זו ולשמור את פעולות הטיפול המתווכמות במקורות המשולבים לשאילתות בחירה אחרת שמתבססת על הפלט של השאילתת האיחוד. באפשרותך לבצע את הפעולות המתקדמות האלה בתצוגת **עיצוב**.

ניתן לראות על פי משפט ה- **SELECT** ו- **UNION** שבו מושרים טבלאות שמכילות נתונים מסווג זהה. זהה דרישת בסיסית בעבודה עם שאילתות איחוד. מספר העמודות, המיקום, וסוג הנתונים בטבלאות, או לפחות שדות שאתה משלב, חייבים להיות זהים. לא ניתן לעדכן את המקור מתוך תצוגת **עיצוב**, אך השאילתת מוחשבת מחדש בכל פעם שפותחים אותה, עם הנתונים העדכנים שהתקבלו מהקלט שלהם.

שים לב ש-Access מבטל זמינות את הסמל **תצוגת עיצוב** בשאילתת האיחוד, ולכן לא ניתן לבחון שאילתת איחוד בתצוגה זו. עצב וערוך אותה בתצוגת **SQL**, ובן את פעולה משפטי SQL בתצוגת **גיליון נתוניות**.



תרשים 4.19: שאלת איחוד בתצוגת SQL

תרשים 4.20 מגלח את פועלות שאילתת האיחוד של תרשימים 4.19. הטבלאות FamilyMembers ו-AdditionsToFamilyMembers מוצגות בחלקו השמאלי של תרשימים 4.20 ; פلت שאילתת האיחוד הוא גילוון הנתונים מצד ימין. כפי שניתן לראות מסדר הציגה של השדה FamID, השאלתה יוצרת גילוון נתונים חדש שמאחד את שתי השורות של הטבלה AdditionsToFamilyMembers לטבלה FamilyMembers.

FamilyMembers : Table			
FamID	Fname	Lname	Relation
1	Rick	Dobson	Me
2	Virginia	Dobson	wife
3	Glen	Hill	son
4	Tony	Hill	son
5	Shelly	Hill	daughter-in-law
*			

AdditionsToFamilyMembers : Table			
FamID	Fname	Lname	Relation
6	Dee	Edelstein	sister
7	Foster	Simmons	nephew
*			

qunFamilyMembersAdditionsToFamilyMembers ...			
FamID	Fname	Lname	Relation
1	Rick	Dobson	Me
2	Virginia	Dobson	wife
3	Glen	Hill	son
4	Tony	Hill	son
5	Shelly	Hill	daughter-in-law
6	Dee	Edelstein	sister
7	Foster	Simmons	nephew
*			

תרשים 4.20: שני גילוונות הנתונים השמאליים הם קלט שאילתת האיחוד של תרשימים 4.19. גילוון הנתונים שמימין הוא פلت שאילתת האיחוד

שאלות משנה

שאלת משנה היא שאלת בחירה המשולבת בשאלת אחרת. שאלת המשנה חייבת להיות משפט SELECT, אך ניתן להשתמש בה בתוך שאלת אחרת כלשהי, כגון שאלת בחירה או שאלת פעולה. קיימות Möglichkeiten מפתח נוספת שמסוגלת למתן את פועלתה של מילת המפתח הסטנדרטית SELECT של SQL בשאלת משנה. המשפט SELECT מחזיר ערכים, ולכן שאלת המשנה מייצגת אפשרות חשובה לבניית קритריונים דינמיים בשאלות שימוש ביהם.

המשפט SELECT המוטבע בשורה **קריטריונים** (Criteria) של העמודה Lname בחולון התיכון של תרגים 4.4 הוא שאלת משנה.

תרגום 4.21 מציג שאלת שמחזירה את ערכי Lname הבלתי משוכפלים שהחזרה שאלת איחוד שנקרהת AdditionsMoreAdditions. השalon השמאלי התיכון בתרגום האיחוד מרשרת שלושה מקורות רשומה נפרדים. החולון השמאלי התיכון בתרגום 4.21 הוא קבוצת החזרה של שאלת האיחוד. שים לב להופעת ערך לא משוכפל אחד בלבד של Lname שנקרה Edelstein. החולון הימני התיכון הוא קבוצת החזרה של שאלת הבחירה.

The screenshot shows three windows from Microsoft Access:

- qryFindNonDuplicatedFamilyNames : Select Query**: This window displays the query design. It includes a top pane showing the table "qunFamilyMembersAdditionsMoreAdditions" with fields FamID, Fname, Lname, and Relation. Below this is a bottom pane where the field "Lname" is selected, and the criteria is set to "Not In (Select Lname from qunFamilyMembersAdditionsMoreAdditions Group By Lname Having Count(*)>1)".
- qunFamilyMembersAdditionsMoreAdditions**: This is a data entry window showing a table with columns FamID, Fname, Lname, and Relation. The data includes various family members like Rick Dobson, Virginia Dobson, etc., with their relationships.
- qryFindNonDuplicatedFamilyName...**: This is a results window showing the Lname column from the previous query. It contains one record: "Edelstein".

תרגום 4.21: החולון העליון הוא שאלת שמצוות ערכי שdots לא משוכפלים במקור רשומות; שני החלונות שמתוחתי הם גלגולות הנתונים של הקטל והפלט של השאלה.

החולון העליון בתרשימים 4.21 הוא עיצוב שאילתת המשנה למציאת ערכי `Uname` לא משוכפלים. שים לב לדמיון בין שאילתת זו לשאילתת שבתרשימים 4.4. בלבד מן העובדה שהשאילתת סורקת מקור רשומות אחר, השוני היחיד בין השאילתות הוא שאילתת המשנה מתחילה במילים `In Not` במקום במילה `Is`. הביטוי המכיל את `In Not` מוחזר שמנות לא משוכפלים, בעוד שהמשפט לאחר מוחזר שמנות משוכפלים.

הערה:



באפשרותך לשלב את קבוצות הרשומות המוחזרות השאילותות כמו אלו שבתרשימים 4.4 ו-4.21 כדי להחזיר את כל השמות השונים, אך Access כולל גם דרכים גרפיות וocaleה שבסיסות על ביטויים כדי להשיג תוצאה זהה. בהמשך הפרק נתיחס בהרחבה לגישה המבוססת על ביטויים. הגישה הגרפית כוללת מתן **הערך כן למאפיין ערכים ייחודיים (unique values)** של השאילתת.

שאילותות פעולה

Access כולל ארבע שאילותות פעולה שניית להפעיל באמצעות ממוקם המשתמש וגם בעורת קוד. אלו הן שאילותות עדכון, הוספה, מחיקה ויצירה. איש פיתוח ברמת ביינימים תיוכח שאילותות פעולה הן מעניינות, מכיוון שהן מספקות דרך מהירה וקלת חישית לשיפור התפקודיות של יישומי Access.

באחד הסעיפים הקודמים בפרק סקרנו את שאילתת הפעולה מהיבטיה התפקודים, מבלי להתמקד בapon היישום שלה. בסעיף זה נתאר כיצד משלבים שאילותות פעולה בישומים מותאמים אישית. רשות עיצוב השאילתת כוללת הרחבות מיוחדות התומכות ברכיבים התפקודים של כל אחד מסווגי שאילתת הפעולה. בסעיף זה נציג את השאילותות ואת אופן העיצוב והשימוש בהן.

שאילותות עדכון

שאילותות עדכון משמשות לשינוי ערכי שדות במקור רשומות של שאילתת. מתחילה ביצירת שאילתת עדכון בדיקת CPI שנותגים במקורה של שאילתת בחירה. לאחר לחיצה כפולה על האפשרות **יצירת שאילתת בתצוגת עיצוב (Create Query In Design View)** בחלון **mse נזונים (Database)**, הוסף מוקור רשומות אחד או יותר לשאילתת באמצעות תיבת הדו-שיכון **מצגת טבלה (Show Table)**. למרות שם תיבת הדו-שיכון, היא מאפשרת להוסיף טבלה או שאילתת אחת או יותר. באופן טיפוסי תרצה בוודאי לצרף את כל הטבלאות והשאילותות שאתא כולן בשאילתת העדכון לאחר סגירת תיבת הדו-שיכון **מצגת טבלה (Create Query In Design View)** (באפשרותך להסיר מקורות רשומות לא רצויים כלשהם על ידי בחרתם ולהיכזה על מקש Delete במקלדת).

כדי ליצור שאלת פועלה, עליך לבחור את הסוג שלה. בחר באפשרות **שאלת עדכון** (Update Query) מהלצת **סוג שאלת** (Query Type) בסרגל הכלים **עיצוב שאלת** (Query Design). חוזת הלוחן תשתנה והוא יציג את סמל שאלת העדכון (עיפרונו ואחריו סימן קריאה). שים לב שפעולה זו מוסיפה שורה מסווג **עדכן ל** (To Update) בראש עיצוב השאלת ומסירה את השורות **מיין** (Sort) והציג (Show). לאחר מכן הוסף שדה אחד או יותר לשורה **שדה** (Field) של רשות עיצוב השאלת. עליך לכלול גם את השדה שברצונך לעדכון. ניתן לצרף לשאלת שדות נוספים למטרות זיהוי או כדי לסייע להגדיר קритריונים לקביעת הרשומות שיעודכו.

לאחר שתסיים לבחור את השדות עבור שאלת העדכון, הכנס ערך או ביטוי באחד התאים של השורה **עדכן ל**. הערך שבתא קבוע את ערכיו החלפה החדש. אם לא סופקו קритריוניים כלשהם, הפעלת השאלת תעדכן את כל הרשומות שבמקור הרשומות של השאלת. שילוב קритריון אחד או יותר יגביל את פעולה העדכון לשורות שעומדות בתנאים מוגדרים.

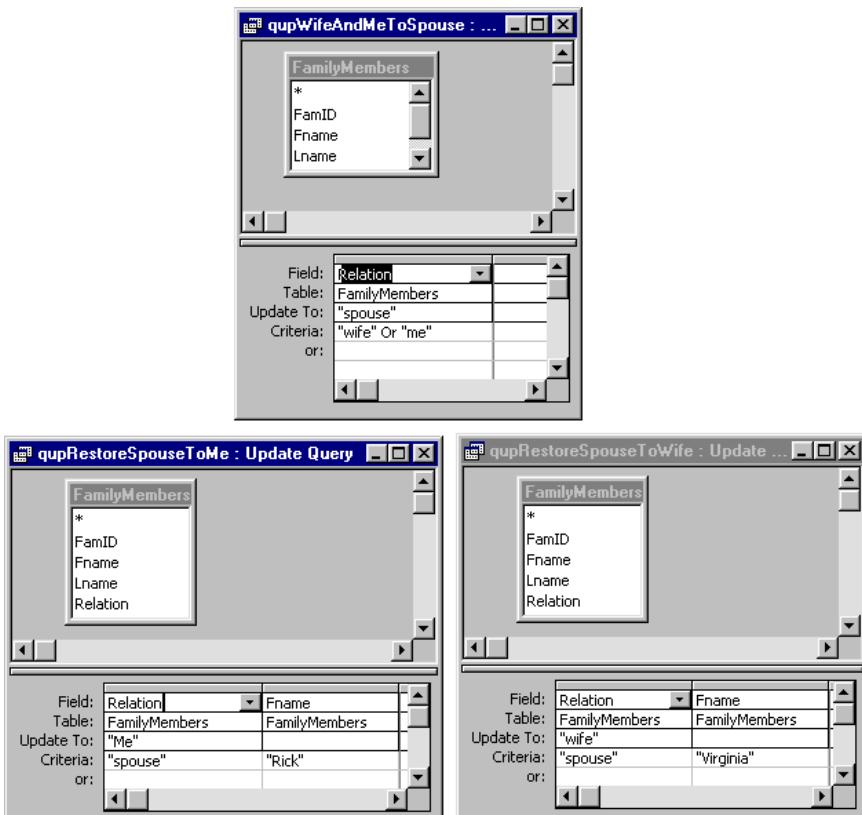
בעת הפעלת שאלת עדכון, לא ניתן לבטל את השינויים בלחיצת עכבר. עליך לשחרר את כל הטבלה מתוך עותק גיבויו, או להפעיל שאלת עדכון אחת או יותר כדי לשחרר את הערכים המקוריים. הלוחן **תצוגה** (View) בסרגל הכלים של תצוגת העיצוב מאפשר לבחון את הרשומות שאלת נשנה מבלי לבצע עדכון בפועל. לאחר שבחרת את הרשומות המתאימות, לחץ על **לחוץ הפעל** (Run) כדי להפעיל את השאלת. תיפתח תיבת הודעה ובה מספר הרשומות שהשאלת תעדכן. באפשרות לבטל את שאלת העדכון בשלב זה על ידי לחיצה על **לא**. אם תלחץ על **כן**, ימשך תהליך העדכון. אם השדה אינו בר-עדכון, מציג תיבת הודעה מתאימה.

Access מציג תיבת הודעה נוספת אם מנסים להפעיל את שאלת העדכון מחוץ לתצוגת העיצוב. באפשרות לדוחות את האפשרות לשנות את ערכי השדות או להמשיך. אם תבחר להמשיך בעדכון, תיבת הדו-שים הבאה דומה לו הראונה שהזוכה בתצוגת העיצוב. התיבה מצינית את מספר הרשומות שישתנו ומאפשרת לשנות את הרשומות פעמיים נספת ואחרונה.

שאלות עדכון יכולות לשנות שדות רבים במקורו הרשומות שלהם. באפשרות לעדכן את השדות במקורו הרשומות כל עוד בידך הרשאה לעשות זאת ומקור הרשומות זמין. נתונים מקשר גומלין מסווג ייחיד לרבים הם לעתים קרובות ניתנים לעדכון, אך לא תמיד. לדוגמה, לא ניתן לשנות את המפתח הראשי, אם עדכנים על פי היררכיה קשרים אינם ניתנים לשימוש. שדות וערכים מוחושבים בשאלות הצלבות (crosstab) אף הם אינם ניתנים לעדכון. אם שדה מחושב, שאלות פועלה איןין יכולות לשנות את ערכיו. העזרה המקוונת כוללת תיאורים מפורטים של שדות שאינם ניתנים לעדכון.

החולון העליון בתרשימים 4.22 מציג את רשות עיצוב השאלת של שאלת עדכון בסיסית שמשנה שני ערכי שדה בטבלה FamilyMembers. שאלת זה משנה את ערך השדה Relation ל-spouse, אם הערך הקודם בשדה מכיל את הערך me או wife. שני החלונות התוצאות בתרשימים מציגים שתי שאלות עדכון נוספות שמשמעותן את הערכים המקוריים של מקור הרשומות. שאלות אלו מציינות גישה אחת לבניית

שאילות עדכון, כך שבאפשרותך לשחזר ערכיהם המקוריים. שים לב כי יש צורך בשתי שאילות כדי לבטל את פעולות השאלתה המקורי. כמו כן, שים לב שכל שלושת המקרים מכילים קритריונים מורכבים. הקритריונים המורכבים מסייעים לזהות טוב יותר את הרשומות המיועדות לעדכון.



תרשים 4.22: שלוש שאילות עדכון

הפעלת שאילות השחזור טומנת בחובה כמה בעיות. ראשית, עליך להפעיל את שתיהן. שנית, Access מבקש את אישורך להפעלת כל שאלה (אם תפעיל אותן מתוך החלון **מסד נתונים**, הוא מבקש את האישור פעמיים). אםعرכת עדכונים מסווגים רבים, כדאי יהיה להפוך את התהליך לאוטומטי כדי לחסוך זמן ולשפר את הדיקוק. שגרת המשנה שלפנייך מבצעת את פעולה העדכון. כל שעל המשתמש לעשות הוא להפעיל אותה.

```
Sub RestoreMeAndWife()
    DoCmd.SetWarnings False
    DoCmd.OpenQuery "qupRestoreSpouseToMe"
    DoCmd.OpenQuery "qupRestoreSpouseToWife"
    DoCmd.SetWarnings True
End Sub
```

הקוד של שתי שאלות העדכון מכיל ארבע שורות. השיטה SetWarnings מבטלת את תיבות הדו-שיך של האזהרה. השיטה OpenQuery משמשת להפעלת שתי השאלות, כל אחת בתורה. השורה الأخيرة מחדשת את השימוש בתיבות הדו-שיך של האזהרה.

שאלות הוספה

שאלות הוספה מוסיפות רשומות חדשות מתוך ערכת רשומות של מקור אל ערכת רשומות יעד. ערכת רשומות היעד יכולה להיות טבלה בודדת או טבלאות שעליין מבוססת שאלת בחירה ברת-עדכו. ערכת רשומות המקור יכולה להיות טבלה או שאלת בחירה בעלת שדות שלפחות חלק מהם תואמים את השדות של ערכת רשומות יעד. ערכת רשומות המקור חייב להיות בסיס הנתונים הנוכחי. ערכת רשומות היעד יכולה להיות בקובץ מסד נתונים כלשהו של Access.

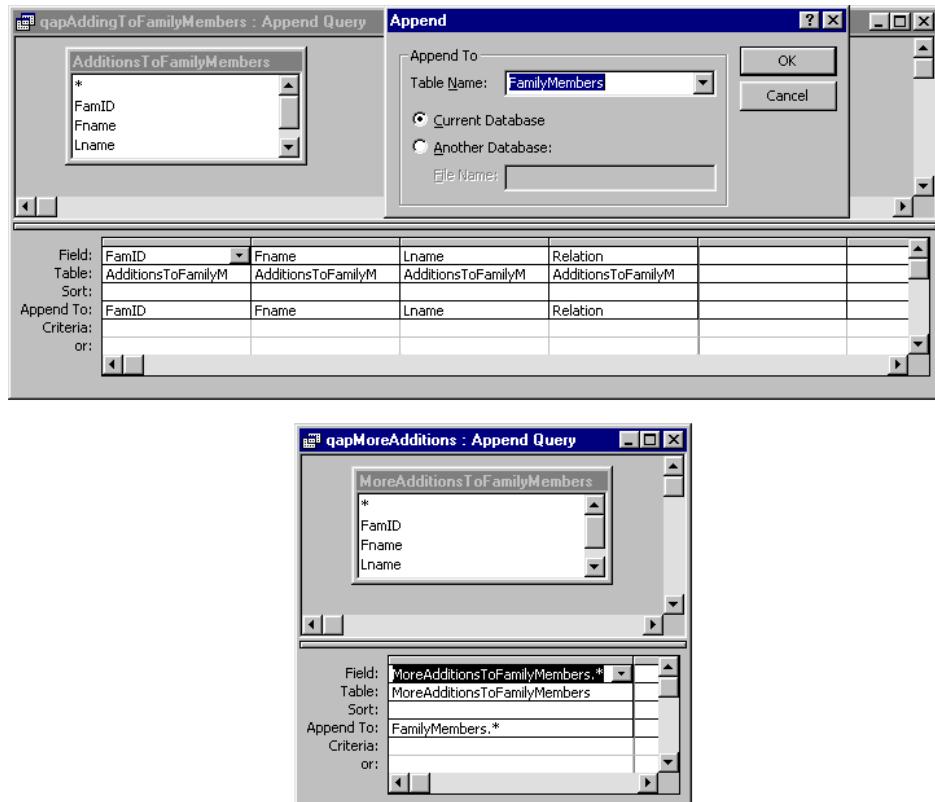
שאלת הוספה אינהחייב להוסיף רשומות המכילות כל שדה ושדה בלבד, אך עליה להכיל שדה מבודק אחד לפחות. אם טבלת היעד מכילה מפתח ראשי, ודא שערכי המפתח ייחודיים בשתי ערכות הרשומות. אם לא, ערכי המפתח הראשי לא יהיו ייחודיים, שאלת הוספה לא תוכל להוסיף רשומות.

מתחלים בעיצוב שאלת הוספה על ידי הוספת ערכת רשומות המקור לتزוגת העיצוב של השאלה. לאחר מכן, מוסיפים שדות מתוך המקור אל השורה שדה (Field) של רשות עיצוב השאלה. לאחר מכן,فتح את הלחצן **סוג שאלת** (Query Type) ברגל הכלים ובחר את **שאלת הוספה** (Append Query) כדי להציג את תיבת הדו-שיך **הוספה** (Append). תיבה זו מפשטת את מישימת העיצוב של טבלה אחרת בסיס הנתונים הנוכחי בתור טבלת היעד. באפשרות גם לציין שברצונך להוסיף רשומות לטבלה בקובץ מסד נתונים אחר של Access על ידי בחירת לחץ בחירה **מסד נתונים אחר** (Another Database) והקלדת הנתיב ושם קובץ של מסד הנתונים המבוקש. לאחר בחירת מסד נתונים, תיבת הרשימה הנפתחת **שם הטבלה** (Table Name) מציגה את הטבלאות בקובץ שבחרת. ציין ערכת רשומות יעד על ידי בחירת שם טבלה מתוך תיבת הרשימה. אם ברצונך להוסיף רשומות לשאלת בחירה, عليك ליצור עצמאך את קוד שאלת הפעולה (עיין פרק 2 ל渴בלת דוגמאות לביצוע פעולה זו באמצעות ADO ו-SQL).

לאחר בחירת מקור רשומות יעד, Access משנה את רשות עיצוב השאלה שלו כך שתכלול את השורה **הוסף ל** (Append To). הוא גם מתאים אוטומטית שדות רבים ככל שהוא מזוהה בערכת רשומות היעד ללא שבערכת רשומות המקור. ניתן לדודס את הבחירה האוטומטית. שדות Sh - Access יינוי מתאים להם בני זוג, תוכל להתאים בעצמך (לדוגמה, שדה שמו שונה, אך סוג הנתונים שלו זהה). לחץ בשורה **הוסף ל** של עמודה, ובחר בשם השדה המתאים מתוך הרשימה הנפתחת של רשומות שדות ערכת רשומות היעד.

החלון העליון בתרשימים 4.23 הוא שאלת הוספה בתציגות עיצוב לאחר הוספת שדות בשורה **שדה והוסף ל** שלה, ותיבת הדו-שיך **הוספה** שלה פתוחה. שאלת הוספה זו

מוסיפה רשומות מתוך הטבלה AdditionsToFamilyMembers. לאחר סגירת תיבת הדוח-שיכון הוספה, לחץ על הלחוץ **הפעל** (Run) כדי להפעיל את השאלתה. בדומה לשאלות פוליה אחרת כלשהי, לחץ על לחוץ **צוגה** (View) בסרגל הכלים כדי להציג את הרשומות שהשאלתה תעבד מבלתי להפעיל את השאלתה.



תרשים 4.23: החילון העליון הוא שאלת הוספה שמציגה את כל השדות שיש להוסיף למקור רשומות היעד. החילון התחתון הוא שאלת הוספה שמנצל את הקוביות כדי ליעד את כל השדות של ערכות הרשומות היעד והמקור.

החלון התחתון של תרשימים 4.23 מציג גירסה שונה של שאלת הוספה שבחלוון היעד. השאלת התחתונה מעתקה את כל השדות מהטבלה MoreAdditionsToFamilyMembers אל הטבלה FamilyMembers. גררתי את הקוביות מרשימה שדה ערךת רשומות המקור לשורה **שדה** (Field) שברשות עיצוב השאלתה. באופן דומה, בחרתי את הקוביות מהרשימה הנפתחת בשורה **הוסף** ל. שתי השאלות מוסיפות את כל השדות הקיימים מערכות רשומות המקור אל ערכות רשומות היעד. לעומת זאת, השאלת התחתונה גמישה יותר, כיון שהיא כוללת אוטומטיות שדות חדשים לשמש לתוסיף לשתי הטבלאות. כאשר מציגים את השדות ברשימה באופן מפורש, השאלת מוגבלת לשדות אלה עד שמעודכנים ידנית את הערכים בשורות **שדה** והוסף ל.

הערה:



במקרים אחדים, Access עלול לא לרענן מיידית את התצוגה של טבלה לאחר השלמת שאלחתת הוספה. לכן, השאלתה יכולה לעדכן בפועל את ערכת רשומות היעד מבל' שהשני יבוא לידי ביטוי בתצוגה. ניתן להתגבר על הבעיה על ידי שני נציגות ערכות רשומות היעד לתצוגת **עיצוב** ולהזור מיד לתצוגת **גילון נתונים** (לחילופין ניתן לרענן את תצוגת ערכות רשומות, על ידי הקשה על F9+Shift).

שאלות מחיקה

שאלות מחיקה מסירות רשומות באופן תמידי מערכת יעד. מסיבה זו עלייך להנוגז זיהירות רבה בעת השימוש בשאלות פועלות למחיקה. הקפד להשתמש בלחצן **תצוגה** (View) כדי לוודא שהרשומות המיועדות להסרה הן אכן אלה שברצונך להסיר בטרם תפעיל את השאלתת. שאלות מחיקה יכולות למחוק גם רשומות בערכות רשומות אחרות פרט לעיד. הדבר נכון במצב שהיעד נמצא בדיוק לצד היחיד של קשר הgomlininus מסווג יחיד לרבים ושבחרה האפשרות למחיקות בהתחאם להיררכיות הקשרים. במצב זה, מחיקת רשומה מצד היחיד מסירה את הרשומות התואמות גם מצד הרבים.

ככלית, עלייך להגן על מקורות הנתונים בעת השימוש בשאלות פועלות, אך הדבר נכון במיוחד לגבי שאלות מחיקה, על שום יכולתן למחוק קבוצות גדולות של רשומות מתוך מקור רשומות. מומלץ לשקלול הכנות עותק גיבוי של מקור הרשומות, וכך תוכל לשחרר רשומות שנמתקו בשוגג. אסטרטגיה נוספת בטיפול בשאלות מחיקה היא להסתיר אותן בחלוון **מסד נתונים** (Database), מה שיצמצם את הסיכוי שהמשתמש יפעיל בטעות שאלתת מחיקה (באפשרות להפעיל באמצעות קוד את שאלות המחיקה שבחרת להסתיר). אובייקטים מסווגים של מסד נתונים אינם מוצגים אלא רק כשtabular להסתיר). אובייקטים מסווגים של מסד נתונים אינם **כלים** (Tools) ובחר את **אובייקטים מסווגים** (Hidden Objects) בכרטיסיה **תצוגה** (View) של תיבת הדוח-שים **אפשרויות**.

הערה:

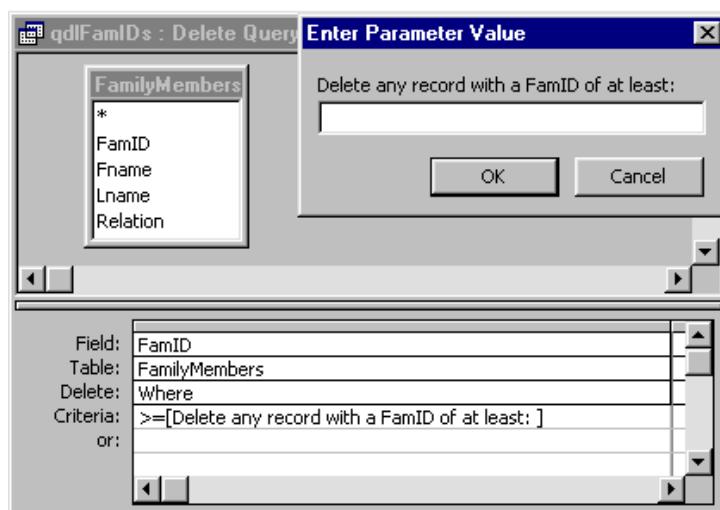
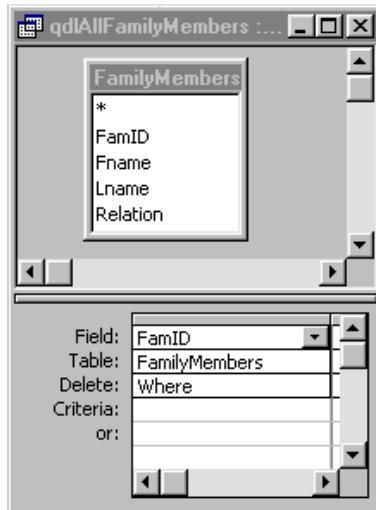


להסתרת שאלחתת פועלה, לחץ לחיצה ימנית על שאלתת בחלוון **מסד נתונים** ובחר באפשרות **מאפיינים** (Properties) מתוך תפריט הקיצור. בחר את תיבת הסימון **מוסתר** (Hidden) בಗילון המאפיינים ולחץ על אישור (יתכן שתצטרך לבטל את גם בחירת תיבת הסימון **אובייקטים מסווגים** בכרטיסיה **תצוגה** של תיבת הדוח-שים **אפשרויות**).

הפעולה הראשונה בעיצוב שאלחתת מחיקה היא לחיצה כפולה על **יצירת שאלתת** **תצוגת עיצוב** (Create Query In Design View) בחלוון **מסד נתונים**. בתיבת הדוח-שים **הציג טבלה** (Show Table), הוסף את מקור הרשומות שמןנו בראצונך למחוק רשומות. ציין את השורות שברצונך למחוק על ידי גרירת שדות מtower רשיימת השדות אל השורה **שדה** (Field) בראש עיצוב השאלתת. הכנס ערכים בשורה **קריטריוניים** (Criteria) של

רשת העיצוב, כדי לציין רשומות יעד המיועדות למחיקה. אם ברצונך ששאלילתת המחיקה תסיר את כל הרשומות ממוקר רשומות היעד, הכנס כוכבית (או שם שדה כשלחו) בשורה שדה ועל תכנית קרייטריון כלשהו. השורה שדה חייבת להכיל ערך אחד לפחות, כדי ששאלילתת המחיקה תוכל לפעול.

תרשים 4.24 מציג את העיצוב של זוג שאליותות מחיקה. השאליטהה העליונה מוחקת את כל הרשומות מתוך הטבלה FamilyMembers; היא אינה מכילה ערכים כלשהם בשורה **קרייטריונים**. לו ידעת את ערכי FamID של הרשומות הספציפיות שברצונך למחוק, יכולת להוסיףם לשורות **קרייטריונים** טרם הפעלת השאליטהה.



תרשים 4.24: שתי שאליותות פרמטר

אם אין יודע איזה שורות למחוק בזמן העיצוב, אפשרות ליצור את שאלתת המחיקה בתור שאלתת פרמטר. החלון התיכון בתרשים 4.24 מציג סוג שאלתת ההנחיה היא ההנחיה של שאלתת פרמטר שמופיעה בזמן ריצה. משתמש יכול להכין ערך FamID כדי למחוק את כל הרשומות שערך FamID שלהן גדול או שווה לערך שצוין.

הSIGERA שלפניך שופכת מעט אוור על שאלות מהיקחה והוספה ייחדיו. השגירה מאפשרת למחוק או לשוחזר את כל הרשומות בטבלה FamilyMembers. כשהיא מסתויימת, הטבלה פתוחה וזמינה להציג. אם מסתיר את השאלות qdlAllFamilyMembers ו-qapRestoreAllFamilyMember, הדרך היחידה להפעיל אותן תהיה באמצעות קוד, כמוואר להלן.

```
Sub ActionQueryDemo()
Dim varYesNo

' Prompt to delete all records.
varYesNo = MsgBox("Do you want to remove all records?", _
vbYesNo, "Programming Microsoft Access 2000")
' If answer is yes, remove them all.
If varYesNo = vbYes Then
    DoCmd.SetWarnings False
    DoCmd.OpenQuery "qdlAllFamilyMembers"
    DoCmd.SetWarnings True
End If
' Prompt to restore all records.
varYesNo = MsgBox("Do you want to restore all records?", _
vbYesNo, "Programming Microsoft Access 2000")

' If answer is yes, restore them all.
If varYesNo = vbYes Then
    DoCmd.SetWarnings False
    DoCmd.OpenQuery "qapRestoreAllFamilyMembers"
    DoCmd.SetWarnings True
End If
' Closing and opening table redraws it while making the table available for viewing.
DoCmd.Close acTable, "FamilyMembers"
DoCmd.OpenTable "FamilyMembers"
End Sub
```

הSIGERA פותחת בפונקציה מסוג MsgBox ששולחת את המשמש אם ברצונו להסיר את כל הרשומות מהטבלה FamilyMembers. אם המשמש עונה בחיוב, השגירה מפעילה את השאלה qdlAllFamilyMembers שמשמעותה בחלק העליון של תרשימים 4.24 (שים לב כי סיסמה ActionQueryDemo מכילה זוג משפטים שמכבים תיבوت דו-שיך של אזהרה ולאחר מכן מפעילים אותו שוב). לאחר מכן, השגירה שולחת את המשמש אם ברצונו לשחרר

את כל הרשומות של FamilyMembers. אם המשתמש עונה בחיוב, הקוד מפעיל את השאלתה qapRestoreAllFamilyMember כדי להחזיר רשותות מתוך עותק גיבוי של FamilyMembers הטרבלה אל הטרבלה .

לפני סיוםה, השיגרה סוגרת את הטרבלה FamilyMembers ופותחת אותה מחדש, פעולה שمبטיחה שהטרבלה תציג את הנתונים העדכניים ביתר, ומשאייה אותה פתוחה לצפיה ב-Access.

שאלות ייצור טבלה

שאלות ייצור טבלה, בניגוד לשלוש שאלות הפעולה האחרות, אינן מיוחדות או מוחקotas מקור נתונים מקורי. המטרה היחידה של שאלת ייצור טבלה היא ליצור טבלה של Access המבוססת על מקור רשומות. שילובו כלשהו של נתונים שנייתן קשור בשאלת בחירה, ייחסב למקור רשומות חוקי שמננו ניתן ליצור טבלה.

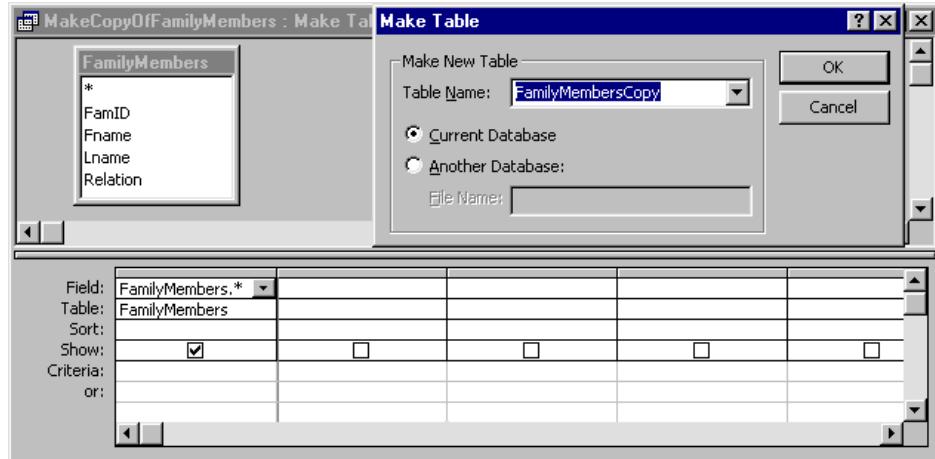
תרחישי יישום אחדים עשויים להזכיר שאלות ייצור טבלה. ראשית, באפשרות להשתמש בשאלת ייצור טבלה כדי ליצור עותק גיבוי של טבלאות קרייטיות. הדבר נכון במיוחד רגע לפנינו שיש למחוק רשומות מתוך מאגר הנתונים של המערכת הפעילה. שנית, שאלת ייצור טבלה יכולה לשמור את מצב הטרבלה או מצב קבוצת טבלאות מצורפות בנקודת זמן קבועה. שלישיית, ניתן להcin עותק נתונים בקובץ מסד נתונים אחד למען המטרות של יישום מסד נתונים אחר. תמיד קיימת אפשרות לקשר נתונים (היתרון של קישור הוא העובדה עדכנים יותר), אך ניתן לשפר את הביצועים באמצעות טבלה בקובץ, במקום טבלה מקושרת שמקורה בקובץ אחר. רבייעית, שאלות עלולות להפוך למורכבות מדי לעיבוד ב-Access, או שעיבודן עלול להימשך זמן רב. במקרים כאלה, החילف את השאלת המורכבות היעייתי בטבלה שנוצרה באמצעות שאלת ייצור טבלה. על ידי כך תעקוף את הودעת השגיאה הנוגעת למורכבות יתר וביצועי השאלת ישתפרקו.

שאלת ייצור טבלה היא במהותה שאלת פעללה, וכך קל בהרבה ליצור טבלה באמצעות שאלת ייצור טבלה לעומת ייצור טבלה בעזרת DAO, ADO, SQL או ODBC. כך יוצרים שאלת ייצור טבלה:فتح שאלת ריקה בתצוגת **עיצוב**, הוסף לה טבלאות ושאלות בטור משאים לטבלה החדשה, צרכ קלטים לפי הצורך ובחר באפשרות **שאלת ייצור טבלה** (Make-Table Query) מתוך רשימת הלחצן **סוג שאלת** (Query Type). בעת, גורר את השדות שתזדקק להם בטבלה החדשה אל השורה שדה (Field) של רשות עיצוב השאלת. לחץ על **חצן הפעל** (Run) כדי ליצור את הטרבלה החדשה. אם קיימת כבר טבלה הנושא את שמה של טבלת היעד, Access ישאל אם ברצונך למחוק את הגירסה הישנה בטרם ייצור טבלה חדשה.

תרשים 4.25 מציג שאלת ייצור טבלה פשוטה שיזכרת עותק גיבוי של הטרבלה FamilyMembers. תיבת הדו-שיך **יצירת טבלה** (Make Table) מופיעה בעת הפעלת שאלת ייצור טבלה. באפשרות נצל את תיבת הדו-שיך כדי להגיד שם טבלה

שנמצאת בקובץ מסד הנתונים הנוכחי, או נתיב ושם קובץ במסד נתונים אחר. לאחר מכן תוכל לבחור טבלה להעתקה מותוך תיבת הרשימה הנפתחת **שם הטבלה** (Table Name).

בדוגמה שבתרשים 4.25, הבחירה FamilyMembersCopy, הטעלה FamilyMembersCopy שנמצאת במסד הנתונים הנוכחי מוגדרת בתור יעד מסד הנתונים. הערך בשורה **שדה** (Field) שברשות עיצוב השאלתה מצין כי כל השדות בטבלה FamilyMembers לוקחים חלק בטבלה החדשה. לא מצוינים קריטריונים כלשהם להכללה או לשורתה של שורתה מערכת רשותה המקור.



תרשים 4.25: שאלתת יצירה טבלה פשוטה

תכנות שאילותות באמצעות ADO ו-SQL

SQL היא שפה מקובלת ביותר לטיפול נתונים יחסיים. רוב המנגנונים לניהול מסדי נתונים יחסיים מבוססים על SQL, ומעטם כל שאילתת שיווצרם בתצוגת העיצוב ניתנת לייצור באמצעות קוד SQL. אם יש לך ידע מעשי כללי ב-SQL (שנקראות גם SQL-92), תוכל לעצב משפטים שאילתיה יעילים עבור השאילותות שלך בשפה הטבעית של מנגנוני מסדי הנתונים הללו.

הסיעיפים שלפניך מציגים שימושים מקובלים של SQL במערכות Access. ראשית, נבחן את משפטי SELECT. לאחר מכן נלמד על פונקציות להגדרת נתונים. משפטים המישימים הגדרת נתונים יכולים להתבסס ישירות על התחרbie של משפט SELECT ולשכל את רמת התפקודיות שלו. לבסוף, נסקור טכניקות SQL ליצירה והפעלה של תצוגות ושורות מאוחסנות.

משפט **SELECT**

משפט SELECT של SQL מחייב קבוצת שורות מתוך טבלה אחת או יותר של מסד נתונים. באפשרות להשתמש בפסוקיות הקשורות, כגון WHERE ו- ORDER BY כדי לשכלל בזרה דרמטית את האופן שמשפט SELECT בוחר ומציג שורות בקבוצת החזרה שלו. פונקציות צבירה של SQL כוללות אפשרות נוספת לניהול קבוצת החזרה.

משפט SELECT בסיסי מצין עמודה אחת או יותר מתוך מקור רשומות. הנה התבנית הבסיסית שלו :

```
SELECT FieldList FROM RecordSource
```

רשימות שדות

בוחרים קבוצת משנה של שדות במקור רשומות על ידי הגדרת רשימה מופרדת בפסוקים עבור העמודה FieldList. בחירות כל השדות במקור רשומות מתבצעת באמצעות כוכבית (*). אם שדה אחד או יותר בעמודה FieldList נקראים בשם זהה, אך מצויים בטבלאות שונות, עליך להקדים את שמות הטבלאות לשמות השדות ולהפריד ביניהם באמצעות נקודה. אם שם הטבלה מכיל تو רוח אחד או יותר, עליך להכניס את שם הטבלה בין סוגרים.

צירופים

כדי לציין את קשר הגומלין בין שתי טבלאות המשמשות כמקור רשומות עבור קבוצת רשומות המוחזרת המשפט SELECT, אתה משתמש **בצירופים** (joins). הפסוקית JOIN של משפט SELECT מציינת צירופים כלשהם לשילוב שתי טבלאות או יותר. צירוף חייב לציין לפחות את העמודות שישמשו לשילוב טבלאות ואת אופן קביעת ההתאמה בין הטבלאות (ניתןukanן צירופים כדי לציין קשרי גומלין בין יותר משתי טבלאות). סוגים הצירופים הטיפוסיים הם צירופים פנימיים, שבוחרים רשומות תואמות מתוך כל טבלה בזוג טבלאות; צירופים שמאליים (left joins) שכוללים את כל הרשומות מתוך הטבלה הראשונה ורק רשומות תואמות מהטבלה השנייה; וצירופים ימניים (right joins) שכוללים את כל הרשומות מתוך הטבלה השנייה ורק רשומות תואמות מהטבלה הראשונה.

פסוקיות **WHERE**

משתמשים בפסוקית WHERE של משפט SELECT להגדרת קריטריונים לבחירת שורות מתוך מקור רשומות. ביגוד לפסקיות FROM, הפסוקית WHERE היא אופציונלית ומשתמשים בה רק כדי להגביל את הרשומות בקבוצת החזרה.

האופרטורים LIKE

האופרטור האופציונלי LIKE משמש למציאת ערכים בשדות שתואימים לתבנית (pattern) מסוימת. אופרטור זה תומך בהתקנת התבנית בין המנגנונים של מסדי הנתונים Jet ו-SQL Server, אך עליך להנוג בזרירות, מכיוון שהמנגנון של שרת קבצים Jet מפרש תווים כליליים (wildcard characters) באופן שונה מזו שעושים מנגנון Microsoft SQL Server. לעומת זאת, שרת-לקות רבים (לדוגמה, MySQL) משתמש בתווים הכליליים % ו-?.

פסוקיות ORDER BY

הפסוקיות האופציונליות ORDER BY משמשת במשפט SELECT כדי להציג ערכות רשומות בסדר שונה מזו שמקتبיב המפתח הראשי. מצינינם את השדה הראשון של פיו רוצים למיין מיד לאחר מילת המפתח ORDER BY. ברירת המניין של סדר המניון היא עליה (ASC), אך באפשרות לציין גם סדר יורד (DESC). אם ברצונך למיין לפי יותר משדה אחד, הפרד בין כל צמד של שם שדה וסדר מיון, באמצעות פסיק.

הפסוקיות HAVING ו-GROUP BY

פסוקיות אלו משלימות זו את זו. הן פועלות עם פונקציות צבירה SQL כגון COUNT ו-SUM. הפסוקיות GROUP BY מצינינן שדות שלפיהם יש לחשב צבירים (aggregates). הפסוקיות HAVING דומות לפסקיות WHERE; היא מגבילה את ערכת הרשומות המוחזרת באמצעות הפסוקיות GROUP BY לרשותם שתואמות לкриיטריונים נתוניים.

מילות המפתח DISTINCT

מילת המפתח DISTINCT היא אחת הדרכים להגבילת מספר הרשומות שמווחזרות באמצעות משפט SELECT. מילת המפתח מונעת כפילות נתונים בשדות שנבחרו. הצב את מילות המפתח DISTINCT בין מילת המפתח SELECT לבין רישימת השדות. התבנית הכללית של מילת המפתח :

Select Distinct *FieldList* from *Recordset*

שימוש באובייקט Command לביצוע SQL

האובייקט Command של ADODB משמש לביצוע פקודות SQL כנגד מקור נתונים באמצעות הבא :

1. הצב במאפיין CommandText של האובייקט Command את משפט SQL שברצונך לבצע.
2. הגדר את המאפיין CommandType של האובייקט בהתאם adCmdText כדי למטר ביצועים.

.3 הפעל את השיטה Execute של האובייקט Command כדי ליצור את קבוצת החזרה של SQL.

.4 פתח אובייקט Recordset המבוסס על האובייקט Command כך שתוכל לטפל בקבוצת החזרה של SQL בישום שלך.

הקוד ש לפניך מנצח את האובייקט Command כדי לבצע משפט SQL פשוט :

```
Sub MySelect()
Dim cnn1 As New ADODB.Connection
Dim cmd1 As ADODB.Command
Dim rst1 As ADODB.Recordset
' Create the connection to another database.
cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office\Office\" & _
    "Samples\Northwind.mdb;"
' Define and execute command to select all ProductID field
' values from a single table.
Set cmd1 = New ADODB.Command
With cmd1
    .ActiveConnection = cnn1
    .CommandText = "Select ProductID from [Order Details]"
    . CommandType = adCmdText
    .Execute
End With
' Assign the return set to a recordset.
Set rst1 = New ADODB.Recordset
rst1.CursorType = adOpenStatic
rst1.LockType = adLockReadOnly
rst1.Open cmd1
Debug.Print rst1.RecordCount
End Sub
```

השлага MySelect מדפסה את מספר הרשומות בקבוצת החזרה (2155) בחלון Immediate (מיידי). קבוצת החזרה מתווך משפט SQL כוללת כל רשותה בטבלה Order Details. כבירת מחדל, המשפט SELECT בוחר את כל הרשומות שבמקור הרשומות שברקע, ומשפט SQL של השירה MySelect אינו מכיל מגבלה כלשהי לגבי הרשומות שיוחזו.

דוגמה באמצעות INNER JOIN

המשפט SELECT שלפניך ממחיש רכיבים חדשים אחדים. ראשית, הוא מציג את התחביר של צירוף פנימי בין הטבלאות Products ו-Order Details. שנית, הוא מספר את השורות בקבוצת החזרה ומדפיס אותן בחלון Immediate.

```

Sub MySelect3()
Dim cnn1 As New ADODB.Connection
Dim cmd1 As ADODB.Command
Dim rst1 As ADODB.Recordset, int1 As Integer

' Create the connection to another database.
cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
"Data Source=C:\Program Files\Microsoft Office\Office\" & _
"Samples\Northwind.mdb;"

' Define and execute command to select distinct ProductName field
' values from a pair of joined tables.
Set cmd1 = New ADODB.Command
With cmd1
    .ActiveConnection = cnn1
    .CommandText = "Select Distinct ProductName from " & _
        "[Order Details] Inner Join Products on " & _
        "[Order Details].ProductID = Products.ProductID"
    . CommandType = adCmdText
    .Execute
End With

' Assign the return set to a recordset
' and print the results to the Immediate window.
Set rst1 = New ADODB.Recordset
rst1.CursorType = adOpenStatic
rst1.LockType = adLockReadOnly
rst1.Open cmd1
Debug.Print rst1.RecordCount
For int1 = 1 To rst1.RecordCount
    Debug.Print rst1("ProductName")
    rst1.MoveNext
Next int1
End Sub

```

המשפט SELECT של השגירה MySelect3 מtabסס על שזרת מתוך שתי טבלאות, ולא אחת בלבד, כמו בדוגמה הקודמת. המשפט מבצע צירוף פנימי על הטבלה Products ועל הטבלה Order Details בין השזרות ProductID של שתיהן הטעבות. מאחר שמשפט SQL משתמש במילת המפתח DISTINCT, לא תראה החזרה של ערכים כפולים מהשדה ProductName. לבסוף, הצירוף הפנימי מאפשר לשגירה להציג את השם התיאורי של המוצר שמתאים לכל קוד זיהוי של מוצר.

דוגמה לשימוש ב- ORDER BY ו-SUM

השיגרה שלפניך משתמש בפונקציית הcentsireה SUM ובאפשרויות הצירוף והמיון שלה. השיגרה מחשבת את סכום מחיר הסיכום לשורה לכל מוצר בטבלה Order Details. קבוצת החזרה של המשפט ממיינת את התוצאות לפי גובה ההכנסה שמניב כל מוצר (מהגבוה עד לנמוכה ביותר).

```
Sub MySelect4()
Dim cnn1 As New ADODB.Connection
Dim cmd1 As ADODB.Command
Dim rst1 As ADODB.Recordset, int1 As Integer
' Create the connection to another database.
    cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office\Office\" & _
    "Samples\Northwind.mdb;"
' Define and execute command to select distinct ProductName field
' values from a pair of joined tables; compute extended price.
    Set cmd1 = New ADODB.Command
    With cmd1
        .ActiveConnection = cnn1
        .CommandText = "Select Distinct Products.ProductName, " & _
            "Sum([Order Details].[UnitPrice]*" & _
            "[Order Details].[Quantity]*" & _
            "(1-[Order Details].[Discount])) As [Extended Price] " & _
            "From Products Inner Join [Order Details] On " & _
            "Products.ProductID = [Order Details].ProductID " & _
            "Group By Products.ProductName " & _
            "Order By Sum([Order Details].[UnitPrice]*" & _
            "[Order Details].[Quantity]*" & _
            "(1-[Order Details].[Discount])) Desc"
        .CommandType = adCmdText
        .Execute
    End With
' Assign Select statement return set to a recordset
' and print the results to the Immediate window.
    Set rst1 = New ADODB.Recordset
    rst1.CursorType = adOpenKeyset
    rst1.Open cmd1
    Debug.Print rst1.RecordCount
    For int1 = 1 To rst1.RecordCount
        Debug.Print rst1("ProductName"), rst1.Fields("Extended Price")
        rst1.MoveNext
    Next int1
End Sub
```

השיגרה MySelect4 יוצרת שדה מחושב באמצעות קוד. השיגרה מפעילה את הפונקציה SUM עם הפסוקית GROUP BY כדי לחשב את ההכנסה שיצר כל אחד מהמושרים. לו לא פסוקית זו, הקוד היה יוצר את השדה המוחשב עבור כל שורה בטבלה המקורית, אך לא היה מספק תוצאות סיכום לפי קבוצה (הוא היה מחשב את ההכנסה הכוללת של כל המושרים, במקום את ההכנסה הכוללת של כל מוצר בנפרד).

הפסוקית ORDER BY שולטת בסדר המיוון של קבוצת החזרה. למרות שניתנו כתוב קוד זה בצורה תמציתית (הקוד זהה לקוד שמחשוב את מחיר הסיכום לשורה), קל יותר להבינו כפי שהוא כתוב, והוא דומה לקוד שיצר רשות עיזוב השאלה.

פונקציות הגדרת נתונים

קיימות כמה דרכים להגדיר נתונים באמצעות משפטים SQL. בסעיף זה נבחן כיצד ליצור טבלה באמצעות שאלות יצירה טבלה. שאלות יצירה הן שאלות פועלות היוצרות טבלה חדשה המבוססת על קבוצת החזרה של שאלתה קיימת. זהה פונקציית הגדרת נתונים.

בסעיף זה גם נדגים כיצד לשנות שדה שגודל בתוספת קבועה באופן אוטומטי באמצעות מילוט המפתח של SQL – ALTER COLUMN ו- ALTER TABLE. Jet 4 SQL (step) סוג נתונים שדה זיהוי שמאפשר לעשות זאת. Jet 4 תומך בהגדרת ערכיו צעד (start) ותחילה (end) עבור שדות מונה אלה (ניתן לאפס אותם בכל עט). שגרות הדוגמה בסעיף זה כוללות כבר את כל הרכיבים התפקידים, אך ערךןعلاה אם תבחן אותן לאחר האמור בפרק 3. CREATE TABLE ו- ALTER COLUMN הן חלק מקבוצה של מילוט מפתח (לדוגמה, CREATE INDEX, DROP CREATE TABLE) שתומכות ישירות בפונקציות הגדרת נתונים באמצעות SQL.

SELECT...INTO

לפניך התחבר שאלות יצירה טבלה באמצעות SQL:

```
SELECT FieldList INTO NewTableName FROM RecordSource
```

אם תיצור טבלה חדשה בקובץ מסד נתונים שונה מזו הנוכחי, הוסף פסוקית IN אחרי הפסוקית INTO ולפני הפסוקית FROM. השימוש בפסוקית IN כדי ליעיד את הנטיב ואת שם הקובץ של מסד הנתונים שישמר את פלט המשפט SELECT...INTO.

RecordSource יכול להתבסס על טבלה אחת או יותר, או שאלתה אחת או יותר ועליו להכיל את כל השדות שברצונך לכלול בטבלה החדשה שאתה יוצר. המשפט SELECT...INTO מעתיק את עיצוב השדות שנבחרו ואת נתוניהם לטבלה החדשה. אין אפשרות לכך ליצור שדות חדשים כלשהם באמצעות המשפט SELECT...INTO. תוכל ליצור בקלות את התבנית הכללית של שאלתה זו על ידי יצירתה בתצוגת העיצוב ולאחר מכן לעبور לתצוגת SQL ולהעתיק את משפט SQL אל שיגרה שמשתמשת באובייקט ADO. אם תאמץ גישה זו, לא תצטרך לתכנן את משפט SQL.

השיגרה שלפניך משתמשת בגרסה פשוטה של משפט SELECT...INTO כדי לאבות את הטבלה FamilyMembers שבבסיס הנתונים הנוכחי אל טבלה חדשה שנקראת .FMBackup

```
Sub MyMakeTable()
Dim cnn1 As ADODB.Connection

' Reference connection.
Set cnn1 = CurrentProject.Connection

' Execute SQL for maketable query.
cnn1.Execute "SELECT FamilyMembers.* INTO " & _
    "FMBackup FROM FamilyMembers"
' This procedure fails if FMBackup already exists.

End Sub
```

למרבה הצער, בכך קל כדי ליצור ולהפעיל משפט SELECT...INTO לא די בשני משפטי ADO הללו. לדוגמה, שיגרה זו תיכשל אם הטבלה FMBackup כבר קיימת. השיגרה יכולה להיכשל גם אם משתמש נוסך יפתח את אחת הطالאות. קשיים אלה ואחרים מחיברים מגנוון לכלכית שגיאות. הדוגמה שלפניך ממחישה מגנוון אפשרי לכלcitת .SELECT...INTO.

```
Sub MyMakeTable2()
On Error GoTo Make2Err:
Dim cnn1 As ADODB.Connection

' Reference connection.
Set cnn1 = CurrentProject.Connection

' Test for unanticipated errors.
' Err.Raise 1

' Execute SQL for make-table query.
cnn1.Execute "SELECT FamilyMembers.* INTO " & _
    "FMBackup FROM FamilyMembers"

Make2Exit:
' Close the connection and set it equal to nothing and exit the sub.
cnn1.Close
Set cnn1 = Nothing
Exit Sub
```

```

Make2Err:
' Trap for table already exists.
If Err.Number = -2147217900 Then
    cnn1.Execute "Drop Table FMBackup"
    Resume
Else
    MsgBox "The program generated an unanticipated " & _
        "error. Its number and description are " & _
        Err.Number & ":" & Err.Description, vbCritical, _
        "Programming Microsoft Access 2000"
End If
Resume Make2Exit
End Sub

```

שיגרה זו מבצעת משימה זהה לזו של השגירה המקורית, אך היא לא תיכשל גם אם הטבלה FMBackup כבר קיימת. המטפל בשגיאות מזוהה שגיאה מסווג זה ומוחק את הטבלה הקיימת בטרם תבצע את הפקודה פעם נוספת. במקרה של שגיאה אחרת, היישום יציג את המידע החינוי בתיבת הודעה כך שהמשתמש יוכל להודיע למפתח היישום על הבעיה.

השגירה כוללת רכיב נוסף ללכידת שגיאות: מיד לאחר הקמת החיבור, הקוד יכול לקרוא לשיטה Raise של האובייקט Err (הרכיב מסומן בתור הערא, אך דאגתי להסיר את הסימון כדי להעריך את ביצועי מגננון לכידת השגיאות). שימוש זה באובייקט Err יוצר שגיאה מלאכותית שמשמעותה לגנות כיצד יש לעצב את השגירה כך שתגיב לשגיאות בלתי צפויות.

הדוגמה גם סוגרת את החיבור ומנדרה את האובייקט Connection בתור Nothing. שתி פעולות אלו דומות בייעודן ומשמעותן זו איז. בעת סגירת חיבור, המשאים לשימושו לצורך החיבור הופכים זמינים לשאר המערכת. יחד עם זאת, האובייקט נשאר בזיכרון, ומופיענו אינם משתנים. בשלב מאוחר יותר, תוכל לפתח את אותו אובייקט Connection עם הגדרות המאפיינים הקודמות שלו או עם הגדרות שונות חדשות. עליך להגדיר אובייקט בתור Nothing כדי להסיר אותו מהזיכרון.

שימוש ב- ALTER COLUMN וב- ALTER TABLE כדי לאפס שדות Autoincrement

פתחים רבים לסייעת Access יתלהבואפשרויות השילטה החדשות בסוגי נתונים Counter (מונה) המשמשים כשדות זיהוי של רשומות. בפרק 3 למדנו כבר כיצד לקבע את ערכי ההתחלה והצעד (תוספה) של שדות מונה. מיד נלמד דרכים נוספות להגדלת רמת השילטה בשדות אלה. השגירה ש לפניה מגדירה את ערכי ההתחלה והצעד של סוג הנתונים Counter.

```

Sub ResetCounter(intStart, intStep)
Dim cnn1 As ADODB.Connection
Dim strSQL
' Reference connection and execute SQL for view.
    Set cnn1 = CurrentProject.Connection
' Create SQL string that references passed arguments.
    strSQL = "ALTER TABLE FamilyMemberNames " & _
        "ALTER COLUMN FamID Identity " & _
        "(" & intStart & "," & intStep & ")"
' Execute the SQL statement to update the counter.
    cnn1.Execute strSQL
End Sub

```

השיגרה ResetCounter מדגימה כיצד להפוך פקודת SQL לדינמית על ידי הוספת ארגומנטים למשפט SQL. תחילה זה דומה להעברת ערכים אל שאלות פרמטר. השיגרה ResetCounter משנה את ערך השדה FamID בטבלה FamilyMemberNames באמצעות שני ארגומנטים. ארגומט אחד הוא ערך שדה המונה החדש של הרשימה והולמת שני ארגומנטים. הארגומנט השני הוא גודל הצעד עבור הרשומות הבאות הבאה שתווסף לטבלה; הארגומנט השני הוא גודל הצעד עבור הרשומות הבאות חדשות. באפשרות לפעיל את השיגרה על ידי הקלדת שורה מעין זו בחולן המידי:

ResetCounter 2,2

שני ארגומנטים אלה מאלצים מתן ערך 2 לשדה ID FamID ברשימה הבאה שמוסיפה לטבלה. כל ערך שדה FamID בכל רשומה נוסף יגדל ב-2 לעומת השדה ברשימה שקדמה לו. היוזר בקביעת ערכי ההתחלה והצעד, מכיוון שאתה עלול לגרום לטעות כללי מפתח ראשי (כגון ערכים כפולים) וכך לשולב מהמשתמשים את הזכות להוסיף רשומות חדשות לטבלה. למקרה המזל, Access 2000 מאפשר לסקור באמצעות קוד את ערכי שדה המונה הקיימים, ולכן תוכל ליצור קוד שיהיה מוגן מפני בעיה זו. השיגרה הבאה מדגימה זאת.

משפט SQL הוא לב-לייה של ResetCounter. כפי שניתן להאות, השיגרה כוללת שלושה מרכיבים קריטיים. מילוט המפתח ALTER TABLE משנות את עיצוב הטבלה שצוינה. מילוט המפתח ALTER COLUMN משנות את עיצוב השדה שצוין. מילוט המפתח IDENTITY מעדכנת את שדה המונה בהתאם לארוגמנטים של הערך ההתחלתי וערך הצעד של השדה.

השיגרה SetResetCounter שלפניך ארוכה, אך אינה מורכבת ביותר. היא מנצלת שיטה שטפלת בערכי שדה מונה מבלי לגרום להפרות כללי מפתח רשומה. תרשימים 4.26 מציג את הטבלה כפי שנראית מיד לאחר הפעלה מוצלחת של השיגרה SetResetCounter.שים לב כי השדה ID ברשימה הראשונה מכיל את הערך 2, ואילו השדה FamID ברשימה הבאה אחרתה מכיל ערך גודל בצעד של 2. השדה FamID ברשימה השלישית מכיל את הערך 8, ולכן איןנו עקייב לערך הצעד. בנוספ, ערך השדה FamID ברשומות הבאות גודל בצעד של 4 בכל רשומה חדשה. השינוי בערכי ההתחלה והצעד של שדה המונה נובע מהקריאה ל-ResetCounter. מחרוזת טקסט בשדה MyMemoField מסמנת את השינוי בערכים אלה.

FamilyMemberNames : Table			
	FamID	PersonName	MyMemoField
▶	2	Rick Dobson	start, step = 2
	4	Virginia Dobson	
	8	Glen Hill	see my new start & step
	12	Tony Hill	
	16	Shelly Hill	
*	(AutoNumber)		

Record: 1 * of 5

תרשים 4.26: הטבלה FamilyMemberNames לאחר ביצוע השגירה SetResetCounter

```

Sub SetResetCounter()
Dim cnn1 As New ADODB.Connection
Dim cmd1 As ADODB.Command
Dim rst1 As ADODB.Recordset, rst2 As New ADODB.Recordset
Dim int1 As Integer
' Reference connection.
    Set cnn1 = CurrentProject.Connection
' Clear FamilyMemberNames table and reset table counter.
    DoCmd.SetWarnings False
    DoCmd.OpenQuery "qdlAllFamilyMemberNames"
    DoCmd.SetWarnings True
    ResetCounter 2, 2
' Add a couple of records to the FamilyMemberNamestable from the
' FamilyMembers table with the initial start and step values.
    Set rst1 = New ADODB.Recordset
    rst1.Open "FamilyMemberNames", cnn1, adOpenKeyset, _
        adLockOptimistic, adCmdTable
    rst2.Open "FamilyMembers", cnn1, adOpenForwardOnly, _
        adLockReadOnly, adCmdTable
    For int1 = 1 To 2
        rst1.AddNew
        rst1(1) = rst2(1) & " " & rst2(2)
        If int1 = 1 Then
            rst1("MyMemoField") = "start, step = 2"
        End If
        rst2.MoveNext
        rst1.Update
    Next int1
    rst1.Close

```

```

' Define and execute command to select all FamID field values
' with highest FamID value first --> last counter value.
    Set cmd1 = New ADODB.Command
    With cmd1
        .ActiveConnection = cnn1
        .CommandText = "Select FamID from FamilyMemberNames " & _
            "Order By FamID Desc"
        . CommandType = adCmdText
        .Execute
    End With

' Save last counter value, and use it to reset start and step to new values.
    Set rst1 = New ADODB.Recordset
    rst1.CursorType = adOpenForwardOnly
    rst1.LockType = adLockReadOnly
    rst1.Open cmd1
    int1 = rst1(0)
    rst1.Close
    ResetCounter int1 + int1, int1

' Add remaining records from the FamilyMembers table
to the FamilyMemberNames
' table with the new start and step values.
    rst1.Open "FamilyMemberNames", cnn1, adOpenKeyset, _
        adLockOptimistic, adCmdTable
'   rst1(2) = "see my new start & step"
    int1 = 3
    Do Until rst2.EOF
        rst1.AddNew
        rst1(1) = rst2(1) & " " & rst2(2)
        If int1 = 3 Then
            rst1("MyMemoField") = "see my new start & step"
        End If
        rst2.MoveNext
        int1 = int1 + 1
        rst1.Update
    Loop

```

השлага SetResetCounter כוללת שישה חלקים המופרדים בהערות. החלק הראשון מקיים חיבור אל מסד הנתונים הנוכחי. החלק השני מפעיל שאילתת מחיקה שمسירה את כל השורות מתוך הטבלה FamilyMemberNames, מאפשר להציג אזהרות מערכת על ידי קריאה לשיטה SetWarning, ולאחר מכן מגדרה את ערכיו ההתחלת והצעד של המונה של הטבלה FamilyMemberNames על ידי קריאה לשлага ResetCounter.

החלק השלישי משתמש בולאה Next...For כדי להעתיק את שתי הרשומות הראשונות של הטבלה FamilyMembers אל הטבלה FamilyMemberNames. מנגנון מסד הנתונים Jet מזכה אוטומטית ערכי FamID לשתי הרשומות הללו באמצעות ערכי ההתחלה והצעד שהוגדרו בחלק הקודם של השירה.

החלק השלישי מפעיל את האובייקט Command כדי למצוא את השדה FamID בעל הערך הגדול ביותר בטבלה FamilyMemberNames. כשהפקודה מתבצעת, היא ממיינת את הרשומות בטבלה בסדר יורד לפי השדה FamID. כתוצאה לכך, הרשימה בעלת הערך הגדול ביותר בטבלה FamID – המפתח הראשי של הרשימה – מוצבת בראש קבוצת החזרה. ידיעת ערך זה מאפשרת לישום לקבוע ערך התחלת חדש עבור רשומות נוספות, כזה שאינו משבכל את ערך המפתח הראשי של רשומה כלשהי בטבלה.

החלק החמישי פותח ערכת רשומות בקבוצת החזרה של הפקודה ומאחסן את ערך השדה FamID של הרשימה הראשונה בקבוצה. לאחר מכן, חלק זה מפעיל את השירה ResetCounter כדי להגדיר ערך התחלת חדש כפול מהערך הגדול ביותר בטבלה FamID וגם ערך חדש שווה לערך הנוכחי בטבלה FamID.

החלק האחרון מוסיף את הרשומות שנותרו בטבלה FamilyMembers לטבלה FamilyMemberNames. כדי להגדיל את גמישות השליטה בשדות מפתח ראשי מסווג מונה, העמודה Memo מכילה הודעה המציינת כיצד השירה SetRetCounter מטפלת ללא בעיה בשדה המונה.

תצוגות ושורות מאוחסנות

Access 2000 כולל תצוגות ושורות מאוחסנות שהיו פעם זמינות רק במערכות מתכדמות לניהול מסדי נתונים. Access 2000 מיישם תצוגות, שאין תומכות בפרמטרים, בתור שאילתות מאוחסנות שמחזירות שורות. Access 2000 כולל שורות מאוחסנות בתור שאילותות פעולה מאוחסנות וגם שאילותות פרטיר הפעולות שאילותות פעולה או שאילותות בחירה. כדי שתיווכח לדעת, ניתן להשיג תפקודיות של פרטיר על ידי העברת קבועי מחוזות לשירה שמזגת את הערכים המועברים אל משפט SQL, וביצוע המשפט.

תצוגות

המשפט CREATE VIEW יוצר תצוגה על ידי הוספת שאילתת בחירה מאוחסנת לחלוון מסד הנתונים תחת קטגוריית האובייקטים **שאילותות** (Queries) (בפרויקט Queries (*.adp), התצוגות שמורות תחת קטגוריית האובייקטים **תצוגות** (Views) של חלון CREATE VIEW מאחר במאפייניהם). תהליך היצירה של שאילתת בחירה שומרה באמצעות CREATE VIEW יואר מאחר באמצעות ADO, מה שחייב יצירת אובייקט מסווג Views והוספתו לאוסף Views.

השירה שלפניך מדגימה את התחבר ו敖פן הפעולה של משפטי VIEW בקוד SQL של ADO.

```

Sub CreateView()
Dim cnn1 As ADODB.Connection
' Reference connection.
    Set cnn1 = CurrentProject.Connection
' Execute SQL for view.
    cnn1.Execute "Create View CategoryView as SELECT * From Categories"
    RefreshDatabaseWindow

'This simple routine fails if CategoryView already exists.
End Sub

```

השлага מפעילה את השיטה Execute של החיבור ומעבירה אליה משפט CREATE VIEW . שס שאלת הבירה בא מייד לאחר מילות המפתח CREATE VIEW ואחריו מילת המפתח AS ומשפט SELECT . המשפט SELECT יכול להיות משפט קלשו מהסוג המקובל. לבסוף, השлага קוראת ל- RefreshDatabaseWindow כדי לעדכן את חלון מסד הנתונים כדי שהמשתמש יוכל להציג את השאלתה החדשה מבלי שיצטרך לרענן מחדש את החלון.

ל- CreateView יש חיסרון : היא נכשלת אם התצוגה שהיא מנשה ליצור, קיימת כבר. באפשרותך, כמובן, להשיר בעצמך את השאלתת היישנה, אך ניתן למחוק אותה גם באמצעות קוד. שתי השגרות שלפניך ממחישות גישה אפשרית לפתרון הבעיה :

```

Sub CreateReplaceView(ViewName As String)
Dim cnn1 As ADODB.Connection
Dim cat1 As ADOX.Catalog
Dim vew1 As ADOX.View
' Reference objects for view
    Set cnn1 = CurrentProject.Connection
    Set cat1 = New Catalog
    cat1.ActiveConnection = cnn1

Enumerate views.
    For Each vew1 In cat1.Views
' Delete named view and replace it with a new version.
        If vew1.Name = ViewName Then
            cat1.Views.Delete ViewName
            CreateCustomView ViewName
            Exit Sub
        End If
    Next vew1

' If the view is not there, create from scratch.
    CreateCustomView ViewName
    RefreshDatabaseWindow
End Sub

```

```

Sub CreateCustomView(MyViewName)
Dim cnn1 As ADODB.Connection, strSQL
' Reference connection and execute SQL for view.
    Set cnn1 = CurrentProject.Connection

' Create SQL string for view.
    strSQL = "Create View " & MyViewName & " as Select * From Categories"
' Create the custom view.
    cnn1.Execute strSQL
    RefreshDatabaseWindow

End Sub

```

השגרה CreateReplaceView אינה מבצעת ישירות את המשפט CREATE VIEW ; היא משאירת מנגינה זו לשגרה השניה, CreateCustomView . CreateCustomView מספרת את התצוגות כדי להערך את הצורך למחוק את התצוגה הישנה לפני שמנסים ליצור תצוגה חדשה בשם זהה. לאחר מכן היא קוראת לשגרה CreateCustomView ומעבירה את שם התצוגה שיש ליצור. CreateCustomView ביצובה לשגרה שבודגמה הקודמת, אך היא מנצלת שרשור מחוזות עם הארגומנט המועבר וקבעו מחוזות, כדי להרכיבמשפט SQL עבור השיטה של החיבור.

יוצרים תצוגה חדשה על ידי הקלדה של פקודה פשוטה בחילון Immediate , הנה כך :

```
CreateReplaceView "NameOfView"
```

שם התצוגה יופיע בין מרכאות כפולות. ניתן לשנות בקלות את View CreateReplaceView כדי לאפשר משפט SELECT דינמי.

במוקם לעבור בזורה מחזורת על מרכיבי האוסף Views , תוכל פשוט לנסתות ליצור תצוגה חדשה וללכוד את השגיאה שנגרמת כתוצאה מתצוגה בעלת שם זהה. מספר השגיאה של תצוגה כפולה הוא 2147467259 . לכוד מספר שגיאה זה ומחק את התצוגה הישנה בטרם תבצע מחדש את המשפט CREATE VIEW FieldView . השגרה CreateReplaceView יישר את המשפט CREATE VIEW עבור מ machisha גישה זו.

שגרות מאוחסנות

שגרה מאוחסנת מפשtot את השימוש החזר בקוד SQL. שגרות מאוחסנות מבצעות פעולות כלויות, כגון שאילתות מחיקה, עדכון או הוספה וקולוטות פרמטרים בזמן ריצה הן עברו פעולות כלויות והן עברו שאילתות החזרת-שורות.

משתמשים במשפט CREATE PROC כדי ליצור שגרה מאוחסנת. התחבר הכללי של CREATE PROC זה זהה לזה של המשפט CREATE VIEW . לאחר המילים מקלדים את שם השגרה המאוחסנת ואת מילת המפתח AS , המשמשת עבור סמן ההתחלה של קוד SQL המגדיר את התנהלות השגרה המאוחסנת. באפשרות לעצב שאילתות פעולה בתצוגת עיצוב השאילתת של Access , להעתיק את קוד SQL בתצוגת SQL ולאחר מכן לבדוק את הקוד שהעתיק אחריו מילת המפתח AS .

שגרות מאוחסנות אינן מופיעות בחולון **מסך נתונים** (Database). לעומת זאת, הן קיימות בתור פריטים באוסף **Procedures** של האובייקט **Catalog**. הקוד יכול להתייחס לשגרות ייחודות באמצעות סכמת אובייקט היררכית זו. לדוגמה, אם היישום יוצר שיגרה מאוחסנת שמוחקת רשומות, השיגרה קיימת באוסף **Procedures** (יחד עם שאלות הפעולה בפרויקט).

השיגרה שלפניך מציגה את התחביר שמיישם את המשפט CREATE PROC. השיגרה מפעילה את השיטה Execute של האובייקט Connection ומעבירה משפט SQL בתור DeleteThese. המשפט CREATE PROC מנסה ליצור שאלתה שנקראת מהרוות. השאלתה מוחקת את כל הרשומות של הטבלה FamilyMemberNames שעריך השדה FamID שלhn הוא 10 או יותר. אם לא נקבעו אמצעים מיוחדים, השיגרה תיכשל במידה שהשאלתה DeleteThese כבר קיימת. כדי לאפשר להפעיל את השיגרה גם אם היא כבר קיימת, הקוד ממשיק להתבצע מהשורה הבאה במקרה של שגיאה.

```
Sub CreateProcToDelete()
On Error Resume Next
Dim cnn1 As ADODB.Connection
Dim cat1 As Catalog
Dim proc1 As Procedure
Dim cmd1 As Command
' Reference connection.
    Set cnn1 = CurrentProject.Connection
    Set cat1 = New Catalog
    cat1.ActiveConnection = cnn1
    Set cmd1 = New Command
    cmd1.ActiveConnection = cnn1

' Execute SQL to make the procedure.
    cnn1.Execute "Create Procedure DeleteThese As " & _
        "Delete From FamilyMemberNames Where FamID>=10"

' Enumerate Procedures collection members.
' For DeleteThese procedure, set command properties and execute it.
    For Each proc1 In cat1.Procedures
        If proc1.Name = "DeleteThese" Then
            cmd1.CommandText = proc1.Name
            cmd1.CommandType = adCmdStoredProc
            cmd1.Execute
        End If
    Next proc1
End Sub
```

השיגרה CreateProcToDelete מתחילה בסדרת הזרות והצבות שתפקידן לתמוך בלוגיקה לייצור והפעלה של השיגרה. השיטה Execute של האובייקט Connection יוצרת את השיגרה על ידי הפעלת משפט SQL. בשלב זה, אם השיגרה כבר קיימת, תתרחש שגיאה. השיטה Execute מוסיפה בפועל את השיגרה DeleteThese לאוסף Procedures בקטלוג של החיבור.

שגורות שכנות באוסף, ולכן ניתן למסוף את חברי האוסף באמצעות לולאת For...Each. הדוגמה משתמשת בלולאה כזו כדי לאטר את DeleteThese. CommandText של האובייקט Command מציין את המאפיין Name (שם) של השיגרה במאפיין CommandType של האובייקט CommandType בטור Command.adCmdStoredProc. שתי הצבות אלו יוצרות את מחרוזת SQL שנקראת Execute DeleteThese. השיטה Execute של האובייקט Command מפעילה את השיגרה ב-Jet ומסירה את הרשומות מהטבלה FamilyMemberNames.

באפשרותך ליצור גירסה גמישה ונבונה של הדוגמה הקודמת בכמה דרכים. ראשית, מנגנון משוכל יותר לכלידת שגיאות יכול לבדוק סוג שגיאות אחרים שמקורם בשיגרה. שנית, אפשרותך לגרום לשיגרה לקבל פרמטרים עבור השדה FamID. הדוגמאות שלפניך מוצלות את המשפט CREATE PROC FamID FamilyMemberNames שלhn שモחקת את כל הרשומות שבטבלה FamilyMemberNames שערץ השדה FamID שלוhn שווה או גדול מערך שהוגדר בזמן ריצעה. הקוד מזכיר את שלבי הגדרת ערך פרמטר של פקודה.שים לב כי עלייך ליצור את הפרמטר, להוציאו לאוסף Parameters ולאחר מכן להציג בו ערך. שגרת הדוגמה גם לוכדת שגיאות שמקורן בניסיון ליצור שיגרה בשם זהה לשם של שיגרה קיימת.

```
Sub CreateProcToDelete2()
On Error GoTo Delete2Err
Dim cnn1 As ADODB.Connection
Dim cmd1 As ADODB.Command
Dim prm1 As Parameter

'Reference connection
Set cnn1 = CurrentProject.Connection

'Test for unanticipated errors
' Err.Raise 1

'Execute SQL to make the procedure
cnn1.Execute "Create Proc DeleteThese " & _
    "(Parameter1 Long) As " & _
    "Delete From FamilyMemberNames " & _
    "Where FamID>=Parameter1"

```

```

'Assign SQL from procedure to command
Set cmd1 = New ADODB.Command
With cmd1
    .ActiveConnection = cnn1
    .CommandText = "DeleteThese"
    . CommandType = adCmdStoredProc
End With

'Set the procedure's parameter
Set prm1 = cmd1.CreateParameter("Parameter1", _
    adInteger)
cmd1.Parameters.Append prm1
prm1.Value = 10

'Invoke the procedure's SQL statement in the command
cmd1.Execute

Delete2Exit:
    Exit Sub

Delete2Err:
    If Err.Number = -2147217900 Then
        'Trap for procedure already exists
        cnn1.Execute "Drop Proc DeleteThese"
        Resume
    Else
        MsgBox "The program generated an unanticipated " & _
            "error. Its number and description are " & _
            Err.Number & ":" & Err.Description, vbCritical, _
            "Programming Microsoft Access 2000"
    End If
    Resume Delete2Exit
End Sub

```

בעת ציון פרמטר במשפט CREATE PROC, הכלל את הצבת הפרמטר לאחר שם השיגורה, אך לפני מילת המפתח AS. באפשרות להציג את שמה ואת סוג הנתונים שלה בפרמטרים, כפי שניתן לראות בדוגמה. מפנים שנייה לפרמטר בפסוקית WHERE, כדי להגביל את פועלות השיגורה. התחבריר זהה של שאלותות פרמטר שיוצרים בתצוגת **עיצוב**.

שים לב כי איןך צריך בעצם לעבור בלולאה על האוסף **Procedures** כדי להפעיל שיגורה ספציפית. באפשרות להתייחס לשמה כל עוד תגדיר את הקבוע 10 בתווך הערך של מאפיין הפקודה, CommandType. הדוגמה מציבה את הקבוע 10 בפרמטר (Parameter1) כדי לשמור על עקביות עם הדוגמה הקודמת, אך תוכל להשתמש במשפט הפונקציה InputBox או בטופס כדי לקלוט את הערך עבור הפרמטר.

הדוגמא גם כוללת מנגנון בסיסי לכידת שגיאות במצב שקיימת כבר שיגרה פעולה שם זהה (מספר שגיאה 2147217900-). השגירה CreateProcToDelete2 עשויה דבר פשוט: היא משליכה את השגירה הקיימת במקורה של שגיאה על ידי הפעלת המשפט DROP PROC *ProcedureName*. תחביר המשפט הוא "DROP PROC *ProcedureName*".

הפעלת שאילותות על מקורות נתונים מרוחקים

עד כה, הדוגמאות שהציגנו בפרק התיחסו למקורות נתונים מקומיים, או כאלה שנמצאים בקובץ Access אחר. Access 2000 מאפשר לעבוד עם גם מקורות נתונים מרוחקים, כגון Microsoft SQL Server או Oracle. בסעיף זה עוסק בנושא בזורה תמציתית; בפרק 12 נתייחס אליו בהרחבה.

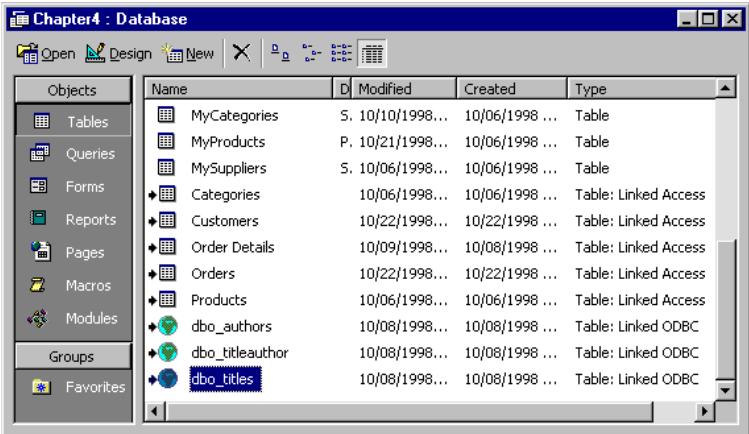
הפעלת שאילותות על מקורות ODBC מוקשיים

באפשרותך לבצע שאילתתה על מקור הקשור בדיזוק כפי שאתה עושים עם מקור נתונים של Access שנמצא בקובץ אחר. תרשימים 4.27 מציג את חלון מסד הנתונים וボן שלוש טבלאות הקשורות במסד נתונים Pubs שמצוורף למשריך SQL Server. פועל בהתאם לשלבים שלහן כדי לקשר את היישום לטבלה שנמצאת במסד נתונים מרוחק:

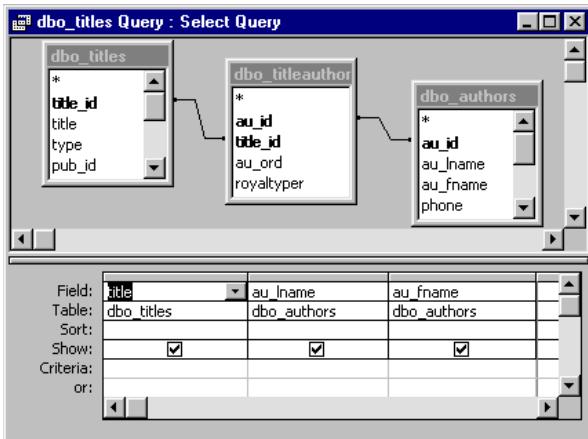
1. בחר באפשרות **קבל נתונים חיצוניים** (Get External Data) מתפריט **קובץ** (File) של Access ולאחר מכן לחץ על **קישור טבלאות** (Link Tables).
2. בתיבת הדיו-שיכון **קישור** (Link), בחר באפשרות **ODBC Databases** מהרשימה הנפתחת **קבצים מסווג** (Files Of Type).
3. בחר ב-DSN בכרטיסיה **Machine Data Source** (באפשרות ליצור DSN חדש אם זה נדרש לך אין זמין).
4. השלם את הקישור על ידי בחירת טבלה ולהיכזה על **OK**.

마חר שכמעט כל מקורות הנתונים תומכים ב-ODBC, דרך טיפול זו בנתונים מרוחקים מספקת גישה אוניברסלית באמצעות המשק המוכר של Access.

החלון העליון בתרשימים 4.28 מציג את שלוש הטבלאות הקשורות שבתרשים 4.27 בתוצאות עיצוב שאילתתה. לאחר יצירת הקישור אל טבלה מרוחקת, באפשרותך להתייחס אליה כאיilo היתה טבלה מקומית. טבלאות במקורות נתונים מרוחקים עלולות להיות גדולות מאוד, ולכן יש להתחשב בהן בזורה מיוחדת בעת עיצוב שאילותות המפנות אליהם. הסעיף הראשון בפרק זה שופך אור על כמה טכניקות אפשריות. גודל הטבלאות וסוג החיבור הפיסי למקור הנתונים המרוחק יסייע לקבוע את עיצוב השאילתת הטוב ביותר. החלון התיכון בתרשימים 4.28 מציג את השאילתת שבחלון העליון בתוצאות **גיליון נתונים**.



תרשים 4.27: חלון מסד נתונים זה מציג שלוש טבלאות הקשורות – dbo_authors – dbo_titles ו-dbo_authors. אפשרות להשתמש במנהל התקנים של ODBC כדי ליצור קישור אל טבלאות שנמצאות במקור נתונים כלשהו באמצעות ניהול המזון של ODBC



This screenshot shows the results of the query 'dbo_titles Query : Select Query'. The results are displayed in a table with columns: title, au_lname, and au_fname. The data consists of 25 records, each representing a book and its author(s). The first few records are: 'The Busy Executive's Database Guide' by Bennett Abraham, 'Fifty Years in Buckingham Palace Kitchens' by Blotchet-Halls Reginald, 'But Is It User Friendly?' by Carson Cheryl, 'The Gourmet Microwave' by DeFrance Michel, 'Silicon Valley Gastronomic Treats' by del Castillo Innes, 'Secrets of Silicon Valley' by Dull Ann, 'The Busy Executive's Database Guide' by Green Marjorie, and 'You Can Combat Computer Stress!' by Green Marjorie.

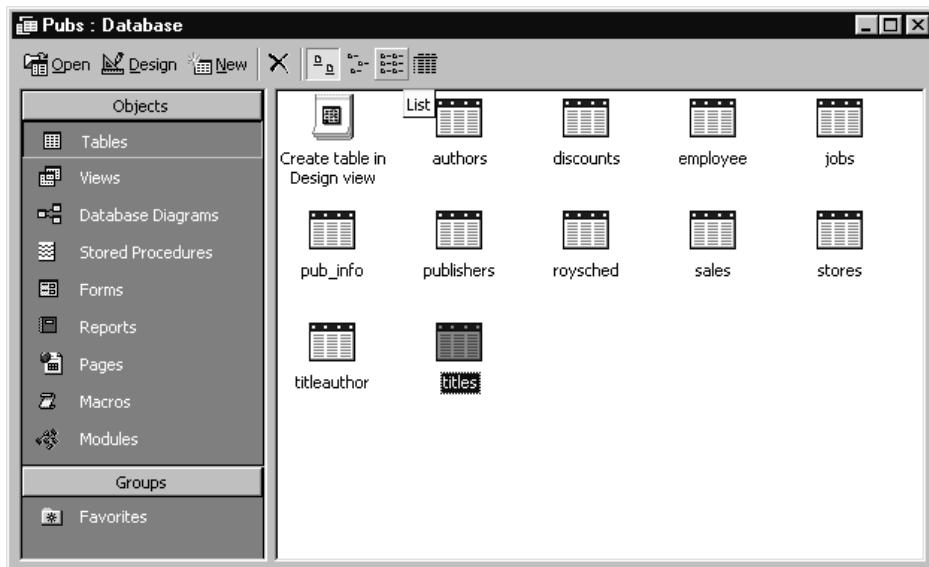
תרשים 4.28: שאלתה פשוטה המבוססת על שלוש טבלאות מקור נתונים מרוחק. החלון העליון מראה כי ניתן ליצור שאלות אל טבלאות הקשורות מרוחקות כאלו היי מקומיות. החלון התחתון מציג את השאלתה בתצוגת גליון נתונים

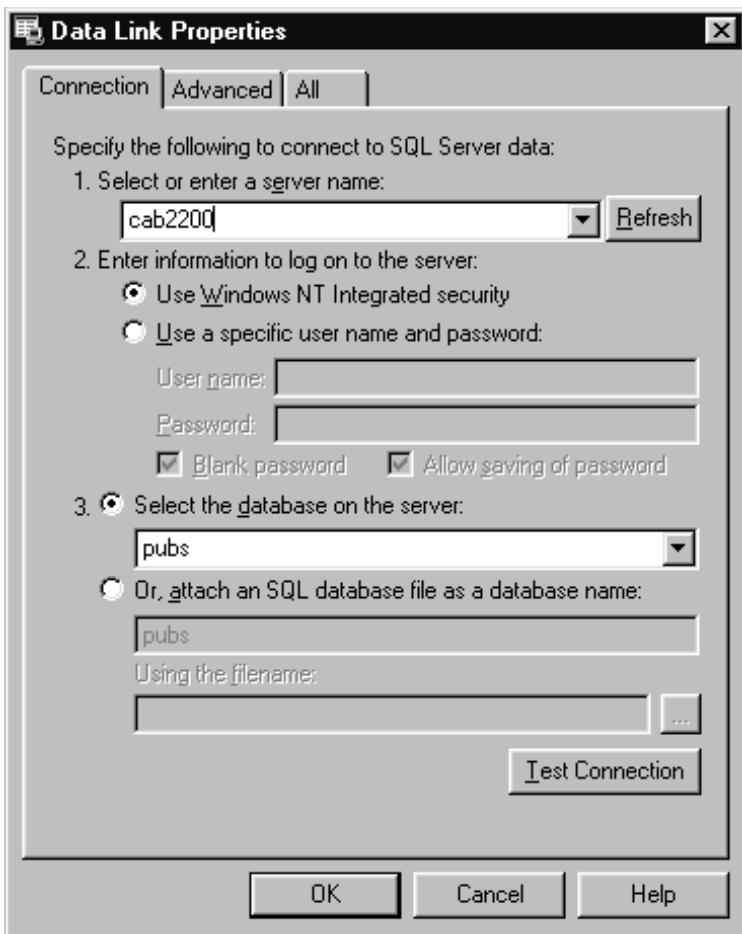
הפעלת שאלות בפרויקט נתונים של Access

Access כולל סוג פרויקט חדש שנקרא Access Project 2000, אשר עוקף את מנגנון מסד הנתונים Jet על ידי יצירת קישור ישיר אל גרסאות 6.5 או 7.0 של SQL Server וגם אל Microsoft Data Engine. Microsoft Data Engine הוא הגרסה החדשה של SQL Server עבור סביבת Windows 95/98. סביבה זו מאפשרת ליצור קישור ישיר אל מסד נתונים מרוחק ולטפל בו כאילו היה מסד נתונים מקומי.

החלון העליון בתרשימים 4.29 מציג את חלון **מסד נתונים של פרויקט Access**. שם לבי השווה זהה במרקחיו לחלון **מסד נתונים של מסד נתונים mdb**.*, אך מקור הנתונים שלו הוא מסד הנתונים Pubs (באפשרות להחליפם את מהויסם **ממקורה אל חלון מסד הנתונים על ידי בחירת התקשרות (Connection) מתפריט קובץ (File)**).

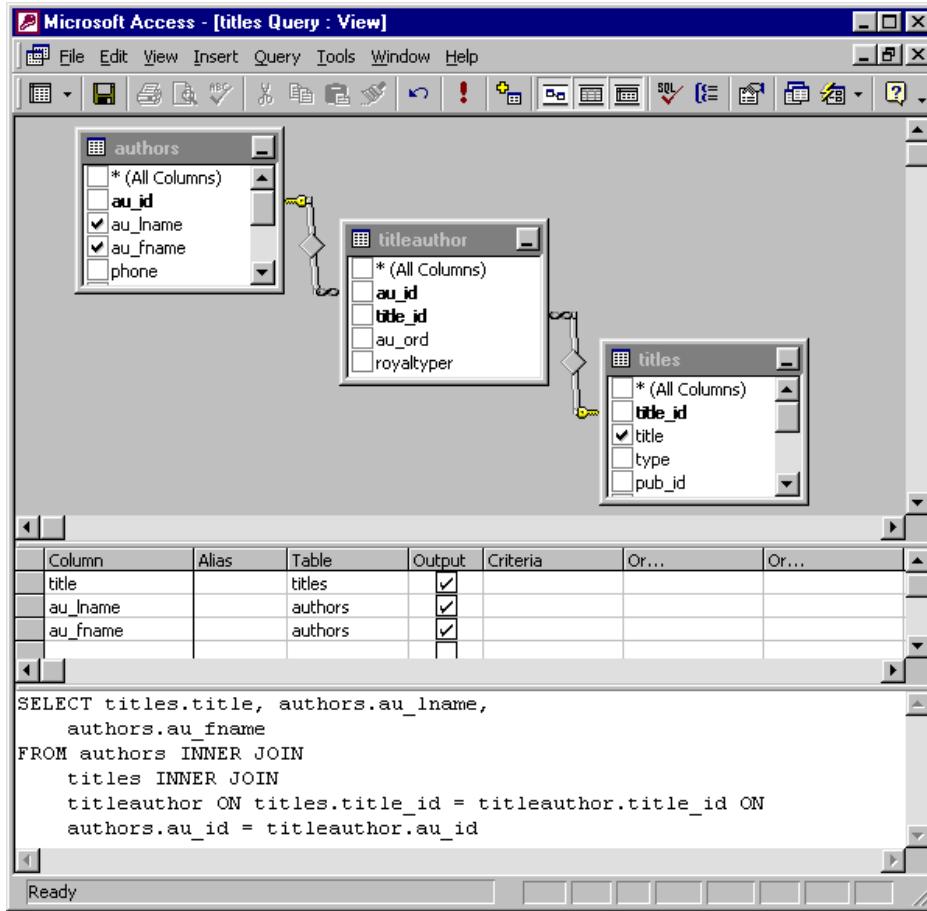
החלון התיכון בתרשימים 4.29 הוא תיבת הדוח-שיך של **Data Link Properties** של הפרויקט. השירות של פרויקט זה הוא cab2200, שם שירות מסד הנתונים SQL Server 7.0 הוא. כדי להקים חיבור פעיל אל שירות, עליך לצין את מידע הכניסה (logon). הדוגמה הנווכחית מנצלת את האבטחה המשולבת של NT Windows. לאחר שסיימית להקים את החיבור, תוכל לפתח מסד נתונים כלשהו שנמצא בשרת שמוטר לך להיכנס אליו. לחץ על **Test Connection** כדי לוודא את חווית הקישור למסד הנתונים המבוקש.





תרשים 4.29: החalon הראשון הוא חalon **מסד נתונים של Access Project**; החalon השני הוא תיבת הדיו-שים **Data Link Properties** של הפרויקט

תרשים 4.30 מציג את תצוגת העיצוב של שאלתה ב- Access Project. רשות עיצוב השאלתה שלו דומה לו של מסד נתונים מסווג *.mdb, אך היא מהוות חלק מאוסף כלי העיצוב החזותיים, Microsoft Da Vinci. באפשרות לחוץ לחייצה ימינית כדי להוסיף טבלאות או שאלות נוספות בתור מקורות רשומות לעיצוב של שאלתה חדשה, ותוכל גם לגרור ולשחרר שדות בין טבלאות כדי ליצור קשרי גומלין. לא קיימת תצוגת SQL מפורשת לעניין זה, אך ניתן להחליף את מצב הבדיקה של חלונית SQL בתצוגת העיצוב באמצעות לחץ בסרגל הכלים **מסך נתונים** (Database). הלחוץ הראשון בסרגל כלים זה עדין מאפשר להחליף בין התצוגה **עיצוב** (Design) והתצוגה **גיליון נתונים** (Datasheet). כיוון רשות עיצוב השאלתה משתנה מאונci לאופק, אך העיצוב נשאר דומה. בונים תצוגות על ידי הוספת קבועים וביטויים לתאי הクリטוריונים מימין לשמות השדות ברשות עיצוב השאלתה.



תרשים 4.30: תצוגת עיצוב של שאלתה ב- Access Project. שים לב להציגת קוד SQL של שאלתה בחולנית נפרדת ולא בתצוגה נפרדת

תכנות שאלות עברו מסדי נתונים מרוחקים

פתחים רבים מעדיפים לעבוד ישירות עם ממשק תכנות ולא עם ממשק משתמש גרפי. Access 2000 מציע תאימות הדזקה יותר בין שאלות של מקורות נתונים מקומיים ומרוחקים ברמת הקוד לעומת זו שבرمת ממשק המשמש הגרפי, מכיוון שנינו להשתמש ב-ADO ו-OLE-DB ו-SQL בעבודה עם שני סוגי מקורות הנתונים. ההבדל הגדול ביותר בין גישה לנתונים מקומיים או מרוחקים, הוא שגישה לנתונים מרוחקים מחייבת לציין את פרטורי OLE DB של ספק OLE DB provider (OLE DB provider) בצוירו הנתונים שברצונך לחשך. העברת יישומים ממוקור נתונים מרוחקים עלולה להצריך טיפול בתאימות שפת SQL, כיוון שמנגנוני מסדי נתונים מרוחקים שונים דורשים לפעמים פרטורי כניסה (logon) שונים ומשתמשים בהרחבות SQL ייחודיות עברו הרכיבים המיעדים שלהם.

כדי להתחבר למקור נתונים מסווג OLE DB, תזדקק לשפק Microsoft SQL Server מיוחד שנקרא SQLOLEDB, אשר מותאם במיוחד לעבודה עם Microsoft SQL Server, במקומם ODBC. SQLOLEDB, ODBC, MSDASQL, המיעוד לעבודה עם מסדי נתונים אחרים. מנהל התקן הכללי של SQLOLEDB, MSDASQL, ותיק יותר מ-OLE DB, SQLOLEDB, אך הוא עדין מנהל התקן רב תכלייתי למכבים בהם לא קיים מנהל התקן OLE DB ספציפי. כל אחד משנה מנהלי התקנים הללו מקבל פרמטרים שונים במקצת (בפרק 3 הסבירנו כיצד לקבוע את הפרמטרים עבור (MSDASQL)).

הדוגמה שלפניך ממחישה את הישירות של תכונות שאילתת למקור נתונים מרוחק מסוג SQL Server 7. SQL Server שים לב שהשיגרה משתמשת במנהל התקן cab2200 שמשתמש במסד הנתונים Pubs בשרת cab2200. לאחר פתיחת החיבור השאלתה מבוצעת מול מסד הנתונים Pubs בהתבסס על מנהל התקן SQLOLEDB והגדורות הפרמטרים שלו, השיגרה יוצרת אובייקט מסוג Command ומגדירה את המאפיינים CommandType ו-CommandText שלו. לאחר מכן היא מבצעת את הפקודה, מציבה את קבוצת החזרה באובייקט Recordset ומדפסה את קבוצת החזרה.

```
Sub SQLOleDBQuery()
Dim cnn1 As ADODB.Connection
Dim cmd1 As ADODB.Command
Dim rst1 As ADODB.Recordset
' Establish a connection to the database.
' Specify the server (cab2200) as the data source and catalog as the database.
Set cnn1 = New ADODB.Connection
With cnn1
    .Provider = "sqloledb"
    .ConnectionString = "data source=cab2200;" & _
        "user id = sa;initial catalog=pubs"
    .Open
End With
' Set up a command object.
Set cmd1 = New ADODB.Command
With cmd1
    .ActiveConnection = cnn1
    .CommandText = "SELECT titles.title, " & _
        "authors.au_lname, authors.au_fname " & _
        "FROM titles INNER JOIN (authors INNER JOIN " & _
        "titleauthor ON authors.au_id = titleauthor.au_id) ON " & _
        "titles.title_id = titleauthor.title_id"
    .CommandType = adCmdText
End With
' Open and print a recordset based on the executed query statement.
Set rst1 = cmd1.Execute
Debug.Print rst1.GetString
End Sub
```

שאילתת זו מפיקה תוצאות זהות לאלו של השאלות המוצגות בתרשיים 4.28 ו-4.30. לחתמי את קוד SQL שבדוגמה הנוכחית ישירות מהתוצאות SQL של השאלה שבתרשיים 4.28. שים לב שהיא דומה, אך אינה זהה בארגון שלו, לקוד SQL שבתרשיים 4.30 של Access Project. יחד עם זאת, אפשרות להמיר את קוד SQL מתרשיים 4.30 לתוך השיגרה שלעיל מבלי לשנות את קבוצת ההחזרה. הדבר מאשר את הדעה שגישת התכונות המבוססת על db.mdb.*. וגישה Access Project יכולות להפיק תוצאות שקולות למרות כמה הבדלי סגנון בייצוגו שלhn ב-QL במקרים מסוימים.

בניהת ממשק משתמש באמצעות טפסים

יישומים משתמשים בטפסים ובדיקות כדי ליעזג נתוניים. טפסים מהווים גם אמצעי לקליטה ומתן משוב על קלט משתמש. משתמשי מסד הנתונים מתקשרים עם היחסום בעיקר באמצעות טפסים, ולכן יש חשיבות רבה לעיצוב טפסים ולאופן תפקודם. פרק זה פותח בהסביר כיצד ניתן רושם ראשוני מלאהיב על ידי שימוש בטופס שנקרא **מסך פתיחה** (Splash Screen). בהמשך הפרק נלמד כיצד לגורום לטפסים (במיוחד טפסי איגוד-נתוניים) לתקשר עם נתוניים. נלמד גם על עיצוב מותנה וביצוע פעולות חישוב מידע עם טפסים. לבסוף, נלמד להפנות לטפסים באמצעות קוד, לכבות ולהפעיל לシリוגין את המצב הגלוי שלהם וגם כיצד לעבוד עם מודולי מחלקה טופס.

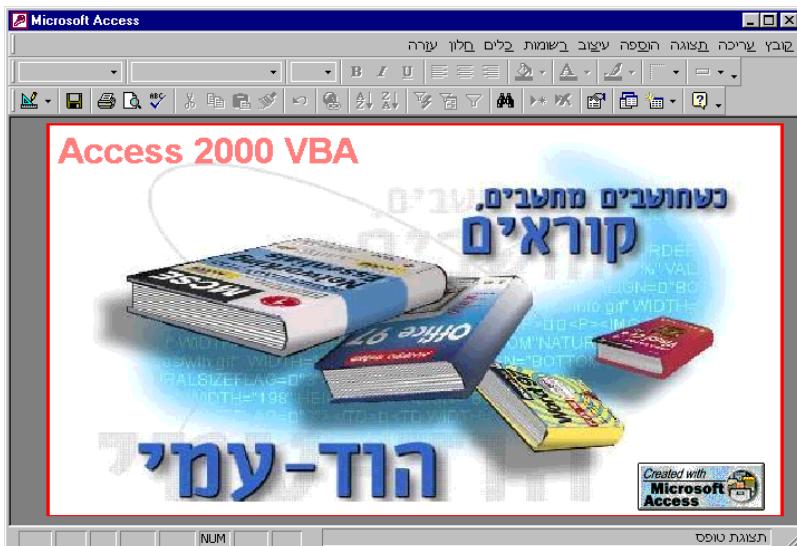
הערה:



Microsoft Access 2000 מציע שני סוגי טפסים, UserForms והמינים ביישומים נוספים של Office, וטופס מותאם שהוא גירסה עדכנית של טופס מגרסאות מוקדמות של Access. Access UserForms אינם משלבים בסביבת הפיתוח של Access בצורה כה הדוקה כמו טפסי Access. לדוגמה, UserForms אינם מאוגדים לטבלאות ולשאלות Access. בנוסף, אשפי הטפסים האשפי הפקדים של Access אינם מטפלים/UserForms. מסיבות אלו הפרק עוסק בטפס Access בלבד. תוכל להרחיב את ידיעותיך על UserForms באמצעות העזרה המקוונת.

טופס מסך פתיחה

אחת הדריכים הקלות להתחילה לubit בקורס בסיסית עם טפסים היא על ידי יצרית מסך פתיחה. מסך פתיחה הוא טופס שמוספי טרם הצגת טופס אחר אינטרקטיבי יותר. מסכי פתיחה מתארים את יייעוד היישום ומציגים בדרך כלל גם את שם מחברו. תוכל לקבוע בקלות רבבה את משך ההציגה של מסך פתיחה. תרשימים 5.1 מציג את מסך הפתיחה שנמצא בקובץ Chapter5.mdb בתיקיה ..\Code\Chap05..\شبתקליטור המצורף. תוכל לאמץ אותו לצרכיך.



תרשים 5.1: מסך פתיחה לדוגמה

יצירת מסך פתיחה

תוכל להתחיל ביצירת מסך הפתיחה על ידי יצרת רקע פרוש. לשם כך עליך להגדיר שני מאפיינים בזמן שהטופס פתוח בתצוגת עיצוב. ראשית, הגדר למאפיין **תמונה** (Picture) (בכרטיסייה **מבנה** (Format)) של גילוון המאפיינים את הנטייב ושם קובץ התמונה. קובץ זה יכול להיות מפת סיביות בתבנית *.bmp, *.ico, *.dib או *.wmf. או *.emf הגրפים הדרושים (תוכל להתקין את המנסננים החסרים באמצעות תוכנית ההתקנה). תמונות המכילות גווני אפור או צבעים מעוממים אחרים מתאימות ביותר כתמונות רקע, מכיוון שהוא מבליות את הרכיבים שבקידמתן – תמונות וטקסט. שנית, תן למאפיין הטופס **ריצוף תמונה** (Picture Tiling) את הערך **בן** (ערך ברירת המחדל שלו הוא לא) אם ברצונך שהתמונה תחזור על עצמה (לא בוצע בתרשימים 5.1).

בשלב הבא, תוכל להוסיף תמונה בקידמה על ידי בחרה באפשרות **תמונה** (Picture). תוכל גם להוסיף פקץ **ציור** (Image) לטופס. אם תנהג כך או

אחרת, תוצג תיבת הדו-שייח **הוספת תמונה** (Insert Picture) שמננה תוכל לבחור תמונה להוספה. לחיצה על אישור מגדרה אוטומטית את המאפיין **תמונה**. Access מאפשר להגדיר מאפיין זה באמצעות קוד VBA שמאפשר לבנות את מראה הטופס בצורה דינמית בתגובה לקלט המשמש (כגון קלט תיבת טקסט), או גורמים סביבתיים (כגון קוד האבטחה של המשמש). תוכל למקם את התמונה, לשנות את גודלה ולהפעיל את האפקטים המיוחדים באמצעות המאפיין **מצב שינוי גודל** (Size Mode). המאפיין יכול לקבל שלושה ערכים: **גזר** (Clip), **מתח** (Stretch) ו**שנה** (Zoom).

נשלים את מסך הפתיחה על ידי הוספת פקד **תוויות** (Label) אחד או יותר. תוכל להשתמש ב-VBA כדי להגדיר מאפיינים עבור פקדי **תוויות** בזמן ריצה. הדבר מאפשר לעצב את טקסט מסך הפתיחה בצורה דינמית.

תוכל להגדיר את מסך הפתיחה כך שייפתח אוטומטית על ידי בחירת האפשרות **הפעלה** (Startup) מהתפריט **כלי** (Tools) ובבחירה בשם הטופס של מסך הפתיחה מתוך הרשימה הנפתחת **הציג טופס/עמוד** (Display Form/Page) שבתיבת הדו-שייח הפעלה. באפשרותך גם להסתיר את חלון מסך הנתונים בעת פתיחת יישום, על ידי ביטול בחירת **תיבת הסימוןציג חלון מסך נתונים** (Display Database Window) בתיבת הדו-שייח הפעלה. לחץ על אישור כדי לשומר את הבחירה.

בקרה על מסך התצוגה

זוג שגרות האירוע שלפניך מציג מסך פתיחה במשך 10 שניות (כדי להציג את קוד VBA שעליו מבוסט הטופס, לחץ לחיצה ימנית על הטופס, בחר באפשרות **אירוע בנייה** (Code Builder) מתפריט הקיצור, לאחר מכן בחר באפשרות **בונה קוד** (Build Event) בתיבת הדו-שייח **בחירה בונה** (Choose Builder) וללחוץ על אישור). שגרת האירוע Form_Open נוננת למאפיין הטופס, TimerInterval, את הערך 10000 (שווה ל-10 שניות; מרוחז הזמן מבוטא ביחידות מילה של אלףית השניה). שגרת האירוע Form_Open סוגרת את הטופס. שים לב כי שגרת האירוע Form_Open משתמשת במילת המפתח Me כדי לצוין את הטופס. באפשרותך להחילף את "frmSplashTimer" Form_Timer Me.Name בשירה Form_Timer; מוסכמת השם Me עמידה יותר, מכיוון שהיא מאפשרת לשנות את שם הטופס מבלי לשנות את הקוד.

```
Private Sub Form_Open(Cancel As Integer)
    Me.TimerInterval = 10000
End Sub
```

```
Private Sub Form_Timer()
    DoCmd.Close acForm, "frmSplashTimer"
End Sub
```

טופס מסך ניוט

טופסי מסך ניוט הם דרך מקובלת לאפשר ניוט ביישום. טפסים אלה יכולים להכיל לחצני פקודה אחדים שהמשתמש יכול ללחוץ עליהם כדי לפתח טופס נוסף. בסעיף זה נציג שתי גישות לשימוש בטופסי מסך ניוט: האחת בעזרת היפר-קישורים והשנייה מבוססת על שגרות VBA.

ניוט בעזרת היפר-קישורים

כל מאוד ליצור ניוט באמצעות היפר-קישורים, מכיוון שהדבר או מחייב כתיבת קוד כלשהו (למרות שנitin להנל היפר- קישור באמצעות VBA). היפר-קישורים יכולים לתפקד בתורה קיזורי דרך אל אובייקטים של מסד נתונים ביחסום, מסמכים בכוון הקשיה, קבצים בראשת או דפי Web באינטרנט או דפים בראשת האינטרא-נט הארגונית. Access מאפשר להשתמש בהיפר-קישורים עם תוויות, לחיצנים ותמונות.

באפשרות להגדיר ולערוך מאפייני היפר- קישור מתוך תצוגת העיצוב של טופס, או באמצעות קוד VBA. כל יותר להגדיר ולערוך היפר- קישורים באמצעות שגרות ידנית (האפשרות גם להפיק טפסים הנטועים במחירות הרבה יותר), מכיוון שתפסים בעלי היפר-קישורים שנוצרו בתצוגת עיצוב אינם מחייבים שמודול יכול קוד VBA. כדי לנצל יתרון זה, צריך מאפיין הטופס **יש מודול** (Has Module) להיות מוגדר בתורה לא.

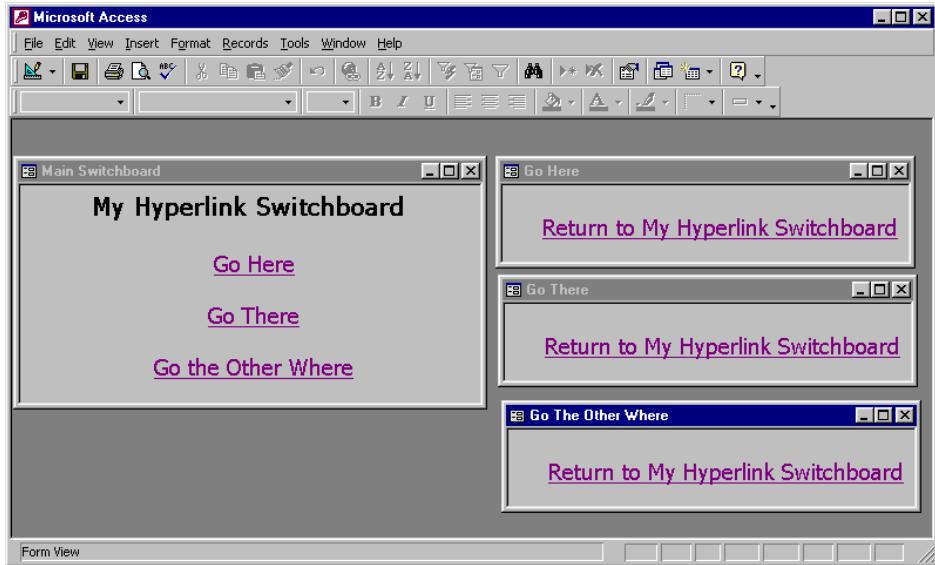
הערה:



סוג הנתונים היפר- קישור מאפשר ליחסום Access להפעיל היפר- קישור מתוך שדות טבלה או שאלתה. שדות מסווג היפר- קישור דומים במובנים רבים למאפייני היפר- קישור, אך משתמשים בהם בצורה שונה. עיין פרק 13 לקבלת מידע נוספת שימוש בהיפר- קישורים לפיתוח בסביבת Web.

תרשים 5.2 מציג ארבעה טפסים המשתמשים במערכת ניוט פשוטה. טופס מסך הניוט הראשי שבצד שמאל מעביר את המוקד אל אחד משלשות הטפסים האחרים כשהמשתמש לוחץ על היפר- קישור בטופס הראשי. ברגע שהמוקד עובר לטופס אחר, יכול המשמש להחזיר את המוקד אל מסך הניוט על ידי לחיצה על ההיפר- קישור המפנה אל הטופס הראשי.

באפשרות להגדיל את מספר הרמות של היררכיה מערכת הניות, בקרה קלה ופשוטה. באופן טיפוסי, כל מסך בן יכול להחזיר את המיקוד למסך האם שלו. מסכים ראשיים כוללים לעיתים קרובות דרך יציאה מהיחסום או מ-Access.



תרשים 5.2: מערכת ניווט פשוטה בעזרת היפר-קישורים

כדי ליצור היפר- קישור באמצעות פקד **תוויות** (Label), فعل כך :

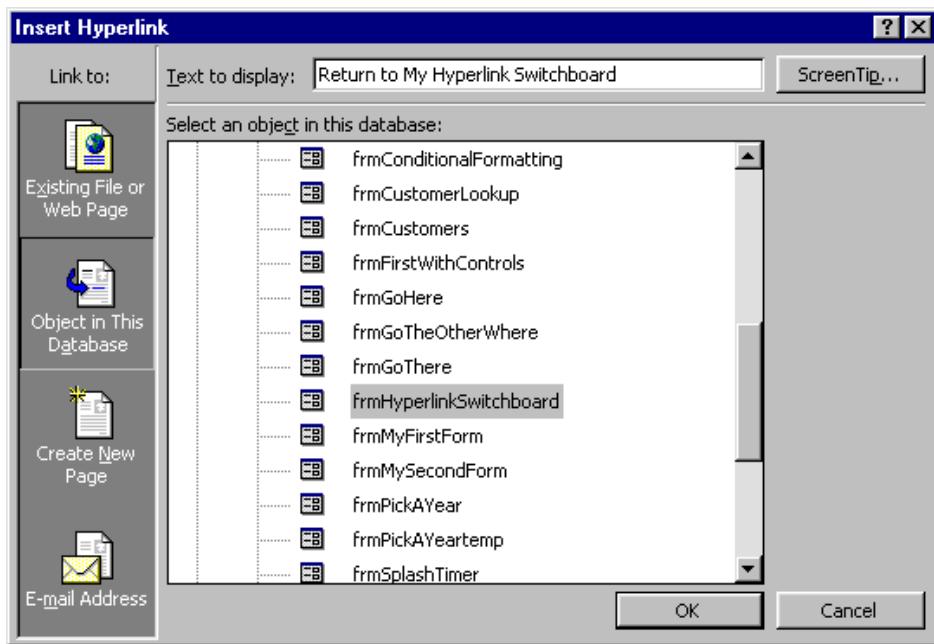
1. לחץ על לחצן **בנייה** (Build) שליד המאפיין **כתובת_היפר- קישור** (Hyperlink) או ליד המאפיין **כתובת משנה של היפר- קישור** (Address) בגילוון המאפיינים של התווית.
2. בתיבת הדו- שיח **הוספת היפר- קישור** (Insert Hyperlink), בחר את סוג האובייקט ל קישור מתוך הרשימה **קישור אל** (To). Link To).
3. הקלד את המידע עבור סוג האובייקט.
4. לחץ על **אישור**.

הערה:



כברירת מחדל, היפר-קישורים שלא לחזו עליהם בהפעלת Access מוצגים כביכול, וכאליה שלחצטו עליהם – בצעב המכונה סיגלית. באפשרות לשנות צבעים אלה על ידי בחירה באפשרויות (Options) מתפריט **כלים** (Tools), בחרירה בכרטיסיה **כללי** (General) ולהיכזה על **אפשרויות אינטרנט** (Web Options). השתמש בשתי תיבות הרשימה שבתיבת הדו- שיח **אפשרויות אינטרנט** כדי לקבוע את הצבעים הרצויים.

תרשים 5.3 מציג את תיבת הדו- שיח **הוספת היפר- קישור** בטופס frmGoHere במאזן הנתונים המדגמי של פרק זה. היפר- קישור מעביר את המיקוד בחזרה אל טופס מסך הניווט.

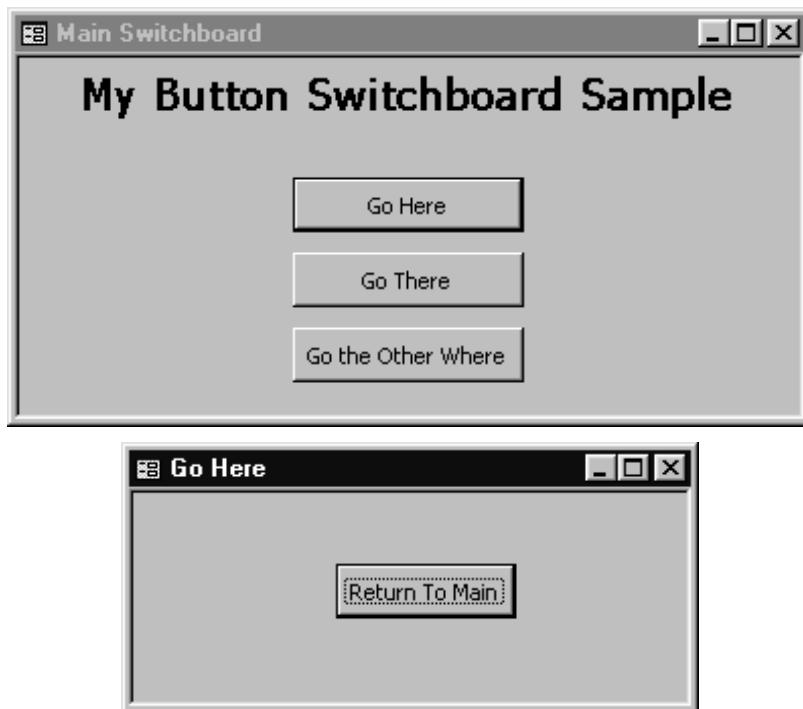


תרשים 5.3: תיבת הדיו-שיך הוסף היפר- קישור עבור התווית בטופס frmGoHere

נירוט באמצעות קוד

דרך מקובלת לנוטפת למימוש נירוט באמצעות טופס מסך היא באמצעות קוד VBA מהחורי אירועי לחיצה של לחץ פקודה. גישה זו מאפשרת חשיפה עשריה יותר לרכיבים התפקודיים של Access לעומת נירוט שmbוסס על היפר- קישורים, מכיוון שכן ניתן לשלב פונקציות נירוט עם אירועים נוספים, כגון סגירת טופס (היפר- קישור מאפשר פתיחת טופס מתוך טופס אחר, אך החזרה אל מסך הנירוט מותירה את הטופס הבן פתוח; נירוט מבוסט VBA מאפשר לישום לסגור את טופס הבן כשהמשתמש סוגר את הטופס הראשי). שגרת אירוע מאפשרת גם להגדיל את השליטה באובייקטים רבים של מסד נתונים. נירוט המבוסס על היפר- קישורים פשוט מעביר את המיקוד לאובייקט אחר, כגון טופס.

תרשים 5.4 מציג זוג טפסים הממשים נירוט בגישה VBA. דוגמה זו מבוססת על אירועי לחיצת לחץ. כשהמשתמש לוחץ על אחד מלחצני מסך הנירוט, קוד אירוע לחיצת לחץ פותח את טופס היעד וסגור את מסך הנירוט הראשי. לחיצה על לחץ **Return To Main** בטופס היעד סוגרת את הטופס ופותחת את טופס הנירוט הראשי (הדוגמה מבוססת על לחצני פקודה, אך אפשרות להשתמש בברורה מסווג אחר כלשהו שמאפשרת למשתמש ליצור אירועים).



תרשים 5.4: טפסים אלה משתמשים בקוד מאחריו אירועי לחיצת הלחצנים, כדי לנoot

זוג שגורות האירוע שלפניך מציג את קוד אירועי הלחיצה של שני הלחצנים שבתרשים 5.4. השגירה הראשונה מעבירה את השליטה ממשך הניוטה הראשי אל המסק השני. היא גם סוגרת את הטופס הראשי. השגירה השנייה מעבירה את השליטה חוזרת אל ממשך הניוטה הראשי ולאחר מכן סוגרת את הטופס השני. השתמשתי באשף לחצני הפקודה כדי ליצור את הגירסה הראשונית של כל אחת מהשגרות; לאחר מכן הוספתי שורת קוד לכל שגירה כדי לסגור את המסק המתאים.

```

Private Sub cmdGoHere_Click()
On Error GoTo Err_cmdGoHere_Click
Dim stDocName As String
    stDocName = "frmButtonGoHere"
    DoCmd.OpenForm stDocName
    DoCmd.Close acForm, "frmButtonSwitchboard"
Exit_cmdGoHere_Click:
    Exit Sub
Err_cmdGoHere_Click:
    MsgBox Err.Description
    Resume Exit_cmdGoHere_Click
End Sub

```

```

Private Sub cmdReturnToMain_Click()
On Error GoTo Err_cmdReturnToMain_Click
Dim stDocName As String
Dim stLinkCriteria As String
    stDocName = "frmButtonSwitchboard"
    DoCmd.OpenForm stDocName
    stDocName = "frmButtonGoHere"
    DoCmd.Close acForm, stDocName, acSaveNo
Exit_cmdReturnToMain_Click:
    Exit Sub
Err_cmdReturnToMain_Click:
    MsgBox Err.Description
    Resume Exit_cmdReturnToMain_Click
End Sub

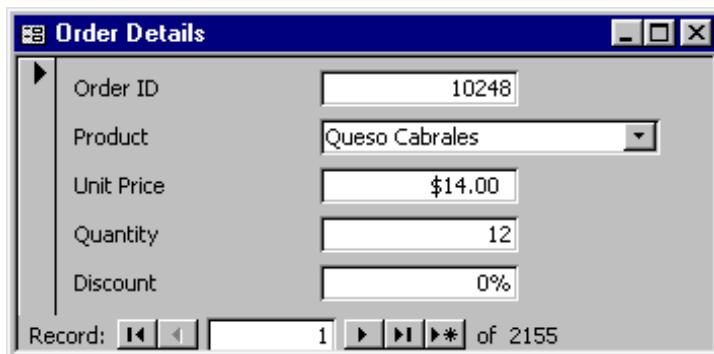
```

קישור טפסים לנתחונים

מאז ומתמיד אפשר Access לאגד טפסים עם נתונים בצורה קלה ופנוטה. זו אחת הסיבות העיקריות להיווטו סביבת פיתוח מהירה.

שימוש באשף הטפסים האוטומטיים

כדי לאגד טופס עם נתונים, ניתן להפעיל את **אשף הטפסים האוטומטיים** (AutoForm). בחר טבלה או שאלתה בחלון מסד הנתונים ולחץ על הלחן **אובייקט חדש**: **טופס אוטומטי** (New Object:AutoForm) בסרגל הכלים **מסד נתונים** (Database). האשף יפתח טופס חדש שמאוגד ישירות למקור הנתונים שנבחר. תרשימים 5.5 מציג טופס דוגמה המבוסס על הטבלה Order Details במאגר הנתונים המדגמי שבפרק זה. באפשרותך לנצל טופס זה לעיון, עריכה, הוספה או מחיקה של רשומות בטבלה Order Details.



תרשים 5.5: טופס שנוצר באמצעות אשף הטפסים האוטומטיים

מקור הנתונים של השדה Product בטופס הוא שדה בדיקת מידע בטבלת הרקע Order Details, ולכן השדה מוצג אוטומטית בהתאם לתיבה משולבת המציגת את שמו המוכרים, במקום את ערכיו הרקע ProductID. כל הרכיבים התפקודיים הללו סופקו אוטומטית על ידי האשף.

עיצוב מותנה

Access 2000 מאפשר לעצב בצורה מותנית את הנתונים המוצגים באמצעות פקד תיבת טקסט או תיבה משולבת ללא תכנות. באפשרות להחיל עיצוב מותנה על פקדי טופס, בשדות מאוגדים או בלתי מאוגדים אחד.

תרשים 5.6 מציג שלושה מופעים של השדות Discount ו- Extended Price. השדה Discount בטופס העליון הפך ללא זמין. הטופס האמצעי מבליט את הערך בשדה Extended Price באמצעות עיצוב של הטיה והדגשה. הטופס התיכון הופך את השדה Extended Price לזמן ומבליט את הערך שבשדה Discount.

השדה Extended Price מחושב; ערכו אינו נגזר ישירות מהטבלה שברקע. המאפיין **מקור פקד** (Control Source) של תיבת הטקסט מכיל ביטוי $((\text{UnitPrice} * \text{Quantity}) * (1 - \text{Discount}))$ באמצעותו הוא מחשב את הערך בעת שהמשתמש עובר לרשומה חדשה, או מעדכן את השדות UnitPrice, Quantity או Discount ברשומה הנוכחי (המנוחים בסוגרים המרובעים מתיחסים לפקדים ואינם שמורות של שדות עבור מקור הנתונים שברקע).

The screenshot shows a Microsoft Access form window titled "frmConditionalFormatting : Form". The main title of the form is "Conditional Formatting Sample Form". Inside the form, there are several input fields and labels:

- Extended Price:** A label above a text input field containing the value "\$168.00".
- Order ID:** A text input field containing the value "10248".
- Product:** A dropdown menu showing "Queso Cabrales".
- Unit Price:** A text input field containing the value "\$14.00".
- Quantity:** A text input field containing the value "12".
- Discount:** A text input field containing the value "0%".

At the bottom of the form, there is a record navigation bar with buttons for navigating between records, currently showing "Record: 1 of 2155".

frmConditionalFormatting : Form

Conditional Formatting Sample Form

Extended Price:

Order ID: \$1,696.00

Product: Manjimup Dried Apple

Unit Price: \$42.40

Quantity: 40

Discount: 0%

Record: 5 of 2155

frmConditionalFormatting : Form

Conditional Formatting Sample Form

Extended Price:

Order ID: \$1,261.40

Product: Manjimup Dried Apple

Unit Price: \$42.40

Quantity: 35

Discount: 15%

Record: 7 of 2155

תרשים 5.6: עיצוב מותנה שולט במראה השדות Extended Price ו-Discount בטופס שלפןיר

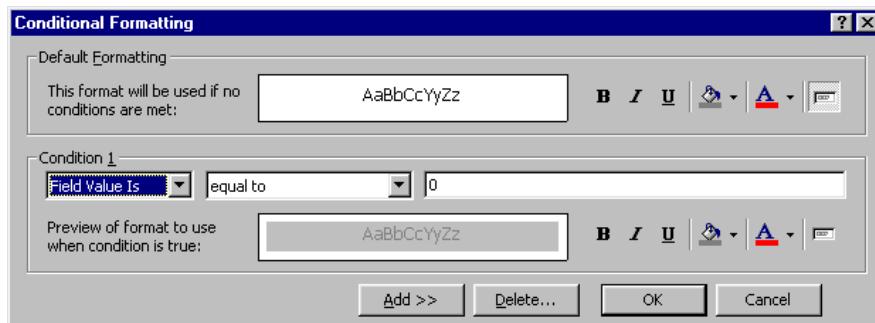
הערה:

פתחים לא מנוטים נוטנים לעתים שמות זהים לשדות ולפקדים. הרgel זה עלול לבבל ולגרום לשגיאות (אשף הטפסים האוטומטיים לוקה אף הוא בחישון זה). מומלץ להוסיף תחיליות לשמות פקדים כדי להבדילם משמות השדות ברקע שלהם. לדוגמה, שמו של פקד תיבת טקסט שמאוגד עם שדה שמו UnitPrice יהיה .txtUnitPrice



כדי להחיל עיצוב מותנה על פקד, בחר את הפקד ובחר באפשרות **עיצוב מותנה** (Conditional Formatting) מתפריט **עיצוב** (Design), כדי לפתח את תיבת הדו-שיח **עיצוב מותנה**, המוצגת בתשרים 5.7. לכל פקד בעל עיצוב מותנה יש שני עיצובים לפחות – עיצוב ברירת מחדל ועיצוב מיוחד שחל כאשר מתקיים תנאי מוגדר. באפשרותך לעצב את הפקד בהתאם לערך השדה של הפקד, לערך הביטוי או רישימת קבלת המיקוד. בעת טיפול בערך השדה של פקד, באפשרותך לבחור מתוך אופרטורים של השוואה, כגון שווה ל (=), גדול מ (>) וקטן מ (<). התנאי של השדה (Field value is equal to 0) Discount שבתרשים 5.7 הוא **ערך שדה הינו שווה ל 0**.

אפשרות העיצוב שנבחרה לתנאי זה מבטלת את זמינות הפקד כאשר ערך ההנחה (discount) הוא 0.



תרשים 5.7: תיבת הדו-שיח **עיצוב מותנה**

אם תחיל עיצוב מותנה על שדה מחושב, כגון Extended Price, עליך לכתוב ביטויו באמצעות אופרטורים תקניים של VBA. התנאי עבור השדה הוא `Extended Price >= 500` (`Text4.value >= 500`) ומציג את הערך המוחשב. כשלעצמו השדה גדול מ-500, התוכן שבתיבת הטקסט יובלט בעזרת הדגשה והטיה.

באפשרותך להחיל על פקד תנאי נוסף ועיצוב מיוחד. כל שימוש לשם כך הוא להזץ על לחץ **הוסף** (Add) בתיבת הדו-שיח **עיצוב מותנה** (Conditional Formatting) וצלין את התנאי החדש ואת נתוני העיצוב שלו.

טפסי משנה

טופס משנה (subform), המספק את אחת הדרכים המקובלות ביותר להציג נתונים ב-Access, הוא טופס המוטבע בטופס ראשי. הטופס הראשי מכיל מידע כללי אודוט אובייקט (כגון הזמן או שם של מטופל). פרט היררכי הקשור אחד או יותר (כגון שורות פרטי בהזמנה או ביקורי המטופל) מופיע בטופס משנה אחד או יותר של הטופס הראשי. שדה משותף אחד לפחות חייב לחבר ייחדיו את מקור הרשומות של הטופס הראשי וכל אחד מטפסי המשנה. השדה המשותף מאפשר לטופס המשנה להציג רק את הרשומות שתואמות לרשומה הנקחית בטופס הראשי. כשהמשתמש עובר לרשומה חדשה בטופס הראשי, טופס המשנה מציג קבוצת רשומות חדשה שקשריה בצורה ייחודית לרשומה החדשה בטופס הראשי.

תרשים 5.8 מציג מסך ראשי, MyOrders, המכיל טופס משנה מוטבע ששמו MyOrderDetails. הטופס מקשר את השאלתה MyOrders אל השאלתה MyOrderDetails בהתאם לשדה המשותף OrderIDField (שאילתות אלו וטבלאות הרקע שלhnן ל��וחות מסד הנתונים Northwind). בעת יצירת הטופס הראשי וטופס המשנה, לא יצרתי קשרי גומליין בין שתי השאלות בחולון **קשרי גומליין** (Relationships) או על ידי שימוש בגלויונות נתונים משניים (עיין בפרק 4 לקבלת מידע נוסף על גלויונות נתונים משניים).

The screenshot shows the 'MyOrders' form in Microsoft Access. The main form fields are:

- Order ID: 11077
- Customer: Rattlesnake Canyon Grocery
- Order Date: 06-May-98
- Ship Name: Rattlesnake Canyon Grocery
- Ship Address: 2817 Milton Dr.
- Ship City: Albuquerque
- Ship Region: NM
- Ship Postal Code: 87110
- Ship Country: USA

Below the main form is a subform titled 'MyOrderDetails subform' containing a table with the following data:

	OrderID	ProductID	UnitPrice	Quantity	Discount
▶	11077	Chang	\$19.00	24	20%
	11077	Aniseed Syrup	\$10.00	4	0%
	11077	Chef Anton's Cajun Seasonin	\$22.00	1	0%

Record navigation buttons at the bottom of the subform indicate 1 of 25 records.

At the bottom of the main form, there are additional record navigation buttons showing 830 records total.

תרשים 5.8: טופס המכיל טופס משנה

כדי ליצור טופס משנה,فتح את הטופס הראשי בתצוגת עיצוב, ודא שהחץ **ASF** בקרה (Control Wizards) שבארגז הכלים (Toolbox) לחוץ, ולאחר מכן גורר טבלה, שאליתה או טופס מתוך חלון מסד הנתונים וחרר אותו בטופס הראשי. טופס המשנה יוצג בתור פקד בטופס הראשי. כדי לטCKERן את הטופס הראשי עם טופס המשנה, עליך להקצות שדה משותף אחד לפחות. בחר את מכולת טופס המשנה והגדיר את המאפיינים **קישור שדות צאצא** (Link Child) ו **קישור שדות מראשי** (Link Master) בתור השדה המשותף. בשאלות MyOrders ו-MyOrderDetails השדה המשותף הוא .OrderID

טופס הראשי יכול להכיל טפסי משנה רבים. הדגישה היחידה היא שמקור הרשומות של כל טופס משנה יחולק שדה משותף אחד לפחות עם מקור הרשומות של הטופס הראשי. לדוגמה, אם המסקן הראשי שבתרשים 5.8 היה מכיל את השדה EmployeeID של הטופס הינה יכול לכלול טופס משנה נוסף המבוסס על הטבלה Employees.

אם אתה מגדיר קשרי גומליין בין טבלאות ושיילות בחלון קשרי הגומליין, או על ידי שימוש במאפייני גיליון שני, באפשרות ליצור מסך ראש עם טופס משנה מוטבע באוֹתָה קלות בה תיצור טופס פשוט. בחלון מסך נתונים,בחר את הטבלה או את השאיילה שעלייה יתבסס הטופס הראשי, ולאחר מכן לחר על הלחץן **אובייקט חדש**: **טופס אוטומטי** (New Object:AutoForm). אשה הטפסים האוטומטיים יבנה טופס ראשי ובו טופס משנה מוטבע. טופס המשנה מנצל את המידע שבחולנות קשרי הגומליין או בגילוון הנתונים המשני. באפשרות לגורר טבלאות, שיילות או טפסים נוספים באופן ידני אל הטופס הראשי בתצוגת העיצוב כדי ליצר טפסי משנה נוספים.

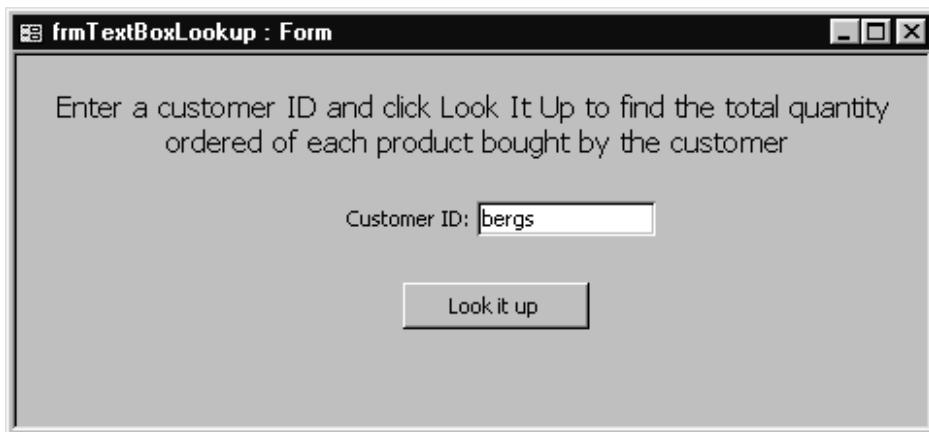
בדיקות מידע והציג נתוניים

טופס בדיקת מידע דומה לשאיילה פרטמר בעלת ממשק קדמי מותאם אישית; הטופס אוסף קלט שפועל שאיילה. בעת שימוש בטפסים וב-VBA, באפשרות לפעול בצורה גמישה בכל הנוגע לאופן איסוף המידע וגם באשר לסוג המידע שתוכל להחזיר למשתמש היישום.

יצירת טופס בדיקת מידע

הדרך הקלה ביותר למימוש טופס שמבצע בדיקת מידע היא לחת למשתמש להקליד את פרטי בדיקת המידע בתיבת טקסט וללחוץ על לחץן כדי להתחיל בחיפוש. תיבת הטקסט חייבת להיות בלתי מאוגדת, מכיוון שהיא אינה מכינה מידע למסד הנתונים; כל שהיא עשוה הוא לקבל נתונים חיפוש מהמשתמש. המידע שבתיבת הטקסט יונצל על ידי שאיילה למציאת המידע המבוקש.

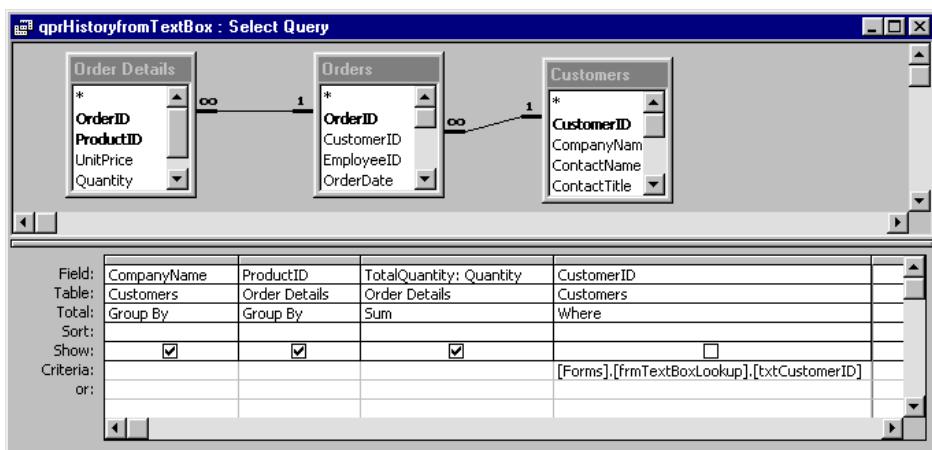
תרשים 5.9 מציג טופס שפותח את השאיילה qprHistoryfromTextBox כשהמשתמש מקליך קוד זיהוי בתיבת הטקסט ולוחץ על לחץן **Look it up**. השאיילה מוצאת את הכמות הכוללת שהזומנה מכל מוצר שקנה הלקוח.



Company Name	Product	TotalQuantity
Berglunds snabbköp	Chai	35
Berglunds snabbköp	Chang	21
Berglunds snabbköp	Aniseed Syrup	30
Berglunds snabbköp	Chef Anton's Cajun Seasoning	12
Berglunds snabbköp	Ikura	22
Berglunds snabbköp	Konbu	6
Berglunds snabbköp	Tofu	16
Berglunds snabbköp	Pavlova	12
Berglunds snabbköp	Alice Mutton	10

תרשים 5.9: טופס שמבצע בדיקת מידע פשוטה

תרשים 5.10 מציג את השאלתה qprHistoryfromTextBox בתצוגת עיצוב. שים לב כי מדובר בשאליתה פשוטה שמסכמת את השדות Quantity בטבלה Order Details עבור כל אחד מה מוצרים שהלקוח רכש (הפסוקית WHERE בעמודה الأخيرة מגבילה את השאליתה לשורות שהשדה CustomerID שלן מתאים לערך שבטיבת הטקסט של הטופס).



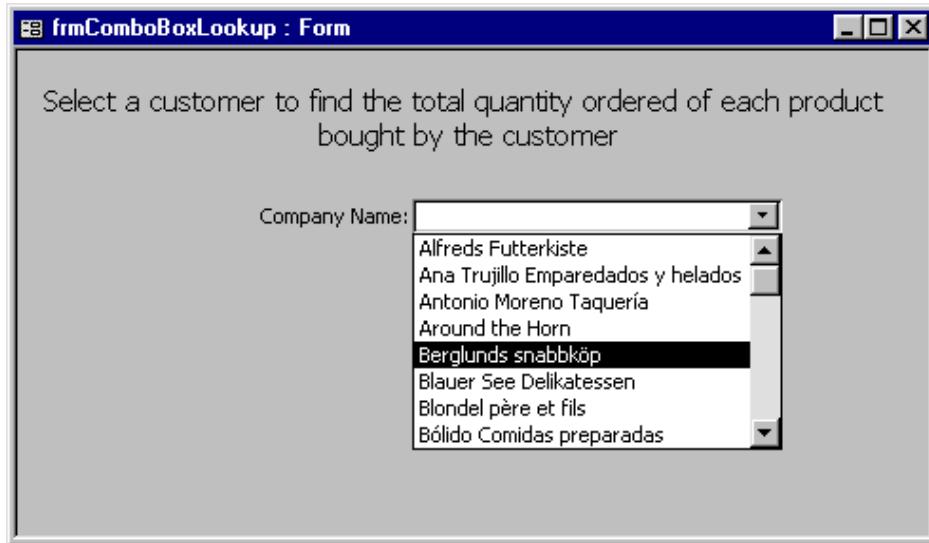
תרשים 5.10: השאלתה qprHistoryfromTextBox בתצוגת עיצוב

האלמנט האחרון בטופס הוא שגרת אירוע VBA המופעלת כשהמשתמש לוחץ על חצן Up It Look. השגירה כוללת שורה ייחודית הפותחת את השאליתה : qprHistoryfromTextBox

```
Private Sub cmdLookup_Click()
    DoCmd.OpenQuery "qprHistoryfromTextBox"
End Sub
```

שימוש בתיבה משולבת עבור קלט משתמש

הטופס שבתרשים 5.11 מעוצב טוב יותר מזה שתרשימים 5.9. במקומות לאלה את המשמש לנחש את קוד זההו הלקוח, הוא מאפשר לו לבחור את הלקוח מתוך תיבת משולבת. שגרת האירוע After Update של התיבה המשולבת פותחת את השאלתה qprHistoryfromTextBox שמשתמשת בתוינוי הלקוח שבחר המשתמש. לכן, לחוץ הפקודה אינו נחוץ יותר.



תרשים 5.11: עיצוב משופר של טופס בדיקת מידע

הערה:



טפסי בדיקת מידע המכילים תיבת משולבת עלולים לפגוע בביטויים ככל שעה מספר פריטי בדיקת המידע. במצב זה עומדות לרשותך שתי אפשרויות. הריאונה, לחזור אל טופס בדיקת המידע בעל תיבת הטקסט שתואר לעיל. לחילופין, תוכל ליצור מערכת מורבדת שתאפשר למשתמשים לבחור אפשרות שייגיבו את תווחה הפריטים הזמינים בתיבה המשולבת.

כדי להעניק תפקודיות זו לפקץ תיבת משולבת, פעולה כך:

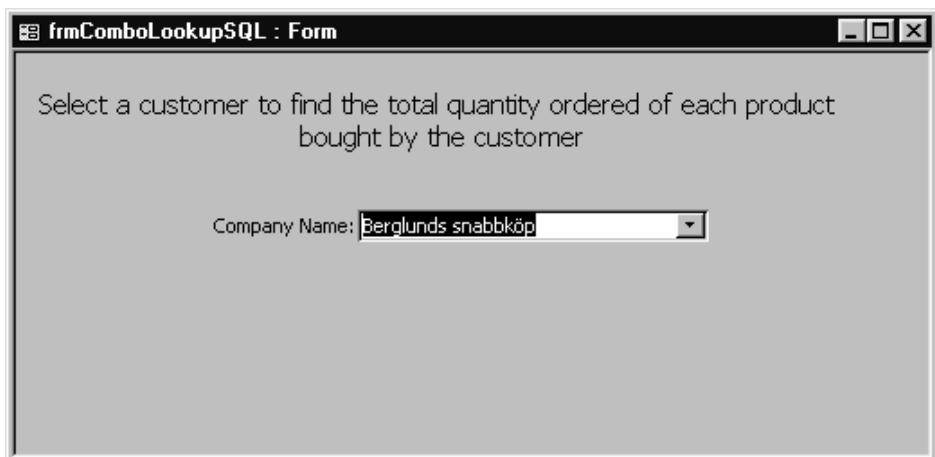
1. הגדר את המאפיין **סוג מקור שורה** (Row Source Type) של הפקץ בהתאם/**שאלילה** (Table/Query) (ברירתם המקורי).
2. הגדר את המאפיין **מקור שורה** (Row Source) של הפקץ בהתאם SQL שמחזירה את השדות המבוקשים (מחזרות SQL של הדוגמה היא `SELECT CUSTOMERID FROM CUSTOMERS`).

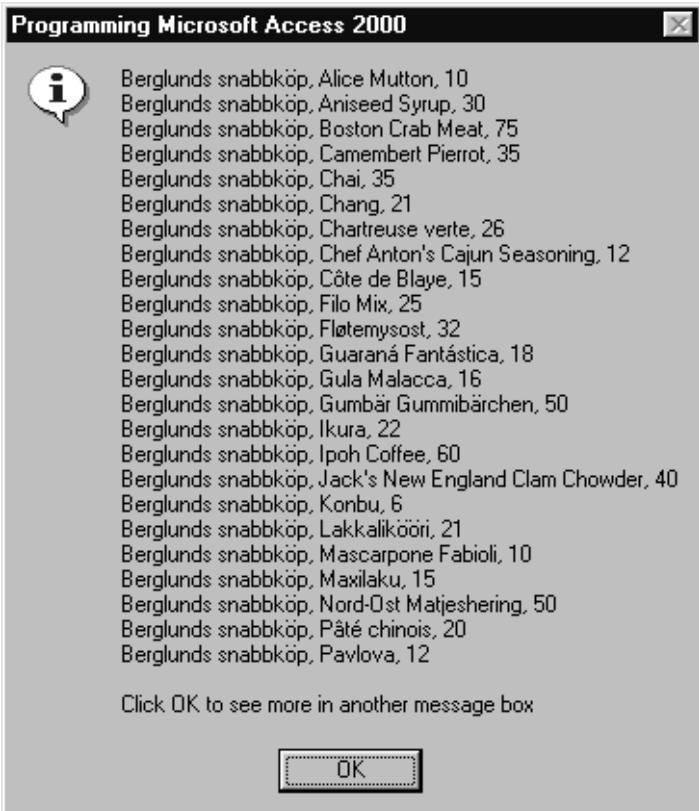
- .3. תן למאפיין **מונה עמודות** (Column Count) של הפקד את הערך 2.
- .4. הגדר את רוחבי העמודות והפרד אותן בפסיקים (הערך הראשון חייב להיות 0). אם תעדיף, אשף התיבות המשולבות יכול ליצור את הפקד עבורך. כל שעליך לעשות הוא לודא שהחצן **ASFי בקרה** (Control Wizards) שבארגז הכלים (Toolbox), במצב לחוץ, ולאחר מכן הוסף את התיבה המשולבת לטופס. עקוב אחר הוראות האשף.

הציג תוכאות בתיבת הودעה

הדוגמאות שהציגנו לוקוט בשתי חולשות. ראשית, הן מעבירות ערכים לשאלות ו呱שנות ערכים הלקחים משאלות. שנית, משמשות הדבר, שימושים עלולים לשਬש בשוגג את עיצוב השאלה. שלישי, משתמש יכול גם לשנות את הנתונים שברקע השאלה.

הטופס המדגמי לבדיקת מידע שבחלקו העליון של תרשימים 5.12 ו-5.12 מתגבר על שני החסרונות הללו על ידי שימוש ב-VBA ו-ADO (ActiveX Data Objects). טופס הקלט דומה במרקחו ובתפקודו לטופס שבתרשים 5.11. קבוצות ההחזרה של השאלות שפותחים שני הטפסים זהות, אך הן מוצגות בדרכים שונות. הדוגמה שבתרשים 5.12 מציגה את קבוצת ההחזרה בתיבות הודעה במקומות בחולון שאלתה בתצוגת **גילוון נתוניים** (הדוגמה עשויה שימוש במספר תיבות הודעה חדש כדי להציג את קבוצת ההחזרה שלה). באופן זה נתוני הרקע מוגנים מפני שימוש שלא במתכוון על ידי המשתמש.





תרשים 5.12: הטעופ הראשון מציג את קבוצת התוצאות שלו בתיבות הودעה כך שהמשתמשים יכולים להציג את קבוצת התוצאות, אך לא לשנות נתונים הרקע שלו

השיגרה שלפניך מופעלת בתגובה לאיורע After Update של התיבה המשולבת בתרשים 5.12. השיגרה מבצעת פקודה בהຕביס על שאלתה, כדי לפתוח קבוצת החזרה. לאחר מכן היא מקצת את קבוצת החזרה מהפקודה אל ערכת רשומות ומציגה את ערכת הרשומות באמצעות תיבת הודעה אחת או יותר.

```
Private Sub cboLookup_AfterUpdate()

Dim ctl1 As Control
Dim cmd1 As Command
Dim rst1 As Recordset, str1 As String

' Set reference to ComboBox control.
Set ctl1 = Me.Controls("cboLookup")

' Create and define command.
' Use ComboBox value in SQL string for command.
Set cmd1 = New ADODB.Command

With cmd1
    .ActiveConnection = CurrentProject.Connection
    .CommandText = "Select Customers.CompanyName, " & _
        "Products.ProductName, " & _
        "Sum([Order Details].Quantity) As TotalQuantity " & _
        "From Products Inner Join ((Customers Inner Join Orders " & _
        "ON Customers.CustomerID = Orders.CustomerID) " & _
        "Inner Join [Order Details] ON " & _
        "Orders.OrderID = [Order Details].OrderID) " & _
        "ON Products.ProductID = [Order Details].ProductID " & _
        "Where Customers.CustomerID = '" & ctl1.Value & "' " & _
        "GROUP BY Customers.CompanyName, Products.ProductName;"
    . CommandType = adCmdText
    . Execute
End With

' Create recordset based on return set from SQL string.
Set rst1 = New ADODB.Recordset
rst1.Open cmd1, , adOpenKeyset, adLockOptimistic

' Loop through return set to display in message box(es)
' in blocks of 925 characters or less.
Do Until rst1.EOF
    str1 = str1 & rst1.Fields(0) & ", " & rst1.Fields(1) & ", " & rst1.Fields(2)
    str1 = str1 & vbCrLf
```

```

If Len(str1) > 925 Then
    str1 = str1 & vbCrLf & "Click OK to see more in another message box"
    MsgBox str1, vbInformation, "Programming Microsoft Access 2000"
    str1 = ""
End If

rst1.MoveNext
Loop

MsgBox str1, vbInformation, _
"Programming Microsoft Access 2000"
End Sub

```

בשיגורה זו, לא יוכלתי לנצל את הקוד SQL מחלון שאליתה בתצוגת **עיצוב**, מכיוון ש MERCHANTABILITY SQL של אובייקט מסוג Command אינה תומכת בשדוחות בדיקת מידע. לכן, הוסףתי את הטבלה Products לעיצוב השאליתה כך שיוכלו לדוח על כל שם של מוצר בקבוצת החזרה במקום על קוד המוצר בלבד, כפי שהגיע מהטבלה Order Details. הטבלה הנוספת מסבכת את לוגיקת הצירוף של השאליטה (עיין בפרק 4 מבוא לתחריר של משפטים SQL).

וללאת So עוברת בזורה סדרנית על מערכת הרשומות וכותבת את תכניתה במחרוזות. בסוף כל רשותה, היא מכינסה تو החזרת גירה (CR) והזנת שורה (LF). אם אוורך המחרוזות עולה על 925 תווים, השירה מכינסה שורה ריקה והוראה להציג קצר היסטוריית המכירות של הל��ו בקטע ההודעה הבא. תיבת ההודעה יכולה להכיל קצר יותר מ-1,000 תווים (השירה MsgBox בasad הנ吐נים המדגמי בפרק זה מסיימת לקובע את מספר התווים המקסימלי שתיבת ההודעה יכולה להכיל; כל אחת מ-19 השורות העליונות כוללת שני תווים שאינם מודפסים). הגבלת הוספת תווים לתיבת ההודעה הנוכחית ל-925 תווים, מאפשרת לתיבה להתמלא מבלילו למשך تو כלשהו.

הציגת מידע דינמית

באפשרותך להציג נתונים, כגון רשותה, בטופס, ואף לעצב טופס שיאפשר למשתמשים להציג את הרשותה, אך לא לעורוך אותה. תרשימים 5.13 מציג זוג טפסים שמאפשרים יחד למשתמש לעיין ברשותם ללקוח. המשתמש בוחר ל��וח מתוך הטופס Show Customer In Form frmCustomerLookup ולחוץ על הלחצן Customers אפשר עריכה (Allow Edits), אשר מציג את רשותה הלוקוח שנבחר (המאפיינים אפשר עריכה Allow Deletions) ואפשר תוספות (Allow Additions) מתקבלים את הערך לא, ולכן המשתמש מנסה לשנות את הנתונים). המשתמש יכול לחוץ על הלחצן Return To Customer Lookup Form כדי להעביר את השליטה חזרה אל טופס בדיקת המידע המקורי. באפשרותו גם להפעיל בדיקת מידע נוספת, או יצאת מהיישום דרך הטופס הנוכחי.

frmCustomerLookup : Form

Select a customer in the combo box
and click Show Customer In Form to open a form that
displays the record for the selected customer

CompanyName:

Customers

Read-only Customer Form

Customer ID

Company Name

Contact Name

Contact Title

Address

City

Region

Postal Code

Country

Phone

Fax

תרשים 5.13: טפסים אלה מאפשרים למשתמש לבחור ולהציג רשומה של לקוחות

שגרת האירוע פשוטה והאלגנטית שלפניך היא הקוד שמאחורי הלחץן .Show Customer in Form

```
Private Sub cmdShowCustomer_Click()
On Error GoTo ShowCustomerTrap
Dim strValue As String, strMsg As String
    strValue = Me.Combo2.Value
    DoCmd.OpenForm "frmCustomers", _
        acNormal, , "CustomerID = '" & strValue & "'"
ShowCustomerTrapExit:
    Exit Sub
End Sub
```

```

ShowCustomerTrap:
If Err.Number = 94 Then
    MsgBox "Select a customer in the combo box " & _
        "before attempting to open the Customer form.", _
        vbExclamation, "Programming Microsoft Access 2000"
Else
    strMsg = "Error number: " & Err.Number & "caused " & _
        "failure. Its description is:" & vbCrLf & Err.Description
    MsgBox strMsg, vbExclamation, "Programming Microsoft Access 2000"
End If
Resume ShowCustomerTrapExit

End Sub

```

יצירת תרשימים מקבוצות משנה של נתונים

האובייקט Chart של Microsoft Graph מאפשר ליצור תרשימים בעלי מראה ממצואי, בצורה קלה ופושטה. האובייקט, המשולב בטופס CPFD אובייקט בלתי מאוגן, יכול להיות מאוגנד לטלילות ושאלילות של Access. באפשרות לבחור מתחז מגוון רחב של סוגי תרשימים ואפשרויות עיצוב (ליחס לחיצה כפולה על האובייקט בטופס בתצוגת העיצוב כדי לחשוף את התפריט המותאם אישית של האובייקט. כשתסייעים להשתמש בתפריט האובייקט, Chart, לחץ על הטופס מחוץ לאובייקט כדי לשחזר את תפריט עיצוב הטפסים המקורי של Access).

יצירת תרשימים באמצעות אשף הטפסים

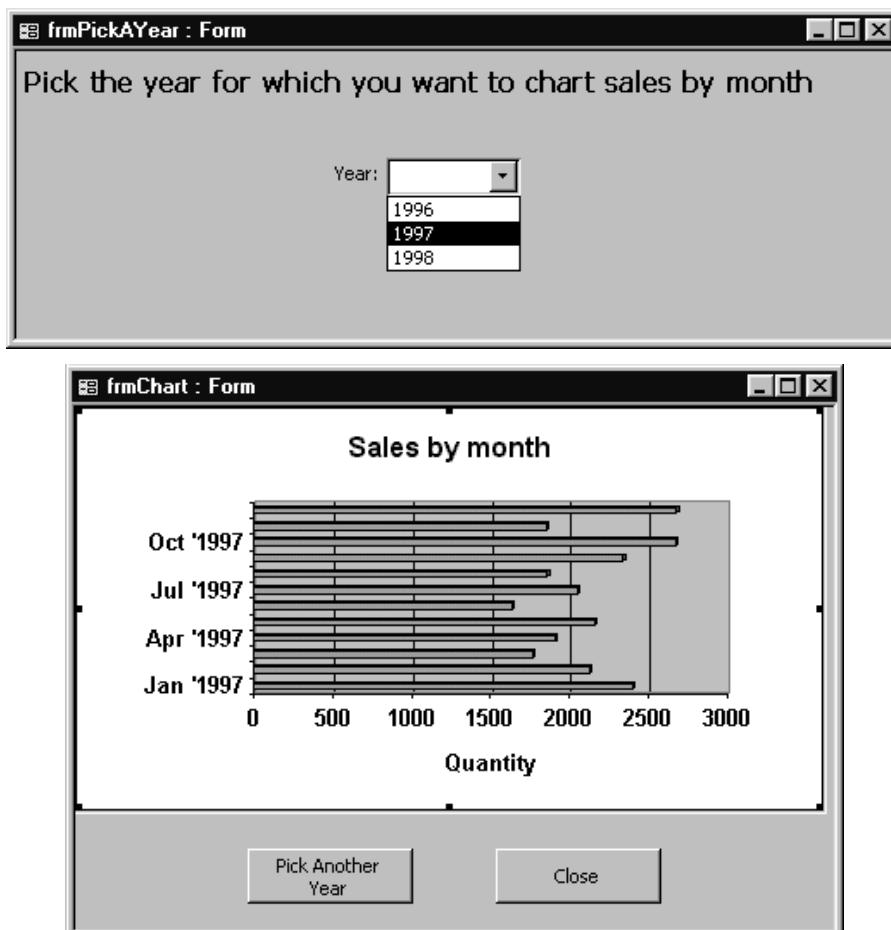
באפשרות להוסיף אובייקט Chart באופן ידני, אך קל יותר לעשות זאת באמצעות אשף התרשימים. כל שעליך לעשות הוא לפעול כך:

- .1. לחץ על האובייקט **טפסים** (Forms) בחילון מסך **נתונים** (Database) ולאחר מכן לחץ על **חדש** (New).
- .2. בחר את **אשף התרשימים** (Chart Wizard) ואת הטבלה או השאלה אליה ברצונך לבסס את התרשימים. לאחר מכן לחץ על **אישור**.
- .3. בחר את השדות שיופיעו בגלגול נתוני התרשימים ולאחר מכן לחץ על **הבא** (Next).
- .4. בחר את סוג התרשימים ולחץ על **הבא**.
- .5. גורר וScheduler את לחצני השדה המבוקשים אל התרשימים ולחץ על **הבא**.
- .6. בחר באפשרות **לשנות עיצוב של טופס או של תרשימים** (Modify The Design Of The Form Or The Chart) ולחץ על **סיום** (Finish).

באפשרות להוסיף פונקציות צבירה ועיצוב על ידי שינוי מחרוזת SQL במאפיין **מקור שורה** (Row Source) של פקד האובייקט הבלתי מאוגד המכיל את האובייקט Chart (האש יוצר עבורך משפט זה).

הציג התרשימים באמצעות קוד

תרשים 5.14 מציג שני טפסים המאפשרים למשתמש להთווות את המכירות בחודש שנבחר. הטופס העליון מאפשר לבחור שנה כלשהי מתוך הטבלה Orders. After Update של התיבת המשולבת פותח את הטופס התחתון המנצל את האובייקט Microsoft Graph 2000 Chart 2000 כדי להציג את סך כל המכירות לפי חודשים בשנה שנבחרה.



תרשים 5.14: טפסים אלה מאפשרים למשתמש להציג נתוני מכירות חודשיים בשנה שנבחרה

התרשים שבתרשים 5.14 מבוסס על שאלתה שמאחזרת את כל ההזמנות של השנה שצינה. השאלתה מתרגמת כל תאריך השנה לאחד בחודש (ambilי לשנות את נתוני הרקע). פעולה זו מאפשרת לצבור את כמות המכירות לפי חודש בזורה קלה ופושטה, ועל ידי כך מאפשרת גם להתוות את המכירות לפי חודש (אשף התרשימים מסכם אוטומטית את כמות המכירות לפי חודש במקור רשותם כמו זה הנוכחי).

שלוש שגרות האירוע שלפניך שולטות באינטראקציה בין שני טפסים. השיגרה cboPickAYear_AfterUpdate טעונה את הטופס עם ציור התרשימים וממעערת את הטופס שהמשתמש בוחר בו את השנה (עליך לסייע את הטופס ולא לסגור אותו, מכיוון ששאלתת התרשימים קובעת את השנה שבחר המשתמש באמצעות התיבה המשולבת של הטופס הראשון).

```
Private Sub cboPickAYear_AfterUpdate()
    DoCmd.Minimize
    DoCmd.OpenForm "frmChart"
End Sub

Private Sub cmdClose_Click()
    DoCmd.Close acForm, "frmPickAYear"
    DoCmd.Close
End Sub

Private Sub cmdPickAYear_Click()
    DoCmd.OpenForm "frmPickAYear"
    DoCmd.Close acForm, "frmChart"
End Sub
```

טיפול בקבצים באמצעות VBA

בסעיף זה נתאר טכניקות אחדות למיכון השימוש בטפסים. נמחיש כיצד ממספרים טפסים ופקדים, כיצד מציגים ומסתירים טפסים באמצעות קוד וגם שיטות להציג טופס בפרויקט אחר.

מספר טפסים ופקדים

האוסף AllForms, שייך לאובייקט CurrentProject, מכיל פריט לכל טופס בפרויקט. יישום יכול למספור את האובייקטים מסווג AccessObject באוסף AllForms כדי למצוא את כל הטעסים בפרויקט. המאפיינים Name ו-IsLoaded של האובייקט AccessObject מציין Name מייצג את השם של כל טופס בפרויקט, וכןם במיוחד לשימוש: המאפיין Name מייצג את השם של כל טופס בפרויקט, והמאפיין IsLoaded מציין אם הטופס פתוח.

השлага שלפניך מנצלת את המאפיין Count של האוסף AllForms כדי לקובע את מספר הטעמים שבפרויקט הנוכחי. לאחר מכן היא מדפיסה את השם ואת מצב הטעינה של כל טופס בפרויקט (האוסף AllForms מאורגן כך שאינדקס החבר הראשון שלו הוא 0; לכן, הלולאה For עוברת על טפסי הפרויקט החל ב-0 ועד אחד לפני אחריו).

```
Sub ListAllForms()
Dim int1 As Integer
' Print the number of forms in the project.
    Debug.Print CurrentProject.AllForms.Count
    Debug.Print
' Enumerate each form in the project.
    For int1 = 0 To CurrentProject.AllForms.Count - 1
        Debug.Print CurrentProject.AllForms.Item(int1).Name
        Debug.Print CurrentProject.AllForms.Item(int1).IsLoaded
        Debug.Print
    Next int1
End Sub
```

האוסף AllForms מכיל את קבוצת כל הטעמים הפתוחים בפרויקט והאוסף Controls מכיל את קבוצת פקדי הטופס. היישומים שתיצור יכולים לנצל אוסףים אלה כדי למצוא טופס מוגדר ופקד בודד בטופס זה.

השлага שלפניך ממספרת את כל הטעמים הפתוחים בפרויקט. היא מציגה את הפקדים של כל טופס פתוח לפי שם ולפי סוג. המאפיין ControlType של האובייקט Control מציין את סוג הפקד. באפשרות להשתמש במילת המפתח TypeOf בדרכו דומה.

```
Sub ListControlsOnOpenForms()
Dim frm1 As Form, ctl1 As Control
' Enumerate all open forms.
    For Each frm1 In Forms
        Debug.Print frm1.Name
' Enumerate each control on a specific open form.
    For Each ctl1 In frm1.Controls
        Debug.Print "    " & ctl1.Name & ", " & _
                    IIf(ctl1.ControlType = 100, "label", "not label")
    Next ctl1
    Next frm1
End Sub
```

שים לב שהשлага מפענחת את ערך המאפיין ControlType. כשערך זה הוא 100, הפקד הוא תווית. בתהילך הפענוח המעייני סביר יותר שתשתמש במשפט Select Case במקומות בוונקציה Immediate If שבודגמה הקודמת. הfonוקציה Immediate If, לעומת זאת, פועלת היטב בפענוח ערך ייחיד (באפשרות לנצל את Object Browser (סורק האובייקטים) כדי למצוא את ערכי ControlType).

הסתירה והציגה של טפסים

באפשרותך להשתמש ב-VBA ובאוסף AllForms יחד עם אובייקטים נוספים כדי להסתיר את הטפסים בחלון מסד הנתונים. אם תהפוך את הטופס לבלי נרא גם ביחסום, המשמש עלול לחשב שシリקת את הטופס לחלוון. טפסים מוסתרים יכולים עדיין לחשוף ערכיהם לשימוש באמצעות אובייקטים אחרים ביחסום.

זוג השגרות שלפניך מסתיר ומ לבטל את הסתרת טופס של Access :

```
Sub HideAForm frmName As String)
```

```
' Close form if it is open so that it can be hidden.  
If CurrentProject.AllForms(frmName).IsLoaded = True Then  
    DoCmd.Close acForm, frmName  
End If  
' Set form's Hidden property and do not show hidden  
' objects in Database window.  
Application.SetHiddenAttribute acForm, frmName, True  
Application.SetOption "Show Hidden Objects", False  
End Sub
```

```
Sub UnhideAForm frmName As String)
```

```
' If form is hidden, set form's hidden property to False and open form.  
If Application.GetHiddenAttribute(acForm, frmName) = True Then  
    Application.SetHiddenAttribute acForm, frmName, False  
    DoCmd.OpenForm frmName  
End If  
End Sub
```

השיטה SetHiddenAttribute מסמנת או מנקה את המאפיין (Hidden) מחלוון מסד הנתונים, של אובייקטי מסד נתונים, כגון טפסים, דוחות ושאלות. שיטה זו כוללת שני ארגומנטים, אובייקט מסווג AccessObject ורגומנט בוליани שמצוין אם יש להסתיר את האובייקט. קריאה לשיטה באמצעות אובייקט וביצורו הערך True זהה לקביעת המאפיין **מוסתר** (Hidden) של האובייקט בחלון מסד הנתונים.

השיטה SetHiddenAttribute לבדה רק מעממת את האובייקט ; המשמשים יכולים עדיין לבחור אותו ולהשתמש בו. כדי להציג שאובייקטים מוסתרים יהיו בלתי נראים למשתמשים, בחר את **אפשרויות** (Tools) מתרחיט **כלים** (Options), לחץ על **אובייקטים מוסתרים** (Hidden Objects) ולאחר מכן לחץ על אישור.

בטרם תפעיל את השיטה SetHiddenAttribute, عليك לבדוק את המאפיין IsLoaded של האובייקט AccessObject. אם האובייקט טוען, عليك לסגור אותו בטרם תנסה להפעיל את SetHiddenAttribute ; קריאה לשיטה כשהאובייקט פתוח גורמת לשגיאה.

מספר טפסים בפרויקט אחר

VBA אינו מוגבל לטיפול באובייקטים של מסד נתונים בפרויקט הנוכחי בלבד. לדוגמה, באפשרותך לבדוק קיום של טפסים ביישום אחר של Access. אחד החלבים החשובים בתהליך זה הוא להשוו את המאפיין Name (שם) של כל חברי האוסף AllForms טופס היעד. قيمة תחולה נוספת שעליך למלוד: פותחים מופיע חדש של טופס היעד. Application של Access כולל את מסד נתונים היעד ולאחר מכן משתמשים בפרויקט הנוכחי של מופיע זה בהתאם המקור עבור האוסף AllForms. שיפור עדין זה מאפשר לטפל באובייקטי מסד נתונים השبيיכים לקובץ מסד נתונים אחר.

שתי השגרות שלפניך מומשוות את הרעיון באמצעות VBA. השירה FormToLookFor מגדרה את נתיב מסד הנתונים אל קובץ מסד הנתונים الآخر וקולטת את השם של טופס היעד. השירה השניה, FormExistsInDB, מחפש טופס יעד. באפשרותך לקרוא לטופס השני מתוך הטופס הראשון.

```
Sub FormToLookFor()
Dim strDB As String
Dim strFormName As String
' Search for forms in the Northwind database.
    strDB = "C:\Program Files\Microsoft Office\Office\Samples\Northwind.mdb"
' Get the name of the form to search for from the user.
    strFormName = InputBox("Enter the name of the form to search for: ", _
                          "Programming Microsoft Access 2000")
' Call FormExistsInDB to check whether the form exists.
    FormExistsInDB strDB, strFormName
End Sub

Sub FormExistsInDB(strDB As String, strFormName As String)
Dim appAccess As Access.Application, int1 As Integer
' Return reference to Microsoft Access Application.
    Set appAccess = New Access.Application
' Open a database in the other application.
    appAccess.OpenCurrentDatabase strDB
' Check whether the form exists.
    For int1 = 0 To (appAccess.CurrentProject.AllForms.Count - 1)
        If (appAccess.CurrentProject.AllForms.Item(int1).Name _ 
            = strFormName) Then
            MsgBox "Form " & strFormName & " exists in the " & strDB & _
                   " database.", vbInformation, "Programming Microsoft Access 2000"
            GoTo FormExistsExit
        End If
    Next int1
FormExistsExit:
```

```

' Report that form does not exist.
MsgBox "Form " & strFormName & " does not exist in the " _
& strDB & " database."
' Close other Access application.
FormExistsExit:
    appAccess.CloseCurrentDatabase
    Set appAccess = Nothing
End Sub

```

השיגרה הראשונה נותנת ל-`strDB` ערך טיפוסי של נתיב אל מסד הנתונים `Northwind`. אם קיימים עותק של `Northwind` במקומות אחרים כלשהו, עליך לעמוד את הנתיב. פונקציה מסווג `InputBox` מנהה את המשטמש להזין את שם הטופס שיש לחפש, ולאחר מכן השיגרה ראשונה קוראת לשיגרה השנייה.

השיגרה השנייה קובעת ופוחתת הפניה אל המופיע החדש של היישום של `Access` ולאחר מכן מגיעה לוולה שבודקת אם טופס כלשהו במסד הנתונים החדש תואם לשם טופס היעד. השיגרה מדוחת אם מצאה את טופס היעד ומשחררת את המשתנים שהוקצו למסד הנתונים החדש לפני חזרתה.

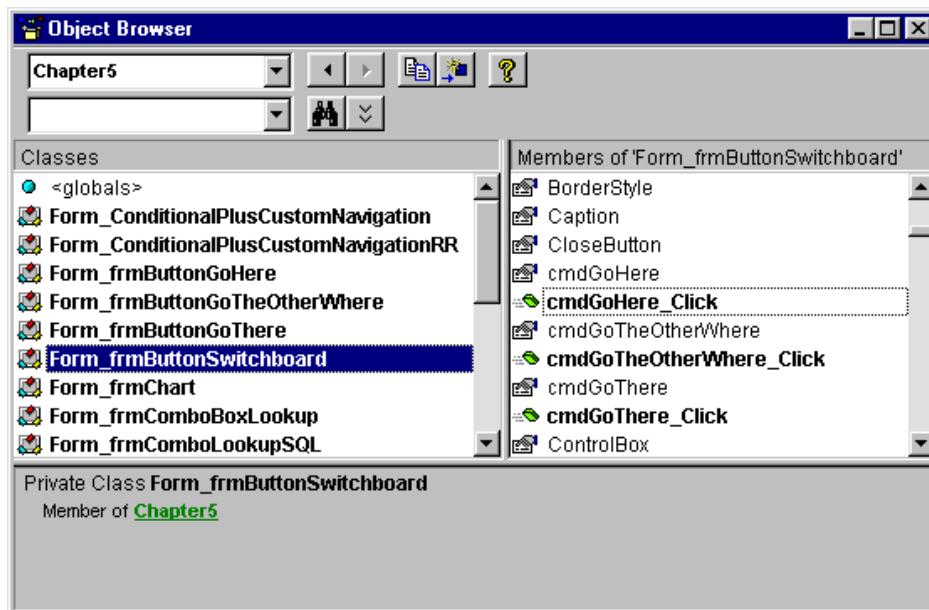
שימוש במחלקות טופס

טופס כלשהו של `Access 2000` בעל מודול מאחוריו הוא מחלקת טופס ; באפשרות ליצור מופעים חדשים של המחלקה בעזרת מילת המפתח `New` בדיק כפי שהיא נוהג עם מחלקה כללית (גנרטית) כלשהי של `Access`. אחד היתרונות של מחלקות טופס הוא שהן כוללות את כל השיטות ומאפייניו הטופס התקניים וגם את התוספות המותאמות אישית שהמפתח מוסיף להן.

הציג מאפיינים ושיטות של מחלקת טופס

באפשרותך להציג את המאפיינים והשיטות של מחלקות טופס באמצעות **Object Browser** (סורק האובייקטים). בחר בשם הפרויקט בתיבת הרשימה הנפתחת **Project/Library** (פרויקט/ספריה) ובחר שם של מחלקת טופס כדי להציג את המאפיינים והשיטות שלו.

سورק האובייקטים שבתרשים 5.15 מציג קבוצת משנה של חברים במחלקה `Form_frmButtonSwitchboard`. כזכור, טופס זה מנוט אל שלושה טפסים אחרים `.cmdGoHere_click`. שגרות האירוע הן שיטות מחלוקת, כגון `cmdGoHere`, `cmdGoHere`, `cmdGoHere`, הם מאפייני מחלוקת.



תרשים 5.15: שיטות ומאפיינים של מחלקה טופס

טיפול במחלקות טופס

השיגרה שלפניך מפנה למודולי מחלקה ב- Access 2000. השיגרה כוללת כמה מקטעי קוד שמתפללים במחלקת הטופס הבסיסית בדרכים מתחכחות וمتקדמות יותר. הטיפול במחלקות טופס ובמופעים שלהם דומה לטיפול בעוגיות באמצעות חותן עוגיות. מכשיר החיתוך הוא מחלקת הטופס והעוגיות הן מופעים שלו. שינוי בעוגיה אינו משפיע על מכשיר החיתוך. לעומת זאת, שינויים במכשיר החיתוך משפיעים על כל העוגיות לאחר ביצוע השינוי.

```
Sub testformclass()
Dim frm1 As Form

' First code segment
' Saves reference to instance of a form class in frm1.
' Can reference with either class or reference name.
Set frm1 = Form_frmCustomers
frm1.Caption = "foo"
MsgBox Form_frmCustomers.Caption
MsgBox frm1.Caption
DoCmd.Close acForm, frm1.Name
```

```

' Second code segment
' Programmatically alters and opens default form instance.
' Does not set reference to instance.
' Clears instance by setting Visible to False.

    Form_frmCustomers.Caption = "foo"
    Form_frmCustomers.Visible = True
    MsgBox Form_frmCustomers.Caption
    If MsgBox("Do you want to close form instance?", vbYesNo, _
        "Programming Microsoft Access 2000") = vbYes Then
        Form_frmCustomers.Visible = False
    End If

' Third code segment
' Open form in Design view to modify class properties.
' Open in Form view to see impact of Design view change.

    DoCmd.OpenForm "frmCustomers", acDesign
    Forms("frmCustomers").Caption = "foo"
    MsgBox Form_frmCustomers.Caption
    DoCmd.Close acForm, "frmCustomers", acSaveYes
    Form_frmCustomers.Visible = True
    MsgBox Form_frmCustomers.Caption

' Fourth code segment
' Restore class caption property.

    DoCmd.OpenForm "frmCustomers", acDesign
    Set frm1 = Form_frmCustomers
    frm1.Caption = "Customers"
    DoCmd.Close acForm, frm1.Name, acSaveYes
    Form_frmCustomers.Visible = True
    MsgBox Form_frmCustomers.Caption

End Sub

```

מڪטע הקוד הראשוֹן מڪצה מחלקה טופס להפניה. הוא יוצר מופיע של המחלקה Form_frmCustomers ומקצה אותו ל-frm1. באפשרות לטפל במופיע באמצעות שם המחלקה או שם ההפניה; התוצאות הזהות שמתקבלות משתי תיבות ההודעה מאשרות זאת. שינוי המופיע של מחלקה טופס אינו משנה את המחלקה עצמה.

אין צורך בהפניות כדי לציין או לשנות את המאפיינים של מופיע מחלקת הטופס. המڪטע השני מבצע שימוש זהה למשימה הריאונה מבלי ליצור מצביע אל מחלקה הטופס. מڪטע זה כולל גם הנחיה ששולת את המשמש אם ברצוננו לסגור את מופיע (מופע מתקיים לאורך חי השיגרה שיוצרת אותו, אלא אם הקוד מסיים אותו קודם לכך).

מڪטע הקוד השלישי פותח את הטופס בתצוגת העיצוב ומטפל במאפייניו. ברגען למקורה בו משנים את מאפייני הטופס באמצעות קוד בתצוגת **טופס** (Form), שינויי מאפיינים שנערכו בתצוגת העיצוב ממשיכים להתקיים גם לאחר שבירת הטופס. שתי השורות האחרונות של המڪטע השלישי פותחות מופע של מחלקת הטופס בתצוגת **טופס**. בפעם הראשונה בשירהה, הטופס נפתח עם הכתובם foo.

מڪטע הקוד הרביעי מחזיר את כתוב מחלקת הטופס לערכו המקורי, "Customers".

הפניות למופיעי מחלקת טופס

דוגמת מחלקת הטופס שלפניך יורדת לעומקן של מחלקות טופס. באפשרות לבדוק بكلות את התנהגות השיגרה על ידי שימוש בטופס frmFirstWithControls, הכלול לחץ שפעל את השיגרה .testformclass2. שיגרה זו מטפלת במופיעים רבים של מחלקת טופס.

```
Sub testformclass2()
On Error GoTo testclass2Trap
Dim frm1 As Form
Dim frm2 As New Form_frmCustomers
Dim int1 As Integer
' Show caption of frm2 before editing the property.
    MsgBox "Frm2 default caption is " & frm2.Caption, _
        vbInformation, "Programming Microsoft Access 2000"
' Set frm1 as a reference to frmCustomers class.
    Set frm1 = Form_frmCustomers
    frm1.Caption = "Caption from frm1 instance"
' Reset caption for frm2 instance.
    frm2.Caption = "Caption from frm2 instance"

' Show the captions of the class instances referenced by frm1 and frm2.
    frm1.SetFocus
    frm2.SetFocus
    MsgBox frm1.Caption, vbInformation, "Programming Microsoft Access 2000"
    MsgBox frm2.Caption, vbInformation, "Programming Microsoft Access 2000"

' Close form instances by their references.
    frm1.SetFocus
    DoCmd.Close
    frm2.SetFocus
    DoCmd.Close

testclass2Exit:
    Exit Sub
```

```

testclass2Trap:
    If Err.Number = 2467 Then
        ' Trap attempt to print caption for closed form.
        MsgBox "Cannot print caption of closed form", _
            vbInformation, "Programming Microsoft Access 2000"
        Resume Next
    Else
        Debug.Print Err.Number, Err.Description
        Resume testclass2Exit
    End If

End Sub

```

נוח מאוד להשתמש בהפניות אל מופעי מחלקה טופס בעת שימושם ביוטר ממופע אחד. הדוגמה לעיל מנצלת את המופעים frm1 ו-frm2. לאחר ההצהרה על frm1 Form_uncharacterized טופס כללית, הקוד מנסה לו בהמשך הפניה אל מופע של המחלקה frmCustomers. משפט ההצהרה על frm2 יוצר הפניה אל מופע של אותה מחלקה טופס, וכן frm1 ו-frm2 הם שני מופעים נפרדים של מחלקה טופס זהה.

משפט Dim השני יוצר מופע המבוסט על מחלקה טופס, באמצעות מילת המפתח New עליך להתייחס אל מחלקה טופס בעלת התכילתית Form. בעת שימוש במילת המפתח New ליצירת מופע). לאחר יצירת ההפניות תוכל להפעיל שייטות ומאפייני טופס רגילים של אובייקטים שאלייהם מתיחסים המשתנים. לדוגמה, השגירה מדירה את המאפיין Caption של frm2 ולאחר מכן מעבירה את המיקוד הראשית אל frm1 ולאחר מכן אל frm2. על ידי כך נפתחים שני הטפסים זה על גבי זה בסך של Access.

באפשרותך לסגור מופע של טופס על ידי העברת המיקוד אליו ולאחר מכן להפעיל את方法.Close של האובייקט DoCmd.

יצירת דוחות

הידע שרכשת בפרק הקודם בנושא טפסים, יהיה לך לעזור רב כשתרצה לשלב דוחות ביישומים שתיצור. למרות שדוחות אינם תומכים בפקדים אינטראקטיביים, תוכל לאכלס דוחות בפקדים שמציגים נתונים, כגון תיבות טקסט ותיבות סימן. דוחות יכולים להכיל גם תמונות גרפיות מאוגדות ובלתי מאוגדות ופקדי ActiveX לתצוגות מיוחדות, כגון תרשימים. ב- Microsoft Access 2000 היכולת ליצור תרשימים בדוח מושלמת בצורה דומה מאוד ליכולת המקבילה בטפסים.

פרק זה עוסק בהיבטים מעשיים של עיצוב ותצוגה בדוח. נסוק בתכנות להציג תוכן דינמי, וגם במספור ובטיפול בדוחות ובפקדים. הפרק מציג מקרה לדוגמה של ספר אורים ב- FrontPage המועד לעסק קטן שלו צרכי מחולקים שבדרך כלל ניתנים לטיפול ב- Access. שני סעיפים בפרק דנים בעיצוב דוחות. סעיף אחר עוסקת בהציג דוחות באמצעות תבנית קובץ תמונה (Snapshot) של Access. תבנית זו מאפשרת לשתף דוחות Access באמצעות דואר אלקטרוני באינטרנט, גם בתחנות שעודשה שאין בהן Access. שלושת הטעיפים האחרונים מציגים דוגמאות בנושאי תכנות ספציפיים. שני הטעיפים הראשונים עוסקים בתוכן דינמי בדוח. הטעיף האחרון מתעמק בטיפול באמצעות קוד בדוחות ובפקדים. סעיף זה מציג טכניקות לטיפול בסוגים רבים של האובייקט AccessObject, כגון חברים באוספים AllReports ו- AllForms.

כיצד ליצר דוח

תוכל ליצור דוח בשלוש דרכים:

- ↳ שימוש באשפים הדוחות האוטומטיים (AutoReport Wizard). זו הדרך הפשוטה ביותר, שאינה דורשת יותר מלחיצות אחדות בעכבר.
- ↳ שימוש באשפים אחרים. אשף הדוחות (Report Wizard) חוסך זמן וailו אשף התוויות (Label Wizard) ואשף התרשיים (Chart Wizard) מסייעים להתמודד עם בעיות עיצוב מיוחדות.
- ↳ שימוש בתצוגת **עיצוב** (Design). יצירת דוחות ידנית בתצוגת עיצוב מאפשרת גמישות מירבית בכל הנוגע לפרישה, עיצוב, מיון וקידוץ.

שימוש באשף הדוחות האוטומטיים

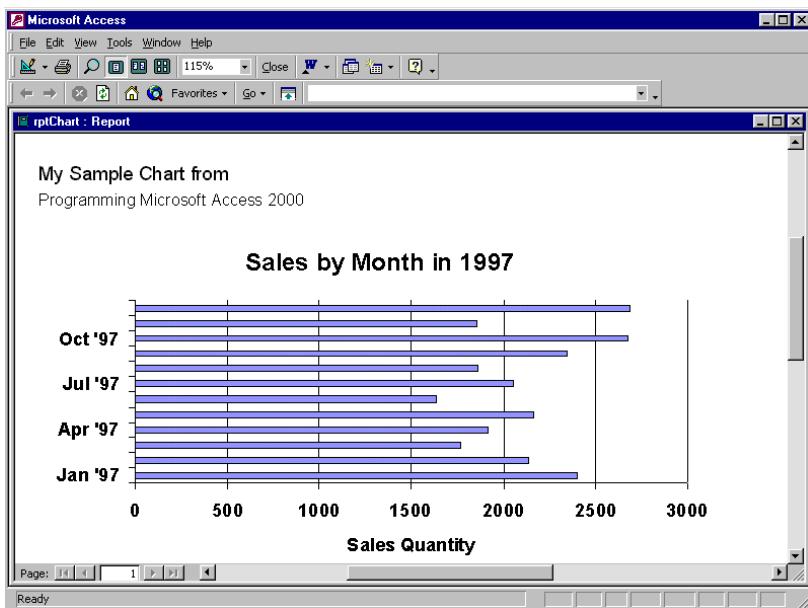
דוח אוטומטי המופעל מסרגל הכלים שבחלונו מסד הנתונים, הוא דוח ערוך בטורים. שדות טקסט ושדות מספריים מוצגים בתיבות טקסט, נתונים מסווג **כן/לא** (Yes/No) מוצגים בתיבות סימון ונ נתונים מסווג **אובייקט OLE** (OLE Object) (כגון תמונות מסווג *.bmp) מוצגים במסגרות אובייקטיבים מאוגדים. הרשומות מוצגות בזווית אחר זו בתבנית זורמת, החל ברשומה הראשונה וכלה באחרונה.

תוכל גם לבצע דוח טبליי בתיבת הדוח-שים **דוח חדש** (New Report). משיקולי מהירות ונוחיות, מפתחים متקדמים וכאליה בעלי ניסיון ממוצע יעדיפו ליצור דוח שונה באופן זה.

שימוש באשפים נוספים

Access כולל אשף דוחות כללי וגם אשפים מיוחדים לציררת תוויות ותרשיים. האשפיםמאפשרים ליצור דוחות וסוגי אובייקטיבים נוספים של מסד נתונים. אשפים מסווגים להאיץ את תהליך הפיתוח ולהנהיג סטנדרטיזציה שופשטת את התחזוקה.

אשף הדוחות (Report Wizard) מאפשר להגדיר דוח בצורה דומה מאוד לזו שבה אשף השאלות הפשוות מסויע בעיצוב שאלתה (ראה פרק 4). האשף חוסך את הזמן והמאיץ בעיצוב השאלה כשלב מקדמי להכנת דוח. בנוסף, הוא מפשט פעולות כגון קביעת סדר מיון, קיבוץ ועיצוב דוחות מותאמים אישית. **אשף התוויות** ואשף **התרשיים** מבצעים פעולות מיוחדות. המקרה לדוגמה של ספר האורחות שיוצג בהמשך הפרק, מציג דוגמת פלט של אשף התוויות. אשף התרשיים של דוחות דומה בפעולתו לאשף התרשיים של טפסים. תרשימים 6.1 מציג דוח תרשימים במצב הצג לפני הדפסה (כדי לרענן את זיכרונו) בנושא אובייקט Chart של Microsoft Graph 2000, *Microsoft Graph*, עיין פרק 5 בקטע העוסק בהעברת קבועות משנה של נתונים לתרשיים).



תרשים 6.1: תרשימים זה הוקן באמצעות אשף התרשיים של דוחות

יצירה ידנית של דוח בתצוגת עיצוב

תוכל ליצור דוח בצורה ידנית על ידי לחיצה כפולה על האפשרות **יצירת דוח בתצוגת עיצוב** (Create Report In Design View) בחלון מסד הנתונים (תחילה בחר את דוחות Reports) מהאוסף **אובייקטים**). יפתח דוח ריק שמאפייני הבסיס שלו משקפים את הערך שבתיבת הטקסט **מבנה דוח** (Report Template) בכרטיסיה **טפסים/דוחות Forms/Reports** של תיבת הדו-שיח **אפשרויות** (Options) (פותחים את תיבת הדו-שיח באמצעות הפקודה **כליים Tools**), **אפשרויות**. ברירת המחדל היא **רגיל Normal**, והיא מציגה דף לבן פשוט ללא צבעים או גופנים מיוחדים. אם יש לך דוח שברצונך לנצל את הגדרות המאפיינים שלו כתבנית לכל שאר הדוחות בפרויקט Access, תוכל להכניס את שם הדוח לתיבת הטקסט **מבנה דוח**. הגדרות דוח זה תהינה הגדרות ברירת המחדל של כל הדוחות החדשניים. דוחות קיימים ישארו ללא שינוי. תוכל לנצל את התבנית כדי לקבוע את מספר המקטעים שיוצגו בהתחלה בדוח ו גם את הגדרות ברירת המחדל של מאפייני מקטי וഫקיים.

הערה:

תיבת הטקסט **מבנה טופס** (Form Template) משמשת לייצור טפסים חדשים כפי שתבנית דוח משמשת ליצירת דוחות.



לאחר שתפתח מופע דוח חדש המבוסס על תבנית, תוכל לאכלס אותו בפקדים. חשוב להבין כי דוחות Access ערוכים במרקעים. בתבנית הרגילה למשל, המקטע **פירוט** (Detail) מוצג בין זוג מקטעים העמוד. כל תוכן הכלול במרקע העליון והמרקע התחתון מופיע בחלק העליון והתחתון בכל עמוד בדוח. המקטע **פירוט** חוזר על עצמו אחת לכל רשומה בעמוד, וגורר ערכיים עוקבים מתוך מקור הרשומות של הדוח.

זוג מקטעים נוספים שיכולים לתהום מקטע **פירוט** בדוח הם המקטעים **כותרת עליונה של דוח** (Report Header) ו**כותרת תחתונה של דוח** (Report Footer). מקטעים אלה מופיעים פעם אחת בלבד בתחילת הדוח ובסיומו. מפעילים ומכבים את המקטעים **דוח** ו**עמוד** התוחמים את המקטע המרכזי **פירוט**, באמצעות **פריט תצוגה** (View).

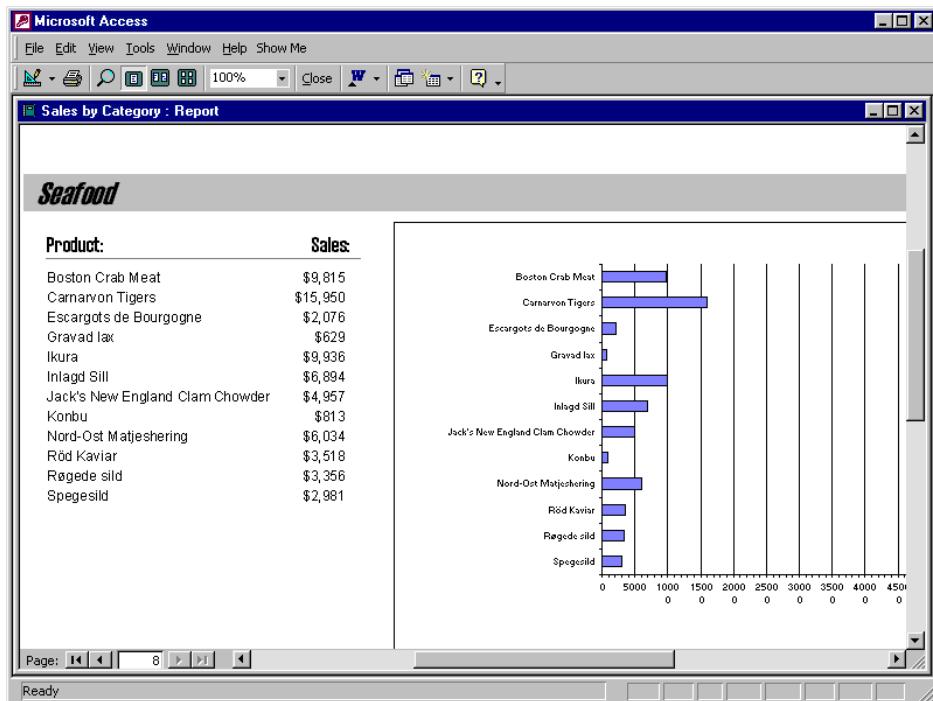
באפשרות לשנות את גודלו של מקטע דוח כלשהו על ידי גירירה של גבולותיו. לדוגמה, כדי להציג מקטע כותרת עליונה של דוח מבלי להציג את מקטע הכותרת התחתונה שלו, צמצם את המקטע לאפס, על ידי גירירת את הגבול התחתון של מקטע הכותרת התחתונה עד שיישק לגבול העליון שלו.

הדווש בתרשימים 6.1 מכיל את המקטע **כותרת עליונה של עמוד** (Page Header) ואת המקטע **פירוט**. המקטע **כותרת עליונה של עמוד** מכיל שתי כותרות: My Sample Chart: Programming MS Access 2000 from ו- **פירוט** של הדוח. פקד המסדרת אינו מאוגד למקור הרשומות, ולכן הוא מופיע פעם אחת בלבד.

פקדי דוח

דוחות של Access מכילים באופן טיפוסי תמהיל של תיבות טקסט ותוויות. תיבות הטקסט משקפות את ערכי השדות של מקור הרשומות שברקע, והთוויות מזהות את השמות של תיבות הטקסט ושל פקדים נוספים. פקדים נוספים שמציגים ערכי שדות בדוח כוללים תיבות סימון ומסגרות אובייקטיבים מאוגדים. המסגרות הן אמצעי הולם לשימרת תמונות גרפיות בטבלה. פקדי תמונה משמשים לשימרת תמונות בלתי מאוגדות שאינן ערכי שדות בטבלה.

דוח ראשי יכול לשמש גם בתור מראה של דוח משנה אחד או יותר ש모ופיעים כפקדים בדוח הראשי. הדוח Sales By Category במסמך הנתונים Northwind (תרשים 6.2) מכיל דוח משנה שמציג מכירות לפי מוצר. הדוח הראשי מציג נתונים זהים אך בצורה גרפית. דוח המשנה מופיע לשמאלו בצדreta בטבלה. התרשימים מנצל את האובייקט Chart של Microsoft Graph 2000 במוגדרת בלתי מאוגדת. דוח המשנה והתרשימים שניהם במרקע הכותרת העליונה CategoryName, מה שגורם לדוח המשנה ולתרשימים לבחור את המוצרים השיעירים לשם הקטגוריה המופיע בcotורת המקטע בכל דף. דוח יכול להכיל מקטעי קבועה מקוננים, כגון מחוזות במדינות או סוגים מסוימים בחטיבות של חברות.



תרשים 6.2: דוח משנה ותרשים בדף הדוח Sales By Category. המוצרים משתנים בכל כל כותרת של קטgorיה.

קוד ברקע של דוחות

ישומים מותאמים אישית דורשים בדרך כלל דוחות ידניים המוצברים באופן שחלום תכנים מוגדרים. Access מציע מגוון רחב של אפשרויות עיצוב באמצעות קוד, אך קרוב לוודאי שתנצל רק חלק קטן מהן :

↳ הדפסה או הצגה מקדימה אוטומטית של דוח על ידי הפעלת השיטה DoCmd, או יצירת היפר- קישור לדוח (סעיף "טופס מסך ניוט" בפרק 5 מציג דוגמאות שניתן לאמץ בדוחות).

↳ הפיכת הדוח לדינמי. ניתן לעדכן מאפיין אחד או יותר של האובייקט Report, באפשרות לשנות את מאפיין מקור הרשומות של דוח ואת הכיתוב של תוכית בדוח.

↳ מספור חברי האוסף AllReports כדי לקבוע אם דוח קיים ואם הוא פתוח. ↳ יצירת מופעים רבים של דוחות (דוחות בעלי מודולים הם מודולי מחלקה).

פתחחים מנוסים וכolumbia בעלי ניסיון ממוצע עובדים לעיתים קרובות עם אירופי דוח, הרגל שתוכל לאמצן כדי לשנות בဓורה דינמית בתוכן הדוח ובעיצובו. השתמש באירועי הדוח המותאמים אישית Deactivate , Close ,Activate ,Open במקומות באירועים

Terminate ו-Initialize לכל מופע חדש של מודול המחלקה הקשור אל דוח. בעת הדפסה או הצגה לפני הדפסה של דוח, מתרחשים האירועים Print ו-Format כנגד המקרים העוקבים שבדוח. שני האירועים הללו שימושיים לשינוי דינמי של תוכן הדוח ועיצובו. הגדרת המאפיין Keep Together בטור True עלולה לגרום לאירוע Retreat להתרחש בזרה סלקטיבית. מקטיע דוח מפרטים רצועות תואמים אך נבדלים זה מזה. לדוגמה, המקטע **כותרת עליונה של דוח** (Report Header) הוא המקטע הראשון כשהוא מוצג בדוח, והמקטע **כותרת תחתונה של דוח** (Report Footer) הוא המקטע האחרון כשמקטע זה כולל בדוח.

מקרה לדוגמה: ספר האורחים של FrontPage

אפשר ליצר בקהלות ספר אורחים מידע על מבקרים (כגון שם ופרטי יצירת קשר) בדף Web בתבנית HTML. גישה זו אינה מחייבת ידע כלשהו במסדי נתונים או בחיבורו קישוריות מסד נתונים פותחה (ODBC). מבקרי האתר רואים טופס בעל חזות מקצועית שאליו הם יכולים להוסיף פרט איש קשר, ומשתמשי FrontPage יכולים ליצור ספר אורחים ולהציגו בזרה קלה ופשוטה. השיטה מתפקדת היטב בכל עוד מספר המבקרים באתר אינו רב מדי.

אם כמות המבקרים באתר גילה משמעותית, ניתן להעביר את כל נתונים אנשי הקשר לקובץ מסד נתונים. גם אם האתר מעביר את הנתונים להפקדה במסד נתונים, עדין נותרה הבעיה לשחרר את המידע המקורי המוצב ב-HTML. מקרה זה מדגים גישה אחת לשחזור נתונים טקסט ישנים ומຕאר כיצד להפיק תוויות מען וטפסים המכילים את המידע לאחר המרתתו לבניית המתאימה.

ייבוא נתונים

ניתן לנצל את אשף יבוא טקסט כדי להעתיק את קובץ ספר האורחים הגלומי אל טבלה של Access. הראשים 6.3 מציג חלקים ובהם שתי רשומות מתוך קובץ HTML הגלומי (עבור האורחים Karl Doe1 ו-Boban Doe2).שים לב למספר העזום של תגי HTML. הסיר הפנימי של FrontPage מנצח תגים אלה כדי לעצב את תצוגת ספר האורחים, אך התגים אינם מאפשרים ייבוא נתונים ישירות לטבלה של הכנסת תוויות מען. כל מידע אנשי הקשר מופיע בעמודה ארוכה יחידה. יש שורות שמכילות נתונים קשר, אך שורות אחרות מכילות תוויות תיאוריות או תווים עיצוב כלליים. כללית העיבוד של מסדי נתונים יחסיים מחיברים להקצות שורה נפרדת לכל אורח שנרשם בספר האורחים. הדבר מחייב לפזר את המידע של אורח אחד הרשום בעמודה יחידה, על פני כמה עמודות.

תרשים 6.3: קטיעים מקובץ ספר האורחים של FrontPage, שיובא אל טבלה של Access

בין אם נתקلت בדרישת המרת נתונים מדויקת זו ובין אם לאו, פיתוח ב-Access מחייב באופן טיפוסי לטפל בזירמת טקסט. עסקים גדולים נדרשים לעיתים קרובות להמיר נתונים מחלקיים ישנים לתבנית של מסד נתונים ייחסי באמצעות תהליך דומה. המיגון העשר של פונקציות עיבוד מחוזות והשילוב של Visual Basic for Applications (VBA) ב-Access מאפשרים להפוך משימה זו לאוטומטית.

המרת מבנה הנתונים

המטרה של עיבוד מחוזות היא להפוך את הנתונים שבטבלה כגון Temp1 שבתרשים 6.3 לטבלת אנשי קשר מקובלת, כגון זו שבתרשים 6.4. אסטרטגיית הפעולה כוללת שתי ערכות רשומות: אחת עבור טבלת המקור ואחד עבור טבלת היעד. השיגרה עברת לאורך שורות טבלת המקור כדי לחלק את נתוני אנשי הקשר ולהחסן את הנתונים שעברו ניתוח במשתנים. לאחר שסיניימה לעבד את כל הנתונים של איש קשר, השיגרה מוסיפה אותן לשורה חדשה בטבלת היעד. מכיוון ששדות השינויים של איש קשר יש דרישות ייחודיות, חוקי העיבוד משתמשים לעיתים משדה לשדה. השינוי בחוקים עשוי לנבוע מכך שינויים גולמיים בשדה גורמים לביעות ייחודיות שלא קורות בשדות אחרים.

קטע השיגרה הבא הופך את תבנית נתונים הטבלה שבתרשים 6.3 למבנה נתונים הטבלה בתరשים 6.4. הנוסח המלא של השיגרה מופיע בתקליטור המצורף בספר. הגירסה המוקוצרת שבפרק מציגה את הקוד שמבצע את המרת שלושת השדות הראשוניים המיצגים באמצעות המשתנים strCname, strFname ו-strLname. תדפיס הקוד ארוך והנוסח המלא שלו ארוך עוד יותר, אך המרת נתונים היא השלב הראשון הكريティ של פרויקטים של מסדי נתונים של Access. אם לא הצליח לטפל בנושא קרייטי זה, פרויקט Access שלך לא יפעיל.

Contact ID	First Name	Last Name	Company Name	Address	Address	City
2	Karl	Doe1	HLC of Virginia	6221 Barrister Rd		Chesterfield
3	Boban	Doe2	Technical School 'Nikola Tesla'	Tabacka carsija 9/10		Pozarevac
4	Quilla	Doe3	Enterprise Connection	22 Willow St.		Marysville
5	Steven	Doe4	Apostolic Faith Church	3823 S. Indiana Ave		Chicago
6	Garland	Doe5	Bassett Furn Ind Inc	662 Westover Dr		Bassett
7	Conrad	Doe6	Saddleback Community College	2214-B Via Mariposa E		Laguna Hill
8	Jason	Doe7	Spydee Internet Services	2240 S. Hwy 29		Pensacola
9	Bill	Doe8	Digital Listing Services	2201 Baker Court		Antioch
10	John	Doe9	Powermatic Sdn Bhd	71 Jln USJ 11/1g		Subang Jay
11	ADIL	Doe10	RESEARCH	31-77	36 STREET	ASTORIA

תרשים 6.4: נתונים ספור האורחים של FrontPage לאחר המרה מtabנית HTML. המירה מסירה את תגי העיצוב ומציבה כל רשומות איש קשר בשורה נפרדת.

```
Sub getfp()
Dim cnn1 As New ADODB.Connection
Dim rst1 As ADODB.Recordset
Dim strFname As String, strLname As String
Dim strCname As String, strSt1 As String
Dim strSt2 As String, strCity As String
Dim strSt As String, strPostal As String
Dim strCountry As String, blSkip As Boolean
Dim rst2 As New Recordset
' Open two recordsets and set references to them.
    cnn1 = CurrentProject.Connection
    Set rst1 = New Recordset
        rst1.ActiveConnection = CurrentProject.Connection
        rst1.CursorType = adOpenKeyset
        rst1.LockType = adLockOptimistic
' Raw contact information is in table temp1.
    rst1.Open "temp1"
    rst2.ActiveConnection = CurrentProject.Connection
    rst2.CursorType = adOpenKeyset
    rst2.LockType = adLockOptimistic
' The application stores parsed contact info in the WebBasedList table.
    rst2.Open "WebBasedList"
' Start a loop through the recordset of raw contact information.
    Do Until rst1.EOF
        blSkip = False
' Start a new contact record when you find
'a label named "SiteEvaluation_FirstName:".
        If InStr(1, rst1.Fields(1),"SiteEvaluation_FirstName:") <> 0 Then
            rst1.MoveNext
            If rst1.Fields(1) <> " <dd>&nbsp;</dd>" Then
```

```

' The length of the first name field is the number of
' characters between ">" and "<" delimiters.
    intFirst = InStr(1, rst1.Fields(1), ">") + 1
    intLen = InStr(6, rst1.Fields(1), "<") - intFirst
    strFname = Mid(rst1.Fields(1), intFirst, intLen)

' Move two records to process last name field.
    rst1.Move 2
    Else
' If the first name is blank, set a Boolean flag to skip the whole record.
        bISkip = True
    End If

' Process last name field.
    intFirst = InStr(1, rst1.Fields(1), ">") + 1
    intLen = InStr(6, rst1.Fields(1), "<") - intFirst
    strLname = Mid(rst1.Fields(1), intFirst, intLen)

' Process company name field.
    rst1.Move 2
    If rst1.Fields(1) <> " <dd>&nbsp;</dd>" Then
        intFirst = InStr(1, rst1.Fields(1), ">") + 1
        intLen = InStr(6, rst1.Fields(1), "<") - intFirst
    End If

' If there is a leading blank in the company name field,
' see if you can find the name after the blank.
    If InStr(2, rst1.Fields(1), "&nbsp;") <> 0 Then
        intLen = InStr(6, rst1.Fields(1), "&nbsp;") - intFirst
    End If

' The parsing rule for the company name field converts with the new
' VBA Replace function html's " into a single apostrophy.
    strCname = Replace(Mid(rst1.Fields(1), intFirst, intLen), """, "")
    Else
' Set company name to zero-length string if there is no entry for the field.
        strCname = ""
    End If

.
.
.

' If Boolean skip flag is False, copy converted contact information
' to rst2, which is reference for WebBasedList table.
    If bISkip = False Then
        With rst2
            .AddNew
                .Fields("FirstName") = strFname
                .Fields("LastName") = strLname
                .Fields("CompanyName") = strCname
                .Fields("Address1") = strSt1
        End With
    End If

```

```

.Fields("Address2") = strSt2
.Fields("City") = strCity
.Fields("StateOrProvince") = strSt
.Fields("PostalCode") = strPostal
.Fields("Country") = strCountry
.Update
End With
End If
End If

' Move to next record in temp1 table and start search
' for a record including label for first name.
rst1.MoveNext
Loop

End Sub

```

כל אחד משלשות השדות המומרים מtabסס על לוגיקת ניתוח שונה במקצת. שאר השדות מטופלים בצורה יותר עקבית. יחד עם זאת, קיימים כמה הבדלים ברורים בחוקי הניתוח בין שדות חלופיים. באפשרותך לקרוא את העזרות ולחזור את הקוד כדי להבין טוב יותר פונקציות VBA שימושיות שמעבדות מחרוזות. בתום תהליך המהרה, הקוד בודק ומבודד שלא הייתה המרת שדה כלשהו שקבעה את דגל הדילוג בטור False. ערך של הדגל מאפשר לשיגורה להוסף רשותה חדשה ל לבטל הקשר היחסית בתבנית המוצגת בתרשים 6.4. לאחר מכן השיגורה עוברת לרשותה חדשה בבטלת המקור (עותק ספר האורחים של FrontPage). כשבטלה המקור מוחזירה את משתנה EOF בטור EOF, משתניימת לולאט Do החיצונית והשיגורה נעצרת.

יצירת תוויות מען

תרשים 6.5 מציג את תצוגת העיצוב של דוח תוויות מען וגם קטע מההצוגה **הציג לפניו הדפסה** (Preview) של תוויות המען. התצוגות מבוססות על נתונים אנשי הקשר מהטבלה שברישום 6.4.

Access כולל אשף תוויות מען גרפיות שמתאימות לmdbcot בשלל גדלים עבור מיוגון דגמי מדפסות לייזר ואחרות. האשף מאפשר גם להגדיר גדלים חדשים של תבנית תוויות. הוא מבטל את הצורך כתוב ביטויים לטיפול במחרוזות כדי ליצור את מבנה התוויות. החלון העליון בתרשים 6.5 מציג את שדות התווית בצורה מוגדלת מהרגיל שמאפשרת להבחין בביטויים שהם מכילים. האשף מתאים אוטומטית את גודל תיבות הטקסט המכילות את ביטויי המחרוזות שלו, כך שהתוויות מתאימות לטופס.

Karl Doe1 HLC of Virginia 6221 Barrister Rd Chesterfield, VA 23832	Boban Doe2 Technical School 'Nikola Tesla' Tabacka carsija 9/10 Pozarevac, SERBIA 12000	Quilla Doe3 Enterprise Connection 22 Willow St. Marysville, PA 17053
Steven Doe4 Apostolic Faith Church 3823 S. Indiana Ave Chicago, IL 60653	Garland Doe5 Bassett Furn Ind Inc 662 Westover Dr Bassett, VA 24055	Conrad Doe6 Saddleback Community College 2214-B Via Mariposa E Laguna Hills, CA 92653

Page: |<|<|>|>|>>|

תרשים 6.5: החלון העליון מציג את דוח תוויות המען בתצורת עיצוב; החלון התחתון מציג קטע מדף התוויות במצב תצוגה לפני הדפסה.

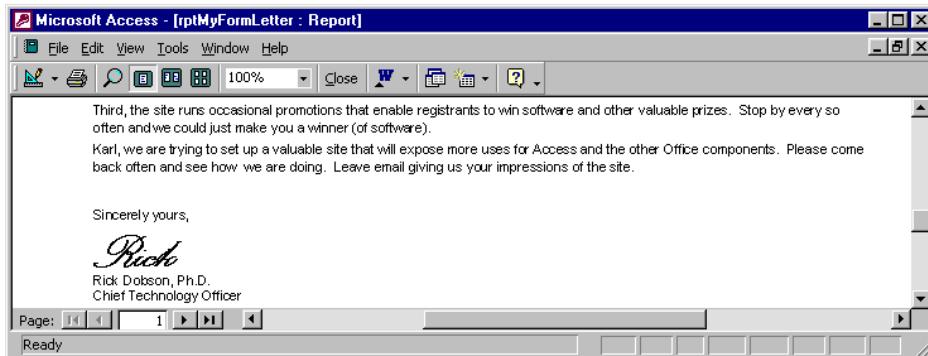
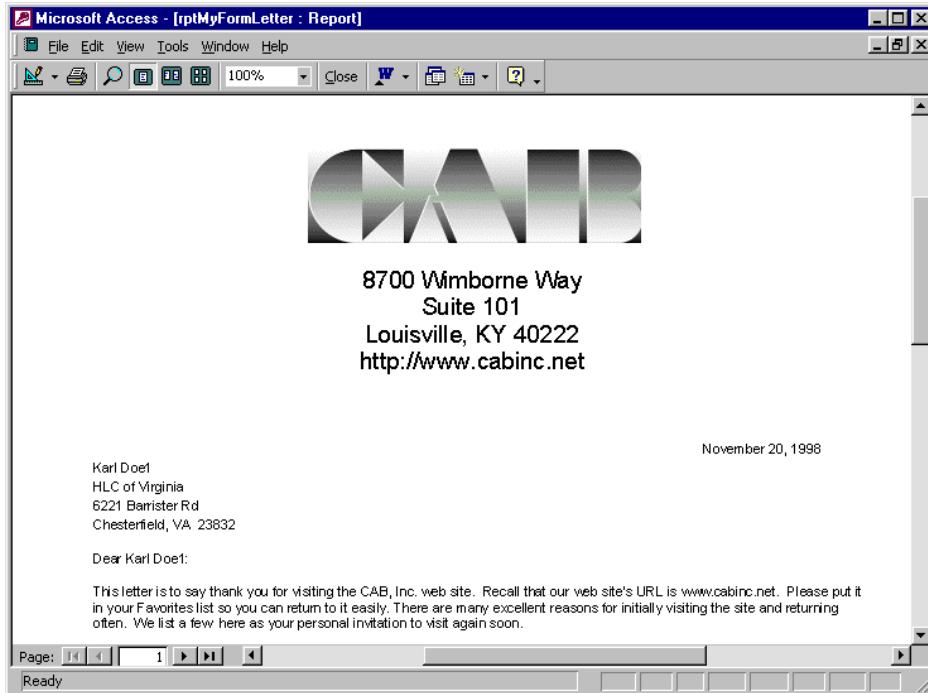
יצירת מכתב אחד

שימוש טיפוסי נוספת בטלטת אנשי קשר הוא במכתב אחד. שילוב של פקדים, קבועים ומשתנים של מחוזות ופונקציות VBA מותאמות אישית ליצרן מכתב אחד, כגון זה שמצוג בתרשים 6.6. המכתב מגינן כמה תכונות ראיות לצוין אופייניות לדוח של Access :

- ◀ בראש המכתב מופיע לוגו.
- ◀ פרטי מען לשיגור מענה מופיעים בגוף שונה מזו של גוף המכתב.
- ◀ תאריך המכתב מציג את שם החודש.
- ◀ כתובת הדואר היוצאת והברכה משתנות מרשותה לרשותה.
- ◀ הפקשה האחורה במכתב משתנה אף היא לכל רשותה.
- ◀ דברי הסיום מלאוים בחתימה.

הمرת מספר חדש לשמו (כגון המרת הערך 1 ל"ינוואר") לא הייתה מעולם בעיה מיווחדת בתכנות, אך ב- Access 2000 וב- VBA 6 הדבר פשוט עוד יותר. תיבת הטקסט המציגה את התאריך מגלה את החידוש ודרך אפשרית לניצולו. היא מכילה את מחרוזת הביטוי הבאה :

```
=ThisMonthName() & " " & Day(Date()) & ", " & Year(Date())
```



תרשים 6.6: קטע ממכתב אחיד

הбиיטוי מכיל זוג פונקציות מקוונות מוכללות עבור היום והשנה, אך פונקציה מותאמת אישית ב-

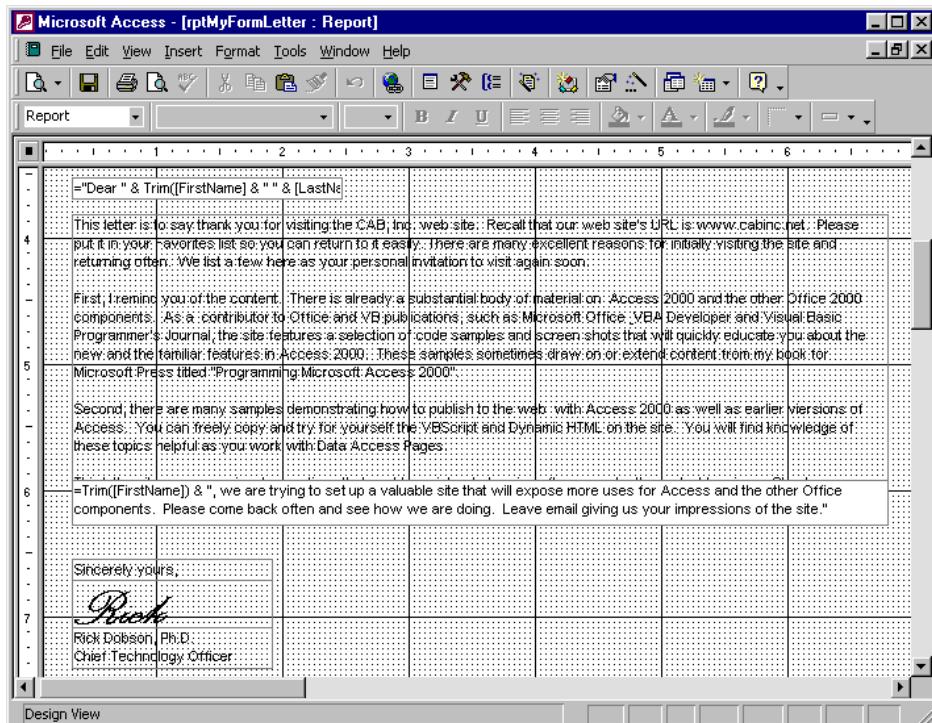
אישית (ThisMonthName) מחזירה את שם החודש. לפניך פונקציה מותאמת אישית ב-

VBA, ThisMonthName, שממירה את מספר החודש בשם המותאים. הפונקציה פוטרת 6 אוטך מהצורך ל כתוב את המשפט Select Case בשגרת פונקציה, או להפעיל פונקציה מסוג Choose כדי לגרום להציג שם חדש בדוח.

```
Public Function ThisMonthName()
    ThisMonthName = MonthName(Month(Date))
End Function
```

יתכן שתהיה מודע לא ניתן לישם את הפונקציה MonthName ישירות בתיבת הטקסט בדוח. לו עשית זאת, היה גורם לשגיאה מכיוון ש- MonthName אינה פונקציה של Access. עליך לקרוא לה בשירה ולהציג את התוצאה לאובייקט של Access, כגון TibetText בדוח.

תרשים 6.7 מציג את תצוגת העיצוב של דוח המכתב האחד המכיל שילוב של פקידי תווית ותיבת טקסט. אם גופו המכתב אינו מכיל רכיבי התאמה לכל רשותה בנפרד, ניתן להציג את כל הטקסט באמצעות תוויות פשוטה. לעומת זאת, לאחר שהפיסקה האחורייה מתחלילה בשם הקשר הראשון, הדוח זוקק בדרך לשחזר ערך השדה בין ערך השדה וקובע המחרוזות. ניתן לראות כי יש לנסוט מהציב את תיבת הטקסט ביחס לפקד התווית שלפנינו. כדי לשפר את זרימת הטקסט בתצוגה המקדימה, עליך לקיים חפיפה בין פקדים בתצוגת העיצוב. יתכן שהייה צורך בכך כמה ניסיונות עד לקבלת עיצוב נאות.



תרשים 6.7: חלקו התיכון של המכתב מתרשים 6.6, והפעם בתצוגת עיצוב

שים לב לשימוש בגוף script בפקד תווית שנועד לדמות חתימה. זהוי אפשרות בת-קיימא למקרים בהם לא נדרשת חתימה בפועל – לדוגמה, לדואר בתפוצה גדולה, כגון מכתבים אחידים.

מיון, קיבוץ וחישוב

אחד הייתרונות הגדולים של תוכנת העיצוב הוא יכולתה לקבץ ולמין רשומות להצגה בדוח. יכולת זו מבוססת על שני גורמים: מקטיע פירוט של הדוח ותיבת הדוח-שייח מיון וקיבוץ (Sorting And Grouping). אפשרות הקיבוץ מפשטת את חישוב סכומי הבינאים לפי קבוצה וגם מחשבת את הסכומים הכלולים.

תרשים 6.8 מציג דוח פשוט שמדגים רכיבי מיון, קיבוץ וחישוב בסיסיים בדוח. הדוח בן שני העמודים מציג את כותרת הדוח בראש העמוד הראשון בלבד. פקד התוויות של כותרת הדוח ממוקם על-כן בmaktיע **כותרת עליונה של דוח** (Report Header). לעומת זאת, שתי כותרות העמודות מופיעות בmaktיע **כותרת עליונה של עמוד** (Page Header) מכיוון שהן מופיעות בראש העמודות בשני עמודי הדוח. המקטיע **פירוט** (Detail) מכיל זוג תיבות טקסט מאוגדות לשדות של מקור הרשומות של הדוח. שתי תיבות הטקסט בדוח מוצגות פעם אחת לכל רשומה במקור הרשומות שברקע. סדר המינו של שורות אלו יכול להיות בלתי תלוי בסדר הרשומות במקור הרשומות שברקע. יכולה זו לשפר את השימושיות הכלולית במקורות רשומות בעובדה עם דוחות רבים.

Orders by Month	
Month and Year	Orders
07/01/1996	22
08/01/1996	25
09/01/1996	23
Sales for quarter: 70	
10/01/1996	26
11/01/1996	25
12/01/1996	31
Sales for quarter: 82	
01/01/1997	33
02/01/1997	29
03/01/1997	30
Sales for quarter: 92	
04/01/1997	31
05/01/1997	32
06/01/1997	30
Sales for quarter: 93	

Month and Year	Orders
07/01/1997	33
08/01/1997	33
09/01/1997	37
Sales for quarter: 103	
10/01/1997	38
11/01/1997	34
12/01/1997	48
Sales for quarter: 120	
01/01/1998	55
02/01/1998	54
03/01/1998	73
Sales for quarter: 182	
04/01/1998	74
05/01/1998	14
Sales for quarter: 88	

תרשים 6.8: דוח שמציג קיבוץ לפי תאריך

השורות שבתרשים 6.8 מוקבצים לפי רביעון. תיבת הדוח-שיח **מיון וקיובץ** בתצוגת העיצוב משמשת לבחירת שדה אחד או יותר שליפוי מוקבצים שורות. קיובץ לפי שדה מוסיף לדוח את המקטעלת כוורת עלינה וכוורת תהוננה של דוח, לכל קבוצה חדשה. יכולת הקיובץ המוכلالת מתאימה אוטומטית לסוג נתוני השדה שליפוי מוקבצים את הדוח. במקורה של שדות תאריך, Access תומך בקיובץ לפי שנה, רביעון, חדש, שבוע, יום, שעה ודקה. שדות מסוג **מספר אוטומטי** (AutoNumber), **מטבע** (Currency) או **מספר** (Number) מאפשרים לקבץ שורות לפי טווח מותאם אישית. באפשרותך לקבע מחירי יחידת מוצר בהפרשים של \$5 בדוח אחד, ובഫזרים של \$10 בדוח אחר. באפשרותך גם לקבע שדות **טקסט** (Text) בהתאם לתווים המובילים שלהם. הדבר מאפשר ליצור דוח בסגנון מיליון מיליון מוצרים מוקבצים בו לפי שם: ראשית מוצרים שמתחילה באות א', אחריהם מוצרים שמתחילה באות ב' וכן הלאה.

יכולות הקיובץ תומכות בחישוב סכומי ביןיים. שים לב שהדוח בתרשימים 6.8 מסכם הזרנות מכירה לפי חדש ולפי רביעון. ניתן להפעיל פונקציית צבירה, כגון SUM, במקטע כוורת תהוננה של קבוצה כדי לחשב את פריטי הקבוצה. קיימות פונקציות צבירה נוספות שיכלולות להוסיף ערך לדוח: COUNT, AVG, MAX ו-MIN. פונקציית הצבירה של סכום הביניים בתרשימים 6.8 מופיעה בתיבת הטקסט במקטע **עלילונה של Date Footer** (Control Source). ביטוי **מקור פקד** (Date Footer) של תיבת הטקסט הוא

```
=Sum([CountOfOrderID])
```

ארגון הפקוד הפונקציה מצבייע על שדה ברקע של הפקוד במקטע **פיוט**. שימוש ב-`txtCountOfOrderID` בתור ארגומנט שמצויב ישרות על פקד תיבת הטקסט גורם לשגיאה.

הערה:



כללי החישוב של סכומי ביןיים נבדלים מאליה של שדות מחושבים אחרים בדוח. שדה מחושב רגיל, כגון צזה שבו מחשבים את הסכום לשורת פריט בהזמנה, מאפשר להפנות לערכיו הפסויים בראשמה הנוכחית. הביטויים שתכתוב יכולים גם להפנות לשדות הרקע של פקדים. בעת חישוב סכומי ביןיים, עליך להפנות לערך שדה הרקע.

יצירת דוחות מרובי עמודות

הדוגמאות שהוצעו עד כה בפרק אינם מציגות דוחות מסד נתונים קלאסיים בעלי עמודות רבות שמקורן בטבלת רקע אחת או יותר. באפשרותך ליצור דוח מסוג זה בשתי דרכים לפחות: באמצעות אשף הדוחות ובטכניקות עיצוב ידניות.

שימוש באשף הדוחות

אשף הדוחות (Reports Wizard), שהממשק שלו דומה לזה של אשף השאלות הפשוטות, מאפשר ליצור דוח מרובה עמודות. תרשימים 6.9 מציג דוח שהוכן באמצעות אשף הדוחות. כדי שניתן לראות, האשף מספק אמצעי חישוב רבים. מקור הרשומות שברקע הדוח מונה את מספר המכירות ומסכם את מחירי הסה"כ לפריט בחתך של לקוחות. הדוח מסכם גם את כמותה המכירה והמחירים לפי לקוח ולכלל הליקות. לאחר מכן הוא מנצל את התוצאות שהתקבלו כדי לחשב את החלק היחסי מסך המכירות הכלול ואת מחיר הסה"כ ללקוח. הוא מסכם גם את מספר הרשומות לכל לקוח ומדפיס הودעה בת שורה אחת ובה שם הליקות ומספר הרשומות שלו.

אשף הדוחות יכול להוסיף לדוח מיגון מיוחד של מאפייני עיצוב. תרשימים 6.9 מציג בנוסף בחריות אפשרי. השימוש בירוח ובקוים אופקיים מאפשר להבדיל את כותרת הדוח משאר חלקיו. באפשרות גם לבחור את אפשרות פרישת השדות הרצוייה מבין אפשרויות אחדות. פרישת הדוח שבתרשים ממקמת את משתנה הקיבוץ, CompanyName, בקצת השמאלי-קיצוני של עמודי הדוח. שם לב כי פרישה זו אינה דורשת כתיבת קוד כלשהו.

CompanyName	ProductName	CustomerID	Orders	ExtPrice
Alfreds Futterkiste	Aniseed Syrup	ALFKI	1	\$60
	Chartreuse verte	ALFKI	1	\$284
	Escargots de Bourgogne	ALFKI	1	\$503
	Flôte myosot	ALFKI	1	\$430
	Grandma's Boysenberry Spread	ALFKI	1	\$380
	Lakkalikööri	ALFKI	1	\$270
	Original Frankfurter grüne Soße	ALFKI	1	\$21
	Raclette Courdavault	ALFKI	1	\$825
	Rössle Sauerkraut	ALFKI	2	\$604
	Spesigfild	ALFKI	1	\$18
	Vegie-spread	ALFKI	1	\$878
Summary for 'CompanyName' = Alfreds Futterkiste (11 detail records)				
Sum			12	\$4,273
Percent			0.56%	0.34%
Ana Trujillo Emparedados y he	Camembert Pierrot	ANATR	1	\$340
	Gudbrandsdalsost	ANATR	1	\$29

תרשים 6.9: קטע הדוח שלפניך מציג חלק אפשרויות החישוב והעיצוב של אשף הדוחות

יצירת דוח מותאם אישית

כשיש צורך בסידור מיוחד של נתונים ושל חישובים ייחודיים בדוח, יש לעצב את הדוח בצורה ידנית. תרשימים 6.10 מציג קטיעים מדויק מותאם אישית שרכיביו התפקודיים דומים לאלה של הדוח שבתרשים 6.9. דוח מותאם אישית זה מעוצב בצורה פשוטה יותר כדי שייהיה קל יותר להבין את מרכיביו. החלון העליון מציג את המכירות לפי מוצר עבור לקוח ייחודי עם סכומי הכספיות של הזמנה, מחיר סה"כ לפרט והאחוזים מסך כל הזמנות עבור שני סוגי החישובים. החלון התחתון מציג את המשך הדוח, בו מתחילה הלקוח השנייה. מקטע הלקוחות התחתונה של הדף מציג את תאריך הדוח, עמוד הדוח הנוכחי ומספר העמודים הכלולים.

Product Sales by Customer

ProductName	Orders	Extended Price
Raclette Courdavault	1	\$825
Aniseed Syrup	1	\$60
Vegie-spread	1	\$378
Rössle Sauerkraut	2	\$604
Original Frankfurter grüne Soße	1	\$21
Lakkaliköri	1	\$270
Grandma's Boysenberry Spread	1	\$380
Fletenmysost	1	\$430
Escargots de Bourgogne	1	\$503
Chartreuse verte	1	\$284
Spegesild	1	\$18
Alfreds Futterkiste Totals	12	\$4,273
Percent of Grand Total	0.6%	0.3%

Sales for Alfreds Futterkiste

Page: 1 / 1 Ready

Sales for Ana Trujillo Emparedados y helados

Konbu	1	\$60
-------	---	------

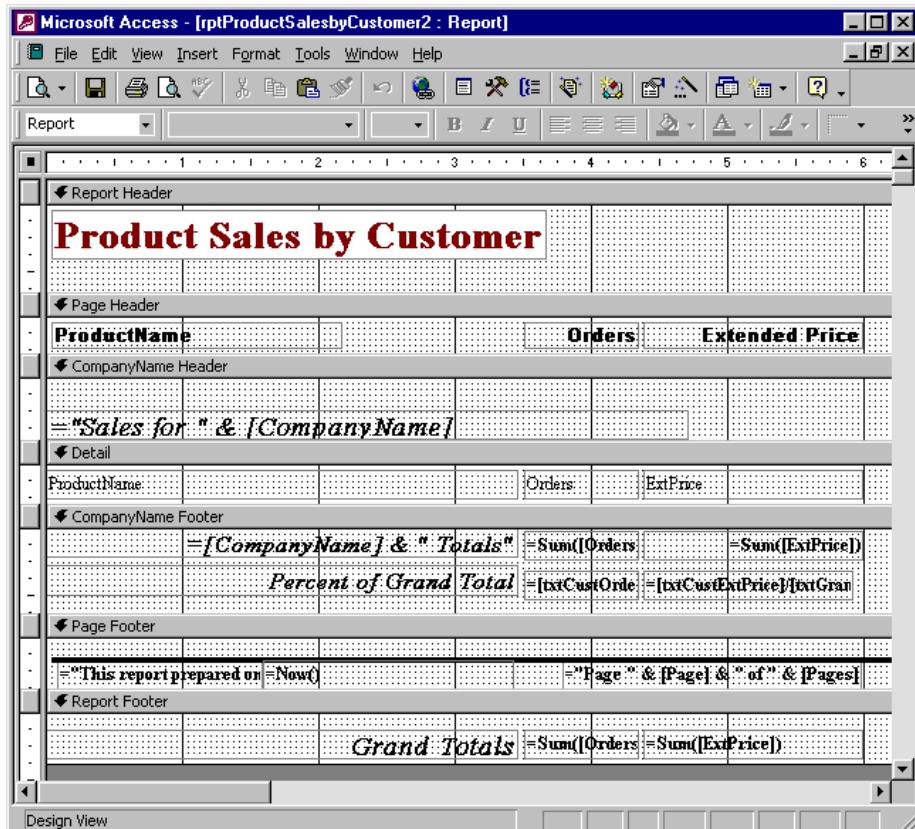
This report prepared on: Saturday, November 21, 1998 Page 1 of 103

Page: 1 / 1 Ready

תרשים 6.10: קטיעים מדויק מותאם אישית מרובה עמודות

תרשים 6.11 מציג את תצוגת העיצוב של הדוח מתרשים 6.10. כמעט כל פקדי הדוח הם תוויות או תיבות טקסט. עליך להזכיר קבועי כותרות לתוויות ולהגדיר את מאפיין הכתיבה שלהם בהתאם קבוע המחרוזת. הטקסט במקטע כותרת הדוח העלינה נמצאת בתווית.

פקדים שמצויגים שדות מסוימים או ביטויי מחרוזות לעולם יהיו תיבות טקסט. ביטויי המחרוזות הראשון מופיע במקטע **כותרת עליונה של CompanyName**. הוא נוצר על ידי שדרור של קבוע מחרוזות והשדה CompanyName במקור הרשומות של הדוח. אחריו מופיע המקטע **פירוט**. כל אחת משלוש תיבות הטקסט שלו מפנה לשדה אחד. פקדים אלה אינם משלבים את השדה בתוך ביטוי, ולכן אין צורך לקדם אותם בסימן שוויון (=), או להציב את שם השדה בסוגרים מרובעים.



תרשים 6.11: תצוגת עיצוב של הדוח מתרשים 6.10

ארבעת הפקדים בצד ימין של המקטע **כותרת תחתונה של CompanyName** מכילים ביטויים לחישוב סכום הזמנות המכירות והסה"כ ללקוח, יחד עם חלקו היחסי של לקוחות בסה"כ מספר היחידות והמחיר של הפריטים. הטבלה שלפניך מציגה את שמות הפקדים והביטויים שהם מכילים. ביטויי חישוב האחיזים מבוססים על שני

פקדים שמופיעים במקטעי הכותרת התחכונה של הדוח, txtGrandOrders ו- txtGrandExtPrice. פקדים אלה מחשבים את סה"כ כמות המכירות וסה"כ מחירי הפריטים של כל הלקוחות.

ביטויים לחישוב סכומים ואחוזים של תרשימים 6.11

ביטוי	שם פקד
Sum([Orders])	txtCustOrders
Sum([ExtPrice])	txtCustExtPrice
= [txtCustOrders]/[txtGrandOrders]	txtOrdersPercent
= [txtCustExtPrice]/[txtGrandExtPrice]	txtExtPricePercent
= Sum([Orders])	txtGrandOrders
= Sum([ExtPrice])	txtGrandExtPrice

הסכוםים של לקוחות ייחדים והסכום הכלול מתיחסים ישירות לערכי שדות הרקע Orders ו-ExtPrice. חישובי האחוזים של לקוחות בודדים מתיחסים לשמות הפקדים אשר תלויים בערכי השדות.

הערה:



כבריתת מחדל Access נותן לפקד תיבת טקסט שם זהה של השדה-orders הוא מאוגד. כדי להימנע מבלבול, תוכל לחתת לשדות שמות שונים. לדוגמה, אם שם השדה הוא Orders, אפשר לשנות את שם פקד תיבת הטקסט שמנוה אליו txtOrders. הדבר יקל עליך לקרוא ביטויים ולקבוע אם הם מתיחסים לערך של שדה או לפקד.

שים לב בתרשימים 6.10 כי תאריך הדוח מודפס בתחתית הדף, אך לפניו בא קבוע מחרוזת. אם ברצונך להחיל עליו תבנית, כגון התבנית Long Date (תאריך ארוך), عليك לבדוק את הפונקציה Now בתיבת טקסט ללא רכיבים נוספים. תרשימים 6.11 מציג שתי תיבות טקסט צמודות המכילות את מחרוזת הטקסט המובייל ואת הערך המוצב של הפונקציה Now הבא אחריה.

צידם הימני של הדוח המוצג בתרשימים 6.11 מנצל שתי מילוט מפתח, Page ו-Pages, שמצוות את מספר דף הדוח הנוכחי ואת מספר הדפים הכלול בדוח (מילוט המפתח אינו מופיע בתרשימים). שים לב כי אפשרות לכלול מילוט מפתח אלו בביטויים מחרוזת.

הפקת דוחות באמצעות תמונות

ניתן להפוך דוחות של Access לזרינים למשתמשים אחרים ב-WWW או בדואר אלקטרוני באמצעות קבצי תמונה (snapshot files). כדי שתחנתה עבודה תוכל להציג תמונה של דוח של דוח של דוח, יש להתקן בה מציג תמונות, אותו ניתן לקבל ללא תשלום בתור יישום נפרד, וגם פקד ActiveX לשימוש Access לא הותקן במחשבים רבים מושך אליו את הדוחות שלך, כאשר ש-Access לא הותקן במחשבים שלהם, וגם כאשר אינם מחוברים לרשת המקומית שלך. הדפדפן Netscape אינו תומך בפקדי ActiveX, אך יש אפשרות לטעון קבצי תמונה דרך חיבור http://. משתמשי Netscape יכולים להשתמש בגרסה העצמאית של מציג התמונות כדי להציג את קובץ התמונה השמור בכונן הקשיח שלהם.

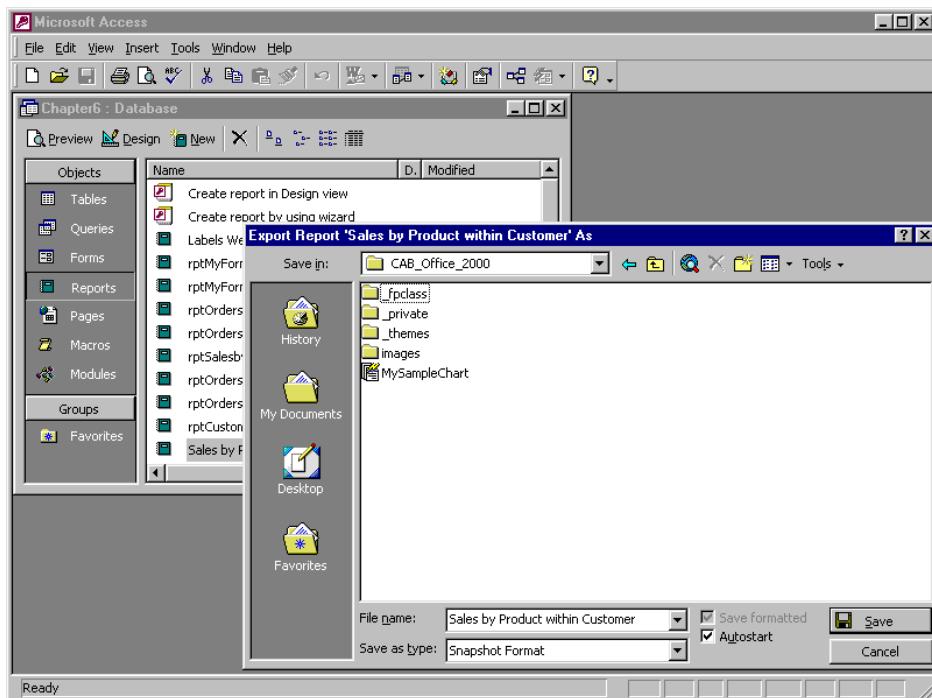
יצירת תמונה

באפשרות ליצור תמונה (snapshot) של דוח על ידי בחירת הדוח בחלון מסד הנתונים ובחירה באפשרות **יצא** (Export) מתפריט **קובץ** (File). הדבר מכניס את שם הדוח בתווך שם קובץ התמונה בתיבת הדוח-שיך **יצא קובץ** (Export File) (תרשים 6.12). בתיבת הרשימה הנפתחת **שמור כסוג** (Save As), בחרת הרשימה **Snapshot Format**. בתיבת הרשימה הנפתחת **שמור ב** (Save In), הקצתה מיקום עבור קובץ התמונה. תרשים 6.12 שומר את הדוח מתרשים 6.9 בספירה וירטואלית (שנקראת CAB_Office_2000) בראשת אינטרא-נט ארגונית. לחיצה על **שמור** (Save) בתיבת הדוח-שיך שבתרשים 6.12 פותחת תיבת דוח-שיך התקדמות ופותחת את הקובץ לאחר שמירתו, במציג התמונות (הכלול ב-Access). הקובץ מאוחסן בשרת, והתבנית שלו ***.snp**.

מצגת תמונה (Snapshot)

תחנת עבודה שימושתנים בה הדפדפן Internet Explorer מגירסה 3 ואילך ומציג תמונות Snapshot Viewer (Snapshot), יכולה לפתח קובץ תמונה. תרשים 6.13 מציג את הדוח ב-Internet Explorer 4. המציג נראה כאילו הוא חלק מהדף, אך למעשה של דבר, הוא משתלט עליו. פקדי ניווט מיוחדים בשורה התחתונה של פקץ ActiveX מאפשרים לנווט בין דפי הדוח. שורת הניווט כוללת גם לחץ הדפסה (מכיון שהפקד מבטל את רכיב ההדפסה המקומי של הדף).

שים לב להתקינה הקורובה בין תמונה הדפדפן שבתרשים 6.13 לבין תמונה הדוח המקורי בתרשים 6.9. רמת התקינה זו אינה מתקיימת כשמייצאים את הדוח למסמך בתבנית HTML. בנוסף, יצואו ל-HTML אינו מספק את תפקודיות הניווט המוכפלת שמציע פקץ מציג התמונות.



תרשים 6.12: ייצרת קובץ תמונה באופן ידני

פקד מציג התמונות מאפשר להציגו של Web Access בדף המכיל תוכן נוסף. קטע קוד HTML שלפניך מציג גוף דף Web ובו תagi H3 ו-H4 לפני הפניות האובייקט לפקץ מציג התמונות. קטעי הטקסט הבאים לפניו הפקד מסבירים כיצד לטעון את המציג אם הדוח אינו מופיע ומציגים היפר- קישור לאתר אינטרנט ממנה יכול המשתמש לטעון את הפקד. ההוראות מסבירות גם כיצד להציג את הדוח באמצעות Netscape Navigator. באופן טיפוסי תרצה לעדכן את הגדרת הערך Param, SnapshotPath, לפि כתובת ה-URL בה נמצא קובץ התמונה שברצונך להציג.

```

<body>
<H3>Snapshot Sample page</H3>
<H4>If you have an IE 3+ browser but cannot see the report below,
download and install the
<a href="http://www.microsoft.com/accessdev/prodinfo/snapdl.htm">
Microsoft Access Snapshot viewer</a>. Then
refresh the page. Netscape users will not even see the report
container in their browsers, but they can open the report
outside their browser using the same snapshot viewer mentioned
above. Netscape users can download the snapshot file from
a web server over http protocol to their workstation.</H4>
<OBJECT ID="SnapshotViewer" WIDTH=640 HEIGHT=480
CLASSID="CLSID:F0E42D60-368C-11D0-AD81-00A0C90DC8D9">
```

```

<PARAM NAME="_ExtentX" VALUE="16722">
<PARAM NAME="_ExtentY" VALUE="11774">
<PARAM NAME="_Version" VALUE="65536">
<PARAM NAME="SnapshotPath"
VALUE="http://cab2200/cab_office_2000/MySampleChart.snp">
<PARAM NAME="Zoom" VALUE="0">
<PARAM NAME="AllowContextMenu" VALUE="-1">
<PARAM NAME="ShowNavigationButtons" VALUE="-1">
</OBJECT>
</body>

```

Sales by Product within Customer

CompanyName	ProductName	CustomerID	Orders	ExtPrice
Alfreds Futterkiste	Aniseed Syrup	ALFKI	1	\$60
	Chartreuse verte	ALFKI	1	\$284
	Escargots de Bourgogne	ALFKI	1	\$503
	Fløtemysost	ALFKI	1	\$430
	Grandma's Boysenberry Spread	ALFKI	1	\$380
	Lakkalikööri	ALFKI	1	\$270
	Original Frankfurter grüne Soße	ALFKI	1	\$21
	Raclette Courdavault	ALFKI	1	\$825
	Rössle Sauerkraut	ALFKI	2	\$604
	Spegetil	ALFKI	1	\$18
	Vegie-spread	ALFKI	1	\$878

Summary for 'CompanyName' = Alfreds Futterkiste (11 detail records)

Sum	12	\$4,273
Percent	0.56%	0.34%

תרשים 6.13: הציג הדוח בדף 4

שימושים נוספים בתמונות

קובצי תמונה בדוחות של Access משמשים למטרות נוספות לאלו שהוצעו. לדוגמה, ניתן לשגר תמונה בדואר אלקטרוני בתווך מסמך מצורף (attachment) להודעת דואר אלקטרוני על ידי לחיצה ימנית על הדוח בחלון **מסך נתוני** (Database) ובחירה

שלח אל (Send To) ולאחר מכן בחירה בגורם נמען דואר (Mail Recipient) מתרפriet הקיצור. זכור לכלול בהודעת הדואר שכך קישור אל אתר האינטרנט ממנה ניתן להוריד את מציג התמונות, עבור נמענים שעדיין לא התקינו את המציג במחשב שלהם.

באפשרותך לנצל גם את השיטות DoCmd OutputTo ו-SendObject כדי למכן את המטרה וההתקפה של קבצי תמונה לאתר שרת Web בראשת אינטרא-נט או גוף אינטרא-נט או אלקטטרוני. הסעיף האחרון בפרק כולל דוגמה שמנצלת קוד כדי לשגר לבקרים רבים בדואר אלקטטרוני אל מען אחד או יותר. המשפט שפנוי מפרסם דוח מתוך מסד הנתונים Northwind באתר מקומי בראשת אינטרא-נט שלו :

```
DoCmd.OutputTo acOutputReport, "Alphabetical List of Products",
    "Snapshot Format", _
    "\\\cab2200\c\inetpub\cab_office_2000\mysnapshot.snp", True
```

הפיתחת דוח לדינמי

שלושה אירועי מקטע של דוח – Print ,Format ו-Retreat – מאפשרים לבנות עיצוב ותוכן דינמיים לתוך דוח. Open ,Close ,NoData ,Page ו-Format הם אירועי דוח וספירים שיכולים לסייע בבניית דוחות נבונים. אירועים אלה יכולים גם לסייע לנו להפעיל את תפקוד היישום לפניו פתיחת הדוח, במהלך ואחריה. באפשרותך להשתמש בצירופים של אירועי דוח כדי ליצור עיצוב דוחות ואפקטים מיוחדים.

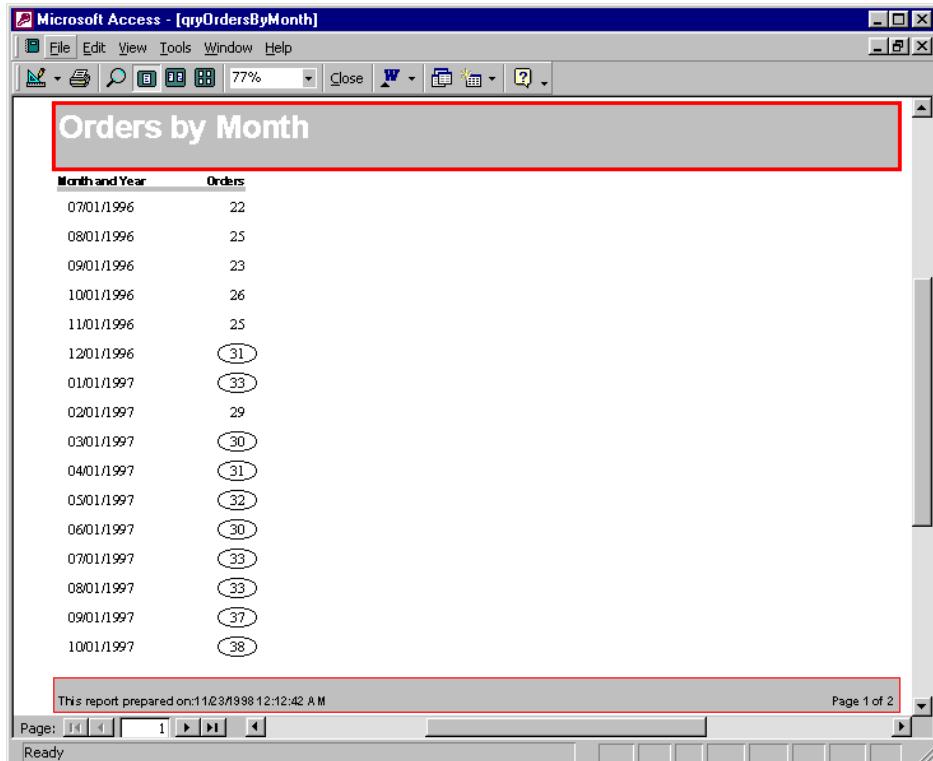
האירוע Open משמש להגדרת מאפייני דוחות ופקדיםthem באמצעות קוד. זה האירוע הראשון של דוח. אם היישום יכול לקיים יותר מdock פתוח אחד בו-זמנית, באפשרותך לנצל את האירועים Deactivate ו-Activate כדי לנטר את העברת המיקוד אל הדוח וממנו. השתמש באירוע Close כדי לבצע פעולה מיוחדת סמוך לסגירת דוח, כגון פתיחה טופס או הצגת תיבת הودעה.

האירוע NoData משמש לזיהוי דוח שמקור הרשומות שלו אינו מכיל נתונים כלשהם. אירוע זה מתרחש לאחר ש-Access מעצב דוח לצורך הדפסה. היישום יכול לנצל זאת כדי לבטל דוח העומד לפני הדפסה והואינו מכיל נתונים. באפשרותך גם לכתנת שגרות אירוע שמנחות את המשטח לספק נתונים על ידי הזנת רשומות או באמצעות מעבר למקור רשומות אחר עבור הדוח.

עיצוב והוספת תוכן

דוגמאות הקוד שלפלני מעצבות ומוסיפות תוכן לדוח באופן דינמי באמצעות האירועים Print ו-Format . תרשימים 6.14 מציג דוח שמנצל את האירוע Print בשלושה מקטעים כדי להוסיף מלבים אדומים סביב מקטעי כותרת הדוח העליונה וכותרת העמוד התחתונה.שים לב להבדלי עובי הגבולות בין המלבנים. המקטע פירוט (Detail) מציג אליפסה שבבב כל הסכומים הכלולים החודשיים הגודלים או שווים ל-30.

بعد שהארוע Print של כותרת הדוח العليונה קורה פעם אחת ויחידה בדף, והארוע Print של כותרת העמוד התחתונה קורה רק פעם אחת בדף, האירוע Print של המקטע **פירוט** מתרחש אחת לכל שורה בדף. משמעו הדבר שהדף שבתרשים 6.14 כולל 16 אירועי Print במקטע **פירוט**. כל אחד מהאירועים מאפשר לישום לבדוק את ערכי הפקד עברו הרשותה הנוכחית. פירושו של דבר שבאפשרות להציג בצורה סלקטיבית אליפסות סביב סכומי הזמן חודשיות מסוימות.



תרשים 6.14: האירוע Print בדף זה מציר מלבים סביב מקטעי הכותרת العليונה של הדוח והכותרת התחתונה של העמוד. אירוע נוסף מציר אליפסות בצורה סלקטיבית סביב סכומי הזמן חודשיות בדף

שלוש שגרות האירוע שלפניך הן הקוד שברקע הדוח של תרשימים 6.6. הישום יכול להחיל את השיטה Line (כמו עבור שגרות האירוע של כותרת הדוח العليונה וכותרת העמוד התחתונה) כדי לציין מלבן סביב מקטע הדוח. ארבעה משתנים מסוג Single מקבלים את הערכים עליון, שמאל, רוחב וגובה של המקטע. משתנה מסוג Long מקבל את מספר הצבע של המלבן (אדום בדוגמה). רגע לפני הפעלת השיטה Line כדי לציין את המלבן, השיגרה ReportHeader_Print מגדרה את רוחב הגבול כ-25 פיקסלים. שני זוגות קווארדינטות מציניות נקבעות עבור השיטה Line. הערך שב-*Color* מצין את

כבר הגבול האחרון של השיטה Line, מורה לשיטה לציר מלבן, או תיבת, תוך התייחסות לשתי הקואורדינטות בתור נקודות קצה הנמצאות בשני קדקודים מנוגדים.

```
Private Sub ReportHeader_Print(Cancel As Integer, PrintCount As Integer)
```

```
Dim sngTop As Single, sngLeft As Single
```

```
Dim sngWidth As Single, sngHeight As Single
```

```
Dim IngColor as Long
```

```
' Set top, left, width, & height.
```

```
    sngTop = Me.ScaleTop
```

```
    sngLeft = Me.ScaleLeft
```

```
    sngWidth = Me.ScaleWidth
```

```
    sngHeight = Me.ScaleHeight
```

```
' Set color.
```

```
    IngColor = RGB(255, 0, 0)
```

```
' Draw line as a box.
```

```
    Me.DrawWidth = 25
```

```
    Me.Line (sngTop, sngLeft)-(sngWidth, sngHeight), IngColor, B
```

```
End Sub
```

```
Private Sub PageFooterSection_Print(Cancel As Integer, PrintCount As Integer)
```

```
Dim sngTop As Single, sngLeft As Single
```

```
Dim sngWidth As Single, sngHeight As Single
```

```
Dim IngColor as Long
```

```
' Set top, left, width, & height.
```

```
    sngTop = Me.ScaleTop
```

```
    sngLeft = Me.ScaleLeft
```

```
    sngWidth = Me.ScaleWidth
```

```
    sngHeight = Me.ScaleHeight
```

```
' Set color.
```

```
    IngColor = RGB(255, 0, 0)
```

```
' Draw line as a box.
```

```
    Me.Line (sngTop, sngLeft)-(sngWidth, sngHeight), IngColor, B
```

```
End Sub
```

```
Private Sub Detail_Print(Cancel As Integer, PrintCount As Integer)
```

```
Dim sngHCtr As Single, sngVCtr As Single
```

```
Dim sngRadius As Single
```

```
' Position and size circle.
```

```
    sngHCtr = (Me.ScaleWidth / 2) - 3670
```

```
    sngVCtr = (Me.ScaleHeight / 2) - 20
```

```
    sngRadius = Me.ScaleHeight / 1.5
```

```

' Conditionally draw circle; last argument sets aspect ratio.
If Me.CountOfOrderID.Value >= 30 Then
    Me.Circle (sngHCtr, sngVCtr), sngRadius, , , , 0.5
End If
End Sub

```

הבדל היחיד בין השגורות ReportHeader_Print ו-PageFooterSection_Print הוא השורה שגדירה את רוחב גבול המלבן. המקטע **כותרת עליונה של דוח** (Report Header) מגדיר רוחב של 25 פיקסלים, אך המקטע **כותרת תחתונה של עמוד** (Page Footer) מציין מלבן ברוחב ברירת מחדל של פיקסל אחד. שתי השגורות מצירות מלבן בשכבה שבקדמת שכבת הדוח הרגיל. ניתן לדעת זאת משום שהגבול האדום של השיטה Line מופיע על גבי צללית הרקע של כותרת הדוח הראשית.

שגרת האירוע של המקטע **פירוט** (Detail) מציירת את האליפסה סביב סכומי ההזמנות באמצעות השיטה Circle לכל שורה במקטע **פирוט**. עליך לקבוע באופן ניסיוני את המרכזים האנכי והאופקי וגם את רדיוס העיגול. ארגומנט יחס רוחב-גובה (aspect) של השיטה Circle משתמש להמרת עיגול לאליפסה ארוכה או צרה. בדומה לשיטה Line, השיטה Circle מציבה את הפלט שלה בשכבה שבקדמת שכבת הדוח הרגיל. הטענה השיטה Circle במשפט If...Then מאפשרת לשיגורה לציר בצורה מותנית את האליפסה סביב חלק מערכי השדה Orders, אך לא כולם.

סיכום ערכי דף

אם הדוח כולל דרישת להציג סכום הערכיים בדף הדוח, عليك לתוכנת את סכומי הדף באמצעות שגורות אירוע, מכיוון ש-Access אינו כולל אפשרות מוכללות לביצוע משימה זו. הדוגמה של הספר למשימה זו מנצלת תיבת טקסט שהמאפיין **סכום** **מצטברים** (Running Sum) של הוגדר בתווך **על הכל** (Over All). הדבר גורם לתיבת הטקסט לסטם את השדה Control Source לאורך כל הדוח. תרשימים 6.15 מציג את הדף הראשון והשני בדף בעל עמודה נוספת לחישוב סכום מצטבר. העמודה הימנית-קייזונית גלויה למטרות למידה, אך בפועל מגדירים את המאפיין **גלווי** (Visible) של פקד הסכום המצטבר בתווך לא.

באפשרותך לחשב סכומי דף בעורת לא יותר מאשר שגורות אירוע (שמוצגות בעמוד הבא). שגרת האירוע PageFooterSection_Format תופסת שתי שורות קוד בלבד. ראשית, היא מעתקה את ערך הפקד בעל הסכום המצטבר (pagesum) אל IngCurrentRSum. לאחר מכן היא מגדירה תיבת טקסט נוספת במקטע כותרת הדף txtpagesizeum (txtpagesizeum) בתווך ההפרש בין IngCurrentRSum ו-IngLastRSum. הערך התחלתי של IngCurrentRSum הוא 0. בתום העיצוב של כל דף, אירוע הדוח התחלתי של IngCurrentRSum מפעיל שיגורה שמעתקה את הערך הנוכחי של IngLastRSum אל IngCurrentRSum, כך שההפרש IngLastRSum ו-IngCurrentRSum בשגרת האירוע Report_Page הוא סכום הדף של הדף הנוכחי.

Orders by Month

Month and Year	Orders
07/01/1996	22
08/01/1996	25
09/01/1996	23
10/01/1996	26
11/01/1996	25
12/01/1996	31
01/01/1997	33
02/01/1997	29
03/01/1997	30
04/01/1997	31
05/01/1997	32
06/01/1997	30
07/01/1997	33
08/01/1997	33
09/01/1997	37
10/01/1997	38

Orders on this page are: 428
This report prepared on: 11/23/1998 1:07:48 AM

Page: [navigation buttons] Ready NUM

Orders by Month

Month and Year	Orders
11/01/1997	34
12/01/1997	48
01/01/1998	55
02/01/1998	54
03/01/1998	73
04/01/1998	74
05/01/1998	14

Orders on this page are: 352
This report prepared on: 11/23/1998 1:09:02 AM

Page: [navigation buttons] Ready

תרשים 6.15: הדוח מנצל שתי שגרות אירוע ותיבת טקסט במקטע **פירוט** שהמאפיין סכומים מצטברים שלה מוגדר בתור **על הכל**, כדי לחשב את סכום הרזמנות בדף

```

Public IngLastRSum As Long, IngCurrentRSum As Long
Public IngPageRSum As Long

Private Sub PageFooterSection_Format (Cancel As Integer, FormatCount As Integer)
    IngCurrentRSum = Me.pagesum
    Me.txtpagesum = IngCurrentRSum - IngLastRSum
End Sub

Private Sub Report_Page()
    IngLastRSum = IngCurrentRSum
End Sub

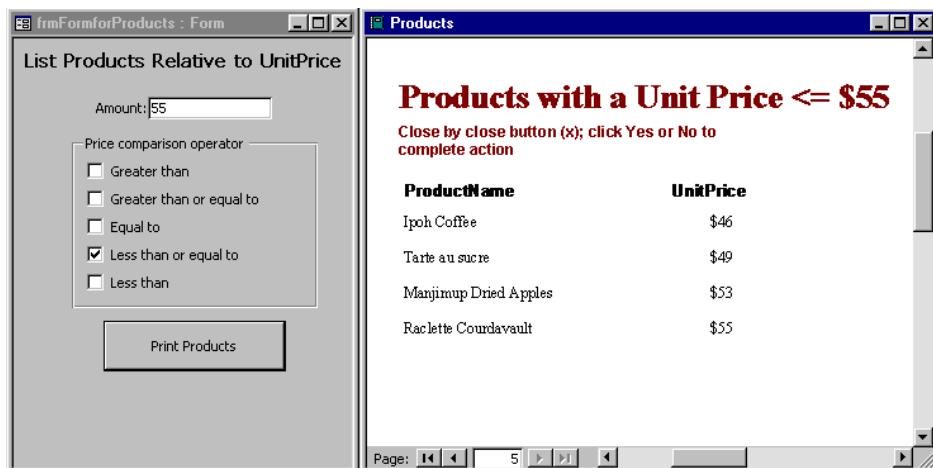
```

שים לב ששגרת האירוע PageFooterSection_Format שבדוגמה מחשבת ומציגה סכומי דפים על ידי כתיבת ערך לתיבת טקסט במקטע כוורת הדף התחתיונה. שגרת האירוע Print אינה מאפשרת פעולה מסווג זה, מכיוון שהאירוע Print מופעל רק לאחר שהדוח עבר את תהליך העיצוב. האירוע Format מופעל במהלך עיצוב הדוח באמצעות היישום.

עדכן דוח באופן דינמי

ניתן להשתמש ב-VBA כדי להוסיף מאפיין מקור רשומות חדש לדוח. בעת שהיישום משנה את תוכן הדוח, אפשר לעדכן את כוורת הדוח הראשית כך שזו תשקף את התוכן החדש. אם תציג את תוצאות השאילתת הפרמטרית בדוח, תוכל גם לעצב את התוצאות באמצעות אפשרות עיצוב רבות.

תרשים 6.16 מציג טופס ודוח. המשמש יכול לטפל בטופס כדי לשנות את תוכן הדוח. הטופס מכיל תיבת טקסט, קבוצת אפשרויות ובה חמישה תיבות סימון ולחוץ פוקודה. לאחר הקלדת מספר בתיבת הטקסט ובחירה סוג השוואה, המשמש יכול ללחוץ על לחוץ הפוקודה כדי לפתח את הדוח שמי민 בתצוגת עיצוב. תצוגה זו נחוצה כדי להוסיף מאפיין של מקור רשומות חדש וכן להגיד את המאפיין Caption של התוויות שמציג את כוורת הדוח הראשית. לאחר עדכון שני מיינר הדוח באמצעות קוד, פותח היישום את התוצאות במצב **הציג לפני הדפסה** כדי להציג את תוצאות המקור הרשומות החדש. ההוראות הבאות לאחר הכותרת מסבירות כיצד לסגור את הטופס.



תרשים 6.16: הטופס שמשמאלי מאפשר למשתמשים להקצות מקור רשומות וכוורתת מתאימה לדוח שמיין.

שגרת האירוע cmdPrintThem_Click של לחוץ הפוקודה (במהשך הסעיף) מבצעת שלוש מושימות. ראשית, היא מרכיבה מחרוזת SQL בהתאם לבחירות שנעשו בטופס. השגירה מוסיפה פסוקיות באופן סדרתי לחלקו הראשון של משפט ה-SQL שמציג את רשימת המוצרים ומהירותם מתוך הטבלה Products. תחילת היא מוסיפה פסוקית WHERE למשפט הבסיסי בהתאם לתיבות הסימון שנבחרו ולכמויות שהוקלדה בתיבת הטקסט. לאחר הוספת הפסוקית WHERE, השגירה מוסיפה פסוקית ORDER BY שמשמעותה את קבוצת החזרה של מחרוזת ה-SQL, לפי מחיר היחידה. אם המשתמש בוחר אופרטור גדול מ (>) או אופרטור גדול או שווה ל (=>), השגירה תקבע סדר מילוי יורד. אחרת – קבוצת החזרה תמיין לפי מחיר היחידה בהתאם לברירת המחדל של סדר המילוי.

הרכיב השני של השגירה cmdPrintThem_Click מעדכן באמצעות קוד את מקור הרשומות ואת כיתוב התווית בדוח. לאחר פתיחת הדוח בתצוגת עיצוב, הרכיב מבצע משפט With...End With המבוסס על הדוח. כדי למנוע ערבותה של מסכים, הרכיב משפט מפעיל את השיטה Echo עם ארגומנט שערכו False. הדבר מעכ卜 את ריענון המסק עד שהמשפט השני מפעיל את השיטה Echo עם ארגומנט שערכו True. בתוך קטע הקוד We...End With, השגירה מדירה את המאפיין RecordSource (מקור רישום) של הדוח בתור מחוזות SQL שנוצרה בחלק הראשון של השגירה. לאחר מכן היא משנה את כיתוב התווית שמציגת את כותרת הדוח הראשית. משימה זו מתאפשרת הדות לביטויי מחוזות שמציג את הערך מקובצת האפשרויות ואת הנקודות מתיבת טקסט.

הרכיב השלישי מבצע שני תפקידים: פותח את הדוח בתצוגת **הציג לפני הדפסה**, כך שהמשתמשים יכולים לראותו. לאחר מכן, הפקודה האחורייה משוחררת את הפונקציה Echo אשר מציגה את הדוח. כיבוי הפונקציה Echo והפעלה מחדש מביאה למעבר מסכים חד וברור.

```
Private Sub cmdPrintThem_Click()
Dim strSQL As String, strOperator As String
Dim strWhere As String
' Set up SQL statement for report record source.
strSQL = "Select ProductName, UnitPrice from Products"
strOperator = Choose(optRule, ">", ">=", "=", "<=", "<")
strWhere = "Where UnitPrice" & strOperator & txtAmount
strSQL = strSQL & " " & strWhere & " Order By UnitPrice"
If optRule <= 2 Then
    strSQL = strSQL & " Desc"
End If
' The commented debug.print statement is convenient for debugging your SQL
statement; remove the comment when you change the SQL statement construction.
'   Debug.Print strSQL
' Open report in Design view to set the report's record source
and its label's caption.
DoCmd.Echo False
DoCmd.OpenReport "rptProductsfromForm", acViewDesign
With Reports("rptProductsfromForm")
    .Visible = False
    .RecordSource = strSQL
    .Controls("lblTitle").Caption = _
        "Products with a Unit Price " & strOperator & " $" & txtAmount
End With
' Now show the form to the user.
DoCmd.OpenReport "rptProductsfromForm", acViewPreview
DoCmd.Echo True
End Sub
```

טיפול בדוחות ובפקדי דוח באמצעות קוד

Access כולל שתי רמות הניתן נתונים בדוחות, בטפסים ובאובייקטים חשובים נוספים של מסד נתונים. פרק 5 למדנו על האוסף AllForms. Access כולל אוסףים נוספים: AllModules, AllViews, AllMacros, AllTables, AllReports, AllDataDiagrams ו-AllDataAccessPages. AllStoredProcedures מהאוסףים הללו הוא אובייקט מסווג AccessObject (סוג אובייקט חדש - Access). באפשרותך להפנות לחבר `AllReports` באחת הצורות הבאות:

`AllReports (0)`

`AllReports ("name")`

`AllReports![name]`

מספר דוחות

הקוד שתיצור יוכל למספר אובייקטים מסווג AccessObject באוסף All כלשהו, כדי לקבוע אם האובייקטים נמצאים בחיבורו של מסד נתונים. אין זה משנה אם האובייקט פתוח או סגור. באפשרותך גם לקבוע אם האובייקט טוען. כאשר אובייקט מסווג AccessObject טוען או פתוח, הישום יכול לעבוד עם מרחב אוסף מקביל. אוסףים אלה הינם כולם רכיבים פתוחים, כגון דוחות וטפסים במסד נתונים של Access. חברי האוסף Reports הם דוחות בודדים פוחדים ביחסם. אובייקטי Report פוחדים אלה חושפים את כל המאפיינים הזמינים באמצעות VBA במקומם הקבועה המוגבלת יותר Reports. באפשרותך לנצל את המאפיין Name של AllReports ושל AllReports באוסף CurrentProject. כדי לעבור בין שני האוספים המקבילים. השימוש במאפיין IsLoaded באוסף AllReports מאפשר לוודא אם יש לפתיחת דוח, טרם הניסיון לטפל במאפיינוReports ובשיטותיו.

השיגרה `ListAllReports` שלפניך ממספרת את חברי האוסף All תוך הצגת שם ומצב הטעינה שלהם. חברי האוסף AllReports שייכים ל-`Application` או `CodeProject`, שהם חברי באובייקט Application. עליך להפנות לאחד מהם כדי לחסוך את חברי האוסף AllReports. לכן, השיגרה `ListAllReports` מתחילה בהגדרת הפניה אל החבר CurrentProject של האובייקט Application. תזדקק להפניה זו כדי לגשת לחבר האוסף AllReports. שים לב שהלולאה `For...Each` עוברת על כל אובייקט AllReports באוסף obj) באוסף obj1 AccessObject מתחיל בהפניה אל Application.CurrentProject

```
Sub ListAllReports()
Dim obj1 As AccessObject, app1 As Object
' Create a reference to the current project instance.
    Set app1 = Application.CurrentProject
```

```

' List each report in the application and
' describe as loaded or not.
For Each obj1 In app1.AllReports
    If obj1.IsLoaded = True Then
        Debug.Print obj1.Name & " is loaded."
    Else
        Debug.Print obj1.Name & " is not loaded."
    End If
Next obj1
End Sub

```

האוספים AllForms ו-AllReports דומים זה לזה בתפקידם. אין מוגבל לבדיקת חברים מסווג AccessObject בפרויקט הפעיל. השגרות ListAllFormsElsewhere ו- ListAllReportsElsewhere שלפניהם מראות כיצד לתקן את שני האוספים כשהם מצביעים על פרויקט אחר. שים לב לדמיון הקוד בין האוספים ובין לדמיון הקוד בין הפרויקט הנוכחי לפרויקט האחר.

השגרה ListAllFormsElsewhere שלפניך מדפיסה את מספר השמות הכלול של כל חברי האוסף AllForms שבקובץ Chapter5.mdb. הקובץ נמצא בתיקitorio המצורף בספר. השירה מבוססת על ההנחה שהעתקת את הקובץ מהתקליטור אל התיקה היא מספרת את החברים (AccessObject) באובייקטים (Programming Access) בקובץ מסד נתונים אחר.

```

Sub ListAllFormsElsewhere()
Dim appAccess1 As Access.Application
Dim obj1 As AccessObject
' Create a reference to another database file.
Set appAccess1 = New Access.Application
appAccess1.OpenCurrentDatabase "c:\Programming Access\" & _
    "Chap05\Chapter5.mdb"

' Print the total number of forms in the database.
Debug.Print appAccess1.CurrentProject.AllForms.Count
For Each obj1 In appAccess1.CurrentProject.AllForms
    Debug.Print obj1.Name
Next obj1
End Sub

```

השירה ListAllReportsElsewhere שלפניך מעוצבת בצורה זהה לקודמתה, למروת שהיא מטפלת באוסף AllReports ולא באוסף AllForms ומנצלת את מסד הנתונים Northwind.mdb במקומם את Chapter5.mdb. עיצוב השגרות כמעט זהה, למעט השימוש במשתני מחרוזת להגדרת שם מסד הנתונים. שינוי זה הוכנס משיקולי נוחות וכליות גרידא – אין כלל ב-Access או ב-VBA שמכטיב שימוש במחרוזות.

```

Sub ListAllReportsElsewhere()
Dim appAccess1 As Access.Application
Dim obj1 As AccessObject
Dim strPath As String, strFile As String, strDBName As String
' Create a reference to another database file.
    Set appAccess1 = New Access.Application
    strPath = "c:\Program Files\Microsoft Office\Office\Samples\" 
    strFile = "Northwind.mdb"
    strDBName = strPath & strFile
    appAccess1.OpenCurrentDatabase strDBName

' Print the total number of reports in the database.
    Debug.Print appAccess1.CurrentProject.AllReports.Count
    For Each obj1 In appAccess1.CurrentProject.AllReports
        Debug.Print obj1.Name
    Next obj1
End Sub

```

שינוי מאפייני פקץ דוח

קוד היישום שכותבת יכול לנצל את האוסף AllReports בתור נתיב לדוחות פתוחים ספציפיים ולפקדים שהם מכילים. לאחר שתעביר בתיב זה, היישום יוכל לקרוא ולשנות את המאפיינים של דוחות פתוחים. השירה ControlsInReports ניגשת דרך חברי האוסף AllReports אל מאפייני תיבות הטקסט והתוויות לדוחות פתוחים ספציפיים.

```

Sub ControlsInReports()
Dim obj1 As AccessObject, ctl1 As Control
For Each obj1 In CurrentProject.AllReports
    If obj1.IsLoaded = True Then
        For Each ctl1 In Reports(obj1.Name)
            If ctl1.ControlType = 100 Then
                Debug.Print ctl1.Name, ctl1.Caption
            ElseIf ctl1.ControlType = 109 Then
                Debug.Print ctl1.Name, ctl1.Value
            Else
                Debug.Print ctl1.Name & " is not a label or a text box."
            End If
        Next ctl1
    End If
Next obj1
End Sub

```

השיגרה ControlsInReports פותחת בלולאת For...Each שעובדת על חברי האוסף AllReports. אם היה מזהה חבר פתוח (המאפיין IsLoaded שלו מכיל ערך True), הקוד נכנס ללולאת For...Each מקוננת כדי למספר את פקדי הדוח. באפשרותך לנצל את המאפיין ControlType כדי לקבוע את סוג הפקד. חשוב לציין את סוג הפקד, מכיוון שהסוג קובע את המאפיינים שחוש הפקד. לדוגמה, פקד תווית מציג את המאפיין Caption שלו, אך פקד תיבת טקסט מנצל את המאפיין Value כדי לתאר מה הוא מציג. באפשרותך לנצל את סורק האובייקטים (Object Browser) ב-VBE כדי להציג את הקודים המספריים של סוגי פקדים אחרים שברצונך לערוך או לבדוק.

שלוח תמונות

הדוגמה שלפניך ממספרת דוחות כדי לקבוע אם הם מסומנים לשיגור בדיוור בתור קבצי תמונה. הדוגמה מתבססת על שתי שגרות. השיגרה הראשונה, SendSnapShots מספרת את חברי האוסף AllReports. לאחר שהקוד בודק אם המאפיין Tag (תג) של הדוח הוא "mail", על הדוח להיות פתוחה. המאפיין Tag אינו זמין באמצעות האוסף AllReports, אלא באמצעות האוסף Reports בלבד. השיגרה SendSnapShots בודקת את המצביע IsLoaded של כל אחד מחברי האוסף AllReports. אם מכיל ערך False. אם מכיל ערך True, השיגרה תפתח את הדוח לפני שתקרה לשיגרה השנייה. שגרת הדוגמה אינה קוראת לשיטה Echo עם הפרמטר False, ולכן המשמש יכול לקבל בקבוק מושב ברגע שהשיגרה השנייה נכנסת לפעולה. הדבר מתאים במיוחד לדוחות ארוכים בהם יצירה ושיגור של קובץ תמונה עלולים להימשך זמן רב.

```
Sub SendSnapShots()
Dim obj1 As AccessObject, app1 As Object
' Create a reference to the current project instance.
    Set app1 = Application.CurrentProject

' Enumerate each member in AllReports to verify if loaded.
' If not loaded, open before calling CheckMailItTag.
    For Each obj1 In app1.AllReports
        If obj1.IsLoaded = True Then
            CheckMailItTag obj1.Name
        Else
            DoCmd.OpenReport obj1.Name, acViewPreview
            CheckMailItTag obj1.Name
            DoCmd.Close acReport, obj1.Name, acSaveNo
        End If
    Next obj1
End Sub
```

```

Sub CheckMailItTag(obj1name)
Dim rep1 As Report
' Set reference to Reports member corresponding to AllReports member.
    Set rep1 = Reports(obj1name)
' If Tag property says "mail it" create a snapshot file and mail it.
    If rep1.Tag = "mail it" Then
        DoCmd.SendObject acOutputReport, obj1name, acFormatSNP, _
            "virginia@cabinc.net", , , "Snapshot Report", "Here is the report.", False
    End If
End Sub

```

השיגרה CheckMailItTag קולטת את שם הדוח שהועבר אליו באמצעות Dim rep1 As Report. היא מנצלת שם זה כדי ליצור הפניה אל חבר האוסף Reports SendSnapShots זהה לשם שהועבר אליה. לאחר מכן היא בודקת את המאפיין Tag של הדוח כדי לקבוע אם הערך שהוא מכיל הוא "mail it". אם כן, השיגרה מפעילה את השיטה SendObject של DoCmd כדי ליצור קובץ תמונה ושלוח אותו כתובות דואר אלקטרוני המוחדרות של virginia@cabinc.net, במקרה שלנו). אפשרו בסדרת כתובות כפי שדורש היישום. חשוב שהארגון כתובות בכתובת כלשי או בסדרה יכול לבצע את קביע החזרות של הבא לאחר גוף הודעה יקבל ערך False. ערך ברירת המחדל True יגרום לשיגרה לעצור אם הודעה פתואה ולהמתין שהמשתמש יערוך את הודעה. הגדרת ערך הארגומנט בתוর False מאפשר לשיגרה לעבור בלולאה על כל הדוחות, ללא צורך כלשהו בהתערבות משתמש.

מודולי מחלקה, טופס ודווח

כדי לכתוב קוד ב- Microsoft Access בצורה ייעילה, עליך לנצל את משאבי התכונות שלך כך יהיה קל לעבד איתם ולעשווים בהם שימוש חוזר. ערך הקוד עולה ביחס ישיר למה שנייתן להפק ממנו.

מודולי מחלקה מאחסנים קוד באופן שמאפשר לעשות בו שימוש חוזר. מודול המחלקה פועל כמו מכולה שחושפת את הקוד והמשתנים הנבחרים שבתוכה בצורה מוכרת למפתחים בסביבת Visual Basic for Applications (VBA). עקרונית, מפעילים שגורות מחלקה ומצביעים ערכיהם במשתנים באמצעות תחביר זהה זהה המשמש לביצוע פעולות על מאפיינים ושיטות של אובייקטים מוכללים של Access. כדי לנצל את הקוד במודול מחלקה, אין צורך לדעת דבר על אופן פועלתו. כמו כן, לאחר שמודולי מחלקה חשובים מאפיינים, שיטות ואיורעים כפי שאובייקטים אחרים עושים זאת, גם מפתחים מתחילה בסביבת VBA יכולים להשתמש בהם.

פרק זה נציג תחילת מודולי מחלקה עצמאיים וגם מודולי מחלקות טופס ודווח. לאחר מכן נדגים דרכי פשוטות לבניית מחלקות ביישומי Access ולפיתוח שיטות ומאפיינים מותאמים אישי. בהמשך נציג מקרה לדוגמה המשמש בשלושה טפסים, Property Let Property Get מהסוגים ו- ActiveX Data Objects (ADO – ActiveX Data Objects) ובטכניקות המבוססות על אובייקטי נתונים (Objects) הטעיף שלآخر מקרה הדוגמה מציג את התchapיר כדי להתחיל ליצור יישום. הסעיף שלآخر מקרה הדוגמה מציג את התchapיר שבאמצעותו מתכוונים אירועים בתוך המחלקות המותאמות אישית ובו נפגש במלת המפתח .WithEvents

בשלב הבא נתמקד במקולות של מודולי המחלקה, הטופס והדווח בעט שנבחן את אוסף All החדשניים ב- Access 2000. פגשו באוספים AllForms AllReports AllModules All； בפרק זה נכיר את האוסף AllModules (לאמייתו של דבר, קיימים בסך הכל עשרה אוסף All). הפרק מסיים בהסביר כיצד לשלב את האוסף AllModules עם האוסף Modules כדי לנצל את הקוד ביישום.

סוגי מודולים

קיימים שלושה סוגים מודולים:

« **מודולים סטנדרטיים** (Standard modules). מודולים אלה מאחסנים שגרות משנה ושורות פונקציה שברצונך להפוך לזריניות בכל חלקי קובץ מסד הנתונים. מודולים סטנדרטיים יכולים להכיל גם משתנים שהוגדרו באמצעות הברה Public (יצורי) שברצונך להפוך לזרינים עבור שגרות במודולים אחרים.

« **מודולי מחלקה עצמאיים** (Standalone class modules). מודולים אלה מאפשרים לייצור אובייקטים מותאים אישית. תוכל להגדיר מאפיינים, שיטות וARIOUIS עבור אובייקטים אלה, ותוכל גם לנצל את מילת המפתח **New** כדי לייצור מופעים של אובייקטי הטופס.

« **מודולי מחלקה של טפסים ודוחות** (Class modules for forms and reports) (נקראים גם מודולי טופס ודוח). כברירת מחדל, טפסים ודוחות מבוססים על מודולים (הגדת ברירת המחדל של המאפיין HasModule שלו היא True). תוכל להשתמש במלת המפתח **Me** בהתייחסות למודול שנמצא מאחורי טופס או דוח.

מודולי מחלקה

מודולי מחלקה עצמאים שונים ממודולי מחלקה טופס ודוח בכמה מובנים.

ראשית, בניגוד למודולי מחלקה של טופס ודוח, מודולי מחלקה עצמאים אינם כוללים ממשק משתמש מוככל. תוכנה זו עשוה אותן מותאים יותר למשימות שאין מחיבות ממשק, כגון חישובים, בדיקת מידע או ביצוע שינוי בסיסי בתונין. מודולי טופס או דוח צריכים לבצע משימות עתרות חישוב, יכולים לקרוא למודולי מחלקה סטנדרטיים.

שנית, מודולי מחלקה עצמאים כוללים את האירועים Initialize ו-Terminate. שמאפשרים פעולות לצרכים התחבצע בעת פתיחה וסגירה של מופע של מחלקה. מודולי דוח וטופס אינם כוללים אירועים כאלה, אך אפשרותך לבצע דברים דומים באמצעות האירועים Load ו-Close.

שלישית, عليك להשתמש במלת המפתח New כדי לייצור מופעים של מודולים עצמאים. מודולי מחלקה דוח וטופסאפשרים ליצור מופעים גם באמצעות השיטות OpenForm ו-DoCmd ו-OpenReport. וגם על ידי הפניה אל המאפיינים או השיטות של מחלקת הדוח או מחלקת הטופס. לדוגמה, Form_MyForm.SetFocus פותחת את הטופס .MyForm.

באפשרותך ליצור מודול מחלקה עצמאי מתוך הטריט **Insert** (הוספה) של VBE (טריט זה כולל גם פקודות לבניית מודול או שירה סטנדרטיים). לאחר בניית מעתפת מודול מחלקה, ניתן לאכפל אותה בשגרות ובחזרות ועל ידי כך לציד אותה בשגרות ובמאפיינים מותאים אישית.

פונקציות מאפיין ושיטות מותאמות אישית

פונקציות מאפיין מיוחדות מאפשרות לפתח בצורה קלה ו פשוטה שילוב כלשהו של מאפיינים מסווג קריאה- בלבד, כתיבה- בלבד וקריאת/ כתיבת עבורה מחלקות. אם היישום שאתה כותב מרשה זאת, תוכל להגדיר מאפיינים באופן פשוט על ידי הגדרת משתנים ציבוריים. מאפיין שמודדר על ידי מודול מחלקה באמצעות משתנה ציבורי, יהיה מסווג קריאה/ כתיבה. היכולת להציג על מאפיינים מותאמים אישית מאפשרת להרחיב את מגוון הרכיבים התפקודיים של Access לטפסים ודוחות. בנוסף, פונקציות מאפיין אלו מאפשרות ליצור מחלקות עצמאיות רבות עצמה.

הយישום יכול גם לבנות שיטות מותאמות אישית לתוך מחלקות. ניתן לנצל שגרות משנה או שגרות פונקציה כדי לבצע זאת. חסיפה סלקטיבית של משתנים וশגרות באמצעות מילת המפתח `Public` מאפשרת להגדיר בצורה מוקדמת את השיטות והמאפיינים שייחספו. הדבר מאפשר ליישומים להגדיר ממשקים ביןם לבין אובייקטי מחלקה שפועלים בדרכים מוגדרות ביותר.

יצירת מופעים של מחלקות

השיטות והשגרות הציבוריות תומכות בגישה באמצעות קוד של שגרות חיצונית למחלקה. עליך ליצור תחילה מופע של המחלקה בשגרה מארחת במודול אחר באמצעות מילת המפתח `New` (밀ת מפתח `New` מפתחת זו משתמש גם ליצירת אובייקטים מותוך מחלקות אחרות, כגון האובייקטים `Recordset` ו-`Connection` של `ADO`. לאמתו של דבר, היישומים שתיצור יכולים ליצור עותקים רבים של מחלקה מותאמת אישית בעט ובוונה אחת – בדיקן כמו המחלקות של `ADO`). לאחר יצירת מופע של מחלקה, קוד השינירה המארחת מטפל במוועג המחלקה, לא במחלקה עצמה. באפשרותך לשנות מאפיין של מופע אחד של טופס, אך כשתיצור מופע נוסף של הטופס, המאפיין שלו קיבל את ערך ברירת המחדל.

מחלקות ואיורים מותאמים אישית

למרות ש-`VBA` מאפשר ליצור מחלקות מותאמות אישית בעלות שיטות ומאפיינים נפרדים, לא ניתן לבנות איורים מותאמים אישית בתוך מחלקות אלו. ניתן, לעומת זאת, לעצב מחלקה שמתחברת למחלקה מוכללת או ספריית סוגים שמחברת אליה. לדוגמה, באפשרותך לבנות מודול מחלקה שמבצעיל קוד `VBA` בתגובה לאיורים `ItemAdded` ו-`ItemRemoved` של האוסף `References` (הפיוiot). אוסף זה עוקב אחר קישוריהם אל ספריות סוג חיצונית ופקדי `ActiveX`. לאחר הפעלת הפניה אל ספריה, כגון `Microsoft ActiveX Data Objects 2.1`, באפשרותך לבנות איורים מותאמים אישית שmbוססים על איורים של `ADO` עבור האובייקטים `Recordset` ו-`Connection`. איורים אלה יכולים להעניק גישה אסינכרונית לנוטונים, גישה שמאפשרת ליישום להגיב לפעולות המשתמשים, גם כשהיא מוכנה להגיב לחיבור מושלים או לזמןות של קבוצת ערכות רשומות שהובאה.

משתמשים במילת המפתח `WithEvents` בהצהרה `Public` כדי להציב על הפניה לאובייקט שמנטר אירועים ומודיע עליהם מותך פקד `ActiveX`. מילת מפתח זו חוקית במודול בלבד. באפשרותן להגדיר משתנים רבים במודול באמצעות מילת המפתח `WithEvents`, אך לא תוכל ליצור מערכיים בעורתה. בנוסף, הצהרה אינה יכולה לכלול בו-זמנית את מילות המפתח `WithEvents` ו-`New`.

שיטות ומאפיינים מותאמים אישית

בעת השימוש במודול בלבד, לא מנו הנמנע שתעבדו עם שני מודולים נפרדים. מודול המחלקה חושף מאפיינים ושיטות וגם מפיץ אירועים. מודול שני מפנה אל מודול המחלקה; הוא מציב וקורא ערכיהם במאפיינים וגם מפעיל שיטות. מודול זה יכול ליזום פעולות שגורמות לאירועים, ואירועים אלה, יכולים להפעיל שגורות אירוע המשויכות להן במודול המחלקה.

חישפת מאפיינים באמצעות משתנה ציבורי

הדוגמאות שלפניך מציגות שני תדפסי קוד. האחד לקוח מהמודול `MyTestClass`. זהו מודול מחלקה שמתחליל בהצהרות על זוג משתנים – אחד עבור הדוגמה הנוכחית והאחר עבור הדוגמה הבאה. השגירה שנקראת `EP` מחשבת את מחיר `סה"כ` לפריט מותך שלושה ארגומנטים שימושיים אליו: `price`, `units` ו-`discount`. השגירה שומרת את תוכנות הביטוי שללה במשתנה שנקרא `ExtendedPrice`. ההצעה באזור הכללי של המודול מגדרה את `ExtendedPrice` בתור משתנה ציבורי. הגדרתו ציבורי מאפשרת לשגירה מארחת שנמצאת במודול אחר העובדת עם המופיע של האובייקט `MyTestClass`, לקרוא את ערך המשתנה.

```
FROM MyTestClass module (a class module)
Public ExtendedPrice As Currency
Private MyComputedPrice As Currency
Public Sub EP(units As Long, price As Currency, discount As Single)
' Compute with result in public variable.
    ExtendedPrice = units * price * (1 - discount)
End Sub

FROM Module1 (a standard module)
Sub ComputeExtendedPrice()
' Create new instance of class module.
Dim MyPriceComputer As New MyTestClass
' Invoke EP method for class, and print Extended Price property.
    MyPriceComputer.EP 5, 5, 0.02
    Debug.Print MyPriceComputer.ExtendedPrice
End Sub
```

השיגרה המארחת, Compute ExtendedPrice , נמצאת במודול סטנדרטי שנקרא Module1 . השיגרה יוצרת מופע של אובייקט, המבוסס על המחלקה שהוגדרה באמצעות האובייקט MyTestClass . בשלב הבא, היא מפעילה את השיטה EP של האובייקט. לבסוף, היא מדפיסה את המאפיין ExtendedPrice של האובייקט.

השיגרה שבדוגמה פשוטה ביותר, אך היא מדגימה מספר נקודות חשובות הקשורות לשימוש במודולי מחלקה:

↳ מודולי מחלקה הם בחירה הולמת לחישוב ביטויים קריטיים לעסקים. מודולי מחלקה משמשים בדרך כלל לאריזות פעולות מתוחכבות יותר מאשר הדורשות לחישוב סה"כ לפריט.

↳ השיגרה השנייה שנמצאת במודול הסטנדרטי, מתחילה בהפניה אל מודול המחלקה, MyTestClass . מילת המפתח New יוצרת מופע של אובייקט שבבסיסו על המחלקה. בדוגמה שלנו, המשתנה שנקרא MyPriceComputer מפנה אל המחלקה.

↳ ניתן לנצל את הפניות האובייקט אל מופע המחלקה כדי להפעיל שיטות ולהגדיר אותו ערכיו מופיעין. מפנים אל השיטה EP של המחלקה באמצעות סימון הנΚודה המקורי. מציגים את הארגומנטים אחרי שם השיטה ומפנים אל מאפיינים בעזרת כללי הסימון הבסיסיים.

↳ יצירת מאפיין של מחלקה יכולה להיות פעולה פשוטה בדיקות כמו הצהרה של משתנה ציבורי במודול המחלקה.

חישפת מאפיינים באמצעות פונקציית מאפיין

הראשון בתדריסי הקוד שלפני מציג גישה שונה למשימה. הגישה מבוססת על מאפיין שהוגדר באמצעות הפונקציה Property Get . השיטה ep2 זהה כמעט בכל פרטיה לשיטה EP מהדוגמה הקודמת. ההבדל היחיד בין השתיים הוא ש- ep2 מפרקידה את תוצאת הביטוי שלה במשתנה ComputedPrice (עיין בהצהרת המשתנה הפרטיה בדוגמה הקודמת). זה כשלעצמו אומר שמוועי המחלקה אינם חושפים את תוצאת הביטוי. הפונקציה Property Get היא זו שחושפת משתנה פרטוי. לאחר שלא הוגדרה פונקציית מאפיין כלשיי עבור ComputedPrice , המאפיין הוא לקריאה בלבד. לו הייתה פונקציה מסוג Property Let בשם זהה, המאפיין היה קרייה/כתיבה. השימוש במאפיינים מסוג קרייה בלבד יכול לסייע לאבטוח את ערכי המאפיינים בישום, או לפחות לספק דרכים להגדיר אותם.

```
FROM MyTestClass module (a class module)
Public Sub ep2(units As Long, price As Currency, discount As Single)
' Compute with result in private variable; expose
' result through Property Get function.
    MyComputedPrice = units * price * (1 - discount)
End Sub
```

```

Property Get ComputedPrice()
' This is how to return a read-only property.
    ComputedPrice = MyComputedPrice
End Property

FROM Module1 (a standard module)
Sub GetComputedPrice()
Dim MyPriceComputer As New MyTestClass
' Using a value defined by a property looks the same
' as one defined with a public variable.
    MyPriceComputer.ep2 5, 5, 0.02
    Debug.Print MyPriceComputer.ComputedPrice
End Sub

```

משתנים ציבוריים לעומת פונקציות מאפיין

תחביר הפעלת השיטה והדפסת ערך המאפיין זהים בשתי הדוגמאות, למروות השוני בחשיפת המאפיין. הדבר מASH את הקביעה שמאפיינים מתנהגים בצורה זהה בין אם מגדריים אותם באמצעות הצהרה ציבורית, או בעזרת פונקציית מאפיין אחת או יותר. ייתכן שפשוט יותר לישם מאפיינים במודולי מחלקה בעזרת משתנים ציבוריים, אך פונקציות מאפיין מצטיינות בGESותן בחשיפת המאפיינים. השתמש בפונקציה Property Get בלבד לטיפול במשתנה מסווג קריאה בלבד, ובפונקציה Set בלבד לטיפול במשתנה מסווג כתיבה בלבד. לטיפול במאפיין מסווג קריאה/כתיבה, השתמש בשתי פונקציות המאפיין. אם המאפיין מפנה אל אובייקט ולא אל משתנה סקלרי, ניתן להשתמש בפונקציה Set במקום בפונקציה Get. Function Get משמש להחזרת מאפייני אובייקט בין אם מטפלים במשתנה הפונקציהGetProperty Get או באובייקט.

מודולי מחלקה ומקורות נתונים

מודולי מחלקה מהווים אמצעי יעיל לאירוע קוד מסווג כלשהו. הם מכילים ערבים מיוחדים שעשוים להיות מועילים למקרים בהם יש צורך להפוך מקור נתונים לזרמי לצורך עדכון או הצגה, אך בה בעת נדרש לאבטח את מקור הנתונים מפני נזק לא מכובן או רשלני.

עדכן נתונים באמצעות מחרוזת SQL

הדוגמה שלפנינו מנצלת מודול מחלקה כדי לעדכן את השדה UnitsInStock בטבלה Products, בהתאם לשדה ProductID והכמות שהזמנה. שיגרה בת שתי שורות מעבירה שני ארגומנטים לשגרת המשנה במודול המחלקה. דוגמה זו עשויה שימוש במודול מחלקה שונה מלאה שהופיעו בדוגמאות החישוב של מחיר סה"כ לפריט

על מנת לבצע פעולה מסוימת על אובייקט מסוים, מחלקים את הפונקציות וההצורות למקומות אחידים של שגרות שיטה ומאפיינים שימושיים מחלקים מודול אחד (MyTestClass2). המודול מכיל פונקציה בשם OrderIt, המשתנה MyTestClass2 מייצג את המודול. המודול מכיל פונקציה שנקראת PO1. הפונקציה קולטת שני ארגומנטים, אחד עבור ProductID ואחד נוסף עבור היחידות שהוזנו.

```
Sub MyOrder()
Dim OrderIt As New MyTestClass
    OrderIt.PO1 1, 10
End Sub
```

השיגרה השניה, PO1, מעדכנת את מסד הנתונים Products. בmphorsh, היא מפחיתה מהשדה UnitsInStock את מספר היחידות שהוזנו. שיגרה זו נמצאת במודול המחלקה (MyTestClass2). שים לב לעיצוב השיגרה: היא מנצלת אובייקט מסוג Command עם מחרוזת SQL שמנגדירה את שאלתת העדכון. למרות שהשיגרה קולטת שני ארגומנטים, היא אינה מישמת שאילתת פרטט. תחת זאת, היא מנצלת את הארגומנטים שהועברו בתור משתנים בביטוי המחרוזת שמנדר את מחרוזת SQL. עיצוב זה מניב שיגרה מרכזת שקל יחסית לקרוא אותה.

```
' A method for updating a table
Public Sub PO1(ProductID, units)
Dim cmd1 As Command
Dim strSQL As String
' Assign the command reference and connection.
    Set cmd1 = New ADODB.Command
        cmd1.ActiveConnection = CurrentProject.Connection
' Define the SQL string; notice
'the insertion of passed arguments.
    strSQL = "UPDATE Products " &
        "SET UnitsInStock = " &
        "UnitsInStock-" & units & " " &
        "WHERE ProductID=" & ProductID
' Assign the SQL string to the command and run it.
    cmd1.CommandText = strSQL
    cmd1.CommandType = adCmdText
    cmd1.Execute
End Sub
```

עדכון נתונים באמצעות שאילתת פרטט

מפתחים רבים מעדיפים לאמץ גישה מקובלת יותר שמתבססת על שאילתת פרטט. השיגרה PO2 שלפנינו מנצלת שאילתת פרטט כדי לבצע את המשימה שביצעה PO1, והפעם באמצעות מחרוזת SQL. שאילתת פרטט מאפשרת להציג על סוג נתונים נוספים בעזרת כללי VBA מקובלים. שים לב שהקבוע adInteger של ADO מייצג את סוג

הנתונים long, והקבוע AddSmallInt מנסה את סוג הנתונים Integer. עלייך ליצור את הפרמטרים באמצעות השיטה CreateParameter באותו סדר בו הצהרת עליהם בפסוקית Parameters במשפט השאלה. אם לא תקפיד לעשות זאת, תקבל שגיאת זמן ריצה (RunTime Error).

הערה:



בדוק את המידע שמציגה העזרה המקורי של Access על המאפיין Type של האובייקט Parameter של ODO, שם תמצא מגוון סוגי נתונים עבור הצהרת של משתנים.

```
Public Sub PO2(ProductID As Long, units As Integer)
Dim cmd1 As Command
Dim strSQL As String
Dim prm1 As ADODB.Parameter, prm2 As ADODB.Parameter

' Assign the command reference and connection.
Set cmd1 = New ADODB.Command
cmd1.ActiveConnection = CurrentProject.Connection

' Write out SQL statement with parameters & assign to cmd1.
strSQL = "Parameters PID Long,Quantity Integer;" & _
    "UPDATE Products " & _
    "SET UnitsInStock = " & _
    "UnitsInStock-Quantity " & _
    "WHERE ProductID=PID;"
cmd1.CommandText = strSQL
cmd1.CommandType = adCmdText
' Declare parameters; must have same order as declaration.
Set prm1 = cmd1.CreateParameter("PID", adSmallInt, adParamInput)
prm1.Value = ProductID
cmd1.Parameters.Append prm1
Set prm2 = cmd1.CreateParameter("Quantity", adInteger, adParamInput)
prm2.Value = units
cmd1.Parameters.Append prm2

' Run update query.
cmd1.Execute

End Sub
```

- משימת העדכון שמבצעת שאלילת הfrmטר מורכבת מארבעה שלבים עיקריים :
- ↳ הצהרה על משתנים והצבת הפניות.
 - ↳ הגדרת מחרוזת SQL עבור שאלילת העדכון והצבתה במאפיין פקודה.
 - ↳ יצירה והצבה של ערכיים בfrmטרים שהוצהרו בשלב 2 לעיל.
 - ↳ הפעלת הפקודה לעדכון מסד הנתונים.

מקרה לדוגמה: תכנות ממשק כניסה בסיסי

המקרה שלפניך מדגים אפשרות לתוכנות ממשק כניסה (login) בعزيزת Access. השיגורה מتبוססת על מודולי מחלקה עצמאיים וגם על מודולי מחלקות טופס. תהליך הכניסה ותוקן מודול המחלקה מנצלים תכניות תכנות שמתאימות לכל שימוש המחייב שימוש בתנונים עם טפסים לא מאוגדים.

כדי להבליט את תפקידם של מודולי מחלקה ולשמור על שקייפות התהליך, שגרת הדוגמה אינה עושה שימוש ברכיב האבטחה המוכל של Access. במקום זאת, היא מتبוססת על זוג טבלאות ועל שלושה טפסים. הטבלה Passwords כוללת שני שדות בלבד : EmployeeID ו-Password. הטבלה Employees מיובאת ישירות מממסד הנתונים Northwind. השדה EmployeeID הוא המפתח הראשי שלה, והוא מכילה, בין היתר, נתונים עסקיים, אישיים ופרטיו קשור של עובדים. שלושת הטפסים מתיחסים לתוכן הטבלאות כדי לנהל את התהליך הכניסה.

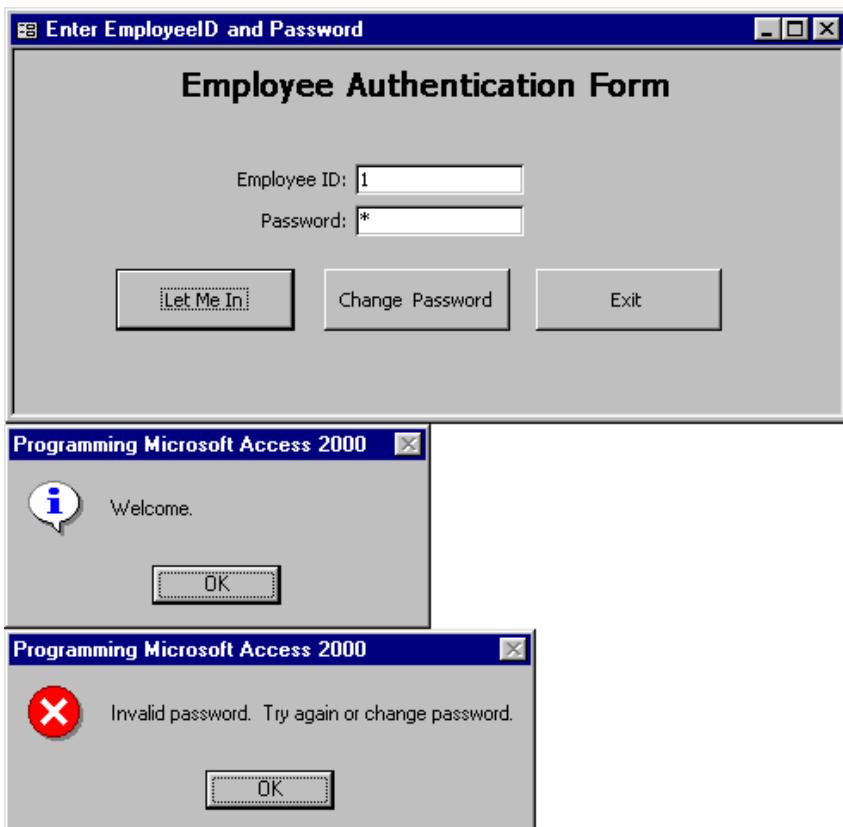
הערה:



מקרה הדוגמה מותר על לכידת שגיאות, פעולה שאפשר בלבדה במערכת מעשית. לכידת שגיאות היא מרכיב חיוני במיזח מערכת שתלויה בקלט משתמש. לאחר שתכיר את המרכיבים הבסיסיים של מודולי מחלקה בפרק זה, כדאי שתறען את ידיעותיך אודוט לוגיקת לכידת שגיאות המתוארת בפרק 1.

טופס הכניסה הראשון

תרשים 7.1 מציג את הטופס הראשון יחד עם שתי תיבות ההודעה שביכולתו ליצור. תרחיש הכניסה נראה כך : המשתמש מקליד את שם המשתמש וסיסמתו בשתי תיבות הטקסט של הטופס ולוחץ על הלוחן **Let Me In**. אם הסיסמה מתאימה לערך השמור עבור השדה EmployeeID, היישום מציג תיבת הודעה פתיחה. אם אין התאמה בין הסיסמה השומרה לבין קוד זיהוי העובד, המשתמש יכול לנפות שנית, או להקליד ערך שונה.



תרשים 1.7: טופס הכניסה הראשוני ובו שתי הודעות תגובה אפשריות

מסכת הקלט של הסיסמה

נוסף לקוד ברקע של הטופס ומודול המחלקה שהופעל באמצעות הטופס, עליך לבדוק בהקפדה את תיבת הטקסט **Password**. התיבה כוללת מסכת קלט של סיסמה שמצוינה כוכבית נגדי כל TWO שמקלידים בתיבת. מקרים את מסכת הקלט לתיבת הטקסט מתוך הרכтиישיה **נתוניות** (Data) של תיבת הדוי-שייח **מאפיינים** (Properties). לחץ על לחץ **בנייה** (Build) במסוק למאפיין **מסכת קלט** (Input Mask) כדי לפתח תיבת דו-שייח שמאפשרת לבחור בה.

הקוד שברקע הטופס

המודול שברקע הטופס, המוצג להלן, מכיל שלוש שגרות אירוע – שיגרה אחת לכל לחץ. לחוץ **Exit** סגור את הטופס ותו-לא. לחוץ **Change Password** פותח טופס נוסף ומעתיק אליו את ערך השדה מתוך הטופס הנוכחי. השיגרה שפותחת את הטופס If frmWhoAmI גם מעבירה את המיקוד אל תיבת טקסט ריקה. לאחר מכן היא סורגת את הטופס הנוכחי.

לחוץ **Let Me In** מפעיל מודול מחלקה עצמאית (MyTestClass3). שם לב כי השיגרה מעבירה את תוכן שתי תיבות הטקסט שלא אל שגרת השיטה `cpw` במודול המחלקה. מודול זה מאתר את סיסמת העובד הקשורה לקוד הזיהוי שלו וקובע אם היא תואמת לסיסמה שבטופס. המחלקה מגיבה באחת משתי הנסיבות אפשריות (ראה טרשים 7.1). מודול המחלקה מפשט את הקוד של שגרת האירוע. כאן מתגלה יתרון נוסף של מודולי מחלקה: האפשרות לעובדות צוות. מפתחים מנוסים יכולים לכתוב שורות בעלות דרגת מעורבות גבוהה במודולי מחלקה, בעוד שפתחים פחות מנוסים יכולים לבצע שימושות פיתוח חדשות יותר ולהפנות למודולי המחלקה כדי לשלב את קטעי הקוד שלהם באלה המתקדמיים יותר.

```
Private Sub cmdExit_Click()
    DoCmd.Close
End Sub

Private Sub cmdNewPassword_Click()
    DoCmd.openform "frmWhoAmI"
    Forms("frmWhoAmI").txtEmpID = Me.txtEmpID
    Forms("frmWhoAmI").txtHireDate.SetFocus
    DoCmd.Close acForm, "frmInputPassword"
End Sub

Private Sub cmdLetMeIn_Click()
    Dim PWT As New MyTestClass3
    PWT.cpw Me.txtEmpID, Me.txtPassword
End Sub
```

הפעלת מודול המחלקה

השיגרה `cpw` של מודול המחלקה `MyTestClass3` מנצלת שאילתת פרמטר כדי לבדוק את הסיסמה של עובד מסוים בטבלה `Passwords`. אחד משני הארגומנטים המועברים לשיגרה הוא קוד העובד; השיגרה נונטנת לפרמטר שלא את ערך הארגומנט. לאחר ביצוע האובייקט `Command` באמצעות שאילתת בחירה, השיגרה מציבה את קבוצת החזרה באובייקט `Recordset`. לאחר שהשדה `EmployeeID` הוא מפתח ראשי בטבלה `Passwords`, שאילתת הבחירה תחזיר תמיד רשותה בודדת.

השיגרה `cpw` מסתiya את הסיסמה שהחזירה השאילתת לסיסמה שהקליד המשמש בטופס, בתור תנאי המשפט `If...Then`. אם יש התאמה, השיגרה מאפשרת למשתמש להיכנס ליישום. בפועל, פותחים טופס נוסף או אובייקט מסד נתונים אחר כלשהו שאליו מגבירים את הגישה באמצעות אבטחה מבוססת-סיסמה. אם אין התאמה, השיגרה מבקשת מהמשתמש להקליד את הסיסמה המדויקת מחדש.

```

Sub cpw(empid As Long, pw As String)
Dim cmd1 As Command
Dim strSQL As String
Dim prm1 As ADODB.Parameter
Dim rst1 As ADODB.Recordset
' Assign the command reference and connection.
    Set cmd1 = New ADODB.Command
        cmd1.ActiveConnection = CurrentProject.Connection
'Write out SQL statement with parameters & assign to cmd1.
    strSQL = "Parameters Secret Long;" & _
        "Select EmployeeID, Password from Passwords " & _
        "Where EmployeeID=Secret"
    cmd1.CommandText = strSQL
    cmd1.CommandType = adCmdText
    Set prm1 = cmd1.CreateParameter("Secret", adInteger, adParamInput)
    prm1.Value = empid
    cmd1.Parameters.Append prm1

' A handy line for catching SQL syntax errors
' Debug.Print cmd1.CommandText

    cmd1.Execute

    Set rst1 = New ADODB.Recordset
    rst1.Open cmd1
    If rst1.Fields("Password") = pw Then
        MsgBox "Welcome on in.", vbInformation, _
            "Programming Microsoft Access 2000"
    Else
        MsgBox "Invalid password. Try again or change password.", vbCritical, _
            "Programming Microsoft Access 2000"
    End If

End Sub

```

טופס הכניסה השני

תרשים 7.2 מציג את הטופס שמוסג בעת שהמשתמש בוחר לשנות את הסיסמה שהוקלדה עבורה קוד העובד המתאים. כל תפקידו של הטופס מותמצה בהנחיית המשתמש לאשר את זהותו. המערכת דורשת את האישור בטרם תאפשר למשתמש לשנות את הסיסמה. בנסיבות רגילות, תיבת הtekst הראשונה מתמלאת על ידי הטופס שטוען אותה (עיין בשגירה הקודמת, cmdNewPassword_Click). כל שעל

המשתמש לעשוט הוא להקליד את תאריך תחילת עבודתו ולחוץ על Submit. הדבר החשוב כאן, הוא לנצל שדה שערכו ידוע לעבוד בלבד. באפשרות לנצל שדה נוסף אחד או יותר כפי שתמצא לנוחך.



תרשים 2.7: טופס הכניסה השני שמנחה את המשתמש לאשר את זהותו

הקוד ברקע הטופס

הטופס מפעיל שאליתה כשהמשתמש לחוץ על לחץ Submit. מודול מחלקת טופס מעבד את נתוני השאליתה ומתאים את תוצאת קבוצת ההחזקה לקלט המשתמש. שגרת האירוע שברקע לחץ Submit כוללת משפט Dim שיוצר עותק של מודול המחלקה MyTestClass3 עם הפניה אל ProcessMe. שורת קוד שנייה מפעילה את השיטה I WhoAmI של המחלקה, כפי שניתן לראות בקוד שפניך.

```
Private Sub cmdSubmit_Click()
Dim ProcessMe As New MyTestClass3
    ProcessMe.WhoAmI CLng(txtEmpID), CDate(txtHireDate)
End Sub
```

הפעלת מודול מחלקת הטופס

שגרת בדיקת המידע של הטופס השני מופיעה בהמשך. היא מנצלת שאליתת פרמטר כדי לבצע בדיקה של תאריך תחילת העבודה של העובד בעל קוד זההו הנתון. הקלדות המשתנים (שים לב לפונקציות CLng ו-CDate בשירה cmdSubmit_Click) לפניו שעוברים למודול המחלקה, מאפשרת לנצל את יתרון אפשרות הקלדת הנתונים בהצורה Parameters וגם את הקלדת הנתונים בטבלה. ללא הקלדה זו, היה על בצע המרות פניות לסוג הנתונים Variant. העיצוב הבסיסי של הבדיקות והחזרה דומה לזה של בדיקת הסיסמה. אם תאריך תחילת העבודה שבטופס מתאים לזה שבטבלה Employees, השירה תפתח את הטופס השלישי.

```

Sub WhoAmI(epid As Long, hd As Date)
Dim cmd1 As Command
Dim strSQL As String
Dim prm1 As ADODB.Parameter
Dim rst1 As ADODB.Recordset

' Assign the command reference and connection.
Set cmd1 = New ADODB.Command
cmd1.ActiveConnection = CurrentProject.Connection

' Write out SQL statement with parameters & assign to cmd1.
strSQL = "Parameters InEmployeeID Long; Select EmployeeID, " & _
    "HireDate From Employees Where EmployeeID=InEmployeeID"
cmd1.CommandText = strSQL
cmd1.CommandType = adCmdText
Set prm1 = cmd1.CreateParameter("InEmployeeID", adInteger, adParamInput)
prm1.Value = epid
cmd1.Parameters.Append prm1

' A handy line for catching SQL syntax errors
Debug.Print cmd1.CommandText

' Execute command.
cmd1.Execute

Check Input vs. Table HireDate.
Set rst1 = New ADODB.Recordset
rst1.Open cmd1
If rst1("HireDate") = hd Then
    DoCmd.openform "frmChangePassword"
    Forms("frmChangePassword").txtEmpID = Forms("frmWhoAmI").txtEmpID
    DoCmd.Close acForm, "frmWhoAmI"
Else
    MsgBox "HireDate not valid for EmployeeID. Try again or Quit.", _
        vbCritical, "Programming Microsoft Access 2000"
End If

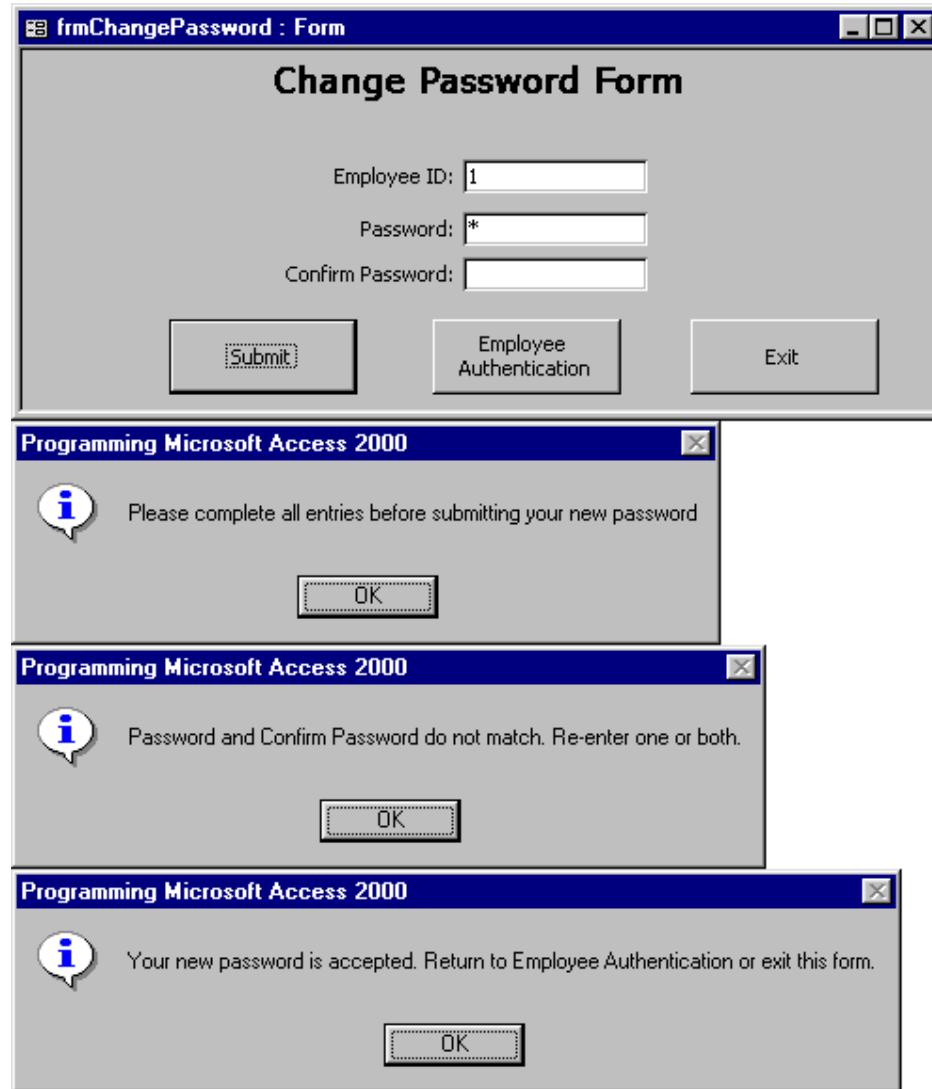
End Sub

```

טופפ הכנסה השלישי

תרשים 7.3 מציג את הטופס האחרון שמוופיע כשהמשתמש לוחץ על לחן בטופס השני, לאחר הקלדת תאריך תחילת העסקה המדויק שלו. הטופס כולל שלוש תיבות טקסט. אחת מיועדת לקוד העובד (תיבה זו מתמלאת עצמאית בנסיבות רגילות). תיבת הטקסט השנייה מיועדת לסיסמה החדשה, והතיבה השלישייה מיועדת לאישור הסיסמה. אם אין התאמה בין תיבות הטקסט הללו, המערכת מציגה למשתמש הודעה

התרעה מתאימה. אם המשתמש לוחץ על לחץ Submit מבלי להקליד ערבים בכל שלוש התייבות, תופיע הודעה תזכורת נוספת. לבסוף, אם הטופס עונה על שתי הדרישות הללו, מודול המחלקה שאליו פנה הטופס מעדכן את הסיסמה עבור העובד בטבלה Passwords.



תרשים 7.3: הטופס השלישי, שמאפשר למשתמש לעדכן את הסיסמה

הקוד ברקע הטופס

המודול שברקע טופס זה הוא המעניין ביותר במקרה שלפנינו. המודול מבצע אימות נתונים, במקומות להעביר אותם למודול המחלקה. גם שיגרה זו מפעילה מודול מחלקה עבור משפט SQL שמעדכן את סיסמת העובד.

ה הפרדה בין אימות הנתונים ועדכון מסד הנתונים מצינה היבט נוסף של יישום מודולי מחלקה – ביצוע מושימות רגישות באמצעות מודול מחלקה. כך ניתן לתקן את המשימות ולהבטיח ביצוע הולם. מרכיבי יישום נוספים שאינם מחיברים תקנון הם מועמדים טיפוסיים להתקאה אישית על ידי מחלקות משתמשי הקצה.

הfonקציות Property Let ו- Property Get – לוגיקת אימות נתונים מיוחדת זו מבוססת על זוג הfonקציות Get ו- Property Let. האירוע AfterUpdate של כל אחת משלוש תיבות הטקסט מפעיל את הfonקציה False או (אם כל התיבות התמלאו נתונים חוקיים, ערך המשתנה הוא True ; אחרת, הערך False).

הfonקציה Property Get משקפת את מצב שלוש תיבות הטקסט בעורת המאפיין filledCheck של הטופס. השירה cmdSubmit_Click בודקת ערך זה כדי לקבוע אם כל שלוש תיבות מסווגות. אם הערך הוא False, השירה מצינה הودעה שモצירה למשתמש להשלים את הקלדות הערכים בכל תיבות. אחרת, שגורת אירוע הלחיצה בודקת אם יש התקאה בין תיבת הטקסט של הסיסמה וזוויל אל אישור הסיסמה. אם לא, תוצג הודעה נוספת שתזכיר למשתמש להקליד סיסמה תואמת. לבסוף, לאחר שהמשתמש עבר את שני השלבים, השירה מפעילה את השיטה NewPS של המופיע .MyTestClass3 המוקומי של המודול3

```
Private AllFilled As Boolean

Private Sub txtConfirm_AfterUpdate()
    Me.filledCheck = txtConfirm
End Sub

Private Sub txtEmpID_AfterUpdate()
    Me.filledCheck = txtEmpID
End Sub

Private Sub txtPassword_AfterUpdate()
    Me.filledCheck = txtPassword
End Sub
```

```

Public Property Let filledCheck(vntNewValu)
    If (IsNull(txtEmpID) Or txtEmpID = "") Or _
        (IsNull(txtPassword) Or txtPassword = "") Or _
        (IsNull(txtConfirm) Or txtConfirm = "") Then
        AllFilled = False
    Else
        AllFilled = True
    End If
End Property

Public Property Get filledCheck()
    filledCheck = AllFilled
End Property

Private Sub cmdSubmit_Click()
    Dim UpdatePW As New MyTestClass3
    If Me.filledCheck = False Then
        MsgBox "Please complete all entries before " & _
            "submitting your new password.", vbInformation, _
            "Programming Microsoft Access 2000"
    ElseIf txtPassword <> txtConfirm Then
        MsgBox "Password and Confirm Password do not " & _
            "match. Re-enter one or both.", vbInformation, _
            "Programming Microsoft Access 2000"
    Else
        UpdatePW.NewPW txtEmpID, txtPassword
    End If
End Sub

Private Sub cmdLogin_Click()
    DoCmd.openform "frmInputPassword"
    Forms("frmInputPassword").txtEmpID = txtEmpID
    Forms("frmInputPassword").txtPassword = txtPassword
    DoCmd.Close acForm, "frmChangePassword"
End Sub

Private Sub cmdExit_Click()
    DoCmd.Close
End Sub

```

העברת ערכים לטופס אחר שתי שגרות נוספות משלימות את רכיביו התפקודים של המודול שברקע הטופס השלישי. שגרת אירוע לחיצה ברקע של לחץ Authentication מחזירה את המשtamsh לטופס הראשון וממלאת את תיבות הטקסט של קוד העובד והסיסמה בערכיהם הלקוחים מהטופס השלישי. רכיב זה משחרר את

המשתמש מהצורך להקליד מחדש את הנתונים מייד לאחר האימות, אך החזרה לטופס הראשון מאפשרת נקודת כניסה ייחודה לישום. הדבר מפשט את תחזוקת היישום בטוחה הארוך. לחוץ היציאה של הטופס, Exit, סוגר את הטופס.

הפעלת מודול המחלקה

מודול המחלקה המופיע על ידי המודול שברקע הטופס השלישי, מונצל ביטוי מחרוזת כדי לחשב משפט SQL שאובייקט Command מעדכן בעורתו את סיסמת העובד. זהה דרך אפשרית לייצוג מחרוזות (כגון ערך הסיסמה) בתוך מחרוזת אחרת (משפט SQL השלם). שים לב לגרשיים הכפולים הרבים, הן לפני והן אחרי ערך הסיסמה החדש. אלה קודי בקרה (escape codes) לייצוג גרש כפול בתוך זוג נספ של גרשים כפולים. מלבד צורת הקינון של מחרוזות בתוך מחרוזת (בהתאם לכללי התחריב של VBA), הקוד קרייא למדי. משפט קטע הודיעה בסוף השיגורה מאשר את שינוי הסיסמה ומנחה את המשתמש כיצד המשיך.

```
Sub NewPW(eid As Long, NuPassword As String)
Dim cmd1 As Command
Dim strSQL As String

' Assign the command reference and connection.
Set cmd1 = New ADODB.Command
cmd1.ActiveConnection = CurrentProject.Connection

' Define the SQL string; notice the insertion of passed arguments.
strSQL = "UPDATE Passwords " &
        "SET Password = "" & NuPassword & """ & _
        "WHERE EmployeeID=" & eid & ";"
Debug.Print strSQL

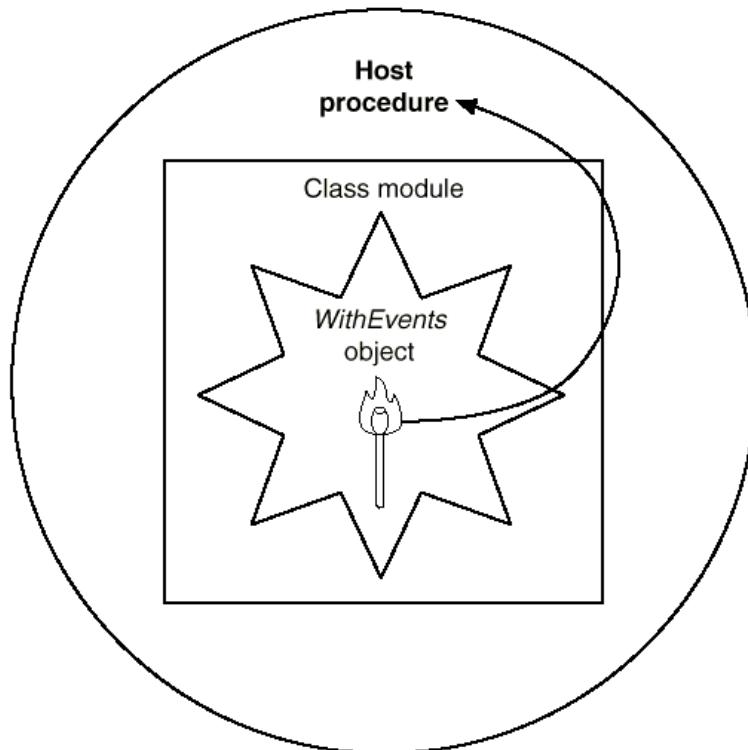
' Assign the SQL string to the command and run it.
cmd1.CommandText = strSQL
cmd1.CommandType = adCmdText
cmd1.Execute

' Confirmation message
MsgBox "Your new password is accepted. " & _
        "Return to Employee Authentication or " & _
        "Exit this form.", vbInformation, "Programming Microsoft Access"
End Sub
```

תכנות אירועים לתוך מחלקות モダム・オブジェクト

באפשרות נצל את VBA ב-Access כדי ליצור מחלקות לייצור מופעים של אובייקטים, אך אובייקטים של VBA אינם מסוגלים ליוזם אירועים בעצמם. יחד עם זאת, תוכל ליצור מחלקות שבסוגן מסוגיות טוג ופקדי ActiveX שמייצים את האירועים שלהם בסביבה מאחרת. במקרה של אובייקטים שמייצים אירועים, יישום VBA יכול לאירוע קוד סביר אירועים שמתרחשים בתוך המחלקה. כשהירה מאחרת של מופע מחלקה מפעילה שיטה שמחוללת הפצת אירוע מתוך המחלקה, שגרת אירוע מגיעה אל המארח של מודול המחלקה (ראה תרשימים 7.4).

הוסף References מתויחס בצורה היררכית לאובייקט Application. מפעילים את השיטות Remove ו-AddFromFile כדי לאפשר לישום להוסיף ולמחוק, באמצעות קוד, הפניות אל ספריות טוג. קבצי ספריה אלה מכילים תיאורים סטנדרטיים של אובייקטים, שיטות ואירועים חסופים. כזכור, מודול האובייקט ADODB תלוי בספריה. ניתן להוסיף הפניות אליו ולספריות נוספות בצורה ידנית או באמצעות קוד.



תרשים 7.4: מילת המפתח WithEvents מפעילה שגרות אירוע של אובייקט אל מארח האובייקט, בעת התறחשות של אירוע

פרק 2 מסביר כיצד להוסיף הפניה באופן ידני אל שלוש הספריות של ADO. הסעיף שלפניך מתאר כיצד להוסיף הפניות באמצעות קוד אל ספריה או פקח ActiveX כלשהם. באפשרותך ליצור הודעה אישור כשהיישום מסיים להוסיף או להסיר הפניה.

שתי מחלקות אירועים מוכללות

מודולי מחלוקת כוללים שני אירועים מוכללים : Initialize ו-Terminate. האירוע Initialize מתרחש כשיצורים מופיע חדש של מודול מחלוקת. יוצרים מעטפת עבור שגרת האירוע Initialize על ידי בחירת Class מותוך התיבה Initialize Object ו-Procedure. באפשרותך לעשות את כל הדרוש בשגרת האירוע Initialize כדי להכין את מופיע המחלוקת לשימוש.

שגרת האירוע Terminate היא כלי הנקיי לאחר ביצוע היישום הנוכחי. פעולת הנקיוי יכולה להסתכם בהגדרת הפניות האובייקט בתווך Nothing. האירועים Initialize ו-Terminate מתרחשים פעם אחת בלבד בתחילת חייו מופיע המחלוקת ובסיומו. זו הסיבה שאינן הם שימושיים ביותר ליצירת התנהגות אינטראקטיבית או דינמית בהזדמנות כלשהי, למעט ביצירה ובפירוק של מodule של מחלוקת.

שימוש במילת המפתח WithEvents לכידת אירועים שהופצו

מודול המחלוקת שלפניך מנצל את מילת המפתח WithEvents כדי ללכוד אירועים שהופצו באמצעות האוסף References. אוסף זה כולל פריט נפרד לכל פריט שנבחר בתיבת הדוא-שייה References. האירועים ItemAdded ו-ItemRemoved מתרחשים רק כשהקובד מוסף או מסיר הפניות. אם משתמש משנה את האוסף References באופן ידני, אירועים אלה אינם מופעלים.

Option Compare Database

' Declare object variable to represent References collection.

Public WithEvents evtReferences As References

' When instance of class is created, initialize evtReferences variable.

Private Sub Class_Initialize()

 Set evtReferences = Application.References

End Sub

' When instance is removed, set evtReferences to Nothing.

Private Sub Class_Terminate()

 Set evtReferences = Nothing

End Sub

```

' Display message when reference is added.
Private Sub evtReferences_ItemAdded(ByVal Reference As Access.Reference)
    MsgBox "Reference to " & Reference.Name & " added.", _
        vbInformation, "Programming Microsoft Access 2000"
End Sub

' Display message when reference is removed.
Private Sub evtReferences_ItemRemoved(ByVal Reference As Access.Reference)
    MsgBox "Reference to " & Reference.Name & " removed.", _
        vbInformation, "Programming Microsoft Access 2000"
End Sub

```

הפעלה וסיום של הפניה מסוג WithEvents

משתמשים במילת המפתח WithEvents בצוירוף מחלקה שמייצגה אירועים. המשפט Public במודול המחלקה שהוצג קודם לכן, מצהיר על הפניה (evtReferences) אל האוסף References באובייקט היישום של WithEvents. מילת המפתח WithEvents שבמשפט מאפשרת למודול המחלקה לכלוד אירועים שהופצו על ידי האוסף References. Shown. שגרת האירוע Class_Initialize מגדרה הפניה. כזכור, לא ניתן לנצל את מילת המפתח New עבור הפניה שהוצאה באמצעות WithEvents.

אריזת קוד סביב אירועים שנלכדו

שתי שגרות אירוע במודול המחלקה, ItemAdd ו-ItemRemove, מפעילות משפטים קטע הודיעה. משפטיים אלה מציגים הודעות שונות שם להפניה ששיתה מוסיפה או מסירה. שגרות האירוע מציגות את תחביר הקוד המותאם אישית לאירועים אובייקטיבים שמפיקים אירועים. במקרה זה, האובייקט הוא האוסף References. שגרות האירוע רק כותבות את שם הספריה שהוספה או הוסרה מהאוסף References.

מודולים סטנדרטיים הגורמים לאירועים

בדומה למודול מחלקה כלשהו, יש צורך בשיגורה אחת או יותר של המודול הסטנדרטי כדי ליצור מופיע של המחלקה (ראה דוגמה בהמשך) וכדי להפעיל שיטות, להציג ערכיו, מאפיין או לקרוא ערכיו מאפיין. באזורי ההצורות של המודול המארח את המחלקה, כוללים את המשפטים Dim ו-Public עם מילת המפתח New ושם המחלקה. על ידי כך יוצרים מופיע של המחלקה ומגדירים הפניה לאובייקט (objRefEvents בדוגמה).

תחביר מודול סטנדרטי של אירועים

אם המופיע של המחלקה מיפוי אירועים מתוך אובייקט מוטבע, עליך להשתמש במשפט Public עם מילת המפתח WithEvents. משפט זה חושף את האירועים למודולים נוספים ונספים שמאחוריהם למחלקה. כמשמעותם מתוך מחלקה הרקע, יש להפוך את

הפניה האובייקט המקומי (objRefEvents), ההפניה שבמודול המחלקה החושפת את האירועים (evtReferences) ולאחר מכן שם שיטה ספציפית, כגון AddFromFile, בנויד להפניה מקובלת למודול מחלקה, ההפניה הנוכחית מצביעה על שיטה Remove. .WithEvents של אובייקט המקור בהצחה.

```
' Create new instance of RefEvents class.  
Dim objRefEvents As New RefEvents  
  
Sub InvokeAddReference()  
' Pass file name and path of type library to AddReference procedure.  
    AddReference "C:\Program Files\Common Files\System\ado\msjro.dll"  
End Sub  
  
Sub InvokeRemoveReference()  
' Pass name of existing reference. (Use internal name from File  
' Properties list; same name appears when adding reference.)  
    RemoveReference "JRO"  
End Sub  
  
Sub AddReference(strFileName As String)  
    objRefEvents.evtReferences.AddFromFile (strFileName)  
End Sub  
  
Sub RemoveReference(strRefName As String)  
    objRefEvents.evtReferences.Remove objRefEvents.evtReferences(strRefName)  
End Sub
```

הדוגמה שלעיל מוסיפה הפניה לטפירה שמאחסנת את המודול JRO (מודול זה מאפשר לבצע שכפול Jet באמצעות O). מפעילים את השיגורה InvokeAddReference כדי ליצור את הפניה. השיגורה קוראת לשיגורה נספה, AddReference, באמצעות תחביר References פריט לאוסף References (DLL) שנמצאת בדרך כלל בתיקיה המאחסנת את מודול JRO הוא ספריית קישור דינמי (ADO) שהנתיב שלו הוא Program Files\System\ADO. הפעלת השיגורה שמנתיב References את הפריט JRO מטבחת את הפריט האוסף References. היעד של RemoveReference הוא המאפיין Name של הפריט באוסף References.

הערה:



כדי לגלוות את הארגומנטים של השיטות AddFromFile, Remove-And-Remove, הוסף הפניות באופן ידני. לאחר מכן, מספר את פרטי האוסף References. Each...For...FullPath שם המשפט המשפטutzת השם שלם, Name ו-FullPath. השתמש בערכי מאפיינים אלה כדי לזרות בצורה ייחודית ארגומנטים של השיטות Remove-And-AddFromFile.

הרחבת היישום

באפשרותך לאמץ בклות את המחלקה RefClass על ידי שימוש בשיטות AddReference ו-RemoveReference כדי לאפשר תחילה בחירה מquiv וგמיש יותר. לדוגמה, היישום שיצרת יכול לגזר את קלט השיגרה AddReference מתוך אוסף של ספריות סוג, קבצי הפעלה, פקדי ActiveX ואפליו מסד נתונים. ניתן להציג למשתמש תיבת משולבת ובה רשימת הפניות להוספה או להסרה. לחילופין, השיגרה יכולה לבחור מתוך רשימה המבוססת על גורמים נוספים, כגון יעד המשימה של המשתמש.

שימוש באובייקט ALL

אם אתה נמנה עם המפתחים הנוהגים לעקוב אחר האובייקטים שלהם בפרויקט מסך נתונים (רובנו רואים זאת כחיוני), תשכח לדעת על האוסף AllModules, מכיר קרוב של האוספים AllReports ו-AllForms אוותם פגשו בפרק 5 ו-6. חברי האוספים שמתחלים ב-All אינם אובייקטיים מסד נתונים, כגון טפסים, דוחות ומודולים, אלא אובייקטים מהסוג AccessObject המכילים כמהות מינימלית של פרטיטים אודוות רוב הסוגים או אובייקטים שמורים במסד נתונים.

מאפייני AccessObject

ניתן למספר את אובייקטי AccessObject במהירות באוסף כלשהו. לאחר שאובייקטים אלה מצבעים על אובייקטים שמורים, לא ניתן להוציא או למחוק חברים. מבצעים שימושות אלו באמצעות האוספים הפתוחים עליהם הם מצבעים.

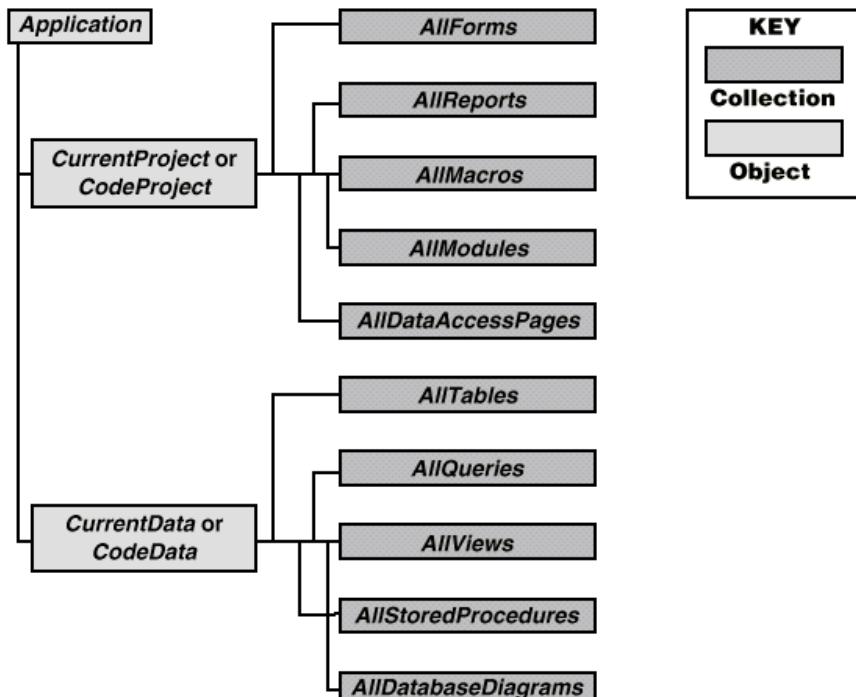
כשאתה פוגש באובייקט מסווג AccessObject שהישום שלך זוקק לפרטיטים נוספים, אפשרו, באפשרותך לנצל את המאפיינים IsLoaded ו-Name כדי לבדוק את מאפייני האובייקט שלוו מצבע האובייקט AccessObject. אוסף אובייקטים פתוחים אלה כוללים מערך מquiv של מאפיינים ושיטות שאינם זמינים באוסף All.

אובייקטי AccessObject כוללים מאפיין מהסוג Type שמתאר את הסוג של AccessObject ולא את הסוג של אובייקט מסד הנתונים. ערך המאפיין Type של כל חבר באוסף AllModules הוא 5. ערך זה מייחד חבר מסווג AccessObject באוסף AllModules מחבר אחר באוסף AllForms, שערך המאפיין Type שלו הוא 2. בכל מקרה, לא תוכל לקבוע אם אתה מטפל במודול מחלוקת או במודול מחלוקת טופס על ידי בדיקה פשוטה של החבר AccessObject באוסף All. عليك לבדוק את המאפיין Type של אובייקט Module ואת המאפיין HasModule של אובייקט Form.

האוספים All

קיימים שבעה אוספים שמתחלים ב-All, פרט לאוספים AllForms, AllModules ו-AllReports. קבוצת עשרת האוספים (ראה טרשים 7.5) מתחולקת באופן טבעי לשתי קבוצות בנות חמישה אוספים כל אחת. האוספים AllMacros, AllReports, AllForms AllMacros.

CodeProject ו- CurrentProject הם חברים באובייקטים AllViews ,AllQueries ,AllTables ,Access Application של האוסףים CurrentData ו- AllDatabaseDiagrams AllStoredProcedures באובייקט Application של Access. כאשרה מציין אובייקטים מסווג AccessObject באוסף כלשהו מתוך שורת האוסףים, עליך להגדיר הפניה שמצוינה על האובייקט הקודם הנכון. אם לא תקפיד לעשות זאת, תקבל שגיאה.



תרשים 7.5: שורת האוסףים All והקרבה ההיורקטית ביניהם לבין האובייקטים Project-Data.

סוג הקובץ של Access מעניק זמינות מוגבלת לאוסףים AllViews ,AllQueries ,AllTables AllDatabaseDiagrams ו- AllStoredProcedures בקובץ המקבול בתבנית *.mdb, או בתבנית החדשה של Access 2000 *.adp (פרק 4 מציג פרטיים בסיסיים של קובץ *.adp, ופרק 12 עוסק בכך בהרחבה). האוסף AllQueries זמין בקובצי *.mdb, אך לא בקובצי *.adp. לעומת זאת, אפשרו לטלל באוסףים AllViews ,AllStoredProcedures AllDatabaseDiagrams ו- *.adp מתוך קבצים מסווג *.adp, אך לא מקובצי מסוג *.mdb. היחסומים יכולים עדין להפנות לתוכנות ושגרות מאוחסנות בקובצי מסוג *.mdb. באמצעות ספריית האובייקט ADOX.



יתכן שתהיה מודע 2000 Access מאפשר לטפל באוספים Procedures ו-Vies AllViews מתוך קבצים מסוג mdb.*. אך איןנו מאפשר לטפל באוספים AllStoredProcedures ו-AllViews בקבצים מסוג db.*. שני האוספים האחרונים לא נחשפו לקבצים מסוג mdb.*. עקב הצורך למלא דרישות שקיים גבוהה יותר. לטפל באוספים AllViews ו-AllStoredProcedures אלה יהיה אפשרי בהדרגה עתידית של Access, או בעדכון מהדורתו הנוכחי.

מספר חברי האוסף All

שלוש השגרות שלפניך מפגינות רמת דמיון גבוהה בתכנות אוספים שונים מהסוג All. השירה הראונה מבצעת מספר פשוט של כל המודולים בפרויקט הנוכחי. שים לב כי היא מצהירה תחילה על obj1 סוג Application.CurrentProject.AllModules, מכיוון שהיא מקבלת את זהות האלמנטים שבאוסף AllModules, כולל אובייקטים מהסוג Application.CurrentProject.Application.AllModules. שים לב גם שלולאות המספר עוברת על האוסף AllModules, אך הקוד מגיע לאוסף זה באמצעות האובייקטים Application.CurrentProject.AllModules.

```
Sub EnumerateAllModules()
Dim obj1 As AccessObject
    For Each obj1 In Application.CurrentProject.AllModules
        Debug.Print obj1.Name & vbTab & obj1.Type & vbTab & obj1.IsLoaded
    Next obj1
End Sub

Sub EnumerateAllForms()
Dim obj1 As AccessObject
    For Each obj1 In Application.CurrentProject.AllForms
        Debug.Print obj1.Name & vbTab & obj1.Type & vbTab & obj1.IsLoaded
    Next obj1
End Sub

Sub EnumerateAllTables()
Dim obj1 As AccessObject
    For Each obj1 In Application.CurrentData.AllTables
        Debug.Print obj1.Name & vbTab & obj1.Type & vbTab & obj1.IsLoaded
    Next obj1
End Sub
```

מבנה השגרות EnumerateAllTables ו-EnumerateAllForms דומה לזה של השירה EnumerateAllModules. ייחד עם זאת, עליך לשים לב להבדלים משמעותיים אחדים בין AllTables והשגרות. ראשית, האוסף הגרפי AccessObject משתנה מ-AllTables אל AllForms בשירה אחת ול>AllTables בשירה האחרת. שנית, הנטיב אל האוסף

AllTables עובר דרך האובייקט CurrentProject,(CurrentData, ולא דרך האובייקט AllViews AllTables או AllStoredProcedures). הקוד היה מסוגל לפעול גם בקובץ מסוג *.mdb, אך לא בקובץ מסוג *.adp.*.

התאמה לסוגי הקבצים *.mdb ו-*.adp

המאפיין ProjectType של האובייקט CurrentObject מאפשר לגלוות את סוג הקובץ בו מטפלים, *.mdb או *.adp.*. הדבר מאפשר כתיבת שגרות בודדות שמתאימות לשביבה בה הוא אמורות לפעול. הדוגמה ש לפניך מציגה את שמן של כל התצוגות והשגרות המאוחסנות בקובץ *.adp, אך במקרה הצורך היא עובה לתצוגת כל השאלות שבקובץ *.mdb. כפי שניתן לראות, התħħolha ħiġiedha d'darosa كانilia hawnia, בדוק את ערך המאפיין ProjectType Type של AccessObject. מאפיין ProjectType מאפשר לבדוק להבדיל אובייקטים של תצוגות שערך הסוג שלהם 7, מאובייקטים שמצביעים אל שגרות מאוחסנות, שערך הסוג שלhn הוא 9.

```
Sub EnumerateAllViews2()
Dim obj1 As AccessObject, dbs1 As Object
    Set dbs1 = Application.CurrentData
    If Application.CurrentProject.ProjectType = acADP Then
        For Each obj1 In dbs1.AllViews
            Debug.Print obj1.Name & vbTab & obj1.Type & vbTab & obj1.IsLoaded
        Next obj1
        For Each obj1 In dbs1.AllStoredProcedures
            Debug.Print obj1.Name & vbTab & obj1.Type & vbTab & obj1.IsLoaded
        Next obj1
    Else
        For Each obj1 In dbs1.AllQueries
            Debug.Print obj1.Name & vbTab & obj1.Type & vbTab & obj1.IsLoaded
        Next obj1
    End If
End Sub
```

AllModules-ו AllForms

הדוגמה הבאה מנצלת את האוספים All והאוספים המתאימים של מודולים וטפסים פטוחים כדי לפתח רשימה של כל המודולים (לפי סוג) ומודולי המחלקה של טפסים בפרויקט. הויל ומאפיין הציון של מודולי מחלקה סטנדרטיים שונה של מודולי מחלקה של טפסים, הקוד מחייב בדיקה של ביטויים שונים של מודולי מחלקה סטנדרטיים לעומת מודולי מחלקה של טפסים.

כזכור, מודולים יש מאפיין Type, אך לטפסים יש מאפיין HasModule. על הקוד לעבור במחזריות על חברי האוסףים AllForms ו-AllModules מכיוון שהקלק מהמודולים והטפסים, או אפילו כולם, עלולים להיות סגורים. בודקים את המצב של IsLoaded של האובייקטים AccessObject ב-AllForms וב-AllModules כדי לקבוע אם יש צורך לפתח מודול או קובץ טרם הعرצת סוג המודול שלו, או כדי לקבוע אם לטופס יש מודול מחלה. השירה שוגרת מחדש טפסים ומודולים לאחר שבדקה אותם.

```
Sub ListAllModulesByTypeAndClassForms()
Dim obj1 As AccessObject, dbs1 As Object
Dim mod1 As Module, frm1 As Form

    Set dbs1 = Application.CurrentProject

    ' Search for open AccessObject objects in AllModules collection.
    ' Open and reclose those that are not open.

    For Each obj1 In dbs1.AllModules
        If obj1.IsLoaded = True Then
            ListTypeOfModule obj1.Name
        Else
            DoCmd.OpenModule obj1.Name
            ListTypeOfModule obj1.Name
            DoCmd.Close acModule, obj1.Name
        End If
    Next obj1

    ' Search for open AccessObject objects in AllForms collection.
    ' Open and reclose those that are not open.

    For Each obj1 In dbs1.AllForms
        If obj1.IsLoaded Then
            DoesFormHaveModule obj1.Name
        Else
            DoCmd.openform obj1.Name
            DoesFormHaveModule obj1.Name
            DoCmd.Close acForm, obj1.Name
        End If
    Next obj1
End Sub
```

```

Sub ListTypeOfModule(modname)
Dim strType As String
' Decode module Type value.
If Modules(modname).Type = 0 Then
    strType = "Standard Module"
Else
    strType = "Class Module"
End If
' Print module name and type.
Debug.Print Modules(modname).Name & vbTab & strType
End Sub

Sub DoesFormHaveModule(frmname)
' Only print form name if it has a module.
If Forms(frmname).HasModule = True Then
    Debug.Print frmname & vbTab & "Form Class Module"
End If
End Sub

```

עריכת מודולים באמצעות קוד

מהחר שניתנו לבצע משימות רבות מאוד באמצעות מודולים סטנדרטיים, מודולי מחלקה עצמאיים ומודולי מחלקה של טפסים, סביר שהיישומים שתיצור יכילו מודולים רבים מסווגים אלה. בבוא הזמן יהיה צורך לתחזק אותם. מטלת תחזוקה שכיחה היא הדרישה להוסיף או למחוק שורת קוד אחת או יותר בקבוצת מודולים. סעיף זה מתאר כיצד להוסיף ולמחוק שורה בכל מודולי המחלקה הסטנדרטיים והעצמאיים ולאחר מכן מסביר כיצד עושים זאת במודולי מחלקה טופס. הקוד של מודולי מחלקה סטנדרטיים ועצמאיים מאוחסן בצורה שונה מזו של הקוד של מודולי מחלקה טופס, ולכן יש הבדלים מסוימים בין התהליכיים.

גישה של ערך

האובייקט **Module** כולל שיטות ומאפיינים שימושיים לשיכולם לסייע בעריכת מודולים באמצעות קוד. הדוגמאות בסעיף זה מבוססות על השיטות **Find**, **InsertLines** ו-**DeleteLines**. שיטות אלו מטפלות במודולים סטנדרטיים ובמודולי מחלקה, כולל מודולי מחלקה עצמאיים וגם מודולי מחלקה דוח וטופס. שיטות אלו מהוות קבוצה משנה של שיטות ומאפיינים שתומכות בניהול תוכן מודול באמצעות קוד.

השיטה **InsertLines** ייחד עם האובייקט **Module** משמשים להוספה שורה אחת או יותר במודול. מספר שורות המודול מתחילה ב-1 ומגיע עד לערך שמכיל המאפיין **CountOfLines** של המודול. השיטה קולטת את מספר השורה וארגון מחרוזת. אם

עליך להוסיף שורות רבות למודול, הוסף את הקבוקים vbCrLf vbCrLf המחרוזת שמייצג את ארגומנט המחרוזת של השיטה. כמשמעותם שורות באמצעות שיטה זו, היא עוברת לשורות הבאות במודול.

השיטה Find מחפש מחרוזת טקסט במודול. היא מחזירה ערך True כשהיא מוצאת את הטקסט המבוקש, או False במקרה שלא. אם ידוע לך לבדוק היכן נמצא קטע טקסט מסוים, אפשרות להגדיר עמודה ושורת פтиחה, ועמודה ושורת סיום. אם איןך יודע את מיקומו של קטע הטקסט במודול, השאר את הארגומנטים של מיקום הטקסט ריקים, והפונקציה תחזיר את ערכי הטקסט שאתה מחפש במודול. אפשרות גם להקצות חיפוש לפי תבנית וחיפוש תלוי רישיות.

השיטה DeleteLines מסירה שורת טקסט אחת או יותר ממודול. השיטה Kollettat שני ארגומנטים: שורת פтиחה ומספר כולל של שורות המיועדות להסרה מהמודול. אפשרות לנצל את השיטה כדי לחפש טקסט במודול. אם תעשה זאת, תוכל לבסס את הבקשה של השיטה DeleteLines על ערך החזרה של השיטה Find.

הוסף טקסט למודולים

הಗורות שלפניך משלבות את האוספים Modules-ו AllModules כדי לעזרך את הטקסט באוסף של מודולים. ליתר דיוק, הן מוסיפות שורת הערא בתחלת כל מודול כדי להזכיר עליו שהוא מודול סטנדרטי או מודול מחלקה. השירהEnumerateAllModulestoInsert לשירה אחרת, שמעדכנת בפועל את מודולי היעד. השירהInsertIntoModules דורשת מודול פתוח, ולכן השירה הראשונה פותחת את המודול אם איןנו פתוח עדין. לאחר שהשירה השנייה מחזירה את השיליטה לשירה הראשונה, היא סוגרת את המודול כדי להחזירו במצבו המקורי.

```
Sub EnumerateAllModulestoInsert()
Dim obj1 As AccessObject
' Loop through AllModules members.
' If module is open, call sub to insert lines;
' else open module first, then close afterwards.
    For Each obj1 In Application.CurrentProject.AllModules
        If obj1.IsLoaded = True Then
            InsertIntoModules obj1.Name
        Else
            DoCmd.OpenModule obj1.Name
            InsertIntoModules obj1.Name
            DoCmd.Close acModule, obj1.Name, acSaveYes
        End If
    Next obj1
End Sub
```

```

Sub InsertIntoModules(modname)
Dim strType As String, mod1 As Module
    Set mod1 = Modules(modname)

' Detect module type to determine which string to insert.
    If mod1.Type = 0 Then
        strType = "Standard Module"
    Else
        strType = "Class Module"
    End If
    mod1.InsertLines 1, strType
    Set mod1 = Nothing

End Sub

```

השירה InsertIntoModules קולטת ארגומנט יחיד – שם המודול אותו יש לעורך. השירה אינה מבצעת מעבר מחזורי כלשהו, מכיוון שהשירה הראשונה קוראת לה פעם אחת לכל חבר באוסף AllModules. השירה פותחת בהגדרת הפניה אל המודול ששמו הועבר בצורת ארגומנט. לאחר מכן קובעת את סוג המודול אליו ההפניה מפנה ומכניסה ערך משתנה מחרוזת להערכה שמצוינת את שם סוג המודול. לאחר שנקבע הטקסט שיש להוסיף, השירה מפעילה את השיטה InsertLines על המודול אליו הוגדרה ההפניה.

מחיקת טקסט ממודולים

שתי השגרות שלפניך מוחקют שורה מתוך שירה. למעשה של דבר, הן מסירות את השורה שהוספה באמצעות זוג השגרות הקודמות. עיצוב שתי השגרות הבאות גמיש דיין, ולבן אפשרותך להרחיבן בקלות כך שיכילו מחיקה של שורות נבחרות רבות מתוך קבוצת מודולים כלשהי.

השגרות מבוססות על לוגיקה כללית זהה לו של זוג השגרות הקודם בהבדל עיקרי אחד: זוג זה מנצל את השיטות Find ו-Delete-*i*. Find כדי להסיר טקסט במקומות השיטה InsertLines. לשיטה Find חשייבות קריטית כشمתוכנים להפעיל את השיטה DeleteLines, מכיוון שהיא מאפשרת לקוד לקבוע את מציאותו של טקסט כלשהו במודול בטרם תמחק אותו. במקרה שלפנינו, השיטה Find מhapus את המילה Module ברכף 40 התווים הראשוניים בשורה הראשונה. השירה DeletefromModules מפעילה את השיטה DeleteLines כדי למחוק שורה אחת החל מהשורה הראשונה של המודול. השיטה DeleteLines מסירה שורות בזורה בלתי מותנית. יחד עם זאת, ניתן להפעיל בזורה ידנית את הפונקציה Undo Delete כדי לשחרר טקסט שהוסר.

```

Sub EnumerateAllModulestoDelete()
Dim obj1 As AccessObject, dbs As Object
Dim mod1 As Module, frm1 As Form

' Loop through AllModules members.
' If module is open, call sub to delete line;
' else open module first, then close afterwards.
    For Each obj1 In Application.CurrentProject.AllModules
        If obj1.IsLoaded = True Then
            DeletefromModules obj1.Name
        Else
            DoCmd.OpenModule obj1.Name
            DeletefromModules obj1.Name
            DoCmd.Close acModule, obj1.Name
        End If
    Next obj1
End Sub

Sub DeletefromModules(modname)
Dim mod1 As Module
    Set mod1 = Modules(modname)

' Delete first line if first 40 characters contain "Module".
    If mod1.Find("Module", 1, 1, 1, 40) = True Then
        mod1.DeleteLines 1, 1
    End If
    Set mod1 = Nothing
End Sub

```

הוספת טקסט למודולי מחלקה טופס

שתי השגרות ש לפניך מוסיפות שורה בתחילת כל מודול מחלקה טופס, ובה הערה כי מדבר במודול מחלקה. במקום לעבור בלולה על כל האוסף AllModules, השירה הראונה עוברת על האוסף AllForms. לכל חבר באוסף, היא קוראת לשירה .InsertIntoForms

שירה שנייה זו מעריכה אם שם הטופס שהועבר הוא מודול מחלקה. אם הוא אכן כזה, השירה מגדרה הפניה למודול שברקע הטופס. על ידי כך היא חושפת את המודול. השירה מסתiyaמת בהוספת שורת הערה למודול והגדרת הפניה אל Nothing. כדי לשחרר את המשאבים.

```

Sub EnumerateAllFormsToInsert()
Dim obj1 As AccessObject

' Loop through AllForms members;
' if form is loaded invoke module to insert line,
' else open form first and then close afterwards.
For Each obj1 In Application.CurrentProject.AllForms
    If obj1.IsLoaded Then
        InsertIntoForms obj1.Name
    Else
        DoCmd.openform obj1.Name
        InsertIntoForms obj1.Name
        DoCmd.Close acForm, obj1.Name, acSaveYes
    End If
Next obj1
End Sub

Sub InsertIntoForms(frmname)
Dim mod1 As Module, strType As String

' If Form has module, set reference to it and insert line into the module.
' Free reference resource when done.
If Forms(frmname).HasModule = True Then
    Set mod1 = Forms(frmname).Module
    strType = "Form Class Module"
    mod1.InsertLines 1, strType
    Set mod1 = Nothing
End If
End Sub

```

מ剔ת טקסט ממודול מחלוקת טופס

שתי השגרות שלפניך מסירות את שורת ההערה Class Module מהשורה הראשונה של המודולים שברקע הטפסים. כפי שניתן להבחין, זוג השגרות מאמצן את התנהגויות של מרכיבים קרייטיים מזוג השגרות הקודם בנוגע להוספה ומחיקה של שורות. זוג השגרות עובר בכוונה מחזוריית על האוסף AllForms, כמו זוג השגרות שהוסיף את שורת ההערה בתחילת כל מודולי מחלוקת הטופס של הפרויקט.

יחד עם זאת, השירה השנייה בזוג זה מנצלת את השיטות Find ו-DeleteLines כדי להסיר את השורה הראשונה במודול, אם המילה Module מופיעה בתחום 40 התווים הראשונים בשורה. בכך היא דומה לשירה שמוחקת שורות מהאוסף Modules.

```

Sub EnumerateAllFormstoDelete()
Dim obj1 As AccessObject

' Loop through AllForms members;
' if form is loaded invoke module to remove line,
' else open form first and then close afterwards.
For Each obj1 In Application.CurrentProject.AllForms
    If obj1.IsLoaded Then
        DeletefromForms obj1.Name
    Else
        DoCmd.openform obj1.Name
        DeletefromForms obj1.Name
        DoCmd.Close acForm, obj1.Name, acSaveYes
    End If
Next obj1
End Sub

Sub DeletefromForms(frmname)
Dim mod1 As Module, strType As String

' If form has module, then check contents of first line
' for "Module", and delete the first line if it is present.
' Free module reference resource when done.
If Forms(frmname).HasModule = True Then
    Set mod1 = Forms(frmname).Module
    If mod1.Find("Module", 1, 1, 1, 40) = True Then
        mod1.DeleteLines 1, 1
    End If
    Set mod1 = Nothing
End If
End Sub

```


אובייקטִי Microsoft Office

במרכיב של 2000 Microsoft Office, חולק Microsoft Access מבחר אובייקטים עם שאר יישומי Office. אובייקטים אלה מאפשרים ביצוע משימות כמו חיפוש קבצים, טיפול בתוכנות העזרה Office Assistant, עריכת שינויים בתפריטים ובסרגלי הכלים הסטנדרטיים, ופיתוח של תפריטים וסרגלי כלים מותאמים אישית. נוסף לכך, היכולות שפותח בתוכנות האובייקטיבים הללו ב-Access תשמש אותך גם ביישומי Microsoft Word, Microsoft Excel, Microsoft FrontPage – והיפלו Office אחרים. רוב האובייקטים פועלים בכל יישומי Office.

פרק זה פותח בסקירה כללית על האובייקטים של Office, ולאחר מכן מתמקד בשלושה אובייקטים: FileSearch, המשמש לניהול תוכניות של חיפושי קבצים; Assistant, CommandBars המשפק משק תוכניתית לתוכנה Office Assistant במסגרת Office; וכן המשמש ליצירת תפריטים וסרגלי כלים מותאמים אישית.

שימוש באובייקטים המשותפים של Office

האובייקטים המשותפים של Office, המופיעים בטבלה שבעמוד הבא, תומכים במספר תחומי פונקציונליות חשובים. חלק מבני האובייקטים הללו אינם זמינים בכל רכיבי Office. הטבלה מတיר את המבנים, מצינית אם זמיןותם מוגבלת לרכיבי Microsoft, ומננה למידע נוסף אודוותיהם בספר זה. לקבלת עזרה מקוונת ו כדי לעבוד מסויימים, ומפנה למידע נוסף אודוותיהם בספר זה. לקבלת עזרה מקוונת ו כדי לעבוד עם אובייקטים אלה, פנה ל-Object Library Tools (כלים) בעורך VBE Visual Basic, או באופן תוכני (כפי ב_amp;quot;אמצעות תפריט Tools") (כלי) בעורך VBE Visual Basic, או באופן תוכני (כפי שיודגס בהמשך פרק זה).

אובייקטים משותפים ב-Office

הערות	תיאור	האובייקט
ראה דיוון נוסף ודוגמאות בהמשך פרק זה.	השתמש באובייקט זה ובօպירט שלוי לייצור סרגלי כלים, שורת התפריטים ותפריטי קיזור, או להכנסת שינויים בפריטים אלה. אפשרותך להוציא שינויים בזמן-עיצוב באופן ידני או באמצעות קוד VBA. שינויים בזמן-ריצה אפשריים בעזרת קוד VBA בלבד.	CommandBar
ראה דיוון נוסף ודוגמאות בהמשך פרק זה.	השתמש באובייקט זה לtimica בדרישות עזרה מותאמת אישית עבור Office Assistant והבלון של Office Assistant.	Assistant
ראה דיוון נוסף ודוגמאות בהמשך פרק זה.	השתמש באובייקט זה לייצוג הפקציינליות של תיבת הדוד-שים (Open).	FileSearch
unin במדריך Microsoft Office .2000 Programmer's Guide	זה יציג של תוסף COM ב-Access וב-Office מארח אחרים.	COMAddin
מנהל מעקב אחר נתוני דיהו מקומיים בפרשיה בינהומית של Office.	אובייקט זה מאפשר החזרה תוכניתית של מידע אודוט התקנת שפות, משק המשתמש והגדרות העדרה.	LanguageSettings
כולל מאפיינים ושיטות לטיפול בקבצים שמחזיר אשף התשובות.	השתמש באובייקט זה לטיפול תוכני באשף התשובות.	AnswerWizard
מיועד למסמכי Word, Excel ותוכנות בעבודה ב-Microsoft PowerPoint בלבד.	אובייקט זה מייצג מאפיין מוכללו מותאם אישית של מסמך Office. לחייבתך עד 28 מאפיינים מוכלים, הcoliים תכונות מסמך כמו כותרת, מחבר, הערות, תאריך הדפסה אחרונה, שעת שמירה אחרונה וזמן עריכה כולל. אובייקט זה תומך גם בתכונות מסמך מותאמות אישית.	DocumentProperty
מיועד למסמכי Word, Excel ותוכנות בעבודה ב-Microsoft PowerPoint בלבד.	אובייקט זה הוא פרויקט ברמה העליונה בעורך ה-Script של Microsoft. השתמש באוסף HTMLProjectItem למעקב אחר מסמכי HTML במסגרת פרויקט.	HTMLProject

הערות	תיאור	האובייקט
מיועד למסמכים Word, Excel ו- Microsoft PowerPoint ולמצגות ב- PowerPoint בלבד.	אובייקט זה מייצג בלוק של Script בערך ה-Script.	Script
מיועד למסמכים Word, Excel ו- Microsoft PowerPoint ולמצגות ב- PowerPoint בלבד.	אובייקט זה מייצג את גוף ברירת המחדל בעת שמירת מסמךCDF Web.	WetPageFont

מאפייני Access Database

שלא כמו ב-Word, Excel ו-PowerPoint, חסר ב-Access האובייקט המשותף DocumentProperty, UserDefinedSummaryInfo :DocumentMSysDB :SummaryInfo ו-UserDefinedContainer DAO (Data Access Object) של Database Container של SummaryInfo. אובייקטים אלה זמינים אך ורק באמצעות SummaryInfo (Summary). האובייקט שבתיבת הדו-שיח מאפייני מסד נתונים (Custom) מכיל את כל המאפיינים המופיעים בכרטיסיה תקציר (Properties). האובייקט UserDefined מכיל את כל המאפיינים הכלולים בכרטיסיה התאמת אישית (Custom Properties) באותה תיבת דוא-שיח. האובייקט MSysDB מכיל את כל המאפיינים המוגדרים באמצעות האפשרות הפעלה (Startup) שבתפריט כלים (Tools), במסד נתונים.

:DAO Database Container של כל אובייקט הדוגמה הבאה מונה את אוסף המאפיינים של

```
Sub enumDBProps()
Dim db As Database, p As DAO.Property

' Set reference to current database.
Set db = CurrentDb

' Print heading for results.
Debug.Print "User defined properties"
Debug.Print "===="
' Iterate through UserDefined database properties.
For Each p In db.Containers!Databases.Documents!UserDefined.Properties
    Debug.Print p.Name, p.Value
Next

' Print heading for results.
Debug.Print
Debug.Print "Summary Properties"
Debug.Print "===="
```

```

' Iterate through SummaryInfo database properties.
For Each p In db.Containers!Databases.Documents!SummaryInfo.Properties
    Debug.Print p.Name, p.Value
Next

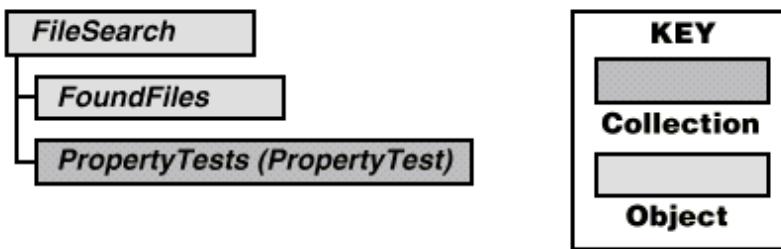
' Print heading for results.
Debug.Print
Debug.Print "MSysDB Properties"
Debug.Print "====="

' Iterate through MSysDB database properties.
For Each p In db.Containers!Databases.Documents!MSysDB.Properties
    Debug.Print p.Name, p.Value
Next
End Sub

```

האובייקט FileSearch

השתמש בתבנית האובייקט FileSearch (המוצגת בתרשים 1) כדי לשלב חיפושם ביחסומי. תוכל לחפש בכוננים הקשיחים של המחשב או בכוננים של מחשבים המוקשרים בראש תקשורת מקומית (LAN). אובייקט זה חושף את הפונקציונליות של תיבות הדוח-שיכון (Find) ו-חיפוש (Open). כפי שמצוואר בתרשים 1, האובייקט .FileSearch ו-האוסף FoundFiles תלויים מבחינה היררכית באובייקט PropertyTests



תרשים 1: האובייקט FileSearch הוא אחד האובייקטים המשותפים של Office.

באפשרותך להגדיר חיפוש קובץ בשתי דרכים בסיסיות, וכל גישה מתאימה לאפשרויות שונות בתיבת הדוח-שיכון **Find**.

↳ חיפוש לפי קרייטריון יחיד (שם קובץ או תבנית, סוג קובץ, או נתיב).

↳ חיפוש תוכניי לפי קרייטירוניים מרובבים, בעזרת האוסף .PropertyTests

בגישה השנייה, תוכל להגדיר ארגומנטים אשר ישמשו קלט לתיבת הדוח-שיכון **פתיחת**. השתמש בתיבת דוח-שיכון כדי לציין, תנאים מרובים לחיפוש, וכן כללים לשරור, כגון האופרטורים And או Or. השתמש באובייקט FoundFiles כדי למנוע את הקבוצה המוחזרת בשתי הגישות.

לאובייקט FileSearch שתי שיטות : Execute ו-NewSearch . השיטה NewSearch מאפסת את כל מאפייני FileSearch ומחזירה אותו לערכי בירית המוחדר. במקרה זה תוכל לערוך את המאפיינים שנדירים בהם ערכים מיוחדים עבור חיפוש קובץ מסוים. אם לא תפעיל את השיטה NewSearch בתחלת המפרט לחיפוש, החיפוש החדש יירש את הגדרות המאפיינים של החיפוש הקודם.

באפשרותך להפעיל את השיטה Execute כדי להפעיל בחיפוש קובץ לאחר שהגדרת אותו. שיטה זאת מקבלת מספר ארגומנטים השולטים בסידור שמות הקובץ באובייקט FoundFiles, ובאפשרות לעדכו, או שלא לעדכו, את אינדקס הקבצים לפני ערכית חיפוש חדש. ערך ההחזרה בשיטה זאת הוא מספר שמות הקובץ התואמים את מפרט החיפוש.

ניהול חיפוש קובץ בסיסי

רבים ממאפייני האובייקט FileSearch מאפשרים מפרט חיפוש גמיש. הקוד שבדוגמה הפשטה הבאה מגדר חיפוש ומוחזר את הקבוצה המוחזרת שלו. הוא יוצר מופע של האובייקט באמצעות המאפיין FileSearch של האובייקט Application . לאחר מכן מוחזר את כל הגדרות המאפיין FileSearch לערכי בירית המוחדר על ידי הפעלת השיטה NewSearch . בשלב הבא הקוד מזכה את המאפיינים LookIn ו-FileName , SearchSubFolders מתקבל ערך mdb . עם שמות כמו Chapter1 ו-Chapter2 . המאפיין הבדיקה לחיפוש זה מכיל סדרה של המגדירים היכן לחפש ומה לחפש. מנגנון הבדיקה לחיפוש זה מוביל ערך Bolianei המציין אם יש להגביל את החיפוש לתיקיה הנוכחית, או להרחיב אותו LookIn . LookIn המשנה שבהגדרת המאפיין SearchSubFolders מתקבל ערך True .

```
Sub FileSearch1()
' Search in My Documents folder and its subfolders for Chapter*.mdb.
With Application.FileSearch
' Start a new search.
    .NewSearch
' Set search criteria.
    .LookIn = "C:\My Documents"
    .FileName = "Chapter*.mdb"
    .SearchSubFolders = True
End With
```

```

With Application.FileSearch
' Execute the Search.
If .Execute() > 0 Then
    MsgBox "There were " & .FoundFiles.Count & " file(s) found."
' Display names of all found files.
    For i = 1 To .FoundFiles.Count
        MsgBox .FoundFiles(i)
    Next i
Else
' If no files found, say so.
    MsgBox "There were no files found."
End If
End With
End Sub

```

לאחר שהשיגרה יוצרת מפרט לחיפוש, היא מפעילה את השיטה Execute עבור האובייקט FileSearch. לשיטה זאת ערך מוחזר המציין את מספר הקבצים העונים על התנאים שהוגדרו לחיפוש. אם הערך הוא 0, החיפוש לא הניב שמות קובץ תואמים והשיגרה תציג הודעה המציין כי לא נמצאו קבצים. אם נמצא קובץ תואם אחד, או יותר, לפי התנאים, השיגרה תציג את המאפיין Count של האובייקט FoundFiles בטרם תציג את השמות ב-FoundFiles, אחד לאחר.

מיזן הקבוצה המוחזרת

דוגמת הקוד הבאה ממיינת את הקבוצה המוחזרת של חיפוש לפי גודל קובץ. שני הפרמטרים הראשונים לשיטה Execute קובעים את תנאי המינון וסדר המינון, בהתאם. השמות הקבועים בפרמטר הראשון מציננים את המשתנה שלפיו יש למין, msoSortByFileName. קבועים אלה הם את שמות הקבצים המוחזרים. הפרמטר השני בשיטה msoSortBySize-ו- msoSortByLastModified, msoSortByFileType מצין סדר עולה או יורדת. דוגמת הקוד מגדרה חיפוש שהמינון בו ייעשה לפי גודל הקובץ, בסדר יורדת, להבדיל מהדוגמה הקודמת, שבה הוחזרו התוצאות בסדר עולה, שהוא ברירת המחדל, לפי שמות הקבצים.

```

Sub FileSearch2()
Dim sngMB As Single
' Search in My Documents folder and its subfolders for Chapter*.mdb.
With Application.FileSearch
' Start a new search.
    .NewSearch
' Set search criteria.
    .LookIn = "C:\My Documents"
    .FileName = "Chapter*.mdb"
    .SearchSubFolders = True
End With

```

```
With Application.FileSearch
```

```
' Return found files in descending order by file size.  
If .Execute(msoSortBySize, msoSortOrderDescending) > 0 Then  
    MsgBox "There were " & .FoundFiles.Count & " file(s) found."  
    For i = 1 To .FoundFiles.Count  
  
' Compute file size in MB and display with filename.  
    sngMB = FileLen(.FoundFiles(i)) / (1024 ^ 2)  
    MsgBox .FoundFiles(i) & vbCrLf & vbTab &_  
        "Filesize (MB): " & Round(CDec(sngMB), 3)  
    Next i  
  
    Else  
' If no files found, say so.  
    MsgBox "There were no files found."  
End If  
  
End With  
  
End Sub
```

תיבת ההודעה המציגת את הקבוצה המוחזרת מראה גם את גודל הקובץ ושמו. לקבלת גודלו של קובץ יש להעביר את האובייקט FoundFiles אל הפונקציה FileLen. גודלי הקבצים מעוגלים אל אלףיה המ"ב הקרובה ביותר.

הערה:



הפונקציה Round היא פונקציה חדשה בגרסה 6 VBA של Visual Basic for Applications. כדי לקבל תוצאות עיקריות עם פונקציה זאת, עליך להעביר תחילה את הארגומנט שלה אל הפונקציה CDec. בדוגמה הקודמת נעשה שימוש בתכנית כזה (פנה לפרק 1 לקבל מידע על הפונקציה CDec).

חיפוש המבוסס על תוכן הקובץ

גם כאשר מדובר בחיפוש פשוט, כמו בשתי הדוגמאות הקודומות, תוכל לחפש קטע טקסט מוגדר במסמך, או באובייקט DocumentProperty שלו. בדוגמה הבאה מתבצעת פעולה זאת. כאן משתמשים במאפיין TextOrProperty של האובייקט FileSearch כדי לסמן מחרוזת טקסט מבוקשת בגוף הקובץ או באוסף Properties של הקובץ. שם לב, שניתן לציין תיקיות במחשבים מרוחקים בעורთ המוסכמות למונן שמות אחידים (הנתיב cab\cab233\cab233) מצבעו אל התקינה cab בשם השיתוף c של מחשב ששמו .(cab233

```

Sub FileSearch3()
Dim sngStart As Double, sngEnd As Double
Dim i As Integer

' Search in cab folder on linked computer for files containing CAB.
With Application.FileSearch
' Start a new search.
    .NewSearch
' Set search criteria.
    .LookIn = "\cab233\d\cab\
    .SearchSubFolders = False
' When searching for text consider restricting the files you search. *.* takes 300
seconds, but msoFileTypeWordDocuments takes 22 seconds.
    .FileName = "*.*"
    .FileType = msoFileTypeWordDocuments
    .TextOrProperty = "CAB"
End With

With Application.FileSearch
' Execute the search.
    sngStart = Now
    If .Execute() > 0 Then
        sngEnd = Now
        Debug.Print DateDiff("s", sngStart, sngEnd)
        MsgBox "There were " & .FoundFiles.Count & " file(s) found."
' Display names of all found files.
    For i = 1 To .FoundFiles.Count
        MsgBox .FoundFiles(i)
    Next I
    Else
' If no files found, say so.
        MsgBox "There were no files found."
    End If
End With

End Sub

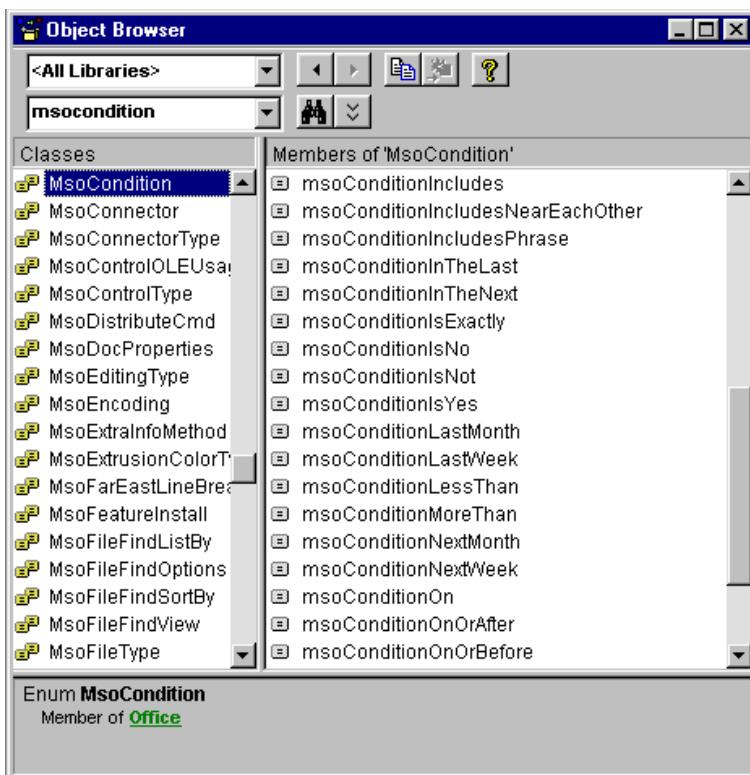
```

חישוב קבצים עלול להימשך זמן רב. תוכל לשפר באורך דרמטי את מהירות הפעולה של השיטה Execute על ידי ציון מאפיין `msoFileTypeWordDocuments`. דוגמת הקוד הקודמת מארת את כל מסמכי Word בתיקיה נתונה, אשר מכילים מחרוזת טקסט מסוימת. השימוש בקבוע `msoFileTypeWordDocuments` במאפיין `FileType` מגביל את החישוב שבדוגמה לקבצים המכילים מסמכי Word בלבד. אל תתפתה לציין `*.*` עבור המאפיין `FileName`, מכיון כוונה לסנן את התוצאות המוחזרות, משום שדבר זה יפגע

מאוד במהירות הביצוע. בדוגמה הקודמת, המתייחסת לקבצי התקינה cab שבמחשב cab233, הבדל במהירות הפעולה הוא 22 שניות כאשר מופעל הקבוע *.msoFileType. לעומת זאת, המוחזרת באובייקט FoundFiles. עליך להשתמש בשיטה Add פעמיים, או יותר, כדי לצלין קרייטריונים מרובים עבור האוסף PropertyTests של כל קרייטריוון Condition Name וCondition .

הכללת קרייטרויי חיפוש מרובים

מבנה החיפוש המתאים הכללה של קרייטרויי חיפוש מרובים לקובוצה המוחזרת באובייקט FoundFiles. עליך להשתמש בשיטה Add פעמיים, או יותר, כדי לצלין קרייטריונים מרובים עבור האוסף PropertyTests של כל קרייטריוון Condition Name וCondition .



תרשים 8.2: השימוש בחברים של קבוצת הספירה msoCondition כדי להגדיר תנאים לקרייטריונים מתאימים באוסף PropertyTests של האובייקט FileSearch. השיטה Add יכולה לכלול גם הגדרת Connector, וכן הגדרת Value אחת, או שתים. כדי לסקור את האפשרויות הזמינים, عليك להציג את איברי המחלקה msoCondition (איור 8.2 מראה קטע כזה). ההגדרות בConnector-Value עשויה לקבל ערך אחד, או שניים, שיצינו כיצד לשלב תנאי נתון עם תנאים נוספים. הגדרה זאת מפעילה את

האופרטורים And או Or המשמשים לミזוג של קרייטריוון מסוימים עם קרייטריווני חיפוש נוספים. השתמש באופרטור Or לטיפול בקרייטריווני החיפוש בנפרד, ובאופרטור And כדי לשלב את הקרייטריוון האמור עם קרייטריוונים אחרים. And הוא ברירת המחדל להגדרת אופרטור. ייחדיו, ההגדרות Connector ,Condition Value ו-Condition MassPackות פונקציונליות דומות לזו של תיבת הדוח-שיכון חיפוש (Find).

תוכל למנוע את חברי האוסף PropertyTests בעזרת LOLATAT For...Each. כל חבר מהווע קרייטריוון חיפוש ייחודי. המאפיין Name מזוהה את הקרייטריוון בשעת מניעת חברי.

בדוגמה הבאה, שהיא האחורה לאובייקט FileSearch, שלושה מקטעים. במקטע הראשון מצוינים הקרייטריוונים, לאחר שהוגדרה הפניה לאובייקט FileSearch. היעד בדוגמה הוא כל קבץ מסד הנתונים בין שני תאריכים. הדוגמה מציגה את התchapir הנכון להפעלת השיטה Add עבור האוסף הראשון קובע את סוג מסד הנתונים. הקרייטריוון השני מציין קבצים שהשינוי האחרון בהם נערך בין 1 בינוואר 1996 ל- 30 ביוני 1999. ההגדירה msoConnectorOr מציינת שעל הקבצים לקיים את שני הקרייטריוונים בנפרד כדי להיכلل בקובצת המוחזרת. אין צורך לציין מאפיין Connector בקרייטריוון השני, לאחר שהAKERTRIOON השני יאמץ את ערך ברירת המחדל msoConnectorAnd. לפני הצגת הקובצת המוחזרת, השגירה מונה את חברי PropertyTests במקטע השני שלו. המקטע השלישי מציג את הקובצת המוחזרת.

```
Sub Search4()
Dim fs As FileSearch, mystring As String
Dim i As Integer
    Set fs = Application.FileSearch

' Set lookin and subfolder properties.
With fs
    .NewSearch
    .LookIn = "c:\My Documents"
    .SearchSubFolders = False
End With

' Set a pair of property conditions.
With fs.PropertyTests
    .Add name:="Files of Type", _
        Condition:=msoConditionFileTypeDatabases, _
        Connector:=msoConnectorOr
    .Add name:="Last Modified", _
        Condition:=msoConditionAnytimeBetween, _
        Value:="1/1/1996", SecondValue:="6/30/1999"
End With
```

```

' Display property tests.
For i = 1 To fs.PropertyTests.Count
    With Application.FileSearch.PropertyTests(i)
        mystring = "This is the search criteria: " _
            & " The name is: " & .name & ". The condition is: " & .Condition
        If .Value <> "" Then
            mystring = mystring & ". The value is: " & .Value
        If .SecondValue <> "" Then
            mystring = mystring _
                & ". The second value is: " _
                & .SecondValue & ", and the connector is" _
                & .Connector
        End If
    End If
    MsgBox mystring
End With
Next i

' Display return set from property tests.
With fs

' Execute the search.
If .Execute() > 0 Then
    MsgBox "There were " & .FoundFiles.Count & " file(s) found."
' Display names of all found files.
For i = 1 To .FoundFiles.Count
    MsgBox .FoundFiles(i)
    Next i
Else
' If no files found, say so.
    MsgBox "There were no files found."
End If
End With

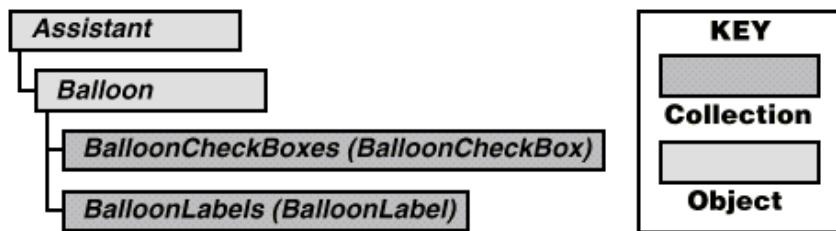
End Sub

```

האובייקט Assistant

המיסייע של Office הוא תוכנית עזרה יידידותית במסגרת Office. אף ששמפתחים ומשתמשים יכבדים אולי איןום אהובים את המיסייע, בעיני המשתמש הטיפוסי יש בו יתרונות רבים. קל למדи לתכנת את המיסייע, שכן תוכל בקלות להעניק לישומי Office המותאמים אישית שלך חזות ותחושה דומה לזה של יישומי Office הסטנדרטיים.

מבנה האובייקט Assistant מוצג בתרשים 3.8. בrama העלiona, עשוי אובייקט Assistant על המסך עם האובייקט Balloon, התלויה בו מבחינה היררכית, או בלבדו. הוואיל והמסיע משמש במגוון רחב של דמיות והנפשות, הוא עשוי לדבר ולהציג מידע גם ללא הסברים. אם ברצונך לכלול הסברים, תוכל לתוכנת אובייקטים מסווג Balloon כך שיופיעו בצדם למשיע. בלוניים עשויים לכלול הסברים, או אפילו לשמש כמנגנון פשוט להזנת נתונים. השתמש באוסףים BalloonCheckBox ו-BalloonLabel עם BalloonLabel פקודה כדי להפוך את המסייעים לאינטראקטיביים.



תרשים 3.8: השתמש באובייקט Assistant להציג והנפשה של מסיע, ולהציג בלון

מסיעים

massuiems עשוים להביא תועלות רבה ביישומים על ידי הצגה חזותית של הפעולות השונות שהיישום מבצע. תוכל להוסיף נוף ייחודי ליישום המותאמים אישית על ידי שימוש עקבי במסיע מסויים, או על ידי שימוש באוסף של מסיעים שונים בנסיבות שונות. לדוגמה, השתמש במסיע הגאון (Genius) להציג עזרה הקשורה בחישובים, ולעומת זאת, כאשר מדובר בעזרה בנוסאים הקשורים במערכות מחשבים, השתמש במסיע הנראה כרובוט (F1). צוות הפיתוח שלך יוכל לקבוע עצמו כלליים בדבר הפעלת הנפשה מסוימת לכל פעולה או משימה האורכות מעל שתי שניות.

הציג והנפשה של מסיע

תוכל לשנות בקלות בתצוגה ובהנפשה של מסיע באמצעות שלוש השגרות הקצרות שנציג להלן : AssistantSearchOn ,AssistantIdleOn ,AssistantNotVisible . השיגרה IdleOn מכילה שורה אחת בלבד, המגדירה למאפיין Visible (גלו) של האובייקט Assistant ערך True . והואיל ו- msоАimationIdle הוא ברירת המחדל לסוג הנפשה (דמות ההנפשה אינה פעילה), השורה המגדירה את המאפיין Animation אינה נחוצה ולכן היא מופיעה רק כהערה. נוסף לכך, הנפשה בסיסית זאת היא חלק מספריית הליבה של Access, יכול תoggle להפעלה ללא הפניה אל הספרייה 9 Microsoft Office . תוכל להפעיל גם את השיגרה AssistantNotVisible ללא הפניה לספריית האובייקטים של Office .

```

Sub AssistantIdleOn()

' Setting animation for idle is optional.
' Assistant.Animation = msoAnimationIdle
    Assistant.Visible = True
End Sub

Sub AssistantNotVisible()
    Assistant.Visible = False
End Sub

Sub AssistantSearchOn()
    If Assistant.Visible = False Then Assistant.Visible = True
    Assistant.Animation = msoAnimationSearching
End Sub

```

כדי להחליף מצב 'לא פעיל' של דמות הנקפה במצב אחר כלשהו, חובה להגדיר במסגרת השירה הנה את המאפיין Animation של האובייקט, והן את המאפיין Visible. השירה AssistantSearchOn גורמת לאובייקט AssistantSearch להציג את הנפשה החיפוש. שלא כמו הנפשות רבות אחרות, הנפשה החיפוש תחזור על עצמה עד שתתקצה הגדרה חדשה למאפיין Animation. בנוסף לכך, חובה לכלול בפרויקט הפניה לספריית האובייקטים של Office כדי שהנפשה תופיע (ניתן לעשות זאת באמצעות ידני, באמצעות התפריט **Tools** (כלים) ב- Visual Basic Editor).

הפניה אוטומטית לספריות האובייקטים של Office

אחד החסרונות של השירה AssistantSearchOn הוא, שהשירה מניחה שיש הפניה בספריית האובייקטים של Office. אם אין הפניה כזאת, השירה נכשלת ללא כל סימן. המסיע מופיע, אולם הוא מציג הנפשה לא פעילה במקומ הנפשה חיפוש. אחד הדריכים לפטור בעיה זאת היא לוודא שיש הפניה בספריית האובייקטים של Office, ולהוציא הפניה אוטומטית במידת הצורך. אם תעשה כך, יוכל היישום להפעיל בבטחה כל הנפשה, ולמקרה כל מאפיין אחר של אובייקט Office משותף. השגורות ReferenceOffice ו-AssistantSearchOn2 שלහן מציגות פתרון זה.

```

Sub AssistantSearchOn2()
    ReferenceOffice
    AssistantSearchOn
End Sub

```

```

Sub ReferenceOffice()
Dim ref1 As Reference
Dim blnOffice9In As Boolean, mso9Library As String
' Enumerate members of References collection to determine
' whether Office Object Library is already referenced.
    blnOffice9In = False
    For Each ref1 In References
        If ref1.name = "Office" Then
            blnOffice9In = True
        End If
    Next ref1

' If Office Object Library reference is missing, reference it.
    If blnOffice9In = False Then
        mso9Library = "C:\program files\Microsoft Office\Office\mso9.dll"
        Application.References.AddFromFile mso9Library
    End If

End Sub

```

השיגרה AssistantSearchOn2 זהה כמעט לchlוטין לשיגרה המקורית להפעלת הנפשות חיפוש. למעשה, היא אכן קוראת לשיגרה AssistantSearchOn, אולם היא מפעילה תחיליה את ReferenceOffice. השיגרה ReferenceOffice מודדת שקיימות בפרויקט הנוכחי הפניה בספריות האובייקטים של Office. היא פותחת במנייה כל חברי האוסף References כדי לבדוק אם המאפיין Name באחד מהם הוא Office. אם כן, השיגרה מגדרה לשנתנה בוליאני ערך True. אם לא, הגדרת המשתנה blnOffice9In נותרת False, לפי ברירת המחדל. המקטע השני של ReferenceOffice יוצר הפניה בספריית האובייקטים של Office כאשר ערך המשתנה הבוליאני הוא False. לאחר שוודאה שקיימת הפניה בספריית האובייקטים של Office, יכולת השיגרה AssistantSearchOn2, יכולת השיגרה AssistantSearchOn וلسירם בכך את הפעלת הנפשות.

הציג הנפשות חיפוש

ניתן להשתמש באובייקט Assistant FileSearch להשלמת פעולה האובייקט. כבר הזכרנו קודם כי בחלק מהmakrims, חיפוש הוא תהליך העשייל לאורך זמן-מה. במקרה זה, רצוי להציג רמזו כלשהו על המסך, כך שהמשתמש יידע שהיישום אכן עוסק בעולה כלשהי. הנפשות החיפוש של המסייע מיטיבה לשרת מטרה זאת.

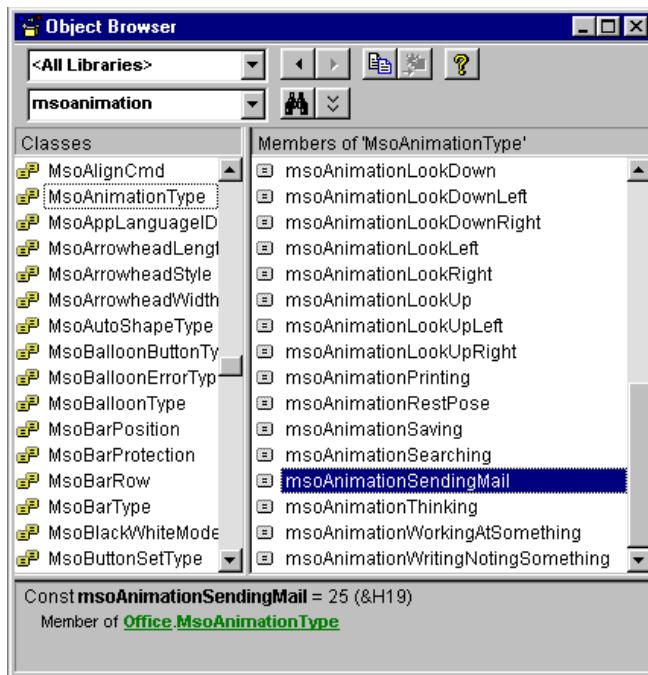
השיגרה הבאה מוחפשת קבצי mdb. בכל רחבי כוון C ומחזירה את מנתין הקבצים שאיתריה. באמצעות הנפשות חיפוש המוצגת מרגע שמתחיל החיפוש והחזרת ההנפשה הלא-פעילה מיד עם גמר החיפוש, המסייע מראה למשתמש متى החיפוש בעיצומו ומתו הוא מסתיים. לשם כך נעשה שימוש בשתי קריאות לשגורות – אחת לפני הפעלת השיטה Execute, והשנייה מיד לאחר סיומה. הויל ו熬 האובייקטים

ו- Assistant תלויים בספריית האובייקטים של Office, השגירה מפעילה בשלב הראשוני את השגירה ReferenceOffice. אם לא קיימת הפניה בספרייה האובייקטים של Office, השגירה יוצרת אותה. ללא אמצעי זהירות זה, השגירה תיכשל אם מסיבה כלשהי בוטלה הפניה בספרייה האובייקטים של Office.

```
Sub FileSearchAct()
' Reference the Office Object Library before using
' either the FileSearch or the Assistant objects.
    ReferenceOffice
' Search on C drive and its subfolders for *.mdb.
    With Application.FileSearch
' Start a new search.
        .NewSearch
' Set search criteria.
        .LookIn = "C:\""
        .SearchSubFolders = True
        .FileName = "*.mdb"
    End With
    With Application.FileSearch
' Execute the search.
' Turn searching assistant on first.
        AssistantSearchOn
        If .Execute() > 0 Then
            AssistantIdleOn
            MsgBox "There were " & .FoundFiles.Count & " file(s) found."
        Else
' If no files found, say so.
            MsgBox "There were no files found."
        End If
    End With
End Sub
```

בחירה סוג הנפשה

ספריית האובייקטים של Office מכילה מעל 30 סוגי שונים של הנפשות. תרשימים 8.4 מציג קבוצת משנה של סוגי הנפשה כפי שהוא נראה דרך **Object Browser** (סורק האובייקטים). רוב ההנפשות, כמו למשל msoAnimationSendingMail ו- msoAnimationPrinting מביצעות מהזור הנפשה אחד וחזרות לנצח לא-פעיל. אחרות, כמו למשל msoAnimationThinking ו- msoAnimationSearching, הזרות על עצמן עד שימושה הנפשה אחרת. בזכות IntelliSense, אין צורך את השמות הקבועים כדי להפנות להנפשות. פעמים רבות, עלייך רק לבחור בהנפשה הרצוייה מתוך רשימה.



תרשים 8.4: החבר `msoAnimationType` בספריית האובייקטים של Office מכיל מעל 30 קבוצות הנפשה

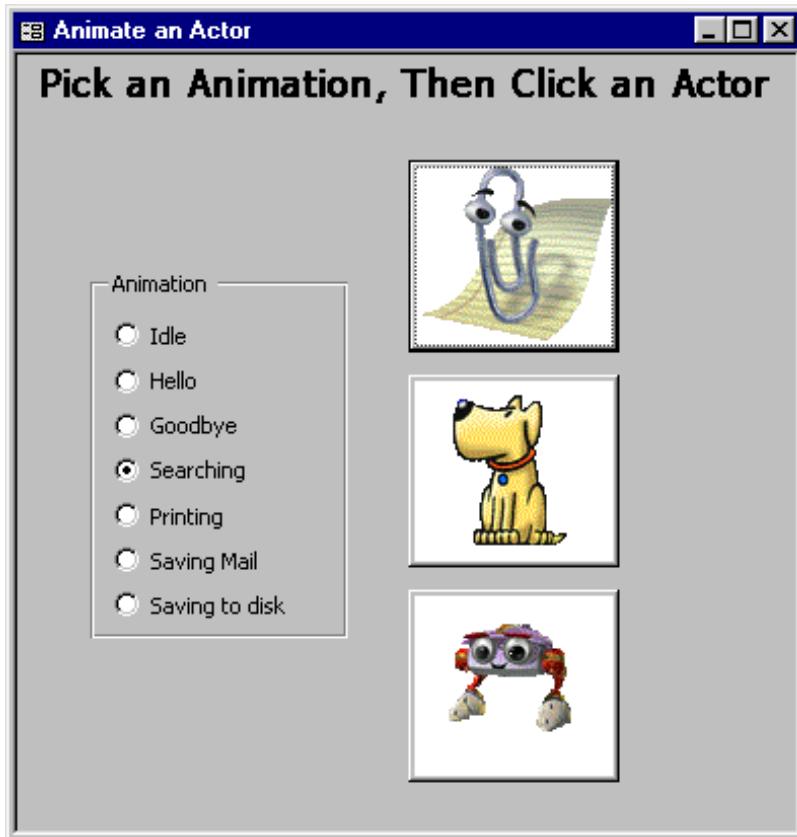
בחירה דמות למסיע

תוכנת Office 2000 משווקת עם שמונה דמויות שונות למסיע, אולס בהחלט ייתכן לחברת Microsoft תצרף דמויות נוספות בעתיד. הצגת הדמות הרצויה של המסיע נעשית על ידי קביעת שם הקובץ המתאים כהגדרת המאפיין `FileName` במסיע. הטבלה הבאה מציגה את שמות הדמויות השונות של המסיע, לצד שמות הקובץ המתאים.

שם הקובץ	דמות המסיע
Clippit.acs	(Clippit) המהדק
OffCat.acs	(Links) לינקס
Rocky.acs	(Rocky) רוקי
Logo.acs	(Office Logo) Office מל
Dot.acs	(The Dot) הנקודה
Mnature.acs	(Mother Nature) אמא אדמה
Genius.acs	(The Genius) הגאון
F1.acs	F1

תצוגה מקדימה של הנפשות במחשב

תרשים 8.5 מציג טופס שבו מופעלות הנפשות של המסייעים. ניתן להציג אותה הנפשה בכל מופיע, או ליצור הבדלים בין מופיע למופיע. לדוגמה, קיימות לפחות גירסאות שונות של הנפשות 'להתראות' עבור הרובוט ששמו F1. הטופס מציג שבעה סוגים הנפשה עבור כל אחד משולשות המסייעים. השתמש בטופס לתצוגה מקדימה של הנפשות על ידי בירה מתוך קבוצת אפשרויות הנפשה, ולאחר מכן לחיצה על אחד מהחצני הפוקודת של המסייעים. ניתן של הקוד המפעיל את הטופס יראה לך כיצד בשלב הנפשות ולהחליפם מסוימים בישומך.



תרשים 8.5: השתמש בטופס זה לייצרת תצוגה מקדימה של הנפשות המסייע.

תרשים 8.6 שלහן מציג מבחר סוגים של מסיעים והנפשות שניתן ליצור בתצוגה מקדימה בעזרת הטופס שבתרשים 8.5. בקצת הימני, הרובוט F1 מבצע את הנפשות 'להתראות' על ידי הילומות בתוך כדור אש. הדמות המהדק (Clippit) מגלאת פיסת נייר ומשתמש בה להציג הנפשות החיפוש. בהנפשה 'שמירה בדיסקט', דמות הכלב רוקי (Rocky) מציגה לראווה דיסקט ולאחר מכן מניחה אותו לשמירה בתוך כיס בקורס. בהנפשה ההדפסה, הרובוט F1 הופך למדפסת.



תרשים 8.8: מבחר סוג הנטשה שմבוצעים שלושה מסוימים שונים. הנפשות אלה הופקו בעזרת הטופס המוצג בתרשים 8.5.

דוגמת הקוד הבאה מציגה את הקוד שמאחורי הטופס המוצג בתרשים 8.5. אזכור החצורה שבראש המודול מכיל את הגדרות זמן-העיצוב של לחצני הפוקודה המראים את סוגים מסוימים, ואת הלחצנים של סוג הנטשה השונים בקבוצת האפשרויות. שים לב שימוש הקבצים המתאימים למסויים הנראים על פני הלחצנים מאוחסנים במאפיין Tag (טג) שלהם. לכל לחץ גם שורת אירוע פשוטה לאירוע הלחיצה המתאים לו. לכל אחד מהלחצנים שבקבוצת האפשרויות ערך המצוין קבוע msoAnimationType של קבוע נתון (לדוגמה, הערך 25 מייצג את שם הקבוע msoAnimationsSendingMail). כאשר הטופס נטען, הוא מגידר סדרה של מאפיינים כדי להבטיח שהמסויים יהיה מוכן בעת שהמשתמש בוחר בלחץ מתוך קבוצת האפשרויות ולוחץ על אחד מהלחצני הפוקודה.

```
Option Compare Database
'
' Design-time Command button settings
' Name: cmdClippit , cmdRocky, cmdF1
' Picture: c:\My Documents\My Pictures\clippit.bmp,
'          or rocky.bmp, or F1.bmp
' Picture Type: Embedded
' Tag: clippit.acs, or rocky.acs, or F1.acs
' On Click: [Event Procedure]
'
' Design-time Option button settings
' Idle button's Option Value = 1
' Hello button's Option Value = 2
' Goodbye button's Option Value = 3
' Searching button's Option Value = 13
' Printing button's Option Value = 18
' Saving Mail button's Option Value = 25
' Saving to disk button's Option Value = 112
'
Private Sub Form_Load()
    With Assistant
        .On = True
        .Sounds = True
        .Visible = True
    End With
End Sub
```

```

Private Sub cmdClippit_Click()
    AnimateActor Me.Controls("cmdClippit").Tag
End Sub

Private Sub cmdRocky_Click()
    AnimateActor Me.Controls("cmdRocky").Tag
End Sub

Private Sub cmdF1_Click()
    AnimateActor Me.Controls("cmdF1").Tag
End Sub

Sub AnimateActor(Fname As String)
    With Assistant
        .FileName = Fname
        .Animation =
            Me.Controls("optAnimation")
        .Visible = True
    End With
End Sub

```

לחצני הפקודה הם גורם מרכזי בתהליך. כאשר לוחצים על לחץ פקודה, שגרת האירוע מעבירה את ערך התג שלו לשיגרה של AnimateActor. שיגרה זאת משתמש בארגומנט המועבר שלא להגדרת המאפיין FileName של המופיע. לאחר מכן היא מגדרה למאפיין Visible של המופיע ערך True ולמאפיין Animation את ערך קבועה האפשרויות. ערך קבועה האפשרויות הוא פונקציה של הלחץ שנבחרת. ערך ברירת המחדל של הפקד הוא 1, במקרה שהמשתמש פ Sach על בחירת הלחץ.

בלוניים

באמצעות בלוניים, תוכל להציג טקסט או גרפיקה, או לאסוף משוב מהמשתמש. ניתן להציג בלוניים כתיבת דו-שיח מודאלית, נטולות מודאליות, או נסגרת-אוטומטית. המאפיין Mode של הבלון קובע את סוג הבלון. ההגדרת msoModeAutoDown סוגרת את הבלון בעקבות לחיצה שלוחץ המשתמש בכל נקודה במשך. ההגדרת נטולת המודאליות (msoModeModal) והמודאליות (msoModeModeless) שכיחות יותר. ההגדרת נטולת המודאליות גורמת לבلون להישאר פתוחה בשעה שהמשתמש מבצע פעולות מחוץ לגבולותיו. ההגדרת המודאלית כופיה בגובה בטרם יכול המשתמש להמשיך בכל פעולה אחרת. ערך ברירת המחדל למאפיין Mode הוא msoModeModal.

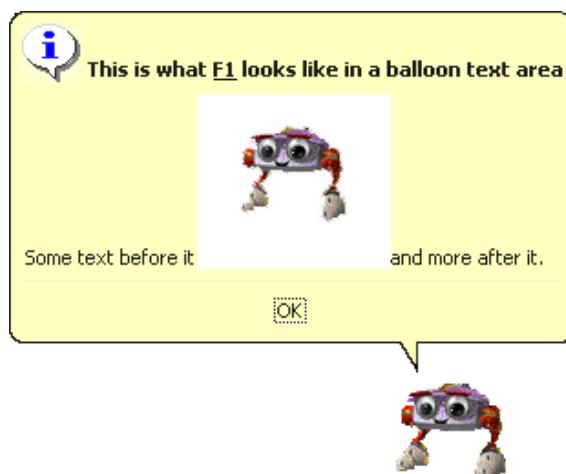
השתמש במאפיין NewBalloon של האובייקט **Assistant** לייצירת אובייקט Balloon. לכל בלון אזורים נפרדים לכותרת, טקסט, תווית, תיבת סימון ולחצנים. תוכל לאכלס אזורים אלה בעזרת הגדרות המאפיין המתאימות, או באמצעות אובייקטים היררכיים כמו BalloonLabels ו-BalloonCheckboxes. הקצתה טקסט לתיבות הסימון ולתוויות

נעשה באמצעות המאפיין **Text** שלו. תוכל לבצע התאמה אישית נוספת בבלון באמצעות המאפיין **Icon** שלו. ישנו שיטה סמלים שונים לשישה סוגים שונים של מאפייני הودעה, כמו למשל התראה, שאלה, מידע ועזה.

גרפיקה בבלונים

השיגרה הבאה מציגה בלון עם כותרת, טקסט וסמל. כמו כן נעשה בה שימוש בלחצן ברירת המחדל (**OK**). מחרוזות עם עיצוב מילוט (escape) שנמצאות בטקסט המאפיין **Heading** (כותרת) מציניות את תחילתו ({1} {n}) וסופה ({n} {0}) של טקסט המעובד בקו-תחתון באזורי הכותרת (ניתן להשתמש במחרוזות עם עיצוב מילוט גם לצורך עיצוב טקסט בקו-תחתון באזורי הטקסט). הגדרת המאפיין **Text** כוללת תמונה מפתח-סיביות באזורי הטקסט.שים לב שניתן לבצע גישת טקסט מסביב לתמונה. הגדרת המאפיין **Icon** (סמל) מצינית שהבלון מכיל מידע. לבסוף, השיטה **Show** פותחת את המסייע ואת הבלון הקשור אליו. תרשימים 8.7 מציג את המסייע והבלון המופיעים עם הפעלת השיגרה.

```
Sub balloonTextImageIcon()
    With Assistant.NewBalloon
        .Heading = "This is what {ul 1}F1{ul 0} looks like in a balloon text area"
        .Text = " Some text before it " & _
            "{bmp ""C:\My Documents\My Pictures\F1.bmp""} " & _
            "and more after it."
        .Icon = msoIconAlertInfo
        .Show
    End With
End Sub
```



תרשים 8.7: ניתן להוסיף גרפיקה לבלון, ולהגדיר את מהות התוכולה באמצעות הגדרות המאפיין **Icon** (כמו למשל הגדרת הסמל **AlertInfo** המשמשת בבלון שבתרשים זה).

תוויות של בלוניים

שתי השגרות שלහן מכינות בלון עם תוויות, אך לא לחצנים. השיגורה setLabelCount מעבירה ארגומנט לשיגורה balloonHeadTextLabel. עבור ארגומנטים חוקיים מ-1 ועד 5, כולל, השיגורה המופעלת מציגה מסיע עם בלון המכיל תוויות כמספר הערך שהועבר לשיגורה. כל תוויות מופיעה עם מחרוזת טקסט פשוטה המכילה על מספר התווית. בעיצוב זה אין צורך בלחצנים, הואיל ולחיצה על אחת התוויות גורמת לסגירת הבלון. השיטה Show פותחת את הבלון ומעבירה את ערך התווית שהמשתמש לחץ עלייה למשתנה i שהוא מסוג Integer. תיבת הودעה מציגה את מספר התווית שהמשתמש לחץ עלייה כדי לסגור את הבלון.

```
Sub setLabelCount()
    balloonHeadTextLabel 5
End Sub

Sub balloonHeadTextLabel(LabelCount As Integer)
On Error GoTo LabelTrap
Dim b1 As balloon, i As Integer

' Check for 0 or negative label count.
If LabelCount <= 0 Then
    Err.Raise 9
End If
Set b1 = Assistant.NewBalloon

' Create balloon with specified number of labels.
With b1
    .Heading = "This is my heading"
    .Text = "Select one of these things:"
    For i = 1 To LabelCount
        .Labels(i).Text = "Label " & i
    Next i
    .Button = msoButtonSetNone
    i = .Show
End With

' Confirm label that user clicked.
MsgBox i, vbInformation, "Programming Microsoft Access 2000"

LabelExit:
    Exit Sub
```

```

LabelTrap:
If Err.Number = 9 Then
    MsgBox "Number of labels more than 5 or less than " & _
        "1. Retry with another number.", vbInformation, _
        "Programming Microsoft Access 2000"
Else
    Debug.Print Err.Number
    MsgBox "Unanticipated error. Error number = " & _
        Err.Number & vbCrLf & "Error Description: " & _
        Err.Description, vbInformation, _
        "Programming Microsoft Access 2000"
End If

Resume LabelExit

End Sub

```

גילוי ארגומנטים לא חוקיים למספר התוויות נעשה באמצעות לוגיקה מוכפלת ומותאמת אישית ליכולת שגיאות. אם הארגומנט המועבר גדול מ-5, הולאה Next...For...המקצתה ערכי תוויות תיכשל בניסיונה להקצחות טקסט לתוויות השישית. Shgiah זהה (שgiaה מס' 9) נובעת מכך שבולונים יכולים להציג חמש תוויות לכל היותר. ארגומנט בעל ערך אפס או ערך שלילי המועבר לשגירה לא יחולל שgiaה מסווג זה, אולם העברתו לא תאלץ את הבלון בתוויות. לכן, השגירה balloonHeadTextLabel לוכדת ערכים אלה בעורת משפט If...Then, ויזכרת שgiaה מותאמת אישית שמספרה 9. המלצות שמכילה השגירה לשgiaה זאת מורה למשתמש להזין ערכים בטוח שבין 5-1 בלבד.

איסוף קלט מהמשתמש בעזרת בלוניים

הוסף BalloonCheckboxes מאפשר הצגה של קבוצת אפשרויות ואיסוף תשובות מהמשתמש. גם כאן הנהן מוגבל לחמשה פקדים – במקרה זה הפקדים הם תיבות סימון. שלא כמו פקדי תווית, פקדי תיבת סימון אינם סוגרים את הבלון עם בחרתם. לכן, אם ציינת בלון עם תיבות סימון, עליך להקצחות ערכות לחצים לסגירת הבלון.

שתי השגורות הבאות מציגות מסגרת כללית להציג פקדי תיבת סימון בתוך בלון. הן אינן כוללות לכידת שגיאות, אולם בפועל, עליך לכלול לוגיקה לטיפול בשגיאות כמו זאת שבדוגמה הקודמת. השגירה setCheckCount מעבירה ארגומנט בטוח שבין 1 ל-5 כדי לציין כמה תיבות סימון יש לכלול בבלון. השגירה השניה נפתחת בהגדרת הפניה 1, בלון חדש. היא מקצתה למאפיין Button של האובייקט Balloon את ההגדרה MsoButtonSetOkCancel כדי לכלול בבלון לחץ אישור (OK) וביטול (Cancel). דבר זה מאפשר סגירה של הבלון בשתי דרכיהם. לאחר הגדרת הערכים עבור הכתובת והטקסט Show בבלון, השגירה מקצתה טקסט לכל אחת מתיבות הסימון. ערך החזרה המשיטה במקורה זה מציין את הלחן שעליו לחץ המשתמש.

```

Sub setCheckCount()
    balloonHeadTextCheck 5
End Sub

Sub balloonHeadTextCheck(CheckCount As Integer)
Dim b1 As balloon, i As Integer
Dim strChecks As String, iChecks As Integer

' Set reference to the balloon.
Set b1 = Assistant.NewBalloon

' Assign text to check box controls.
With b1
    .Button = msoButtonSetOkCancel
    .Heading = "Here's the Heading"
    .Text = "This is some text in the balloon."
    For i = 1 To CheckCount
        .Checkboxes(i).Text = "Check box # " & i
    Next
    i = .Show
End With

' Did user click Cancel button?
If i = msoBalloonButtonCancel Then
    MsgBox "You cancelled the balloon.", vbInformation, _
        "Programming Microsoft Access 2000"
    Exit Sub
End If

' Record individual boxes checked and count of all checked boxes.
For i = 1 To CheckCount
    If b1.Checkboxes(i).Checked = True Then
        If strCheck = "" Then
            strCheck = CStr(i)
        Else
            strCheck = strCheck & ", " & CStr(i)
        End If
        iChecks = iChecks + 1
    End If
Next i

```

```

' Present a message box with the results.

If iChecks = 0 Then
    MsgBox "No boxes checked.", vbInformation, _
        "Programming Microsoft Access 2000"
ElseIf iChecks = 1 Then
    MsgBox "You checked box " & strCheck & ".", _
        vbInformation, "Programming Microsoft Access 2000"
Else
    MsgBox "You checked " & iChecks & " boxes. " & _
        "These were boxes: " & strCheck & ".", _
        vbInformation, "Programming Microsoft Access 2000"
End If
End Sub

```

שארית השירה מעבדת את התגובה לבלוון. תחילה, השירה בודקת את ערך החזרה מהשיטה Show. אם ערך זה שווה לקבוע msoBalloonButtonCancel, המשמש לחץ על הלחצן ביטול. לאחר הצגת הודעה המתאימה על כן, השירה מסתיימת.

הויל והדרך היחידה לסגור את הבלון היא ללחוץ על הלחצן אישור או ביטול, אם המשמש לא לחץ על ביטול, מן הסתם לחץ על אישור. קטע הקוד הבא מבצע שתי משימות: הוא מפתח מחרוזת טקסט שמצוינת תיבות מסוימות, ולאחר מכן מונח את התיבות הללו. קטע הבלוק האחרון בשירה משתמש במחרוזות ובתוצאת המספרה להכנת תיבת הودעה שבה ייאמר, בהתאם לתוצאה, שאף תיבה לא סומנה, שסומנה תיבה אחת בלבד, או שסומנו מספר תיבות.

בלונים לא מודאליים

הדוגמה הבאה מעבדת בלוניים נטולי מודאליות, שנשאים פתחים בשעה שהמשמש מבצע פעולה כלשהו בחלק אחר של היישום. כל הדוגמאות הקודמות עסקו בבלוניים מודאליים. אחת התכונות המותאמות אישית לעובודה עם בלוניים נטולי מודאליות היא המאפיין Callback (התקשורת חוזרת). מאפיין זה מקבל כערך את שמה של שירה אחרת המבצעת פעולה כלשהו עם התגובה לבלוון. על השירה הראשון כוללת משתנה המיציג את הפניות הבלון. הfrmtr הראשון ציינו הפך שהמשמש לחץ עליו. הfrmtr השלישי משתמש במשתנה ארוך המזיהה את המאפיין Private של הבלון המפעיל את שגרת ההתקשרות חוזרת. בבלון נטול מודאליות, חשוב במיוחד ליצור פקד לשגירה. אם אין מציג פקד כזה, הבלון עלול להישאר פתוח ללא הגבלת זמן.

השירה שלහן מציגה בלון עם ארבע תוויות.שים לב, שהגדרת המאפיין Mode של הבלון היא msoModeless. הגדרה זאת מחייבת הגדרה של המאפיין Callback.

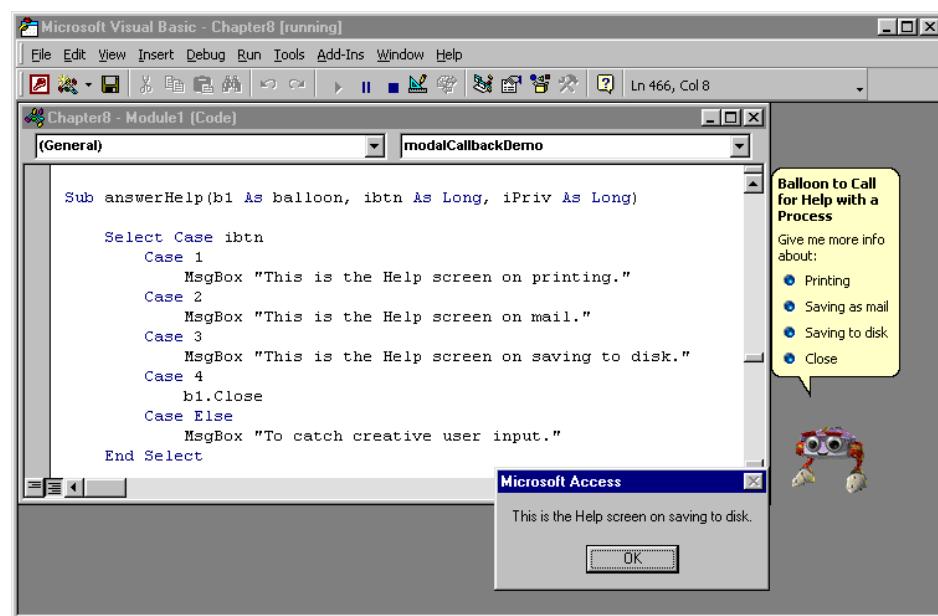
לא הגדרה כזו, השירה תחולל שגיאה. שים לב גם, שאחת התוויות מצוינה

במפורש אפשרות סגירה. ניתן להציג אפשרות סגירה גם באמצעות הגדרה של המאפיין Button. לחיצה על תווית לא תגרום באופן אוטומטי לסגירת הבלון; היא תפעיל את שגרת ההתקשרות חזרה הקרויה answerHelp. שגירה זאת מוצגת בתרשים 8.8, להלן.

```
Sub modalCallbackDemo()
Dim b1 As balloon
Set b1 = Assistant.NewBalloon

With b1
    .Heading = "Balloon to Call for Help with a Process"
    .Text = "Give me more info about:"
    .Labels(1).Text = "Printing"
    .Labels(2).Text = "Saving as mail"
    .Labels(3).Text = "Saving to disk"
    .Button = msoButtonSetNone
    .Mode = msoModeModeless
    .Callback = "answerHelp"
    .Show
End With

End Sub
```

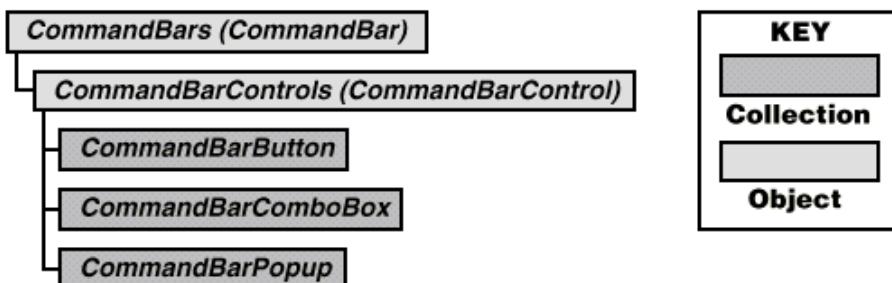


תרשים 8.8: בלון נטול מודאליות ושגרת ההתקשרות חזרה שלו.

השיגרה `answerHelp` מציגה תיבת הودעה שתכולתה נקבעת על ידי התווית שהמשתמש לחץ עליה בבלון. המשפט `Select Case` המעבד את התגובה קורא את התווית שבחר המשמש מתוך הfrmटר השני המועבר לשגרת ההתקשרות חזרה. לאחר שהמשתמש סוגר את תיבת ההודעה, הבלון נשאר פתוח. בכלל, יכול המשמש להמשך ולנוט בטוק טופס, בעוד הבלון נטול המודאליות מציג עזירה על אופן התגובה לשדות השונים בטופס. אם המשמש ילחץ על החלוץ הרביעי, השיגרה תשמש בהפניה `b1` לבלון `.Close` להפעלת השיטה `Close`.

האובייקט CommandBar

מבנה האובייקט `CommandBar` (המוצג בתרשימים 8.9) עשיר ביותר – הוא כולל סרגלי כלים מוכללים ומותאמים אישית כאחד. **Command Bar** (סרגל פקודות) הוא מונה כללי המציג שורת תפריטים, סרגל כלים או שורת תפריטים מוקפצת. פקדיו `CommandBar` מאפשרים למשתמש לפעול עם יישום נתון, באמצעות משק לסרגלי פקודות של היישום. פקדיו `CommandBar` נחלקים לשושן מחלקות אובייקטים רחבות: `CommandBarPopup`, `CommandBarComboBox`, `CommandBarButton`.



תרשים 8.9: השימוש במבנה האובייקט `CommandBar` להטאה אישית של סרגלי פקודות מוכללים, וליצירת סרגלי פקודות אישיים

ספרת האלמנטים בסרגלי פקודות

ספרה של האלמנטים בסרגל הפקודות חיונית בעבודה עם סרגלי פקודות. הספרה היא כלי חשוב לבירור ההיררכיה של האובייקט `CommandBar`. המידע שיתברר לכך באמצעות הדוגמאות הבאות יסייע בידך לעורך שינויים בסרגלי פקודות מוכללים, ובפיתוח סרגלי פקודות מותאמים אישית.

השיגרה הקצרה `ShlLln` מפיקה את מנתין סרגלי הפקודות ביישום. אם אין ביישום סרגלי פקודות מותאמים אישית, השיגרה מפיקה את מנתין סרגלי הפקודות המוכללים, 140 במספר. מספר זה גבוה יותר כאשר יישום מכיל גם סרגלי פקודות מותאמים אישית.

```

Sub countCommandBars()
    MsgBox "There are " & CommandBars.Count & _
        " bars in the CommandBars collection."
End Sub

```

הרשוטך שלושה סוגי של סרגלי פקודות. ספריית האובייקטים של Office מכילה את קבועי msoBarType להפניה אל סרגלי פקודות מסוימים אלה : msoBarTypeNormal, msoBarTypePopupMenu ו- msoBarTypeMenuBar. כמו כן ניתן לבחון בין סרגלי פקודות מוכללים לסרגלי פקודות מותאמים אישית. השגירה הבאה מונה את סרגלי הפקודות המוכללים, לפי סוג.

```

Sub builtinCommandBarCount()
Dim cbr1 As CommandBar
Dim iMbars As Integer, iTbars As Integer
Dim iPbars As Integer, ibuiltin As Integer
For Each cbr1 In CommandBars
    If cbr1.BuiltIn Then
        ibuiltin = ibuiltin + 1
        If cbr1.Type = msoBarTypeMenuBar Then
            iMbars = iMbars + 1
        ElseIf cbr1.Type = msoBarTypeNormal Then
            iTbars = iTbars + 1
        Else
            iPbars = iPbars + 1
        End If
    End If
Next
MsgBox "There are " & ibuiltin & " command bars. " & _
    "IMBars & "is a menu bar, " & iTbars & " are toolbars, and " & _
    "iPbars & " are popup bars."
End Sub

```

רשימת סרגלי הפקודות הכלואים לעין

מכיוון שקייםים 140 סרגלי פקודות מוכללים, מן הסתם לא תרצה למן את כלם בדרך כלל. עם זאת, עשוי להיות חשיבות לקבוצות משנה שונות בישום נתון. לדוגמה, הישום עשוי להזדקק לנตอน, אלו סרגלי פקודות נראים לעין. השגירה enumerateVisibleCommandBarsImmediate כותבת שורה לחלון (מיידי) עבור כל סרגל פקודות נראה לעין. השורות מציגות שלושה מאפיינים לכל סרגל פקודות – שם סרגל הפקודות, סוגו ומספר הפקדים שהוא מכיל. זוג פונקציות If מקוונות מפענחו את המאפיין Type. במקום למנוחת את הפקדים בסרגלי הפקודות כדי להגיע למספרה, השגירה פושט מציגה את המאפיין Count של סרגל הפקודות.

```

Sub enumerateVisibleCommandBars()
Dim cbr1 As CommandBar
For Each cbr1 In CommandBars
    If cbr1.Visible = True Then
        Debug.Print cbr1.name, _
            (IIf(cbr1.Type = msoBarTypeNormal, "toolbar", _ 
            IIf(cbr1.Type = msoBarTypeMenuBar, "menu bar", "popup bar"))), _
            cbr1.Controls.Count
    End If
Next cbr1
End Sub

```

כל למדи להרחיב את קטע הקוד שיעיל כך שייננה את הפקדים הנפרדים בכל סרגל פקודות נראה לעין. לכל סרגל פקודות אוסף Controls, והאלמנטים של אוסף זה הם אובייקטי CommandBarControl. השגירה הבאה מחייבת אובייקט CommandBarControl תוך הצגת רשימת הכתובים של הפקדים בכל סרגלי אובייקט CommandBarControl הנראים לעין.

```

Sub enumerateControlCaptions()
Dim cbr1 As CommandBar
Dim ctl1 As CommandBarControl
For Each cbr1 In CommandBars
    If cbr1.Visible = True Then
        Debug.Print "Command bar name: " & cbr1.name & _
            " and control count: "; cbr1.Controls.Count
        For Each ctl1 In cbr1.Controls
            Debug.Print cbr1.name, ctl1.Caption
        Next ctl1
    End If
Next cbr1
End Sub

```

הצגת פקודות תפריט

לבסוף, ייתכן שתידרש לך רשימה של הפקודות הנפרדות בתפריט כלשהו. במקרה זה יש לטפל בתפריט כבסרגל פקודות, כך שהפקודות ייחשפו כפקדים. תוכל לקבוע את השם עבור סרגל פקודות המציג תפריט מסוים באמצעות השגירה העוברת על הפקדים של תפריט נתון. השגירה הראשונה מעבירה שם של סרגל פקודות לשיגרה השנייה, שעוברת בלולאה על פני הפקדים של אותו סרגל פקודות.

```

Sub listCommands()
    enumerateCommandsOnMenu ("Help")
End Sub

Sub enumerateCommandsOnMenu(menuName)
Dim cbr1 As CommandBar
Dim ctl1 As CommandBarControl

' Set a reference to a command bar.
Set cbr1 = CommandBars(menuName)

' Loop through the controls for that command bar.
For Each ctl1 In cbr1.Controls
    Debug.Print ctl1.Caption
Next ctl1
End Sub

```

טיפול בסרגלי פקודות מוכלים

ניתן לשנות סרגלי פקודות מוכלים במספר דרכים. בסעיפים הבאים מוצגת כמה מהן.

שינוי מצב הזמינות של סרגלי הפקודות ופקדיהם

באפשרותך להפוך סרגלי פקודות לבלי זמינים, ולאחר מכן להשיב אותם למצב זמין. שתי השגרות הבאות מבטלות את הזמינות של שורת התפריטים המוכנת (המכונה Menu Bar), ולאחר מכן מכך מחזירות אותה למצב זמין. כדי לבטל את זמינותו של סרגל פקודות זה בטופס, עליך להגדיר למאפיין Enable שלו ערך False בתוך שירה של אירוע טופס. יישומיך יוכל להיתנו את ביטול הזמינות של סרגל פקודות בגורמים שונים, כמו למשל זיהוי משתמש.

```

Sub disableMenuBar()
Dim cbr1 As CommandBar

For Each cbr1 In CommandBars
    If cbr1.name = "Menu Bar" Then
        cbr1.Enabled = False
    End If
Next cbr1

End Sub

```

```

Sub enableMenuBar()
Dim cbr1 As CommandBar

    For Each cbr1 In CommandBars
        If cbr1.name = "Menu Bar" Then
            cbr1.Enabled = True
        End If
    Next cbr1

End Sub

```

כמו כן ניתן להפוך לבתאי זמינות פקודות מסוימות בשורת התפריטים או ברגל כלים. השגירה הראשונה מתוקן שתி השגורות הבאות הופכת לבתאי זמין את הפקודה **תצוגה** (View) שבשורת התפריטים וברגל הכלים **תצוגת טופס** (Form View). פעולה זאת מסיימת לשמור על עיצוב הטופס, על ידי ביטול שני נתיבים מוכרים למעבר בין **תצוגת טופס** (Form) ל**תצוגת עיצוב** (Design). נוסף לנטרול הפקד **תצוגה** (View), מגינה השגירה הראשונה על השינוי על ידי קביעת הגדרה **msoBarNoCustomize** של אפיון Protection של רגול הפקודות. הגדרה זאת הופכת לבתאי זמין (МОЗГ в АПРО) את הלחץ **איפוס** (Reset) בתיבת הדו-שייח**התאמה אישית** (Custom) של רגול הפקודות **שורת תפריטים** (Menu Bar) ו**תצוגת טופס**. השגירה השניה מחזירה את הפקודות בשני רגול הפקודות במצב זמין. שים לב כי שמות הפקודות מופיעים בקוד כשמות הפקדים במשק Access (למשל, במשק עברי שם הפקודה בקוד שלחלה הוא "תצוגה", ובמשק אנגלי שמה "View").

```

Sub disableViewMenuAndControl()
Dim ctl1 As CommandBarControl

' Disable and protect View Menu.
    Set ctl1 = CommandBars("Menu Bar").Controls("תצוגה")
    ctl1.Enabled = False
    CommandBars("Menu Bar").Protection = msoBarNoCustomize

' Disable and protect View Control.
    Set ctl1 = CommandBars("Form View").Controls("תצוגה")
    ctl1.Enabled = False
    CommandBars("Form View").Protection = msoBarNoCustomize

End Sub

Sub enableViewMenuAndControl()
Dim ctl1 As CommandBarControl

' Enable View Menu.
    Set ctl1 = CommandBars("Menu Bar").Controls("תצוגה")
    ctl1.Enabled = True

' Enable View Control.
    Set ctl1 = CommandBars("Form View").Controls("תצוגה")
    ctl1.Enabled = True

End Sub

```

הציג סרגלי פקודות בלתי נראים

פעולה נוספת, פשוטה אך חשובה, שנitin לבצע היא חשייפת תפריט מוכפל שבדרך כלiano מוצג. השירה הבאה מציגה את השם, הסוג ומספר הפקדים בכל סרגל פקודות נראה לעין. אם סרגל הכלים **אינטרנט** (Web) אינו נראה לעין, השירה מגדרה את המאפיין שלו **Visible** (גלו) ומשאיר רישום של פעולה זו בחולון **Immediate** על ידי הדפסת השם, הסוג ומניין הפקדים של סרגל הכלים. תוכל להעלים את סרגל הכלים **אינטרנט** שוב על ידי הקצתה ערך **False** למאפיין **Visible** שלו.

```
Sub showToolBar()
Dim cbr1 As CommandBar
    For Each cbr1 In CommandBars
        If cbr1.Visible = True Then
            Debug.Print cbr1.name, cbr1.Type, cbr1.Controls.Count
        ElseIf cbr1.name = "Web" Then
            cbr1.Visible = True
            Debug.Print cbr1.name, cbr1.Type, cbr1.Controls.Count
        End If
    Next cbr1
End Sub
```

הוסף פקודות לסרגלי פקודות מוכלים

מלבד ניהול בחרים מוכלים באוסף **CommandBars**, באפשרות להוסיף פקודות מותאמות אישית לכל סרגל כלים מוכלל. דרך פשוטה לעשות זאת היא להוסיף אובייקט **CommandBarButton**. עליך לדעת במדוקיך את שמו של סרגל פקודות כדי שתוכל להוסיף בו לחץ חדש בעזרת השיטה **Add** (זכור שבאפשרותך להפעיל את השירה **enumerateControlCaptions** כדי ליצור רשימה של שמות סרגלי הפקודות). לאחר הוספת החץ, הגדר את המאפיינים עבור אובייקט **CommandBarButton** החדש כך שיציביע אל שירה או פונקציה מותאמת אישית.

השירה **newMenuItem** שלහן ושלוש השגרות הנלוות אליה מוסיפות פריטים חדשים לתפריט. השירה **newMenuItem** מוסיפה אובייקטי **CommandBarButton** בסוף של סרגל פקודות בשם **Tools** (כלים). שלוש השגרות הנלוות מאפשרות למשתמש לציין איזו דמות ילبس המסיע – המהדק (Clippit), רוקי (Rocky) או F1. אובייקטי **CommandBarButton** החדשים מאפשרים למשתמש להפעיל את השגרות השולטות בדמותו המסיע שתוצג.

```
Sub newMenuItem()
Dim newItem As CommandBarButton
' Set reference to new control on the Tools command bar.
    Set newItem = CommandBars("Tools").Controls.Add(Type:=msoControlButton)
```

```

' Start new group with command to invoke showClippit.
With newItem
    .BeginGroup = True
    .Caption = "Show Clippit"
    .OnAction = "showClippit"
End With

' Set reference to new control on the Tools command bar.
Set newItem = CommandBars("Tools").Controls.Add(Type:=msoControlButton)

' Assign command to invoke showRocky.
With newItem
    .Caption = "Show Rocky"
    .OnAction = "showRocky"
End With

' Set reference to new control on the Tools command bar.
Set newItem = CommandBars("Tools").Controls.Add(Type:=msoControlButton)

' Assign command to invoke showRocky.
With newItem
    .Caption = "Show F1"
    .OnAction = "showF1"
End With

End Sub

Sub showRocky()
    With Assistant
        .Visible = True
        .FileName = "Rocky.acs"
        .On = True
    End With
End Sub

Sub showClippit()
    With Assistant
        .Visible = True
        .FileName = "Clippit.acs"
        .On = True
    End With
End Sub

Sub showF1()
    With Assistant
        .Visible = True
        .FileName = "F1.acs"
        .On = True
    End With
End Sub

```

השתמש בשיטה Add עבור האוסף Controls של סרגל פקודות כדי להוסיף פקד חדש בתפריט מוככל. שיטה זאת מקבלת מספר ארגומנטים, לרבות הפרמטר Type. נוסף לפקד הלחץ (msoControlButton) שבדוגמה, תוכל לצין תיבת טקסט פשוטה (msoControlComboBox), תיבה מושלבת (msoControlEdit), ועוד. לפי ברירת המחדל, השיטה Add מוסיפה את הפקד החדש בסוף של סרגל הפקודות, אולם ניתן לעקוף תוכנה זאת ולהציג את הפקד במקום אחר בסרגל הפקודות. פרמטר נוסף, ID, מפשט את ההוספה של פקודות מוככליות מתפריטים אחרים לסדרל הפקודות המותאמים אישית.

לאחר הוספת פקד לסדרל פקודות מוככל, ניתן לקשר אותו לפקוד מותאם אישית בפעולות המאפיין OnAction. קבוע כערך המאפיין את שם של שיגרה שאתה רוצה להפעיל באמצעות הפקד החדש. באמצעות המאפיין Caption של הפקד תוכל בנקל לתת תוויות לפקד החדש. השתמש בשיטות CopyFace ו-PasteFace כדי לסמן את הפקדים המתאימים אישית. כאשר המאפיין BeginGroup מקבל ערך True, יופיע על סרגל הפקודות פקד, ככלפננו קו מפריד. בדוגמה, מאפיין זה מקבל ערך True עבור הראשון בין שלושת הפקדים המתאימים אישית, אולם ערך ברירת המחדל False יותר בעינו עבור שני הפקדים האחרים.

שיעור סרגלי פקודות

בהתחלת הlient של יישומייך המותאים אישית, ניתן שתרצה להסיר פקדים מותאים אישית שהוספה בתפריטים מוככליים. עשה זאת באמצעות השיטה Reset. השיגרה הבאה מסירה את כל הפקדים המתאימים אישית מסרגל הפקודות Tools.

```
Sub removeMenuItem()
    CommandBars("Tools").Reset
End Sub
```

יצירת סרגלי פקודות מותאים אישית

יצירת סרגל פקודות מותאים אישית נעשית בשלושה שלבים לפחות:

1. הוספת סרגל פקודות חדש לישום. הסרגל החדש יהיה ריק בתחילתו.
2. הצבת פקדים בסרגל הפקודות. פעולה זאת דומה להוספה פקדים בסרגל פקודות מוככל.
3. הגדרת ערך True למאפיין Visible של סרגל הפקודות אם ברצונך להציגו. כמו כן תוכל לאפשר למשתמשים לחסוך את סרגל הפקודות המותאים אישית שלך באמצעות התוכנות הרגילים (כמו למשל תיבת הדו-שים התאמת אישית Customize).((Customize))

שתי השגרות הבאות יוצרות סרגל פקודות מותאם, ובו פקד לחצן אחד הגורם להופעת המסיע בדמota רוקי. השגירה newCommandBarAndButton מעבירה הלאה את שני שלביים הראשוניים של ייצירת סרגלי פקודות אל השגירה הדוגמאית הבאות. השגירה .addShowAssistantsAndRocky מטעינה לסרגל הפקודות החדש את שם Show Assistants. בעת השגירה מוסיפה פקד מותאם אישית. כאשר מגדירים פקדים לסרגלי פקודות מותאמים אישית, חובה להקצות ערך למאפיין Style哉 ערכי המאפיינים האחרים שמדוברים בסרגלי פקודות מוכלים. אם תשים לעשוות זאת, השגירה addShowAssistantsAndRocky עלולה להציג בסרגל הפקודות לחצן ריק.

```

Sub newCommandBarAndButton()
On Error GoTo CBarBtnTrap
Dim cbr1 As CommandBar
Dim cbr1btn1 As CommandBarButton
Dim cbr1Name As String
' Add command bar to show Rocky.
    addShowAssistantsAndRocky
' Make CommandBar visible.
    Set cbr1 = CommandBars("Show Assistants")
    cbr1.Visible = True
CBarBtnExit:
    Exit Sub
CBarBtnTrap:
    Debug.Print Err.Number; Err.Description
    Resume CBarBtnExit
End Sub

Sub addShowAssistantsAndRocky()
Dim cbr1 As CommandBar
Dim cbr1btn1 As CommandBarButton
' Add a command bar named Show Assistants.
    Set cbr1 = CommandBars.Add("Show Assistants", msoBarTop, , True)

'Add a button control to the command bar.
    Set cbr1btn1 = cbr1.Controls.Add(msoControlButton, , , True)
' Set button properties.
    With cbr1btn1
        .Caption = "Show Rocky"
        .BeginGroup = True
        .OnAction = "showRocky"
        .Style = msoButtonCaption
    End With
End Sub

```

לאחר שהשליטה שבה אל השירה newCommandBarAndButton, היא מדירה ערך True למאפיין Visible (גלו) של הפקד. ללא שלב זה, הדרך היחידה בה יכול משתמש לראות את סרגל הפקודות המותאמים אישית החדש היא על ידי הצגתו במפורש (למשל, בלחיצת עכבר ימנית על סרגל פקודות ובחירה שם סרגל הפקודות הרצוי). הקודвлיכידת שגיאות בשירה newCommandBarAndButton מאפשר לישום להפעיל את השירה גם כאשר סרגל הפקודות כבר קיים. ללא הלוגיקה ללבידת שגיאות, השירה כבר קיים. הוואיל ושגיאה זאת אינה קריטית (סרגל הפקודות כבר קיים) ניתן בהחלט להתעלם ממנה.

שינויי סרגלי פקודות מותאמים אישית

שלוש השגורות הבאות מוסיפות פקדים חדשים בסרגל פקודות מותאמים אישית קיים. כמו כן הן חופפות גישה נוספת לטיפול בסוגיות סרגל הפקודות הקיים. השירה addCbrBtns מציבה זוג לחצנים נוספים בסרגל הפקודות שייצרנו בדוגמה הקודמת. אם סרגל פקודות זה אינו קיים, השירה מפעילה את השירה addShowAssistantsAndRocky היוצרת את סרגל הפקודות Show Assistants לפי ערך החזירה של שגרת הפונקציה doesCbrExist. שירה זאת מבירת אם סרגל פקודות נתון כבר קיים. בין אם סרגל הפקודות Show Assistants מציבה בסרגל הפקודות עוד שני לחצנים. מגדר הפניה לסרגל זה. יתרת השירה מציבה בסרגל הפקודות עוד שני לחצנים. בסיום השירה addCbrBtns, סרגל הפקודות הופך נראה לעין אם עד כה היה מוסתר.

```
Sub moreButtons()
    addCbrBtns "Show Assistants"
End Sub

Sub addCbrBtns(cbrName As String)
Dim cbr1 As CommandBar
Dim cbr1btn1 As CommandBarButton

' Optionally create Show Assistants command bar.
' Reference it with a variable.
If Not doesCbrExist(cbrName) Then
    addShowAssistantsAndRocky
    Set cbr1 = CommandBars(cbrName)
Else
    Set cbr1 = CommandBars(cbrName)
End If
' Add a new button to Show Assistants command bar.
    Set cbr1btn1 = cbr1.Controls.Add(msoControlButton, , , True)
```

```

' Set properties for button to show Clippit.
With cbr1btn1
    .Caption = "Show Clippit"
    .OnAction = "showClippit"
    .Style = msoButtonCaption
End With

' Add a new button to Show Assistants command bar.
Set cbr1btn1 = cbr1.Controls.Add(msoControlButton, , , True)

' Set properties for button to show F1.
With cbr1btn1
    .Caption = "Show F1"
    .OnAction = "showF1"
    .Style = msoButtonCaption
End With

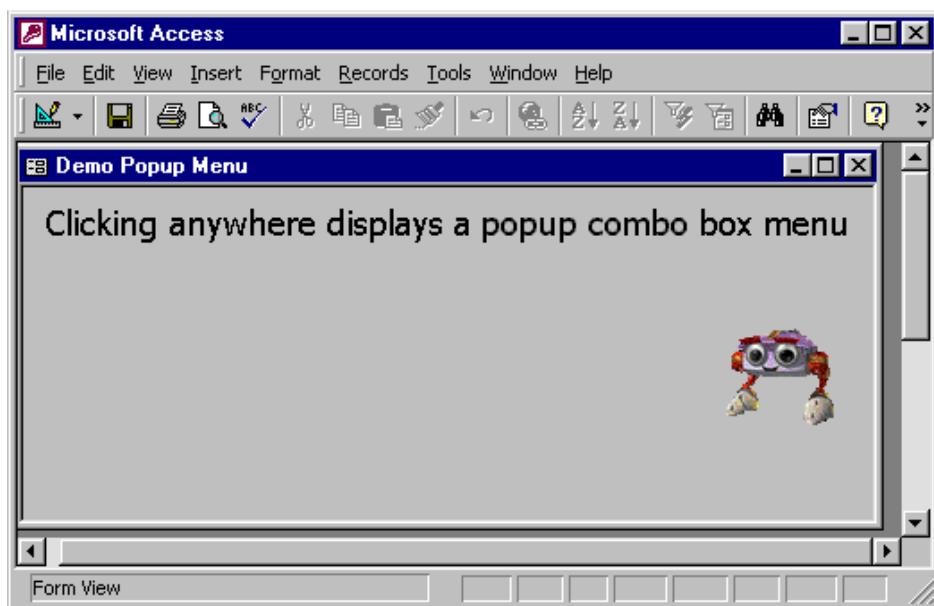
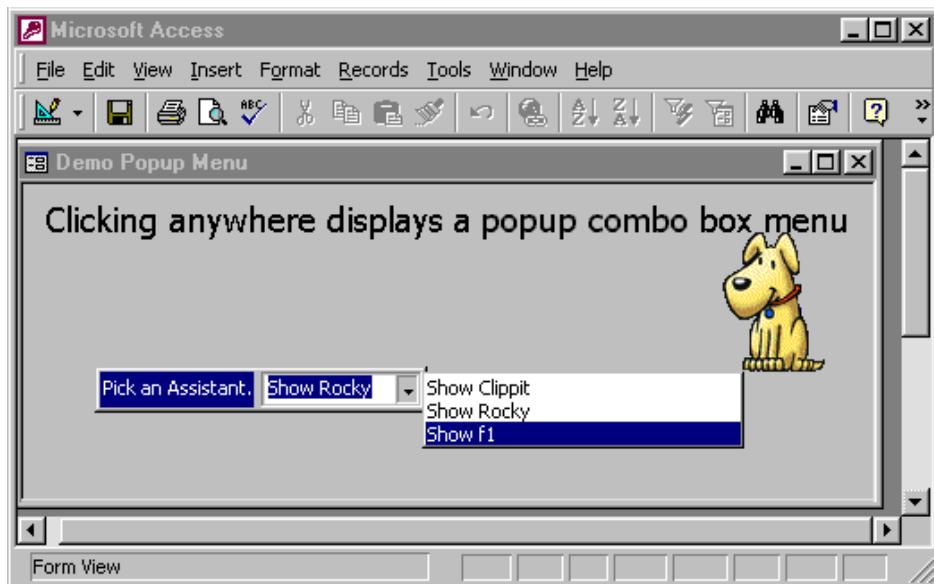
' Make the Show Assistants command bar visible.
If Not cbr1.Visible = True Then cbr1.Visible = True
End Sub

Function doesCbrExist(cbrName As String) As Boolean
Dim cbr1 As CommandBar
doesCbrExist = False
For Each cbr1 In CommandBars
    If cbr1.name = cbrName Then
        doesCbrExist = True
    End If
Next cbr1
End Function

```

יצירת סרגל פקודות מוקפץ

דוגמת השיגרה הראשונה שלහן הופכת לזמן פקד תיבת-משולבת בסרגל פקודות מותאמת אישית, והופכת את סרגל הפקודות לשורת תפריטים מוקפצת. תרשימים 8.10 משלහן מציג את התנהוגות התפריט המוקפץ על גבי טופס. לחץ בכל נקודה-שהיא בטופס כדי להקפיים סרגל פקודות מותאמת אישית ובו פקד יחיד. פקד זה הוא תיבה משולבת עם ערכיים לבחירת מסיע בדמות המהדק (Clippit), רוקי (Rocky) או F1. התהילך מתחילה באירועו לחיצה עבור מקטע הפירוט של הטופס. שלוש השגרות הבאות מכילות את הקוד יצירת הדוגמה המוצגת בתרשימים 8.10.



תרשים 8.10: שורת תפריטים מוקפצת בהתאם אישית, עם פקד תיבה-משולבת. לפתיחה שורת התפריטים המותאמת אישית, לחץ בכל נקודה-שהיא בטופוף

```
Private Sub Detail_Click()
    ShowAndProcessComboBox
End Sub
```

```

Sub showAndProcessComboBox()
Dim cbr1 As CommandBar
' Call from Click Event in form.
If doesCbrExist("Custom1") Then
    CommandBars("Custom1").ShowPopup
Else
    createAndShowPopUpMenu
End If
End Sub

Sub createAndShowPopUpMenu()
Dim cbr1 As CommandBar
' Add command bar named Custom1.
Set cbr1 = CommandBars _
    .Add(name:="Custom1", Position:=msoBarPopup, Temporary:=True)
With cbr1
    .Controls.Add Type:=msoControlComboBox
    With .Controls(1)
        .Style = msoComboBoxLabel
        .Caption = "Pick an Assistant."
        .AddItem "Show Clippit"
        .AddItem "Show Rocky"
        .AddItem "Show F1"
        .OnAction = "processComboBoxChoice"
    End With
End With
cbr1.ShowPopup
End Sub

Sub processComboBoxChoice()
Dim caseValue As Integer
' Decode selected item and implement corresponding method.
Select Case CommandBars("custom1").Controls(1).ListIndex
    Case 1
        showClippit
    Case 2
        showRocky
    Case 3
        showF1
End Select
End Sub

```

השיגרה הראשונה היא שגרת האירוע שמאחורי הטופס, והיא מפעילה את השיגרה showAndProcessComboBox, השוכנת במודול סטנדרטי (standard module). שיגרה זאת קובעת אם סרגל הפקודות Custom1 כבר קיים. אם כן, השיגרה מפעילה את

השיטה ShowPopup כדי להציג את סרגל הפקודות כשורטת תפריטים מוקפצת. אם לא, היא יוצרת את סרגל הפקודות Custom1 על ידי הפעלת השירה createAndShowPopupMenu של השירה createAndShowPopupMenu. המשמעה של השירה השלישייה, היא יוצרת את סרגל הפקודות המותאמים אישית וממיד מציגה אותו כשורטת תפריטים מוקפצת.

השירה createAndShowPopupMenu אמנים קצרה, אולם היא פעולה טכנית מעניינית. ראשית, נעשה בה שימוש במשפט With...End With...End. המשפט החיצוני מוסיף חבר חדש לאוסף CommandBars, והמשפט הפנימי מוסיף פקד לאותו חבר. הקצאות המאפיינים בתוך משפט With...End הפנימי מגדרות לפקד סגנון של תיבת משולבת, מגדרות את האלמנטים בראשית התיבה המשולבת, ומציניות את השירה processComboBoxChoice, המופעלת בעקבות בחירה של המשתמש מתוך תיבת המשולבת. שירה אחרת זאת משתמשת במשפט Select Case המבוסס על האלמנט שנבחר מתוך התיבה המשולבת כדי להפעיל אחת משלוש השגרות המותאמות אישית המציגות מסיע.

מחיקת סרגלי פקודות מותאמים אישית

אם אתה עוסק בבניית סרגלי פקודות מותאמים אישית ביישומים, מן הסתם תctrיך לעיתים גם למחוק אחד מהם, או יותר. בדוגמה הבאה נמחק סרגל כלים באמצעות יצירת לולה העוברת על כל סרגלי הפקודות כדי לאתר את אלה המותאמים אישית, ככלומר אלה שבهم המאפיין BuiltIn מכיל ערך False. כאשר השירה מוצאת סרגל פקודות מותאמים אישית, היא שואלת את המשתמש אם למחוק אותו. אם המשתמש משיב כן (Yes), סרגל הפקודות נמחק והשירה מוסיפה אחד למנין סרגלי הפקודות שנמחקו. בכל מקרה, השירה מוסיפה אחד לשנתנה המונה את סרגלי הכלים המותאמים אישית.

```
Sub deleteCustomCbr()
Dim cbr1 As CommandBar, delFlag As Boolean
Dim delBars As Integer, cusBars As Integer

' Not necessary to initialize delFlag, delBars, or cusBars because their default
' values (False and 0) are OK.

' Conditionally delete custom menu bars.
For Each cbr1 In CommandBars
    If (cbr1.BuiltIn = False) Then
        If MsgBox("Are you sure that you want to " & _
                  "delete the " & cbr1.name & " command bar?", _
                  vbYesNo, "Programming Microsoft Access 2000") = _
                  vbYes Then
            cbr1.Delete
    End If
Next
```

```

        delFlag = True
        delBars = delBars + 1
    End If
    cusBars = cusBars + 1
End If
Next cbr1

' Report outcome of command bar enumeration.
If Not delFlag Then
    If cusBars > 0 Then
        MsgBox "No custom command bars deleted" & _
            "out of a total of " & cusBars & ".", _
            vbInformation, "Programming Microsoft Access 2000"
    Else
        MsgBox "No custom command bars.", vbInformation, _
            "Programming Microsoft Access 2000"
    End If
Else
    MsgBox delBars & " custom command bar(s) deleted.", _
        vbInformation, "Programming Microsoft Access 2000"
End If
End Sub

```

השлага deleteCustomCbr מסוימת בהציג אחד משלושה משפטים אפשריים, לפי מספר סרגלי הפקודות שנמכו וסך כל סרגלי הפקודות המותאמים אישית. זוג משפטי If...Then מקוונים מטפל בניתוב אל המשפט המותאים עבור תיבת הודעה. אם אין מחיקות, אולם יש לפחות סרגל פקודות מותאמים אישית אחד, המשפט מציג הודעה המדוחת כי לא נמחקו סרגלי פקודות, ומציג את מספר סרגלי הפקודות המותאמים אישית. אם אין מחיקות ואין סרגלי פקודות מותאמים אישית, השлага מציגה הודעה ברוח זאת. לבסוף, אם השירה גילתה סרגלי פקודות, תיבת הודעה מציגה את מספרם.

שילוב Access עם יישומי Office אחרים

בתחילת דרכה, זכתה Microsoft Access לפופולריות עצומה כאחד מרכיבי מערכת התוכנות Microsoft Office. אלמנטים רבים במשק המשמש של Access משותפים גם למשקי המשמש ביישומי Office אחרים, וקל למדи להעביר נתונים בין Access לבין אחרים Microsoft Office. בנוסף לכך, ניתן לשלב את Access עם שאר מרכיבי Office גם לשאר מרכיבי Office. נוסף לכך, ניתן לשלב את Access עם שאר מרכיבי Microsoft Office. דבר זה מאפשר ליהנות מה יתרונות של מערכת תמלילים וגלגולות אלקטרוניים.

פרק זה מסביר כיצד לבצע שילוב תוכניתו של Access עם יישומי Office אחרים בעזרת תכונות Access מוכללות:

«**יישומי** יכולו להגעה למנהל ISAM באמצעות האובייקט Connection כדי לטפל בתוניהם של גלגולות אלקטרוניים ב- Microsoft Excel. אובייקט Connection המבוסס על מנהל ISAM יכול לשמש כערוץ דו-כיווני לשיתוף נתונים בין Access ו- Excel.

«**תוכל** להגעה באמצעות תוכניתו למקורות נתונים של Access בעזרתו יכולת מיזוג הדואר ב- Microsoft Word, המאפשרת יצירה של תוויות מען, מכתבים אחידים וקטלוגים של מוצריים.

«**בעזרת** אוטומציה, **יישומי** יכולו לנצל בו-זמנית את מבני האובייקט משני יישומי Office, או יותר. לדוגמה, **יישום** יכול ליציא שמות וכתובות מתוך מחסן נתונים Microsoft Outlook (Contacts) ב- Microsoft Outlook. בדומה לכך, ניתן לאחסן נתונים של Microsoft Access באמצעות Microsoft Word.

הדוגמאות בפרק זה מתמקדות ב- Access, Word ו- Outlook, אולם העקרונות הכלליים חלים גם על יישומי Office אחרים, וכן על ערכות של יוצרים אחרים, שנחappsות בהן תכונות האובייקט באמצעות אוטומציה ומתקפשר בהן טיפול בתוניהם באמצעות Microsoft Visual Basic for Applications (VBA).

קישור Office ליישומי אחרים

בסעיף זה תערוך הিירות עם שולש טכניקות (מנהל ISAM) ניתנים להתקנה, השיטה OpenDataSource באובייקט MailMerge (ווטומציה) להפעלת Access בשילוב עם יישומי Office אחרים. בסעיפים הבאים ייעשה שימוש בטכניות אלה בהקשר מעשי.

מנהל ISAM ברה התקנה

השתמש באובייקט Connection המוכר ב- ADO (Microsoft ActiveX Data Objects) המשמש ל קישור עם מקורות נתונים אחרים באמצעות מנהלי ISAM. מקורות מידע כאלה יכולים להיות כאלה שאיןם Jet ואים ODBC, כמו למשל Excel, dBASE ו- Paradox. בסעיף זה נשתמש במנהל ISAM ב- Excel כדי ליצור קישורים בין ISAM בקבץ החוברות עבורה ב- Excel. יש לשים טכניות דומות בכל הנוגע למנהלי ISAM בקבצים dBASE, Lotus 1-2-3, Paradox, HTML, טקסט ו- XML, נדרים ומוגבלות משלו. תוכל להרחיב את ידיעותיך על ידי עיון בסיקום אודזות המופיע Connect בתוכנית העזרה המקוונת.

הערה:



היקף התמיכה ל- ISAM בר התקנה ממשיך לגדול בהתאם לצורכי המשתמש וההתקויות הטכנולוגיות. ב- 2000 Access בוטלה התמיכה ל- ISAM עבור מסדי נתונים של Microsoft FoxPro לטובת מנהל ODBC FoxPro. מנהלי ISAM המקבילים ימשיכו לפעול עם נתונים dBASE ו- Paradox בגרסה 5 ומתה. אם דרישה לך גישת קריאה/כתיבה לגרסאות אחרות, עליך לרכוש בנפרד את תוכנת Borland Inprise Database Engine במערכות הexas.

כאשר אתה משתמש במנהל ISAM, מחרוזות החיבור שלו מכילה שלושה ארגומנטים, כל אחד מהם חייב להסתאים בנקודה-פסיק. תחילת ערך קבוע ספק. אם אתה משתמש במנהל ISAM בר התקנה, תחל את מחרוזות החיבור בהפניה לשפק Jet. המשך במפרט המצביע אל הקובץ עבור מקור הנתונים. אם מדובר ב- Excel, על הפירוט לכלול את הכוון, הנתיב ושם הקובץ. קבוע את הfrmטර האחרון על ידי הצבת שם מנהל ISAM הרצוי כהגדירה לfrmטר המאפיינים המורחבים. קיימים מנהלי התקן מוגדרים לגרסאות השונות של Excel ולסוגים אחרים של מקורות נתונים שנינעו ליצור עם קישור. ההפניה אל חוברת עבורה ב- 2000 Excel נעשית באמצעות המחרוזת Excel 8.0," ולאחריה נקודה-פסיק.

בדוגמת הקוד הפושא שלහן נעשה שימוש במנהל ISAM כדי ליצור קישור עם חוברת עבורה ב- 2000 Excel בתוך יישום של Access 2000. המשפט Dim מצהיר על אובייקט Connection חדש ויוצר אותו. המשפט הבא מתחילה את החיבור על ידי הצבעה אל חוברת עבורה של Excel באמצעות מנהל ISAM ב- 8 Excel. לאחר שנוצר החיבור עם מקור הנתונים, על היישום להגדיר טווח של תאים בחוברת העבודה. בדוגמה הקוד

מוקצת הטווח customers שבתוך הקובץ לערכת הרשומות ששםה rst1. משמש בקשר זה כדי לטפל בתונינים בחוברת העבודה. בסיום הדוגמה מודפסות שתי העמודות הראשונות של השורה הראשונה מתוך הטווח שצוין בחוברת העבודה של Excel בתוך חלון Immediate ב- Access.

```
Sub connect2XLPrintFromFirst()
Dim cnn1 As New ADODB.Connection, rst1 As ADODB.Recordset
' Make connection to Excel source.
cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Programming Access\Chap09\customers.xls;" & _
    "Extended Properties=Excel 8.0;"
' Open read-only recordset based on Excel source.
Set rst1 = New ADODB.Recordset
rst1.CursorType = adOpenForwardOnly
rst1.LockType = adLockReadOnly
rst1.Open "customers", cnn1, , , adCmdTable
' Print selected fields from first record.
Debug.Print rst1.Fields(0).Value, rst1.Fields(1).Value
' Close connection to source.
cnn1.Close
End Sub
```

כאשר עובדים עם מנהל ISAM אין צורך בפתיחת מקור הנתונים ב- Excel (או אפילו בפתיחת Excel עצמו). כמו כן לא נדרש לישום הפניה אל תבנית האובייקט של Excel. על אף הדרישות המזעריות של מנהל ISAM ב- Excel, ניתן להשתמש בו הונן לקריאה והן לעדכו של מקורות נתונים ב- Excel.

השיטה OpenDataSource

באפשרות להשתמש בשיטה OpenDataSource של האובייקט MailMerge כדי ליצור קישור עם מקור נתונים Access מתוך יישום של Word. עלייך להשתמש ב- Access – וליתר דיוק, ב- Jet – כמחסן נתונים ליישומי מיזוג דואר היוצרים תוויות מען, מכתבים אחידים, קטלוגים של מוצרים, וכו'. אם כי ניתן לבצע ב- Access חלק מהפעולות הללו באמצעות האובייקט Report, סביבת Word היא הטבעית יותר לכתיבה תכנים. כמו כן תמצא ב- Word כלים מעולים לעיצוב טקסט ומאפייני WYSIWYG ('ימה שאתה רואה הוא מה שאתה מקבל') החסרים באובייקט Report של Access. תוכל לנצל משאביים אלה בעזרת שגרות VBA מבוססת-Word, וכן בעזרת אוטומציה מתוך Access.

כאשר אתה מפנה מקור נתונים Access באמצעות השיטה OpenDataSource, עלייך להפנות תחילת הקובץ של מסמך Word, וכן אל האובייקט MailMerge של Word. יש לציין שני פרמטרים עבור השיטה עם Name (שם), המציין את הconoן, Access:

הנתיב ושם הקובץ של מקור הנתונים ב- Access, והפרמטר Connection (חיבור), המגדיר מקור נתונים מסווג 'טבלה' או 'שאילתת', ואת שם אובייקט מסד הנתונים ב- Access. חובבה לכלול במסמך Word סימניות או שדות מיזוג דואר המצביעים על השדות שבמסד נתונים Jet. כדי להפעיל מיזוג השוואב נתונים ממוקור הנתונים שצוין (למשל, טבלה ב- Access) אל תוך מסמך Word, עליך להפעיל את השיטה Execute עבור האובייקט MailMerge.

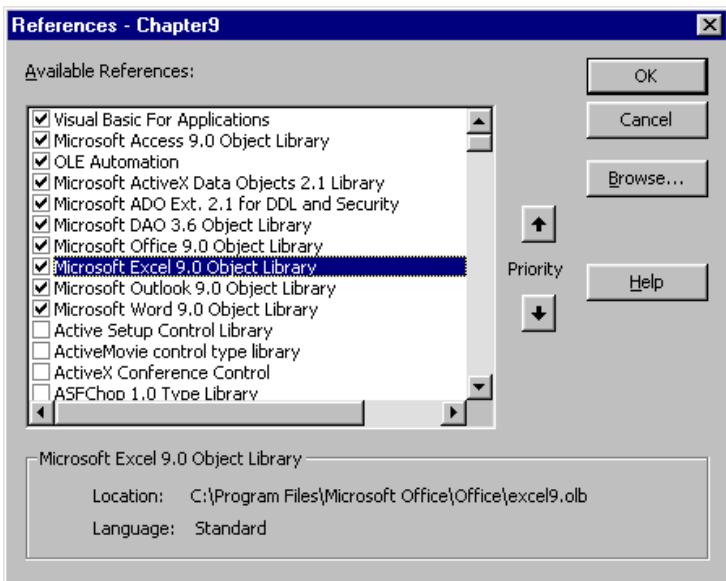
הרשוטץ מספר זרכים לסייעו ערכיהם המופיעים במסמך מיזוג דואר ב- Word. לדוגמה, תוכל השתמש בפרמטר SQLStatement של השיטה OpenDataSource כדי לציין אילו רשומות לשולף מתוך מקור הנתונים. כאשר עושים פעולה זאת עם מקור נתונים Jet, הפניה אל הפעלה Access נעשית באמצעות Manager ODBC, תוך הצבת Constr כהגדרת הפרמטר Connection. השתמש בתחביר של משפט SQL לסייעו רשומות מתוך טבלה או השאילה Access.

גישה אחרת לסייעו כרוכה בשאילתת מיוחדת במסמך Access. הפרמטר Connect של השיטה OpenDataSource רק מפנה אל אותה השאילתת. השתמש במאפיינים FirstRecord ו- LastRecord של האובייקט DataSource כדי לציין את הרשומה הראשונה והאחרונה האמורה להופיע בתוך מסמך Word לאחר מיזוג. האובייקט DataSource מצביע אל יעד שצוין על ידי השיטה OpenDataSource.

אוטומציה

בעזרת אוטומציה ניתן להפעיל יישום מסוים כך שיישלו ביחסום אחר. Microsoft Component Object Model, או COM, מגדר את ה프וטוקול עבור יכולת זאת. היישום השולט פועל עם היישום הנשלט על ידי טיפול בשיטות ובמאפיינים החשופים של היישום הנשלט, ותגובה לאירועים המתרחשים בו. כדי לעשות זאת, על היישום השולט להכיל הפניה לסדריית האובייקטים של היישום השני, וליצור מופע של אותו יישום (עיין פרקים 7 ו-8, שם תמצא פרטים על יצירה וניהול של הפניות באמצעות קוד). היישום השולט מפעיל שיטות ומקרה ערכי מאפיינים באמצעות המופע של היישום הנשלט.

תרשים 9.1 מציג ניבת דוא-שיח References (הפניות) מתוך יישום ב- Access, עם הפניות ל- Excel, Outlook ו- Word, כמו גם לסדריות Office המכילה את מבני האובייקטים המשותפים. מבוסן מסויים, אוטומציה הופכת את כל מבני אובייקט הרכיב של Office למשותפים. Access מסוגל לחושף את מבנה האובייקט שלו כشرط אוטומציה, ولנצל את מבני האובייקט של יישומים אחרים בכך שהוא פועל כמוLKוח של אוטומציה.



תרשים 9.1: תיבת הדוחש References ב- Excel, Access, ובה הפניות ל- Word ו-Outlook.

הfonקציה CreateObject לעומת הפונקציה GetObject

השתמש בfonקציות CreateObject ו- GetObject לייצרת מופעים של יישומים אחרים. הפונקציה GetObject משתמש לבירור אם כבר יש מופע פתוח של יישום מסוים. אם כן, ניתן ליצור הפניה אליו. אם איש אינו משתמש בפועל במופע, דבר זה יכול להיות קביל. אם יישום שרת האוטומציה עדיין אינו פתוח, או אם אתה מעודיף להימנע שימוש במופע פתוח, השתמש בfonקציה CreateObject כדי ליצור מופע חדש של יישום נתון. תוכל להשתמש בfonקציה GetObject גם כדי לפתח מופע של יישום, שבתוכו פתוח קובץ מסוים.

שתי השגרות הבאות יוצרות מופע של Excel מתוך יישום ב- Access. השירה השנייה, `isAppThere`, מפעילה קישור מאוחר (קיים בזמן ריצה) כדי לברר אם קיים מופע של יישום Office כלשהו. משתנה `objApp` עם ציון `objApp` אם ציון אובייקט כללי לייצג כל יישום של Office (ואפילו אובייקט COM אחר). השירה הראשונה, `ax` משמשת בקישור מוקדם (קיים בזמן הידור) המשתנה `ax` יכול לייצג רק אובייקט Application של Excel. לא ניתן להחליף את Excel.Application בfonקציה CreateObject או Excel.Application בOBJECT Application אחר של Office, כמו למשל `Word.Application`. עם זאת, ניתן ליצור שירה חדשה ונפרדת, למשל `wordThere`, שתכלול משתנה שיוחזר כאובייקט מסוג `Word.Application`. השירה החדשה תוכל להפנות אל השירה הכללית `isAppThere`, בדומה לפעה שעשוה השירה `ax`.

```

Sub xlThere()
Dim xlApp As Excel.Application
If IsAppThere("Excel.Application") = False Then

' If no, create a new instance.
    Set xlApp = CreateObject("Excel.Application")
    xlApp.Visible = True
Else

' Otherwise, reference the existing instance.
    Set xlApp = GetObject(, "Excel.Application")
End If

' If user wants instance closed, close app and set reference to Nothing.
If MsgBox("Close XL ?", vbYesNo, _
    "Programming Microsoft Access 2000") = vbYes Then
    xlApp.Quit
    Set xlApp = Nothing
End If
End Sub

Function IsAppThere(appName) As Boolean
On Error Resume Next
Dim objApp As Object
IsAppThere = True
Set objApp = GetObject(, appName)
If Err.Number <> 0 Then IsAppThere = False
End Function

```

בדרך כלל אוטומציה אינה הופכת יישום של Office לנראה לעין כאשר היא פותחת אותו. אם רצונך שהיישום הנפתח ייראה, יהיה عليك בדרך כלל להגדיר למאפיין Visible (גלווי) של אותו יישום ערך True. ישומים שונים חושפים אובייקטים שונים לאוטומציה. Excel חושף אובייקטים כמו Workbook, Application ו- Worksheet כموון, שני האובייקטים האחרונים אינם זמינים ביישומי Office האחרים.

סגירת הפניה של אוטומציה

השיגרה `xlThere` משליקת באופן מוגנה הפניה ליישום Office אחר. תחילת, יש לסגור את היישום או לצאת ממנו (Excel תומך בשיטה `Quit`). לאחר מכן יש להגדיר בהפניה ערך `Nothing`. שני השלבים נחוצים כדי לאחזר את המשאבים שתפסה הפניה של אוטומציה.

עבודה עם Excel מתוך Access

שתי הדוגמאות שלහן מראות כמה יכולות ש-ISAM בר התקנה ב-Excel מוסיפה לישומים. הדוגמה השלישי מציגה שיטה פשוטה אך רבת-עוצמה לשימוש באוטומציה. במקום לטפל במישרין באלמנטים מפורטים של בניית האובייקט של שרת אוטומציה, השיגרה מפעילה שיגרה בתוקף שרת האוטומציה. שיגורה זאת, בתורה, מעדכנת את קובץ גיליוו העבודה, אולם בזמן ובאופן שנקבע על ידי יישום של Access.

עבודה עם ערכים מתוך גליונות עבודה של Excel

שלוש הדוגמאות הבאות פועלות עם חוברת העבודה המוצגת בתרשים 9.2. שם הקובץ הוא MyGas.xls. ארבע העמודות הראשונות בגיליון מכילים נתונים שהזנו ידנית, וארבע העמודות הבאות מכילות נתונים ביוטיים המבוססים על ארבע הרשומות. בשורת הנתוחאות מוצג הביטוי עבור ערכים בעמודה MPG. הנתונים שוכנים בטוחה בשם gas. תיבת הדו-שיה הגדרת שם (Define Name) מציגה את גבולות הטוחה בגilioון.

The screenshot shows a Microsoft Excel window titled "MyGas". The main sheet displays a table of gas log entries from July 1995 to July 1996. The columns are labeled: Date, Miles, Gallons, Price per, MPG, cents per, last refill, and Total. The "MPG" column contains values like 26.029, 24.464, 26.976, etc. The "gas" name is defined in the "Define Name" dialog box, which also shows the formula =Sheet1!\$A\$1:\$H\$45.

Date	Miles	Gallons	Price per	MPG	cents per	last refill	Total
10/28/1995	281.4	10.811	\$ 1.44	26.029	\$ 0.055		\$ 15.56
11/11/1995	260	10.628	\$ 1.44	24.464	\$ 0.059	14	\$ 15.29
11/22/1995	230	8.526	\$ 1.45	26.976	\$ 0.054	11	\$ 12.35
12/06/1995	298	9.742	\$ 1.45	30.589	\$ 0.047	14	\$ 14.12
12/16/1995	274.3	10.099	\$ 1.45	27.161	\$ 0.053	10	\$ 14.63
01/07/1996	295.4	10.229	\$ 1.45	28.879	\$ 0.050	22	\$ 14.82
01/20/1996	268.29						
02/03/1996	247.5						
02/17/1996	281.7						
02/29/1996	281						
03/09/1996	304.29						
03/27/1996	286.6						
04/28/1996	321.4						
05/06/1996	321						
05/19/1996	266.1						
05/25/1996	361						
06/02/1996	290.5						
06/15/1996	286.29						
06/26/1996	306						
07/04/1996	303.39						
07/08/1996	376.2						

תרשים 9.2: גilioון אלקטרוני של Excel עם טוחה נקוב, gas, המשתרע על פני התאים H45 עד A1

הדוגמה הראשונה קוראת ערכים מתוך Excel, מבצעת חישובים שונים ב- Access ומדפסה את התוצאות בחילון Immediate. לאחר שנוצר חיבור עם מקור הנתונים של Excel, היחסום יכול לטפל במקור הנתונים באמצעות קוד, ממש כפי שהוא מTEL בטבלה פנימית ב- Access. לדוגמה, ניתן לננות את הרשומות בטבלה, או לחשב ערכים על סמך הערכים המופיעים במקור הרשומות. הדוגמה מדפסה בחילון Immediate את הערכים הקיימים בגיליון העבודה של Excel, לצד התוצאות של ביטויים שחובבו ב- Excel. פעולה זאת מאשרת שחישובים ב- Access יכולים להניב תוצאות זהות לחישובים ב- Excel. יכולות זאת עשויה לצמצם את כמות הנתונים שעל היחסום לקרוא מתוך גיליון אלקטרוני עתיק-חישובים.

```

Sub openXLComputePrint()
Dim cnn1 As New ADODB.Connection
Dim rst1 As ADODB.Recordset
Dim computedMPG As Double, computedTotal As Currency

' Make connection to Excel source.
cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Programming Access\Chap09\mygas.xls;" & _
    "Extended Properties=Excel 8.0;"

' Open read-only recordset based on Excel source.
' Recall default is read-only.
Set rst1 = New ADODB.Recordset
rst1.Open "gas", cnn1, , , adCmdTable

' Enumerate records and compute with field values.
Do Until rst1.EOF
    computedMPG = rst1.Fields("Miles") / rst1.Fields("Gallons")
    computedTotal = rst1.Fields("Gallons") * rst1.Fields("Price per Gallon")
    Debug.Print rst1.Fields("Date"), _
        rst1.Fields("Miles"), _
        rst1.Fields("Gallons"), _
        rst1.Fields("Price per Gallon"), _
        rst1.Fields("MPG"), computedMPG, _
        rst1.Fields("days since last refill"), _
        rst1.Fields("Total"), computedTotal
    rst1.MoveNext
Loop

' Close connection to source.
cnn1.Close
End Sub

```

השיגרה המשנית מצהירה על אובייקט Connection חדש, יוצרת אותו, ולאחר מכן פותחת אותו. דבר זה הוא בעל חשיבות מכרעת כאשר עובדים עם מנהל ISM, מכיוון שהוא הדריך שבה תנהל את הקישור עם מקור הנתונים שמחוץ ל- Access. מחרוזת החיבור מצביעה אל האובייקט בקובץ MyGas.xls של Excel (קטע מתוך נתוני מופיע בתרשימים 2.9). עליך לסייע את מחרוזת החיבור בצדן הימני (Extended Properties) של המצביע אל ISAM המשמש את היישום שלך. בדוגמה נעשה שימוש במנהל ISAM (Excel 97 ו- Excel 2000) שפועל עם קבצי חוברת עבודה ב-

הфункция Recordset בהמשך השיגרה היא מרכיב קריטי, אם כי סטנדרטי למדי, בישומי ISAM. על ידי הגדרת רככת רשות בחיבור, מתאפשר לישום לקרוא מתיק, ולכתוב אל תוכן, מקור הנתונים המרוחק. אם יישום ה- Access שלך יוצר קשר עם מקור הנתונים של Excel, עליך להשתמש בשיטות AddNew ו- Update של האובייקט Recordset כדי להוסיף שורות חדשות לגילוון אלקטרוני מתוך Access. במפרט רככת הרשות עליך לקבוע איזה חלק של חוברת העבודה יקשר. אם תכלול הפניה לטוווח gas, רככת הרשות תוכל להשתמש המכלול עבור גילוון העבודה. התחבר הפעינה אל טבלה חיצונית ב- Excel זהה לתחריב המשמש להפעינה לטבלה פנימית ב- Access.

הערה:



אם מוטל על היישום שלך לכתב אל מקור נתונים ב- Excel מתוך Access, או להכניס בו שינויים, הקפד להגדיר סמן שיתמוך בפונקציונליות זאת (לדוגמה, העבר את הקבוע adOpenKeyset עבור סוג הסמן, ואת הקבוע adLockOptimistic עבור סוג הנעה). שלא כמו ב- DAO, סמן ברירת המחדל ב- ADO אינו תומך בעדכון.

האלמנט החשוב הבא בשיגרה הוא לולאת Do המונה את כל הרשותות בטוווח gas. שתי הרשותות הראשונות בלולה מחשבות ביטויים עבור שני ערכיהם מחובבים. המשתנים computedMPG ו- computerTotal משווים את חישובי Access לאלה של Excel, ומאמתים את יכולת לטפל בנתונים שנקרו ממקור נתונים ב- Excel. שבע הרשותות הבאות בלולאת Do מדפיסות בחולון Immediate את ערכי שדות הטבלה מה-Excel, לצד שני משתנים מחובבים עבור כל שורה. הניווט בטוווח טבלה ב- Excel זהה לניווט בטבלה פנימית. כמובן, עליך להפעיל שיטת MoveNext בטור הולאה כדי להתקדם בין שורות גילוון העבודה.

תרשים 9.3 שלහן מציג את חמישה העמודות הקיצוניות מימיין בפלט המופק מן הדוגמה הקודמת. שתי העמודות הראשונות מציגות תוכאות זהות עבור MPG ב- Excel (העמודה הראשונה) וב- Access (העמודה השנייה). דבר זה יפה גם לחשבון חדלך הכלול בכל מילוי. דבר זה מאשר שסוגי הנתונים double ו- currency ב- Access מסוגלים להגיע לתוצאות זהות ללאו של Excel.

26.0290444917214	26.0290444917214	Null	15.557	15.557
24.4636808430561	24.4636808430561	14	15.2937	15.2937
26.9763077644851	26.9763077644851	11	12.3542	12.3542
30.5892013960172	30.5892013960172	14	14.1162	14.1162
27.1611050599069	27.1611050599069	10	14.6335	14.6335
28.8786782676703	28.8786782676703	22	14.8218	14.8218
26.3987011709141	26.3987011709141	13	14.6246	14.6246
23.0790749720254	23.0790749720254	14	13.287	13.287
26.3123482159537	26.3123482159537	14	13.0506	13.0506
26.7237280076082	26.7237280076082	12	12.8178	12.8178
30.6682120540214	30.6682120540214	9	12.2934	12.2934
25.8874537078855	25.8874537078855	18	13.6063	13.6063
30.6357830521399	30.6357830521399	32	14.6769	14.6769
30.2174526969783	30.2174526969783	8	15.2865	15.2865
25.7350096711799	25.7350096711799	13	15.1895	15.1895
34.9129593810445	34.9129593810445	6	15.0861	15.0861
32.2347980470484	32.2347980470484	8	13.1485	13.1485
28.8366236905721	28.8366236905721	13	14.3857	14.3857
31.3974964087831	31.3974964087831	11	14.122	14.122
31.8385979641096	31.8385979641096	8	13.7122	13.7122

תרשים 9.3: קטע פלט של השירה openXLComputePrint. שים לב לתוצאות החישוב הזרות ב- Access וב- Excel

יצירה דינמית של טבלאות Access המבוססות על גליונות עבודה של Excel

הדוגמה הקודמת חשפה את הערכים בגילוון אלקטרוני באמצעות ערכת רשומות. אם מוטל על היישום שlk לפעול דרך קבוע עם נתוניים מתוך גילוון אלקטרוני, תוכל לשפר את רמת הביצועים שלו על ידי העתקת הערכים מגילוון העבודה אל תוך טבלה מקומית ב- Access. בנוסף לכך, היישום ימצמצם בדרך זאת את הביקוש שלו למשאבי חיבור על ידי העתקת הערכים מגליונות עבודה אל תוך טבלאות מקומיות כאשר דרושה לו גישה בו-זמנית למספר טוחנים שונים בגלויון העבודה. הדוגמה הבאה יוצרת באמצעות קוד טבלה בעלת אינדקס לטוחן של גילוון אלקטרוני, ומאלסת את הטבלה בערכים מתוך אותו טוחן. שים לב שבדיקה נעשו בסוג הנתוניים החדש Identity כדי לציין את ערכי ההתחלה וערך הצעד בשדה האינדקס של הטבלה .(MyID)

```
Sub createTableFromXL()
On Error GoTo createTableTrap

Dim cnn1 As ADODB.Connection
Dim cnn2 As New ADODB.Connection
Dim rst1 As ADODB.Recordset
Dim rst2 As ADODB.Recordset
Dim cat1 As ADOX.Catalog
Dim tbl1 As ADOX.Table
Dim pk1 As ADOX.Index
```

```

' Set connection, catalog, and table objects.
    set cnn1 = CurrentProject.Connection
    Set cat1 = New ADOX.Catalog
    cat1.ActiveConnection = cnn1
    Set tbl1 = New ADOX.Table

' Define table named gas and append it to the Tables collection.
    With tbl1
        .Name = "gas"
        .Columns.Append "Date", adDate
        .Columns.Append "Miles", adDouble
        .Columns.Append "Gallons", adDouble
        .Columns.Append "PricePerGallon", adCurrency
    End With
    cat1.Tables.Append tbl1
    strSQL = "ALTER TABLE Gas ADD COLUMN MyID Identity(2,2)"
    cnn1.Execute strSQL

    Set pk1 = New ADOX.Index
    With pk1
        .Name = "MyPrimaryKey"
        .PrimaryKey = True
        .Unique = True
        .IndexNulls = adIndexNullsDisallow
    End With
    pk1.Columns.Append "MyID"
    tbl1.Indexes.Append pk1

' Make connection to Excel source.
    cnn2.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Programming Access\Chap09\mygas.xls;" & _
        "Extended Properties=Excel 8.0;"

' Open read-only recordset based on Excel source.
' Recall default is read-only.
    Set rst1 = New ADODB.Recordset
    rst1.Open "gas", cnn2, , , adCmdTable

'Open read-write recordset based on local table
'named gas.
    Set rst2 = New ADODB.Recordset
    rst2.ActiveConnection = cnn1
    rst2.CursorType = adOpenKeyset
    rst2.LockType = adLockOptimistic
    rst2.Open "gas"

```

```

Do Until rst1.EOF
    With rst2
        .AddNew
        .Fields("Date") = rst1.Fields("Date")
        .Fields("Miles") = rst1.Fields("Miles")
        .Fields("Gallons") = rst1.Fields("Gallons")
        .Fields("PricePerGallon") = rst1.Fields("Price Per Gallon")
        .Update
    End With
    rst1.MoveNext
Loop

createTableExit:
    Exit Sub
createTableTrap:
    If Err.Number = -2147217857 Then
        cat1.Tables.Delete "gas"
        Resume
    Else
        Debug.Print Err.Number; Err.Description
        Resume createTableExit
    End If
End Sub

```

השיגרה המתוארת לעיל ארוכה מכיוון שהיא מבצעת מספר פונקציות נפרדות, אם כי קשריות זו בזו. כדי ליצור טבלה מקומית שתכיל ערכי גיליון אלקטרוני של Excel, DrOsh לדוגמה זוג אובייקטים : Connection ו- Recordset. אובייקטים אלה מספקים קישוריות בו-זמנית אל גיליון העבודה ועל הטבלה המקומית, כך שהשיגרה מסוגלת להעתיק שורה מקור נתונים אחד למשנהו. כדי להגדיר באמצעות קוד טבלה מקומית ב- Access, הקוד מצהיר על אובייקטי Catalog ו- Table .Index

לפני העתקת הנתונים מ- Excel, השיגרה מכינה טבלה מקומית שתכלול אותם. היא פותחת בהצבת החיבור לפרוייקט הנוכחי בהפניה 1.הויל ו-1ачות מפנה לחברו הפרוייקט המקורי, אין צורך לכלול בהצהרה שלו את מילת המפתח New. מנגד, השיגרה יוצרת מופעים חדשים של האובייקטים Table ו- Catalog (דבר שמשתף בzechrot שליהם, בהכללת מילת המפתח New). לאחר מכן השיגרה משתמשת בקוד ADO כדי להגדיר ולצרך שדות שיקלטו את ערכי גיליון העבודה. כדי לציין את ערכי ההתחלה וערכי הצעד לאינדקס, השיגרה חזרה לקוד SQL. יכולת זאת תלואה לחלוטין בפונקציונליות מוכללת של מנגנון Jet. לפיכך, קוד SQL מותאם ספציפית למנגנון מסד הנתונים Jet. לאחר השלמת הגדרת האינדקס וצירוף לטבלה, השיגרה מכוננת חיבור אל גיליון העבודה (בדוגמה זאת נעשה שימוש באותו גיליון אלקטרוני שישמש בדוגמה הקודמת).

כל ניסיון להגדיר מחדש טבלה קיימת יחולל שגיאה שמספרה 2147217857-. השגירה מוחקקת את הטבלה הישנה ומשיכת בהוספת הטבלה החדשה. ביישום אמיתי, ייתכן שעדיף לשמר את הטבלה הישנה בארכיו.

השיגרה מתכוונת להעתיק הערכים על ידי יצירת שני אובייקטי Recordset – האחד עבור הגילוון האלקטרוני, והשני עבור הטבלה המקומית. הקוד משתמש בסמן בריית המחדל עבור הגילוון האלקטרוני, מכיוון שמוסטל עליו רק לקרוא מהגילוון האלקטרוני ערכים לפי הסדר. לעומת זאת נעשה שימוש בסמן מסווג adOpenKeyset עבור הקישור לטבלה המקומיית, כך שניתן יהיה להוסיף רשומות. מכיוון ש-Access מסוגל לחזור על חישובי Excel עם תוצאות זהות, אין צורך להעתיק שדות מחושבים. דבר זה הופך את ערכי השדה בטבלה לבלי תלוים זה בזה, כך שהטבלה מנורמלת.

הפעלת שגרות Excel מתוך שגרות

בשיגרה הבאה, runXL משתמש בפונקציה GetObject כדי ליצור מופיע של האובייקט Application של Excel, המכיל את חוברת העבודה MyGas המוצגת בתרשים 9.2. השיגרה מדירה ערך True למאפיין Visible של האובייקטים Application ו- Window. לאחר מכן היא מפעילה את השיטה Run של האובייקט Application עבור השיגרה computeOnGas בתיקיה ThisWorkbook של הקובץ MyGas.xls. אחרי שהשיגרה computeOnGas מקובץ-Excel מחרירה את השליטה ל- Access, runXL מפעילה את השיטה Save עבור ActiveWorkbook ב- Excel. פועלה זאת שומרת את השינויים ומונעת הופעתה של שאלת שמירה זאת כאשר השורה הבאה מפעילה את השיטה Quit. אם אתה מעוניין לסגור את Excel מבלי לשמור את השינויים ולא השאלת אם לבצע שמירה, הציב ערך True במאפיין Saved של חוברת העבודה וטרם תפעיל את השיטה Quit (עיין בשורת הערה כדי לברר מהו התכ婢 הנכון). איזוזר משאבי האוטומציה נעשה על ידי הצבת ערך Nothing בהפניה של אובייקט האוטומציה.

```
Sub runXL()
Dim myXLWrkBk As Excel.Workbook
' Open connection to XL workbook and make visible.
    Set myXLWrkBk = GetObject("c:\Programming Access\Chap09\MyGas.xls")
    myXLWrkBk.Application.Visible = True
    myXLWrkBk.Application.Windows("MyGas.xls").Visible = True
' Run procedure in ThisWorkBook folder.
    myXLWrkBk.Application.Run "ThisWorkBook.computeOnGas"
' Close automation object.
' Either invoke the Save method or set the Saved
' property to True to avoid a prompt about saving changes.
    myXLWrkBk.Application.ActiveWorkbook.Save
'    myXLWrkBk.Application.ActiveWorkbook.Saved = True
    myXLWrkBk.Application.Quit
    Set myXLWrkBk = Nothing
End Sub
```

תרשים 9.4 מציג את גילוון העבודה לאחר הפעלת השגירה computeOnGas. שים לב שהשגירה מחשבת נתונים מסוימים שתו שורות מתחילה לשורה האחרונה בטבלה, ומוסיפה עמודה חדשה המציגת את הקילומטראז' היומי (במילילים) בין מילוי למים מילוי הבא. השגירה מתאימה את גודל העמודות לפי הרוחב ביותר שלן.

	A	B	C	D	E	F	G	H	I
1	Date	Miles	Gallons	Price per Gallon	MPG	cents per mile	days since last refill	Total	Miles per Day
32	10/11/1996	346.1	10.407	\$ 1.38	33.256	\$ 0.041	8	\$ 14.35	43.26
33	10/20/1996	323.1	10.140	\$ 1.38	31.864	\$ 0.043	9	\$ 13.98	35.9
34	10/30/1996	291.79	9.365	\$ 1.42	31.158	\$ 0.046	10	\$ 13.29	29.18
35	11/05/1996	291.6	9.606	\$ 1.36	30.356	\$ 0.045	6	\$ 13.05	48.6
36	11/11/1996	313.7	9.695	\$ 1.36	32.357	\$ 0.042	6	\$ 13.18	52.28
37	11/18/1996	299.39	9.705	\$ 1.36	30.849	\$ 0.044	7	\$ 13.19	42.77
38	11/27/1996	336.3	10.721	\$ 1.46	31.368	\$ 0.047	9	\$ 15.64	37.37
39	12/05/1996	286.5	9.510	\$ 1.32	30.126	\$ 0.044	8	\$ 12.54	35.81
40	12/12/1996	324	10.554	\$ 1.30	30.699	\$ 0.042	7	\$ 13.71	46.29
41	12/17/1996	330.1	10.740	\$ 1.34	30.736	\$ 0.044	5	\$ 14.38	66.02
42	12/22/1996	318	9.717	\$ 1.28	32.726	\$ 0.039	5	\$ 12.43	63.6
43	01/07/1997	361	10.993	\$ 1.30	32.839	\$ 0.040	16	\$ 14.28	22.56
44	01/12/1997	347	10.314	\$ 1.32	33.644	\$ 0.039	5	\$ 13.60	69.4
45	01/18/1997	313	10.217	\$ 1.33	30.635	\$ 0.043	6	\$ 13.58	52.17
46									
47	Summary	13483.5	443.828	1.380191903	30.38	0.045	448	\$ 612.57	30.097

תרשים 9.4: קטע פלט של השגירה computeOnGas; שים לב לעמודת הנתונים החדשה, ולשינוי ברוחב העמודות

השגירה computeOnGas כרוכה בקוד VBA סטנדרטי בלבד, אולם היא משתמשת באובייקטים, במאפיינים ובשיטות ייחודיים ל-Excel. אם תבצע פעולה אוטומטית, תידרש לך באורך בלתי נמנע היכרות כלשהו עם מבנה אובייקט נספף, אחד לכל הפחות, במקרה זה מבנה האובייקט עבור יישום Office שאתה מבצע בו פעולה אוטומטית. אחד היתרונות של השימוש בשיטה Run, כמו בשירה XLChx, הוא שהשיטה מאפשרת למפתחים להתמחות במבנה אובייקט מוגדרים. כאשר מפתחת מעוניין להשתמש בפונקציה רגילה ביחסם בלתי מוכר, באפשרותו להעתיק שגירה שיעצבה בידי מפתח אחר. גם ללא היכרות יסודית עם יישום מסוים, המפתח יוכל להפעיל את השיטה Run עבור השגירה שהעתיק.

```
Sub computeOnGas()
Dim mySheet As Worksheet
Dim iRow As Integer, lastRow As Integer
Dim sumDays As Long

' Set reference to first worksheet.
Set mySheet = Worksheets(1)
lastRow = Range("gas").Rows.Count
```

```

' Assign column heading.
mySheet.Cells(1, 9) = "Miles per Day"
' Compute miles per day.
For iRow = 3 To lastRow
    mySheet.Cells(iRow, 9) = _
        Format(Range("gas").Cells(iRow, 2) / Range("gas").Cells(iRow, 7), "0.###")
    sumDays = sumDays + mySheet.Cells(iRow, 7)
Next iRow
' Compute summary statistics.
mySheet.Cells(Range("gas").Rows.Count + 2, 1).Select
ActiveCell.Formula = "Summary"
' Compute total miles.
ActiveCell.Offset(0, 1).Activate
ActiveCell.Formula = "=Sum(b2:b" & lastRow & ")" & ""
' Compute total gallons.
ActiveCell.Offset(0, 1).Activate
ActiveCell.Formula = "=Sum(c2:c" & lastRow & ")" & ""
' Compute total gas dollars.
ActiveCell.Offset(0, 5).Activate
ActiveCell.Formula = "=Sum(h2:h" & lastRow & ")" & ""
' Compute days since last refill.
ActiveCell.Offset(0, -1).Activate
ActiveCell.Formula = "=Sum(g3:g" & lastRow & ")" & ""
' Compute price per gallon.
mySheet.Cells(Range("gas").Rows.Count + 2, 4).Select
ActiveCell.Formula = "=H" & (lastRow + 2) & "/C" & (lastRow + 2)
' Compute miles per gallon.
ActiveCell.Offset(0, 1).Activate
ActiveCell = Format(mySheet.Cells(lastRow + 2, 2) / _
    mySheet.Cells(lastRow + 2, 3), "0.###")
ActiveCell.Font.Bold = True
' Compute cents per mile.
ActiveCell.Offset(0, 1).Activate
ActiveCell = Format(mySheet.Cells(lastRow + 2, 8) / _
    mySheet.Cells(lastRow + 2, 2), "0.###")
' Compute miles per day.
ActiveCell.Offset(0, 3).Activate
temp = mySheet.Cells(lastRow + 2, 2)
temp2 = sumDays
ActiveCell = Format(mySheet.Cells(lastRow + 2, 2) sumDays, "0.###")
' Resize columns to show values.
Worksheets("Sheet1").Columns("a:I").AutoFit
End Sub

```

עבודה עם Outlook מתוך Access

תוכנת Outlook משוכנת עם ערכת תיKİות סטנדרטית, הכוללת תיKİות ללוח שנה, אנשי קשר, פריטים שנמחקו, טיוות, תיבת דואר אלקטרוני נכנס, יומן, הערות, תיבת דואר אלקטרוני יוצא, פריטי דואר שנשלחו ומשימות. המשמש יכול להויסף תיKİות מותאמות אישית, ולקמן תיKİות זו בזו. המשמש מבצע פעולות בפריטים בתוך התיקיות – הוספה,CHANKE, צגוה ופונקציות נוספות.

הגירסה הראשונית של Outlook שווה עם תמייה תוכניתית רק באמצעות script של Microsoft VB. ב- 2000 מותוספת שilityה תוכניתית, בעורת VB. Outlook 2000 תומך ביצירת script הנו באמצעות VBA והן באמצעות VB Script. למען תאימות עם יתרת הספר, השעיף מתמקד ביצירת script ב- Access מותוך Outlook (Contacts).
בעורת VBA. נוסף לכך, בכל הדוגמאות נעשה שימוש בתיקיה **אנשי קשר** כדי ליצור הקשר מוכר.

באפשרותך ליצור ב- Access הפניה למופיע של Outlook באמצעות הפונקציה CreateObject. בטרם תוכל להפנות לתיקיה מסוימת, בדרך כלל عليك להחיל את השיטה GetNameSpace על האובייקט Application. האובייקט NameSpace הוא אובייקט בסיס מופשט שמתקיים בין האובייקט Application לבין התיקיות הנפרדות. השיטה מקבלת ארגומנט יחיד, שלווי להיות MAPI במהדרה הנוכחית. عليك להחיל את השיטה GetDefaultFolder על האובייקט NameSpace כדי לקבל את תיKİית ברירת המחדל של סוג נתון. השתמש בקבוע כדי להקצתו תיKİית ברירת מחדל שהישום שלך יטפל בה. הקבוע לתיקיה **אנשי קשר** הוא FolderContacts.
.iof

ספירת פריטים בתיקיה אנשי קשר

השיגרה הבאה מטפלת בתיקיה **אנשי קשר** וסומרת את הפריטים שהיא מכילה. הכו
תיקיית **אנשי קשר** לדוגמה, וככלול בה מספר עצמים כדי לבדוק את אופן פועלתה של דוגמה זאת והדוגמאות הבאות. התקליטור המצורף לספר זה כולל נתוני **אנשי קשר**. דוגמה, לצורך איכלוס של תיKİית **אנשי קשר**.

```
Sub listContacts()
Dim myOlApp As Outlook.Application
Dim myNameSpace As NameSpace
Dim myContacts As Items
Dim myItem As ContactItem
' Create an instance of Outlook.
' Reference its MAPI Namespace.
' Reference MAPI's Contact folder.
    Set myOlApp = CreateObject("Outlook.Application")
    Set myNameSpace = myOlApp.GetNamespace("MAPI")
    Set myContacts = _
        myNameSpace.GetDefaultFolder(olFolderContacts).Items
```

```

' Enumerate items in Contact folder and print selected fields.
For Each myItem In myContacts
    Debug.Print myItem.FirstName, myItem.LastName, myItem.Email1Address
Next
End Sub

```

השיגרה פותחת בהצהרה על ארבעה משתנים : אחד לישום Outlook, אחד לאובייקט NameSpace שלו, אחד לאוסף הפריטים שבתיקיה **אנשי קשר**, ואחד למספרם אותם פריטים. דרישים שלושה משבטי Set כדי לחושף את פרטי התיקיה **אנשי קשר**. האחزوן משתמש בשיטה GetDefaultFolder כדי לחזור לתיקיה **אנשי קשר**, וכן במאפיין Items כדי לגשת אל הפריטים הנפרדים. הספירה מתרחשת בלולאת For...Each. לפיריטים בתיקיה **אנשי קשר** סדרה של מאפיינים וביהם פריטים אודוות **אנשי קשר**. בדוגמה נעשה שימוש בשלושה מהמאפיינים הללו כדי להדפיס שם פרטי, שם משפחה וכתובת דואר אלקטרוני ראשונה עבור כל ערך בתיקיה **אנשי קשר**.

הוספת פריט לתיקיה אנשי קשר

באפשרות לבנות גם פתרונות המבוססים על Access, המטפלים בתוכולת התיקיה **אנשי קשר**. מבין שלוש השגורות הבאות, הריאונה, הריאונה, addOneContact, מוסיפה איש קשר חדש בתיקיה. השיגרה משתמשת בקבועי מחuzeות להגדרת השם הפרט, שם המשפחה וכתובת דואר אלקטרוני עבור איש קשר, אולם אין קווי לשנות את השיגרה כך שתעביר נתונים אלה כארוגומנטים. זה בדוק מה שעשוות שתי השגורות הבאות, removeOneEmail ו- deleteAContact. השיגרה removeOneEmail מעבירה כתובת דואר אלקטרוני לשיגרה deleteAContact, מ attractiveness פריט 'איש קשר' בעל כתובת תואמת, ומוחקת פריט זה.

```

Sub addOneContact()
Dim myOlApp As Outlook.Application
Dim myItem As ContactItem
' Create an instance of Outlook.
    Set myOlApp = CreateObject("Outlook.Application")
' Create an item for the folder.
' Populate the item with values.
' Save the item.
    Set myItem = myOlApp.CreateItem(olContactItem)
With myItem
    .FirstName = "foo"
    .LastName = "bar"
    .Email1Address = "foobar@yourcompany.com"
    .Save
End With
End Sub

```

```

Sub removeOneEmail()
    deleteAContact ("foobar@yourcompany.com")
End Sub

Sub deleteAContact(strEmail)
Dim myOlApp As Outlook.Application
Dim myNameSpace As NameSpace
Dim myContacts As Items
Dim myItem As ContactItem
' Create an instance of Outlook.
' Reference its MAPI Namespace.
' Reference MAPI's Contact folder.
    Set myOlApp = CreateObject("Outlook.Application")
    Set myNameSpace = myOlApp.GetNamespace("MAPI")
    Set myContacts = _
        myNameSpace.GetDefaultFolder(olFolderContacts).Items
' Enumerate to search for item to delete.
    For Each myItem In myContacts
        If myItem.Email1Address = strEmail Then
            myItem.Delete
            Exit Sub
        End If
    Next
' No entry found
    MsgBox "No entry found with email of " & strEmail, vbCritical, _
        "Programming Microsoft Access 2000"
End Sub

```

לשיגרה זאת דרישים שני אובייקטים בלבד – אובייקט Application של Outlook ואובייקט ContactItem שייצג פריט בתיקיה **אנשים קשר**. השיגרה יוצרת הפניה לאובייקט Application בעזרת הfonקציה CreateObject. הפניה זאת תומכת בשיטה CreateItem, שיוצרת מופע ריק של פריט עבור כל תיקיה שצוינה. עליך לקבוע את סוג התיקיה של הפריט באמצעות קבוע שיעבור לשיטה CreateItem. להגדרת מאפייניו של איש קשר,בחר מתוך רשימה של מעל 140 מאפיינים. הדוגמה מקצת קבועי מחרוזות עברור המאפיינים LastName ו- Email1Address (בוחלט ייתכן שלא יהיה קשר מסוים יותר מכתובת דואר אלקטוריונית אחת). לאחר מכן השיגרה מפעילה את השיטה Save כדי לשמור את הערך החדש בתיקיה **אנשים קשר**.

מחיקת פריט מהתיקיה אנשים קשר

השיגרה deleteAContact (המופיעה לעיל) מקבלת כารגוمنت מחזרות המייצגת את ערך המאפיין Email1Address של פריט 'איש קשר' שיש למוחקו. השיגרה מונה את איברי התיקיה **אנשים קשר**, עד שהיא מאתרת פריט בעל מאפיין Email1Address תואם

לארוגומנט המועבר. כאשר נמצאת התאמה, השיגרה מוחקת את הפריט על ידי הפעלת השיטה Delete, ויזכאות מהשיגרה כדי למנוע את המשך החיפוש. אם נמנעו כל פריטי התקינה **אנשי קשר** מבלי שנמצא ערך תואם, השליטה עוברת למשפט של תיבת הودעה, המדוחחת שלא נמצא ערכים תואמים לכטובת הדואר האלקטרוני שהועברה לשיגרה.

הוספת פריטים מרובים לתיקיה אנשי קשר

אחד המשימות הנפוצות המבוצעות בעזרת מנהל מסד נתונים, כמו למשל Access, היא הוספת פריטים מרובים לתיקיה **אנשי קשר**. פריטים אלה יכולים להתווסף ממוקורות מגוונים, כמו למשל התקינה **אנשי קשר** במחשב אחר, כתובות שהזנו דרך האינטרנטי, ואפילו קובץ אנשי קשר יישן ב-Access. השיגרה addContacts המובאת להלן משתמש באחת הגישות לעדכון תיקיית **אנשי קשר** של Outlook בתוויני אנשי קשר מתוך טבלה .Access של

```
Sub addContacts()
Dim myOlApp As Outlook.Application
Dim myItem As ContactItem
Dim rst1 As New Recordset
' Open the Contacts folder in Outlook.
    Set myOlApp = CreateObject("Outlook.Application")

' Open the table with the new contacts.
    With rst1
        .ActiveConnection = CurrentProject.Connection
        .Open "oe4pab"
    End With
' Create a ContactItem for adding contacts and
' loop through the table records to add them to the folder.
    AssistantWorkingOn
    Do Until rst1.EOF
        Set myItem = myOlApp.CreateItem(olContactItem)
        With myItem
            .FirstName = IIf(IsNull(rst1.Fields(0)), "", rst1.Fields(0))
            .LastName = rst1.Fields(1)
            .Email1Address = rst1.Fields(2)
            .Save
        End With
        rst1.MoveNext
    Loop
    AssistantIdleOn
End Sub
```

השיגרה מציבה הפניה לישום של Outlook, ולאחר מכן פותחת ערכת רשומות המבוססת על הטבלה oe4pab. זהה הטבלה המקומית בתיקיה **טבלאות** (Tables) שב-Access. הטבלה מכילה 34 ערכים בלבד, ואולם השיגרה מסוגלת לקלוט רשימה ארוכה הרבה יותר של כתובות. סיבת זאת השיגרה מפעילה שירה נוספת, המפעילה את המשיע בהפנשת 'עובד' ומשאירה אותו במצב פעיל עד ש- Access ו- Outlook יסיימו את עדכון התקינה **ארגוני קשר** ב- Outlook בערכיהם מתוך הטבלה oe4pab (פרק 8 מתאר כיצד לבנות שגורות שישלטו בהפנשת המשיע. תוכל למצוא את השגורות הללו בספריה Chap08 בתקליטור הנלווה). בין שתי הקריאות, האחת להפעלת הפנשת המשיע והשנייה לביטולה, לוatta Do עוברת במחזריות על כל הרשומות שבטבלה oe4pab. הלואה יוצרת אובייקט ContactItem חדש בכל מעבר, ולאחר מכן מציבה את הרשומות ממעבר זה בפריט, ושומרת את הפריט.

מחיקת פריטים רבים מהתיקיה ארגוני קשר

השיגרה הבאה, removeEmails, היא גירסה מותאמת של השיגרה שהובאה מוקדם יותר. שיגרה זאת מוחקת רשומות רבות מתוך תיקיית **ארגוני קשר**. היא מוחקת את הפריטים אחד לאחר, על ידי קריאות חוזרות לשיגרה שבטבלה oe4pab כמקור לארגומנטים. לשיגרה זאת שני יתרונות: קל לכתוב אותה, ויש בה ניצול חוזר של השיגרה deleteAContact.

```
Sub removeEmails()
Dim rst1 As New Recordset
' Open the table with the new contacts.
With rst1
    .ActiveConnection = CurrentProject.Connection
    .Open "oe4pab"
End With
' Loop through the table records to add them to the folder.
AssistantWorkingOn
Do Until rst1.EOF
    deleteAContact (rst1.Fields(2))
    rst1.MoveNext
Loop
AssistantIdleOn
End Sub
```

אם כי שיגרה זאת יכולה לבצע את המשימה, יש בה לפחות שני חסרונות. ראשית, השיגרה עוברת על כל ערכיו התקינה **ארגוני קשר** עבור כל פריט המועד למחיקה. דבר זה הופך יקר יותר בהتمדה ככל שגדל מספר הפריטים המיועדים למחיקה, מספר הפריטים בתיקיה עצמה, או שניהם גם יחד. שנית, אם לא נמצא פריט תואם, השיגרה מציגה תיבת הודעה המחייבת את המשתמש לחוץ על לחץ כדי להמשיך. אם מספר

רב של ערכים ברשימה הפריטים למחיקה איןנו נמצא עוד בתיקיה **אנשי קשר**, הכוורת להזען על לחוץ עבר כל פריט שחשר בתיקיה יכול להיות מייגע. אחת הדרכים להתגבר על חסכנות אלה היא להחליף את השגירה deleteAContact בקריאה לשגירה deleteAContact2.

```
Sub deleteAContact2(strEmail)
On Error GoTo delete2Trap
Dim myOlApp As Outlook.Application
Dim myNameSpace As NameSpace
Dim myContacts As Items
Dim myItem As ContactItem
Dim strFilter As String
' Create an instance of Outlook.
' Reference its MAPI Namespace.
' Reference MAPI's Contact folder.
    Set myOlApp = CreateObject("Outlook.Application")
    Set myNameSpace = myOlApp.GetNamespace("MAPI")
    Set myContacts = myNameSpace.GetDefaultFolder(olFolderContacts).Items

' Find target item and remove it.
    strFilter = "[Email1Address] = """ & strEmail & """
    Set myItem = myContacts.Find(strFilter)
    myItem.Delete
delete2Exit:
    Exit Sub
delete2Trap:
    If Err.Number = 91 Then
        ' If item is not there, just keep on going.
        Resume Next
    Else
        ' Otherwise, pause with a message box.
        MsgBox Err.Number & ":" & vbCrLf & Err.Description, vbCritical, _
            "Programming Microsoft Access 2000"
        Resume Next
    End If
End Sub
```

שיגרה זאת מחישה את החיפוש אחר פריט המועד למחיקה באמצעות השיטה Find. היא בונה קритריון לשיטה Find, המבוסס על כתובות האלקטרונית המועברות אליה. לאחר שהיא מאתרת פריט בעל כתובות אלקטרונית תואמת, השגירה מchildה את השיטה Delete על אותו פריט. השימוש ב- Find מניב חיסכון ניכר בזמן, גם כאשר מדובר ברשימת פריטים קצרה, כמו זו את שבבלה 0e4pab, אולם יתרון המהירות גדול ככל שגדלה רשימה הכתובות האלקטרונית, או מספר הפריטים, כמו בתיקיה

ארגוני קשר. השגירה לוכדת גם מקרי כשל, כאשר השיטה Find אינה מחזירה פריט. דבר זה קורה כאשר אין בתיקיה **ארגוני קשר** פריט תואם לפי כתובות אלקטרוניות. במצב כזה, השגירה deleteAContact2 מחזירה שקט את השילטה לשגירה הקוראת, כדי שזו תוכל לחפש כתובות אלקטרוניות חדשות, ללא צורך בהთערבות המפעיל.

עבודה עם Word מתוך Access

הדוגמה הראשונה בסעיף זה מראה כיצד לבנות ולבצע טבלה ב- Word, ולאחר מכן בנתונים מתוך טבלה ב- Access. בדוגמה נעשה שימוש באוטומציה כדי לשלוט ב- Word מתוך Access, והיא כוללת אפליו טופס פשוט להפעלת שגרת האוטומציה. שתי הדוגמאות הבאות מטפלות באמצעות קוד בשתי דרכים של מיזוג:Dואר : הפקת תוויות מען, והפקת מכתב אחד.

דוגמת המכתב היחיד המובאת כאן מבצע פעולה זהה לזו של דוגמת המכתב היחיד בפרק 6. תוכל להשוות בין שתי הגישות כדי לברר איזו מתאימה יותר לצרכיך. בכלל, הרעיון שמאחורי רכיבים מרובים הוא שרצוי להשתמש בכל אחד מהם לביצוע המשימה שהוא מיטיב לבצע. הדוגמה בפרק זה מאפשרת שבירת נתוניים ב- Access, והפקת מכתבים איחדים להדפסה ב- Word. אם אתה מעדיף לעבוד עם מבנה האובייקט של Word ועם האובייקט MailMerge, ייתכן שגישה זאת תהיה לשרת אותך. לעומת זאת, אם אתה מעדיף להתרץ בהתקומות ב- Access, ייתכן שעדיפה הגישה המובאת בפרק 6.

אוטומציה של Word מתוך Access

דוגמת האוטומציה הבאה מעבירה נתונים מערכת רשותם המבוססת על טבלה ב- Access, לטבלה ב- Word. כאשר יש לך הפניות למבני אובייקטים מרובים בעלי מונחים דומים, עליך להוסיף בהצהרה על האובייקט קידומת לפני הקצתה סוג הנתוניים של האובייקט: לדוגמה, השתמש ב- Word.Table במקומו ב- Table. פעלזה זאת מורה למתרגמים Aiזה אובייקט Table נדרש לך. זכור, שם Access יכול להציג על פניו אובייקט Table מתוך ספריית ADOX. וכור גם שהתנהגות האובייקט Table ב- Word Range שונה Aiזה אובייקט Table Excel. Intellisense VBA ו- VBA מספקים לך לבנות פתרונות על פניו שונה מהתנהגותו ב- Excel. מספר יישומים, אולם אין הם פוטרים לכך מהכרת מבני האובייקט ביישומים השונים.

```
Sub fromAccessToWordTable()
Dim myWDApp As Word.Application
Dim myRange As Word.Range, myTable As Word.Table
Dim acell As Word.Cell, emailCol As Integer
Dim rst1 As New Recordset, irow As Integer
```

```

' Open the table with the new contacts.
With rst1
    .ActiveConnection = CurrentProject.Connection
    .Open "oe4pab", , adOpenKeyset, adLockOptimistic, adCmdTable
End With

' Create a Word application instance and turn on
'the Assistant's working animation.
AssistantWorkingOn
Set myWDApp = CreateObject("Word.Application")

' Add a document to the application and a table to the document.
' Specify rows to equal one more than e-mail address table in Access.
myWDApp.Documents.Add
Set myRange = myWDApp.ActiveDocument.Range(0, 0)
myWDApp.ActiveDocument.Tables.Add Range:=myRange, _
    NumRows:=rst1.RecordCount + 1, NumColumns:=3

' Insert column headings for table.
With myWDApp.ActiveDocument.Tables(1).Rows(1)
    .Cells(1).Range.Text = rst1.Fields(0).Name
    .Cells(2).Range.Text = rst1.Fields(1).Name
    .Cells(3).Range.Text = rst1.Fields(2).Name
End With

' Insert first name, last name, and e-mail from Access table.
' Insert contact info in the second through the last row.
For irow = 2 To myWDApp.ActiveDocument.Tables(1).Rows.Count
    emailCol = 0
    For Each acell In myWDApp.ActiveDocument.Tables(1).Rows(irow).Cells
        acell.Range.Text = IIf(IsNull(rst1.Fields(emailCol)), "", _
            rst1.Fields(emailCol)))
        emailCol = emailCol + 1
    Next acell
    rst1.MoveNext
Next irow

' Format table to fit content, turn on idle animation, and
'make Word visible so user can see table in Word.
myWDApp.ActiveDocument.Tables(1).AutoFitBehavior wdAutoFitContent
AssistantIdleOn
myWDApp.Visible = True

End Sub

```

השיגרה מתחילה בפתיחת אובייקט ערכת רשומות המבוסס על טבלה ב- Access, הטבלה oe4pab המוכרת מהדוגמאות על עובדה עם Outlook מתוך Access. מכיוון שהשימוש משתמש במאפיין RecordCount, עליך להימנע משימוש בסמן מסוג קדימה-בלבד (forward-only). בדוגמה נעשה שימוש בקבוע adOpenKeyset לציון סוג הסמן. לאחר פתיחת ערכת הרשותות, היישום מפעיל את הנפשה העובדה של המסייע, ומריץ את הפונקציה CreateObject כדי ליצור מופיע חדש של Word.

בשלב הבא, השיגרה בונה את הטבלה ב- Word. היא יוצרת מסמך חדש ומוסיפה Table בפינה הימנית העליונה שלו (במסך עברי). הפרמטרים של השיטה Add באובייקט Table קובעים שבטבלה החדשה מספר השורות יהיה גדול באחד מספר השורות בטבלה של Access. דבר זה מותיר מקום (שורה אחת) לכותרות העמודות, נוספת לכל הנתונים בטבלה oe4pab. לפני שהיא מתחילה לטפל בערכי ערכות הרשותות, השיגרה כותבת את כותרות העמודות בשורה הראשונה. כותרות אלה הן שמות השדות בטבלה של Access.

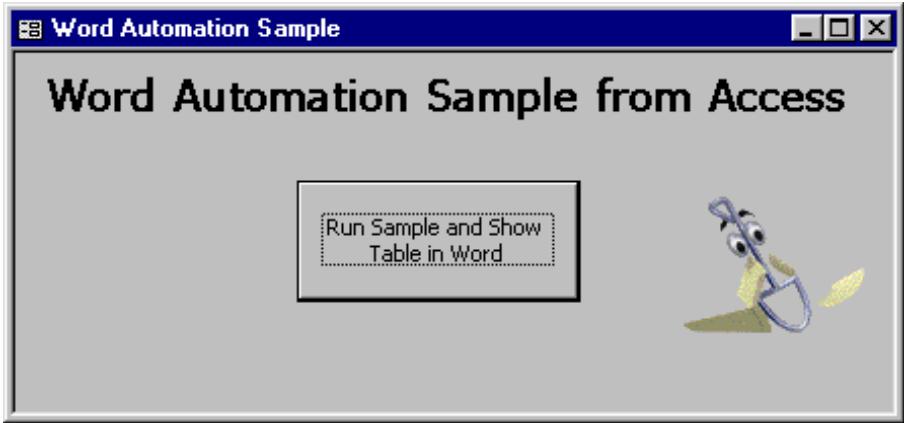
זוג לולאות For מקונות מסיירות על פני כל התאים בטבלה. הלולאה החיצונית מתקדמת בין השורות, לפי הסדר. הלולאה הפנימית 'צעדי' לרוחב העמודות בתוך שורה נתונה. שים לב, שב- Word קיים אוסף Cells עבור התאים הנפרדים בתוך שורה נתונה. הלולאה הפנימית מסיירת בין התאים הנפרדים בתוך שורה נתונה בטבלה. היררכיה אל תא נפתחת באובייקט האוטומציה myWDAppl, ולאחר מכן עוברת לאופן הפניה אל תא ActiveDocument, לטבלה הראשונה באותו מסמך, ולשורה נתונה בתוך הטבלה. לאחר שזוזה תא המועד לטיפול בתוך שורה מסוימת, פונקציית IIf המתבססת על ערכי ערכות הרשותות מחשבת ערך עבור המאפיין Text של אותו תא.

לאחר שהשיגרה עברת במחזוריות על פני כל תא הטבלה, היא מסתiyaמת בכך שהיא מבצעת שלוש פעולות. תחילת היא מעצבת מחדש את רוחב העמודות כך שיהיה רחבות די הצורך להציג ערכי העמודות ללא גליישת שורות. שנית, היא מקצת הנפשה מצב לא-פעיל למשיע. בפועל, דבר זה מפסיק את הנפשה העובדה המתחילה ממש לפני הפעלת האובייקט Application של Word. לבסוף, השיגרה מציבה ערך True במאפיין Visible של האובייקט Application. פעולה זאת מעבירה את המיקוד מ- Word ל- Access.

תרשים 9.5 מציג טופס של Access המאפשר למשתמש להפעיל את השיגרה בלחיצה על לחץ. התרשים מראה את הטופס מייד לאחר שהמשתמש לחץ על הלחץ. דמותו המשיע עדיין נמצא בהນפשה העובדה שלה. כאשר המשיע עבר למצב לא-פעיל, המיקוד עבר מהטופס של Access למסמך Word המכיל את הטבלה שיצרה השיגרה .fromAccessToWordTable

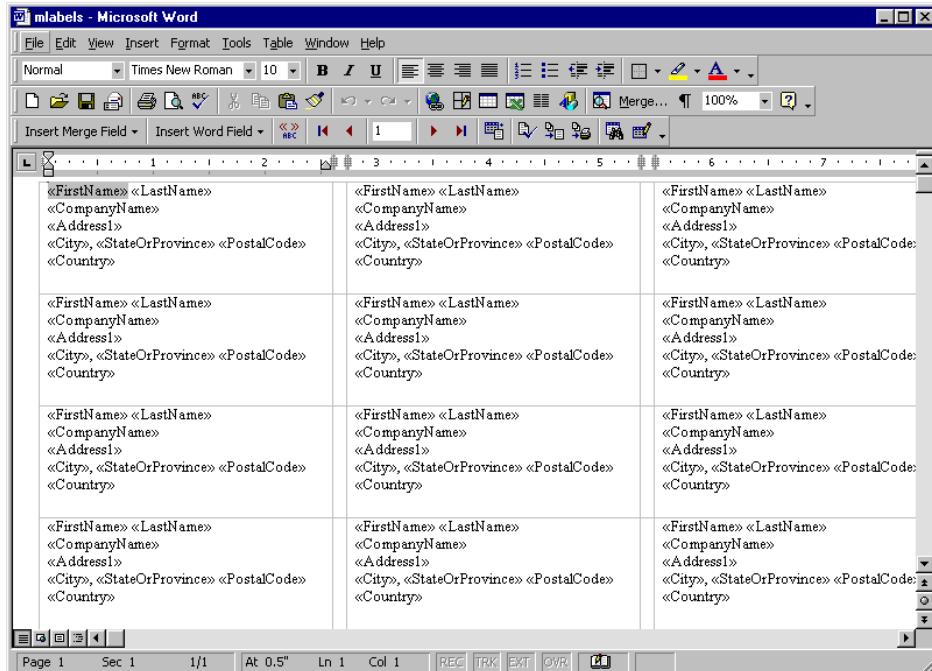
הפקת תוויות מען

תוכנת מיזוג הדואר המוכללת ב- Word יכולה לשאוב נתונים מתוך שאילתת או טבלת מסד נתונים של Access, המשמשים כמקור רשותות להפקת תוויות מען. אומנם ניתן למקם באמצעות קוד את שדות מיזוג הדואר, או הסימניות, כך שיתמכו בהצבת נתונים Access במסמכי Word, אולם פעמים רבות קל יותר למקם אותן ידנית במקומות שבהם אתה מבקש להוסיף נתונים במסמך. נוסף לכך, ניתן להיעזר באשפים מוכללים לצורך הפרישה של פקדים בתבניות עם תוויות מרובות בכל עמוד.



תרשים 9.5: טופס זה מפעיל את השגירה `fromAccessToWordTable`; כאשר נסकת הנפשת העבודה במסיע, המיקוד עובר לטליה שב-Word

תרשים 9.6: שלහלו מוצג קטע מתוך הקובץ `mlabels.doc`. שדות מיוזוג הדואר מוקמו בעזרת אשף מיוזוג הדואר המוכלל, אשר ניתן להפעיל מתפריט **כליים** (Tools). לאחר שימוש שפרסת את הפקדים, תוכל לשולט באמצעות קוד בהדפסת תוויות `.MailMerge` על טבלאות או שאלות של `Access`, באמצעות האובייקט



תרשים 9.6: קטע מתוך מסמך Word שנוצר לצורך הדפסת תוויות מען בתבנית Avery 5260

שגרת Word הבאה מיעדת להפעלה מתוך התיקיה ThisDocument של התבנית Normal. היא מתחילה בפתיחת הקובץ mlables.doc, המכיל את שדות מיזוג הדואר הנראים בתשרים 9.6. לאחר מכן היא מציבה הפניה, myDoc, אל מסמך המבוסס על אותו קובץ. למעשה, הקובץ mlables.doc אינו מכיל קוד שלו.

```
Sub printPreviewLabels()
Dim myDoc As Document

' Set reference to the document.
Documents.Open FileName:="mlables.doc"
Set myDoc = Documents("mLabels")

' Reference the data source in Access.
myDoc.MailMerge.OpenDataSource Name:= _
    "C:\Programming Access\Chap09\Chapter9.mdb", _
    Connection:="TABLE WebBasedList"

' Send the labels to a new document.
With myDoc.MailMerge
    .Destination = wdSendToNewDocument
    .Execute Pause:=True
End With

' Either Print Preview or Print labels.
If MsgBox("Do you want to preview before printing?", _
    vbYesNo, "Programming Microsoft Access 2000") = vbYes Then
    ActiveDocument.PrintPreview
Else
    ActiveDocument.PrintOut
End If
End Sub
```

המפתח למיזוג דואר עם נתוניים מ- Access הוא להפעיל את השיטה OpenDataSource האובייקט MailMerge בתחום מסמך, כמו למשל המסמך שההפקה myDoc מצביעה עליו. פעמים רבות תידרש להקצות שני ארגומנטים, Name (שם) ו- Connection (חיבור). עליך להציב את Name כך שיביע על קובץ המקור ב- Access (חיבור). המכיל את הנתונים למיזוג הדואר. השירה printPreviewLabels מפנה אל הקובץ Chapter9.mdb, פרק זה בתקליטור הנווה. השתמש בארגומנט Connection כדי להקצות לאובייקט מסד הנתונים סוג ושם. בעזרה הקוד תוכל להקצות סוג Table או סוג Query. הדוגמה מצביעה אל טבלה בשם WebBaseList.

השתמש באובייקט MailMerge לכתיבה על גבי התבנית עם שדות מיזוג הדואר, או ליצירת מסמך חדש שיכיל את הנתונים לאחר המיזוג. עליך להשתמש במאפיין Destination של האובייקט MailMerge כדי להקצות את האפשרות שבחרת. בדוגמה

נעשה שימוש בקבוע כדי ליצר מסמך חדש שיכיל את הנתונים לאחר המיזוג. אחרי שתגדיר את כל מאפייניו MailMerge, החל את השיטה Execute על האובייקט MailMerge. פעולה זאת תגרום למיזוג הנתונים מתוך Access במסמך Word היוצר את תוכיות המعن. אם אין צורך עוד את המסמך עם השדות הממוזגים ואת המסמך השולט בפריסת השדות במסמך, סגור את שניהם ידנית, או בעורת תוכנית אחרת.

ניתן להפעיל בקלות כביכול שירה כמו printPreviewLabels מתוך Access. השירה הבאה עשויה זאת בשלוש שורות בלבד. דוגמה זאת מופעלת מתוך מודול סטנדרטי בקובץ Access.

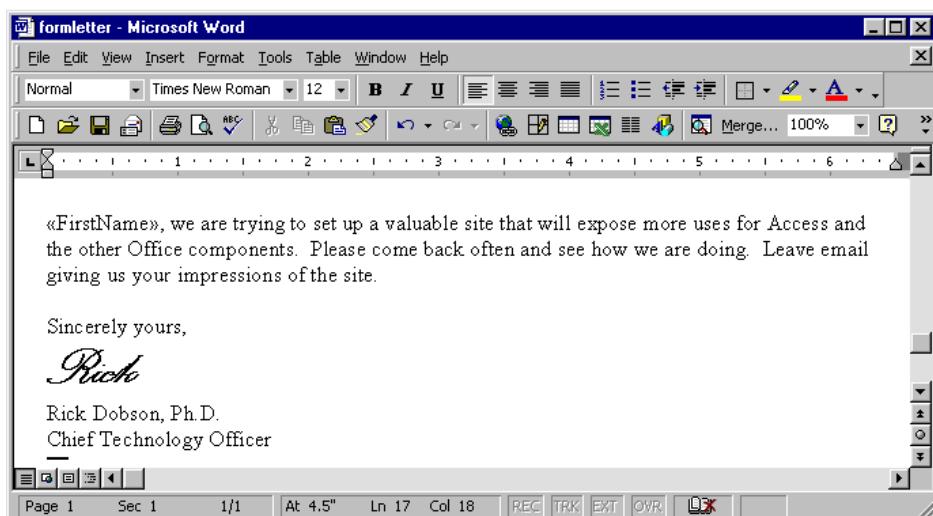
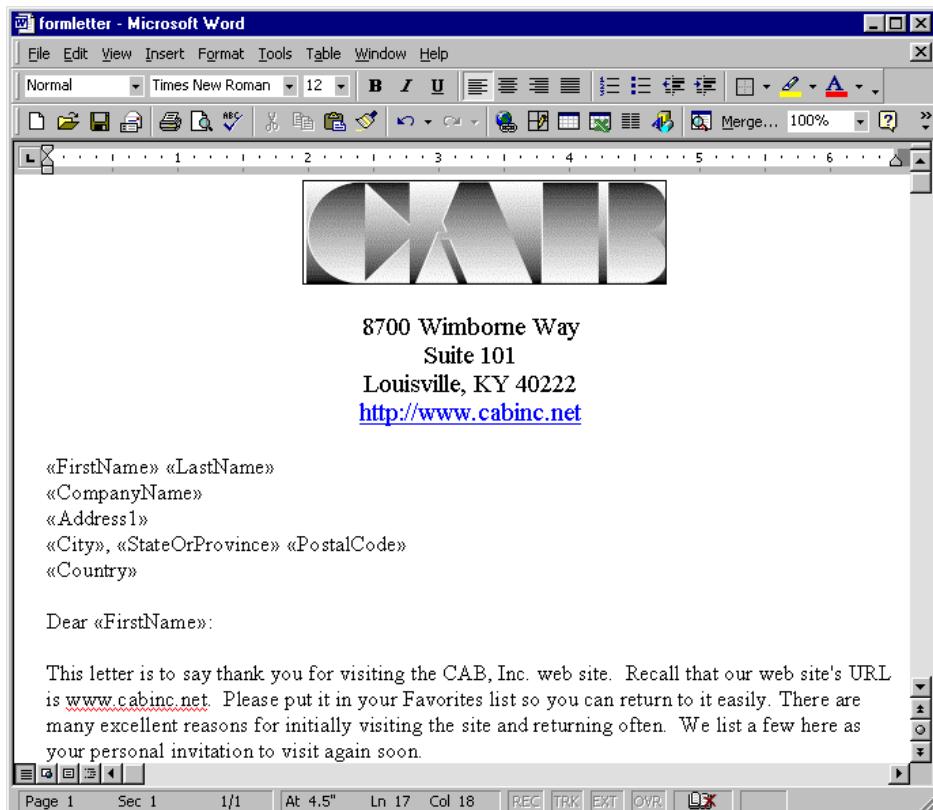
```
Sub runMLabels()
Dim myWDApp As Word.Application
' Open connection to Word and make Word visible.
    Set myWDApp = CreateObject("Word.Application")
    myWDApp.Application.Visible = True
' Run mailing label procedure.
    myWDApp.Application.Run "printPreviewLabels"
End Sub
```

הפקת מכתב אחיד

השירה להפקת מכתב אחיד המובאת להלן זהה, בעיקרה, לשירה המפיקה תוכיות מען, להוציא פריסת שדות מיזוג הדואר, הנעשית במתכונת ישירה יותר. דבר זה נובע מן העובדה שבדרך כלל אין יותר מימוש אחד של כל רשומה בעמוד נתון, בניגוד לרשומות רבות בכל עמוד. תרשימים 9.7 מציג פריסת של מכתב אחיד ב-Word, שהינה זהה לפрисת המכתב שהוזג בפרק 6. אין קושי בהצבת שדה מיזוג דואר בגוף המכתב. דבר זה נראה בלוח התחתון של התרשים.

אל תשמש באשף מיזוג הדואר אם בכונתך לשנות בהדפסת התוכיות באמצעות קוד. הקוד שהוא מפיק מכל פרמטרים רבים נוספים, ומיועד במפורש לשימוש עם האשף, אך לא בהכרח עם יישומים מותאמים אישית. הדוגמה הבאה וכן זאת שבסעיף הקודם מיועדות לשימוש מתוך התיקייה ThisDocument של התבנית Normal. דוגמאות אלה הן נקודות זינוק מצוינות לעיצוב יישומי מכתבים אחידים ותוכיות מען משלך. עם זאת, عليك להפעיל את אשף מיזוג הדואר כדי להוסיף בחולון במסמך את סרגל הכלים **מיזוג דואר** (Mail Merge). לאחר שחשפת את סרגל הכלים, תוכל להוסיף שדות מיזוג דואר בכל מקום רצוי במסמך.

השירה הבאה פועלת עם המסמך הנראה בתרשימים 9.7. תרשימים זה מציג קטיעים מראש הקובץ formletter.doc ומחתתיתו. השירה פותחת קובץ זה. השימוש בשיטה OpenDataSource זזה להפקת מכתב אחיד או תוכיות מען. בדוגמה זאת, מוקצים ערכיים למאפיינים FirstRecord ו- LastRecord של האובייקט DataSource. השיטה OpenDataSource מצינית פרמטרים המצביעים אל מקור הנתונים שהאובייקט DataSource מייצג. הצבת ערכים במאפיינים FirstRecord ו- LastRecord מגדירה טווח של רשומות שעבור כל אחת מהן יש להציג מכתב אחיד. דוגמה זאת פועלת עם סדר ברירת המחדל למילון רשומות במקור הנתונים.



תרשים 9.7: קטעים מטור מסמך Word המשמש להדפסת מכתבים אחידים

```

Sub printPreviewLetters()
Dim myDoc As Document
' Load file and set a reference to it.
    Documents.Open FileName:="formletter.doc"
    Set myDoc = Documents("FormLetter")
' Reference the data source in Access.
    myDoc.MailMerge.OpenDataSource Name:= _
        "C:\Programming Access\Chap09\Chapter9.mdb", _
        Connection:="TABLE WebBasedList"
' Send the labels to a new document.
    With myDoc.MailMerge
        .Destination = wdSendToNewDocument
        With .DataSource
            .FirstRecord = 5
            .LastRecord = 9
        End With
        .Execute Pause:=True
    End With
' Either Print Preview or Print labels.
    If MsgBox("Do you want to preview before printing?", _
        vbYesNo, "Programming Microsoft Access 2000") = vbYes Then
        ActiveDocument.PrintPreview
    Else
        ActiveDocument.PrintOut
    End If
End Sub

```

ניתן להפעיל את השגירה מתוך Access על ידי הפעלת השיטה Run עבור אובייקט Application של Word, כפי שקרה השגירה הבאה. לשם כך דרישות שלוש سورות בלבד. בעוד שהשגרות runLabels ו- runFormletters ישרו שגרות למודול סטנדרטי ב- Access, תוכל להציב כל אחת מהן, או את שתיהן, מאחרורי טופס של Access ולהפעיל אותן באמצעות לחצני פקודה. אם הקוד שלבelow מצבב שאלתה בפרמטר OpenDataSourceConnection, תוכל לשנות את השאלה בטרם תפעיל את מיזוג הדואר ב- Word, וזאת כדי לשפר את שלוות המיזוג.

```

Sub runFormLetters()
Dim myWDApp As Word.Application
' Open connection to Word and make Word visible.
    Set myWDApp = CreateObject("Word.Application")
    myWDApp.Application.Visible = True
' Run form letter procedure.
    myWDApp.Application.Run "printPreviewLetters"
End Sub

```


עבודה עם מסדי נתונים מרובי-משתמשים

בחלק ניכר מشارית ספר זה, עוסוק בבנייה יישומי Microsoft Access לקבוצות של משתמשים. פרק זה עוסק בכמה סוגיות מרכזיות הנוגעות לריבוי משתמשים: שיתוף קבצים ושליטה באבטחה. תוך סקירה נושא שיתוף הקבצים, הפרק עוסק בבדיקה מפורשת של טכניות ושאלות בנוגע לשיתוף קבצים וערכות רשומות. בפרקים הבאים נבחן טכנולוגיות לשיתוף מסדי נתונים של Access, כמו למשל שכפול מסדי נתונים, שימוש בשרת SQL, ושימוש במסדי נתונים של Access באמצעות האינטראנט. בכתיבת פרק זה הנחנו כי יש לך ידע מעשי בסוגיות בסיסיות של ריבוי משתמשים ואבטחה, וכן התחמכנו בהתקויות חדשות בתחום הקוד. לדוגמה, הפרק כולל דוגמאות המשוות את הנעה החדשנה, ברמת השורה, לנעה ברמת העמוד שהיתה מקובלת עד כה בטיפול בערכות רשומות, וכן דוגמאות המראות כיצד לשЛОט באבטחה באמצעות קוד דרך הספריות ADODB ו- ADOX החדשנות. הפרק מסתיים בדוגמה של טרנזקציות בסביבה מרובת משתמשים.

שיטוף קבצים

הצעד הראשון בשיתוף יישום של Access הוא הפעלת מצב ברירת המחדל לפתיחת לשיתופי. תוכל לעשות זאת באמצעות הדרטistica מתקדם (Advanced) בתיבת הדו-שיח אפשרויות (Options). הדרטסיה זאת מכילה עוד אפשרות המקדמת את השיתופי. השיטות SetOption ו- GetOption של האובייקט Application של Access מאפשרות ליישומיך לקבוע הגדרות אלה ולקראן אותן.

הצב קבצי Access מרובי-משתמשים בתיקיות של ספריות קבצים מסוותפות. מכיוון ש- Access עם מגנון Jet הוא שרת קבצים ולא מסד נתונים שרת-לקוח, عليك לשאוף לצמצם את גודל הקבצים העוברים בראשת הפיסיטה. אחת השיטות לעשות זאת היא לפצל יישום לשני קבצים. שמור את טבלאות הנתונים בקובץ Access בתיקיית הספריות המשותפות, והפץ בין שאר המשתמשים קובץ Access אחר, שימושו אותן בתחום העבודה המקומיות. בקובץ המופץ ייווצר קישור בחזרה אל טבלאות הנתונים בתיקיות המשותפות. עיצוב מסוג זה מחיש את העבודה ומפחית את נפח התעבורה בראשת. לרוב ניתן לשפר את רמת הביצועים על ידי הכללת עוד תכנים בקובץ המופץ, כמו למשל נתונים שאינם משתנים לעיתים קרובות. תוכל להשתמש בשגרות אוטומטיות לעדכון נתונים מקומיים בעקבות אירועים קבועים מראש, כמו למשל עם פיתוח היישום, או בתגובה להחיצת לחץ על ידי המשתמש.

הערה:



תוכל להשתמש באשף פיצול מסדי הנתונים כדי לפצל קובץ Access לשני חלקים (בחר באפשרות **עררי מסד נתונים מתוך טפריט כלים**). אחד הקבצים יכול את הטבלאות, ואילו الآخر יכול שאלות, טפסים, דוחות, פקודות מאקרו, מודולים וקיים דרך לדפי גישה לנתונים (דף גישה לנ נתונים הם אלמנט חדש ב- 2000 Access). הצב את הקובץ הראשון בתיקיה משותפת בראשת, וקשר אליו את הקובץ השני. הפץ את הקובץ השני בין המשתמשים בתחום העבודה השונות.

באפשרותך לקבוע הגדרות אבטחה ברמת המשתמש, כך שימוש נתון יוכל לפתחו קובץ מסוים (ומستخدمים אחרים לא יוכלו להשתמש בו). שימוש בלבדי במסד נתונים עשוי להיות נחוץ למשתמש על מנת לבצע שינויים, כמו למשל הוספה מודולים חדשים, או בדיקה חוזרת של מודולים קיימים. בסביבה מרובת משתמשים, הגישה הבלתי למסד נתונים מתבצעת באמצעות המאפיין Connection Control והתקינה רשימת משתמשים (User List).

המאפיין Connection Control

המאפיין החדש Connection Control מקל על השליטה ב涅גשות למסדי נתונים באמצעות קוד. ניתן להגדיר את המאפיין זהה עבור האובייקט Connection של ADO (ActiveX Data Objects) כך שלא יתאפשר למשתמשים חדשים להתחבר אל מסد

נתונים, וממשמשים קיימים לא יורשו להתחבר אליו מחדש לאחר שסגורו את הקובץ. המאפיין Connection Control מאפשר לך להשיג גישה בלעדית אל מסד הנתונים כדי לבצע בו שmericה של מודולים חדשים ושינויים אחרים בעיצובי היישום לאחר שאחרון המשמשים הנוכחיים ניתק את הקשר עם מסד הנתונים.

שתי השגרות הבאות משנות את הגדרות המאפיין Connection Control. השתמש באוסף Properties של האובייקט Connection כדי להגדיר את המאפיין. ערך של 1 כהגדרת המאפיין גורם לתכונה לסגירה פסיבית כאשר המשתמשים מתוקים את הקשר עם קובץ היישום. ערך של 2 מאפשר למשמשים אחרים להקים מחדש את הקשר עם מסד הנתונים. תכונה זאת שימושית למפתחים המבקשים להעניק את השינויים שביצעו לאחר ששמרו אותם בקובץ.

```
Sub closeDBConnection()
Dim cnn1 As ADODB.Connection
    Set cnn1 = CurrentProject.Connection
    cnn1.Properties("Jet OLEDB:Connection Control") = 1
End Sub

Sub openDBConnection()
Dim cnn1 As ADODB.Connection
    Set cnn1 = CurrentProject.Connection
    cnn1.Properties("Jet OLEDB:Connection Control") = 2
End Sub
```

רשימת המשתמשים

רשימת המשתמשים, שהוכנסה לשימוש ב- Jet, היא פעולה של נתוני משתמשים נשמרו בקובץ הנעה של קובץ mdb.*. נתונים אלה זמינים באופן בלעדי באמצעות Jet, אולם תוכל לקבל גישה אליהם רק באמצעות DAO (אם מסד הנתונים שלך שוכן בקבצים שהשיטוף שלהם הוא לקריאה בלבד, תכונה זאת לא תהיה זמינה, הויל ו- Jet לא יוצר קובץ נעה עבור שיטופים לקריאה בלבד. עליך להפעיל את השיטה OpenSchema עבור אובייקט Connection כדי להציג ערכת נתונים ובה ארבעה שדות נתונים עבור כל משתמש מן העת האחורה. השיטה שולפת נתונים מקובץ הנעה של קובץ mdb.*. ארבעת השדות בערכת הרשומות הם :

- ◀ computer_name שם המחשב שמננו מבצע המשתמש את החיבור.
- ◀ login_name שם הכניסה לצורך אבטחה ברמת המשתמש.
- ◀ connected שדה זה מצינו אם המשתמש מחובר כרגע (לפעמים, זיהויי משתמשים נשארים במסד הנתונים הנעלם גם לאחר שהמשתמש כבר ניתק את החיבור עם מסד הנתונים של Jet).
- ◀ suspected_state שדה זה מציין סיום בלתי-ריגיל של החיבור אל מסד הנתונים.

השיגרה הבאה מפעילה את רשימת המשתמשים. צור הפניה אל רשימת המשתמשים בעזרת זיהוי ייחודי גלובלי (Globally Unique ID - GUID) המצביע על הפקד להחזירת ערכות הרשומות של משתמשים פעילים. הדוגמה מחזירה ערכים לשדות המשתמשים ברשימה זאת עומד על 1, המפתח יכול לבצע שירה של שינויים שנערכו במודולים של מסד הנתונים. נוסף לכך, המפתח יכול להשתמש ברשימה כדי לברר אל מי עליו לפנות כדי לקבל שליטה בלעדית במסד הנתונים.

```
Sub listActiveUsers()
Dim cnn1 As ADODB.Connection, rst1 As New ADODB.Recordset
' Set the connection for the current project and invoke the
' OpenSchema method to return the current list of users.
    Set cnn1 = CurrentProject.Connection
    Set rst1 = cnn1.OpenSchema(adSchemaProviderSpecific, , _
        "{947bb102-5d43-11d1-bdbf-00c04fb92675}")

' Print a heading for the User List recordset and enumerate list members.
    Debug.Print "Machine Name    " & "User Name"
    Debug.Print "=====    ====="
    Do Until rst1.EOF
        Debug.Print rst1.Fields("computer_name") & rst1.Fields("login_name")
        rst1.MoveNext
    Loop

End Sub
```

שיתוף טפסים

Access מאפשר למשתמשים מרובים לעורך, להוסיף ולמחוק רשומות במסד נתונים, גם ללא קוד. עם זאת, תוכל לשפר כושר זה בסביבה מרובה משתמשים על ידי הפעלת מודולים מאחוריו טפסים.

טפס אוטומטי פשוט המבוסס על טבלה או מקור אחר לערכות נתונים עשוי לפעול בסביבה מרובה משתמשים. שני משתמשים, או יותר, יכולים לפתוח טופס כזה ולדף בין הרשומות. כמו כן יכולים המשתמשים לשנות את ערכת הנתונים המאוגדת של הטופס בכל הדריכים האפשרות ב- Access. ניתן לתוכנת את Access ולשנות את ערכי ברירת המחדל השולטים באופן וב嗣ותיו שבו משתמש אחד יכול לראות את השינויים שהכניס משתמש אחר. קובץ מסד נתונים המשמש אותנו בפרק זה מכיל חמשה טפסים (WebBaseList) ו- WebBaseList1, WebBaseList2, WebBaseList3 ו- WebBaseList4. המשקפים סוגים שונים של שליטה באמצעות קוד.

נעילת רשומות באופן ידני

בכרטיסיה **מתקדם** (Advanced) בתיבת הדוח-שים **כלים** (Tools), **אפשרויות** (Options), תמצא פקדים לשיתוף טפסים. הלחוץ **רשומה בעריכה** (Edited Record) מחייב נעילה פסימית ברמת העמוד. בנסיבות מסווג זה, Access נועל את העמוד (או העמודים) המכילים רשומה מרגע שהטופס פותח את הרשומה לצורך עריכה. אם יש בעמוד או בעמודים אלה עוד רשומות, גם אלה ננעלות. תיבת הסימון **פתח מסך נתונים עם נעילה ברמת רשומה** (Open Databases With Record-Level Locking) מחייבת נעילה אופטימית על הרשומה הנוכחית בלבד. רשומות אחרות באוטו עמוד איןן ננעלות. תכונה זאת, של נעילה ברמת הרשומה, נוצרה לבקשת מפתחים רבים במטרה לצמצם את הסיכון להתקשות בגלל טיפול מקביל בסביבה מרובת משתמשים. Microsoft הוסיפה תכונה זאת כדי לבטל את ההשפעה של הגדרת קיבולת העמוד מ- 4KB ל- 2KB (דבר שהוא הכרחי כדי לאפשר עבודה עם תבניות Unicode בשדות המבוססים על תווים במקומות שבהם הם מופיעים).

רענון ערכי שדה

בברירת המחדל בכרטיסיה **מתקדם** (Advanced), מרוחה הזמן לרענון מצינו את מספר השניות המירבי העשויה להשפיע עד שערך שדה שעודכן בטופס מסוים יופיע לעיני משתמשים אחרים. המשתמשים יכולים לרענן שדה נתון באופן יידי בחריה בפקודה **רענן** (Refresh) מהתפריט **רשומות** (Records), או בכך שיעבירו את הסמן על פניו השדה. אם הסמן איינו פעיל בטופס, ייתכן שייחלו זמן רב יחסית עד שהשינוי יופיע לעיני משתמש אחר המופיע באותו רשותה רשומה.

הטופס WebBasedList מציג את התנהלות הגדרות הנעילה והרענון הקיימות בטופס Access בסביבה מרובת משתמשים. אם תפתח שני עותקים שונים של הטופס ותוכניש שינויים באותה רשומה באמצעות שני מחשבים שונים (או שתי פעולות שונות של Access באותו מחשב), תוכל לראות את התנהלות הנעילה ומרוחקי הזמן לרענון/עדכון.

הטופס WebBasedList1 הוא גירסה משופרת של הטופס WebBasedList נטול המודלים; הוא משתמש בשתי שיגורות אירועים קצורות, המובאות להלן. שגרת האירוע Form_Open קובעת ערך של 2 שניות באפשרות מרוחק זמן לרענון. שגרת האירוע Form_Close מחזירה לתקנו את מרוחק זמן ברירת המחדל בן 60 השניות. שים לב, שהשיגרה משתמשת בשיטה SetOption עבור האובייקט Application של

```
Private Sub Form_Open(Cancel As Integer)
    Application.SetOption "Refresh Interval (Sec)", 2
End Sub

Private Sub Form_Close()
    Application.SetOption "Refresh Interval (Sec)", 60
End Sub
```

כasher Sheti haPulot shonot potchot at haTopf WebBasedList1, haDbar nerah kaiilo Udeconim netonim p'shotim uborim mahpula achot lemashia mahri yotter masher batopf WebBasedList. Merovot zman hruneun haKzr yotter aiino meshk夫 shinuiyi sdr, hoshfa ou machika shel rishomot b'in haPulot, olim hoa meshk夫 udconim b'mahirot raba, v'zrikat haMashabim shlo nmocha.

haedchen haPshut shmebzia haShigra Open_Form ushi l'hshpia baofen drmati ul haHinut haNerait lein shel yi'usom Access li'rebiyi m'shatmishim. Le'musa, la'hadra zot ain shom haShpia be'ut taful bi'di m'shatmish yachid. Metrata haYichida haia l'hshpia ul ha'utuvi shbo shinuiim she'uracim m'shatmishim achrim yavo li'yo bi'toi batopim (v'bgalionot netonim).

פקד רענון ושאלתה חוזרת

haTopf WebBasedList2 Mcil l'hazn Refresh haMefu'el at haShita ubor haTopf. haKod shma'achori ayruu haLchicha shel hazon zo hoa Me.Refresh. haShita Refresh fu'let b'diok como psak haZman shel morroch haZman lruneun. Gam ca'an, haM'shatmishim ainim ro'aim fu'lot shel sdr miyon chadsh, haHospat rishomot ao machikat rishomot, olim haM ro'aim midyil kl shinuii sh'hoCNS brishomot bi'di m'shatmishim achrim. Lemro'ot shnitun l'hshig tu'zot domot ul idy ha'ubrat haSman ul pan haShda, haHazon nkar Refresh cdi sheM'shatmish yidu midyil aiyo fu'ola hoa m'bazu.

haShita Refresh ma'irah, olim haia minha m'udbina topf ck shi'shuf rishoma chada, v'haia matziga rishomot sh'machko um smu' m'hika. haFul at haShita Requery shel haTopf cdi ha'zig at urcet haRishomot cpi she'ha nra'it la'achar sh'mashim achd, ao yotter, haosif ao m'hak rishomot. B'matzav beriyat ha'machdl, haShita Refresh m'shira'at haRishoma ha'nochiah b'matzav pe'iel, olim haShita Requery bo'chrat at haRishoma ha'ra'ona la'achar shi'ymha at fu'letta. Am tra'za l'chazar l'mikom haKodom (asher krov l'wadi aiino haRishoma ha'ra'ona), u'li'z l'haosif logika matayima cdi l'no'ot b'chazar l'mikom haRizzi. Am mesh'tmesh mas'ivim shi'na at haKriyutivim miyon, como l'mash'urci m'petach rashi'ao m'petach miyon, haShita Requery t'shuf at shinuiim ha'llo, olim haia minha m'udbina at sdr haRishomot. haShita Requery fu'let b'sabibat mesh'tmesh yachid v'sabibah morobat m'shatmishim ca'achet (laShita Refresh ain haShpia bi'yishomim l'mesh'tmesh b'od).

Sheti haGurot ha'baot mosipot l'hazon Requery l'cd l'hazon Refresh batopf WebBasedList2. LaShigra cmdRequery_Click shelosha chlakim. Tchihla haia shomerat at urk haM'petach ha'ra'asi ha'noch'i cdi l'si'yu ba'itor haRishoma ha'machdl la'achar shahshita Requery mu'virah at nkodat urcet haRishomot l'Rishoma ha'ra'ona batopf. Shnit, haia machihla at haShita Requery. Shlishit, haia mu'virah at haMikod al shada haM'petach ha'ra'asi (ContactID), v'mp'aliha at haPekoda DoCmdFindRecord. Zabar zo m'hazir at haTopf m'Rishoma ha'ra'ona al haRishoma shahshita haRishoma ha'nochiat l'pani haFu'let haShita Requery. Am mesh'tmesh achor m'hak at haRishoma ha'zot, haTopf y'zuir at haTzonga al haRishoma ha'ra'ona. Tocel l'hafeil logika m'shocollet yotter cdi la'chibat haRishoma ha'nochiat b'mikom ach.

```

Private Sub cmdRefresh_Click()
    Me.Refresh
End Sub

Private Sub cmdRequery_Click()
Dim pkBefore As Long
' Save primary key value before requery.
    pkBefore = Me.ContactID.Value
' Apply requery method.
    Me.Requery
' Attempt to find primary key value before requery.
    Me.ContactID.SetFocus
    DoCmd.FindRecord pkBefore
End Sub

```

נעילת רשומות באמצעות קוד

נעילת רשומות פירושה נעילת ערכים ברשומה, באופן שמנוע משתמשים אחרים להכניס בהם שינויים. Access 2000 מאפשר לראשו נעלת רשומות, במקום נעילת עמודים בלבד. השיטה בעיתוי הנעליה נעשית באמצעות אפשרויות הנעליה בכרטיסיה מתקדם (Advanced) שבתיבת הדו-שייח אפשריות (Options). כמו כן ניתן להקצות באמצעות קוד ערכים למאפיין RecordLocks (נעילות רשומה) של טופט נתון, כדי לקבוע אם הנעליה תהיה אופטימית או פסימית, ואם תחול ברמת העמוד או ברמת הטבלה. המאפיין RecordLocks עשוי לקבל אחד משולשה ערכים. הערך 0, שהוא ברירת המחדל, מצין נעילת עמוד אופטימית. הערך 2 מצין נעילת עמוד פסימית. הערך 1 גורם לנעליה ייחודית של הטבלה כולה.

הכרטיסיה מתקדם מכילה גם תיבת סימון חדשה לנעליה ברמת הרשומה במקומות ברמת העמוד. תיבת סימנו זאת נבחרת לפי ברירת המחדל. ההגדרה שהיא מכילה אפשרות לשני משתמשים לעדכן בו-זמנית שתי רשומות שונות באותו עמוד, באמצעות טפסים. אין אפשרות לנעליה ברמת השורה בתנאים מסוג תזכיר (memo) או בעמודי אינדקס. בדרך כלל, Access אינו תומך במתן שמורות משתמשים עם נעילות סותרות. ניתן להגדיר בקוד באופן ידני נעילה ברמת השורה באמצעות השיטה SetOption של האובייקט Application ב- Access. השיטה GetOption מאפשרת החזרת הגדרות לנעלית שורה, לעומת הגדרות לנעלית עמוד.

השורות Open Form_Open ו- Close Form_Close המובאות להלן מטפלות בסוגנותות שונים של נעילת עמוד ונעלית רשומה לעומת נעילת עמוד. האירוע Open Form_Open מאפשר שימוש טפסים לבצע נעילה אופטימית או פסימית של עמודים. כאשר המשתמש בחור באחת האפשרויות, השורה מצבה ערך False בהגדרת הנעליה ברמת השורה. אם המשתמש לא בחר באחת משתי האפשרויות לנעלית עמוד, השירה מצבה ערך True בהגדרת הנעליה ברמת השורה. ערכי True ו-False אלה מקבילים לבחירה, או הסרת הבחירה, בתיבת הסימון המייצגת נעליה ברמת הרשומה בכרטיסיה מתקדם.

```

Private Sub Form_Open(Cancel As Integer)
' Present a series of message boxes to let a user set the
' page-locking style and record-level locking for a form.
If MsgBox("Optimistic page locking?", vbYesNo, _
    "Programming Microsoft Access 2000") = vbYes Then
    Me.RecordLocks = 0
    Application.SetOption "Use row-level locking", False
ElseIf MsgBox("Pessimistic Page locking?", vbYesNo, _
    "Programming Microsoft Access 2000") = vbYes Then
    Me.RecordLocks = 2
    Application.SetOption "Use Row Level Locking", False
Else
    MsgBox "Setting new row-level locking.", _
        vbInformation, "Programming Microsoft Access 2000"
    Application.SetOption "Use row-level locking", True
End If
' Handy for confirming selected states
Debug.Print Me.RecordLocks
Debug.Print Application.GetOption("Use row-level locking")
End Sub

Private Sub Form_Close()
' Restore default locking.
Me.RecordLocks = 2
Application.SetOption "Use row-level locking", True
End Sub

```

הערה:

הגדרות הנעילה שקבע המשתמש הראשון במצב של ריבוי משתמשים, יחול על כל שאר המשתמשים באותה הפעלה. כאשר אתה בודק שגרות המציגות ערכיהם באפשרויות הנעילה השונות, עליך לצאת מ- Access ולהפעילו מחדש עבור כל בדיקה כדי שההגדרות הנעילה החדשות יחולו.



לא לכל ההגדרות במאפיין RecordLocks הגדרות מקבילות בכרטיסיה **מתקדם**. שני משפטי ההדפסה שבתחתיו האירוע Form_Open משקפים הגדרות נוכחות לסוגנו וסוג נעלית העמוד, וכן אם חלה במערכת נעליה בرمמת השורה במקום ברמת העמוד.

ב-Access 2000 מכיל מאפיין אופציוני להחלפת נעליות מרובות ברמת העמוד בנעליה אחת של טבלה כולה. עם הגידול במספר הנעליות בכל טבלה, Access יוכל להחליף באופן אוטומטי את נעליות הרשומה הנפרדות בנעליה אחת לכל טבלה. קידום זה בנעליה האוטומטית עשוי להפחית באופן דרמטי את "העלות" של ניהול נעליות, בתוספת היתרונו של נעלית רשומות כאשר כמה הנעליות מועטה. תוכל לבחור תוכנה זאת על ידי ביטול האפשרות לנעליה ברמת הרשומה כברירת מחדל, ושימוש בשיטה

עבור האובייקט Application. כמו כן עליך להציב ערך בערך הרישום החדש PagesLockedToTableLock הניתנה האוטומטית. כל ערך גדול מ- 0 מציין את המספר המירבי של נעילות לטבלה לפני טרם ינסה Jet להחליפן בנסיבות טבלה אחת. ערך של 100 פירושו, ש- Jet ינסה להחליף נעילות עמודה בנסיבות טבלה בוודאות עם הצבתה של הנעילה ה- 101, ה- 201 וכן להלאה, בכל טבלה נתונה. הנסיניות יצילחו רק אם תינטו ל- Jet גישה לעדית לטבלה (הואיל והצבת נעילה על הטבלה נועלת את הטבלה בפני כל המשתמשים, מלבד בעל הנעילה). הטיפול בתנאים ב- Jet מהיר הרבה יותר כאשר ניתנת לו גישה לעדית לטבלה. תוכנה זאת גם חוסכת מהמפתח את הצורך לקבוע מראש מתי להחיל נעילות טבלה לעדיתות.

שיתוף ערכות רשומות

בערכות רשומות ובמשפטים SQL, Jet משתמש בערכי ברירת מהдел 'חכמים' כדי לקבוע אלו הגדרות נעילת עמודה ונעילת רשומה להחיל. באמצעות קוד DAO המסתמך על סמניטים, עוברת ברירת מהдел ב- Jet לנעילה ברמת הרשומה. בשפטים SQL, העשוויים במקל להשפייע על/API שורות, Jet עובר לנעילה ברמת העמוד כברירת מהдел. דבר זה מאפשר לישום שלך ליהנות מקידום נעילה (אם אפשרות זאת זמינה).

אם אתה מחייב נעילה ברמת השורה על ערכות רשומות, תוכל להגיע לשיפור בהתקאה וברמת הביצועים רק אם תקפיד במפורש על גילישת משימות תחזקה של ערכות רשומות בטרנוזקציות. השתמש בשיטות BeginTrans, CommitTrans ו- Rollback (שנדוון בהמשך פרק זה) כדי להבטיח גילישה בכל פעולת רשומה. ללא אמצעי זה, יצטברו עוד ועוד נעילות, עד שיגיעו לערך של הגדרת הרישום FlushTransactionTimeout.

אם אתה מחייב נעילה ברמת הרשומה, عليك לשקל בזיהירות אלו סוג נתוני להקצות לשדות ואלו אורכי רשומה לקבוע, כדי שתוכל לקיים בקרה על גודל קובץ מסד הנתוניים. נעילה ברמת הרשומה גורמת לכתיבת שורות במקום עמודים; שורות המשתרעות מעבר לגבול העמוד יש להעביר לעמוד חדש. באמצעות סוג נתוניים בעל אורך קבוע, תוכל לקבוע במדויק את אורכו של רשומות; לא ניתן לעשות זאת כאשר טבלאות הנתוניים ב- Access מכילות תווים בעלי אורך משתנה (עם זאת, שפת הגדרת הנתוניים ב- SQL Jet מכילה סוג נתונים CHAR, שמאפשר לך לקבוע את אורכם של שדות מסווג 'מחוזת'. דבר זה עשוי, בטורו, לסייע בקרה על גידול הנפח של רשומות כתוצאה מרשותם שעמוד ייחידי יכול להכיל).

נעילה ברמת השורה

הSIGROW הבאה מראה כיצד מופעלת נעילת שורות ביישום מרובה משתמשים. השיגרה מדמה יישום לשני משתמשים על ידי הפעלת שני חיבורים כנגד אותו קובץ נתונים. לישום דרוש קובץ נוסף, שמו TestRowLock.mdb. הפעל את השיגרה במצב **עד כדי** לראות כיצד נעילת השורות מצילה, או נכשלת, בהתאם לנسبות השונות.

```

Sub RowLockingPessimistic()
' Use TestRowLocking.mdb to run this demo of row locking;
' DO NOT HAVE TESTROWLOCK.MDB OPEN WHEN RUNNING THE DEMO.
Dim cnn1 As New ADODB.Connection
Dim cnn2 As New ADODB.Connection
Dim rs As New ADODB.Recordset
Dim rs2 As New ADODB.Recordset
Dim j As Integer
On Error GoTo ErrHandler
' Open two connections against the same database, both using row locking mode;
' Jet OLEDB:Database Locking Mode info - 0 is "page mode", 1 is "row mode"
' Open first connection to TestRowLocking.mdb.
cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data " & _
    "Source=c:\Programming Access\Chap10\TestRowLocking.mdb;" & _
    "Jet OLEDB:Database Locking Mode=1;"
' Open second connection to TestRowLocking.mdb.
cnn2.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=c:\Programming Access\Chap10\TestRowLocking.mdb;" & _
    "Jet OLEDB:Database Locking Mode=1;"
' Open a recordset in the first connection and begin editing the first row.
rs.Open "TestRowLocking", cnn1, adOpenKeyset, _
    adLockPessimistic, adCmdTableDirect
' Edit first row field value.
Debug.Print rs.Fields("col1")
rs.Fields("col1") = 2
' rs2.Update; putting an update here would close the record lock
Debug.Print rs.Fields("col1")
' Open a recordset in the second connection.
rs2.Open "TestRowLocking", cnn2, adOpenKeyset, _
    adLockPessimistic, adCmdTableDirect
' Attempt to edit the first row; even in row locking, since we are in the pessimistic
' locking mode, this will fail.
Debug.Print rs2.Fields("col1")
rs2.Fields("col1") = 3
Debug.Print rs2.Fields("col1")
' Move to another row in the same page.
rs2.MoveNext
' Attempt to edit the next row; this works under the row locking mode because
' the lock only applies to a single row, not the whole page.
Debug.Print rs2.Fields("col1")
' Should not fail (row locking)
rs2.Fields("col1") = 4
Debug.Print rs2.Fields("col1")

```

```

' Update recordsets.
rs.Update
rs2.Update
' Close connections and exit.
cnn1.Close
cnn2.Close
Set cnn1 = Nothing
Set cnn2 = Nothing

Exit Sub

ErrorHandler:
For j = 0 To cnn1.Errors.Count - 1
    Debug.Print "Errors from cnn1 connection"
    Debug.Print "Conn Err Num : "; cnn1.Errors(j).Number
    Debug.Print "Conn Err Desc: "; cnn1.Errors(j).Description
Next j
For j = 0 To cnn2.Errors.Count - 1
    Debug.Print "Errors from cnn2 connection"
    Debug.Print "Conn Err Num : "; cnn2.Errors(j).Number
    Debug.Print "Conn Err Desc: "; cnn2.Errors(j).Description
Next j
Resume Next
End Sub

```

שים לב שהניסיון שנעשה בחיבור השני לכטוב ברשותה הראשונה של הקובץ TestRowLock.mdb נכשל, והרשותה עדין פתוחה בחיבור הראשון. ניסיון זה עשוי להצליח אם החיבור הראשון יסיר את העיליה מהרשותה (למשל, על ידי הפעלת משפט `rs.Update`). נסינוו של החיבור השני לכטוב ברשותה השנייה מצליח מייד, מכיוון שלא חלה כל עיליה על רשותה זאת. הלוגיקה לטיפול בשגיאות מציגה את הערכים מהאוסף Errors בחיבור של Jet.

נעילה בرمת עמוד

להלן קטע מהשיגרה (`PageLocking()` בעיצובה דומה), העושה שימוש בעילית עמוד במקומות בעילית רשותה. שים לב שהערך של **Database Locking Mode**, שהוא 1 בדוגמה הקודמת, התחלף ב- 0 בדוגמה זאת. ערכים אלה מציינים, בהתאם, מעקב בرمת השורה וברמת העמוד. שני נסיעותינו של החיבור השני לעדכן רשותות נכשלים, והויל וchipbor הריאשו פתח נעילה, ועמוד ייחיד מכיל את כל הרשותות של טבלה קצרה זאת.

```

' Open first connection to TestRowLocking.mdb.
cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data " &_
    "Source=c:\Programming Access\Chap10\TestRowLocking.mdb;" & _ 
    "Jet OLEDB:Database Locking Mode=0;" 
' Open second connection to TestRowLocking.mdb.
cnn2.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _ 
    "Data Source=c:\Programming Access\Chap10\TestRowLocking.mdb;" & _ 
    "Jet OLEDB:Database Locking Mode=0;" 
' Open a recordset in the first connection and begin editing the first row.
rs.Open "TestRowLocking", cnn1, adOpenKeyset, _
    adLockPessimistic, adCmdTableDirect
' Edit first row field value.
Debug.Print rs.Fields("col1")
rs.Fields("col1") = 2
Debug.Print rs.Fields("col1")
' Open a recordset in the second connection.
rs2.Open "TestRowLocking", cnn2, adOpenKeyset, _
    adLockPessimistic, adCmdTableDirect
' Attempt to edit the first row; because of pessimistic page locking mode
' setting, this fails.
Debug.Print rs2.Fields("col1")
rs2.Fields("col1") = 3
Debug.Print rs2.Fields("col1")
' Move to another row in the same page.
rs2.MoveNext
' Attempt to edit the next row; this fails too because the next row is on the
' same page.
Debug.Print rs2.Fields("col1")
rs2.Fields("col1") = 4
Debug.Print rs2.Fields("col1")
' Update recordsets.
rs.Update
rs2.Update

```

אבטחה

Access מכיל מערך מגוון של תכונות אבטחה, התומכות בצריכים השונים של יישומי Access השונים. רוב יישומי Access למשתמשים מרובים עשויים להפיק תועלות באבטחה ברמת המשתמש, המאפשרת למפתחים ליצור קבוצות של משתמשים. אלא שבמערכות מסויימים יש צרכים מיוחדים נוספים. בסעיף זה נביא סקירה קצרה על תכניות אבטחה נוספות, מעבר לאבטחה ברמת המשתמש. הseiיפים הבאים עוסקים בפתרונות שונים לניהול אבטחה ברמת המשתמש באמצעות ADO, דרך הקוד.

חלופות לאבטחה ברמת-משתמש

אחד היתרונות של Access הוא יכולתו לשרת קהלי יעד שונאים. יישומים מסוימים הם עתירי-קוד, על כן עליך להבטיח את השקעתך בקוד מקור. יישומים אחרים משרותם קבוצות עובדה קטנות עם כשרים טכניים מוגבלים, אך עדין נדרש אבטחה ברמה מינימלית כדי למנוע גישה לנוטונים. ביישומים אחרים, היתרון נובע מהפעלת משק משתמש מותאם אישית המגביל את הפונקציונליות על ידי חסipaט סדרה מוגבלת של פקודות.

שימוש ממשק מותאם אישית

לעתים ניתן ליצור אבטחה מספקת ליישום על ידי החלפת ממשק Access הרגיל במשק מותאם אישית. בתפריט **כליים** (Tools), בחוץ אפשרות **הפעלה** (Startup) כדי לפתח תיבת דו-שיח המאפשרת לך לציין כוורת וסמל ליישום מותאם אישית, טופס פתיחה מותאם אישית, ותפריטים מותאמים אישית שיכליפו את אלה המקוריים ב-Access. תיבת דו-שיח זאת מאפשרת גם להשבית את החלון **מסד נתונים** (Database) ואת שורת המצב. כמו כן תוכל לטפל באמצעות קוד בתוכנות של תיבת הדו-שיח **הפעלה**. אם סוג זה של טיפול מתאים לצורכי האבטחה שלך, عليك לשකול אם כדאי לתגבר אותו בשיטות **Show** ו- **Hide**, ובמאפיינים **Visible** ו- **Hidden** לאובייקטים בחלוון **מסד נתונים** (עיין בפרקים קודמים, שם תמצא פרטיהם על טכניקות לייצרת תוכנות מותאמות אישית: פרק 8 מתריך כיצד לבנת את תבנית האובייקט CommandBars כדי לבנות תפריטים מותאמים אישית, ולהתאים תפריטים מוכלים; פרק 5 מראה כיצד לבנות טופס פתיחה מותאם אישית).

הגדרת סיסמה למסד נתונים

באפשרותך לחיבר משתמשים להזין סיסמה כדי לקבל גישה בלתי מוגבלת לכל אובייקטי הנתונים ומסדי הנתונים ב-Access. בהשוויה לאבטחה ברמת המשתמש, אין קושי בניהול סיסמות. אבטחה באמצעות סיסמה היא אמצעי הולם כאשר פועלת במשרד קבוצה שלחבריה דרושה גישה זהה לכל האלמנטים של קובץ מסד נתונים, אולם לא כל עובדי המשרד יכולים בקבוצה.

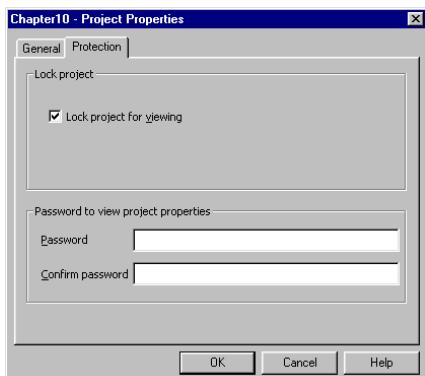
לא ניתן להשתמש בקובץ מאובטח בסיסמה כחבר בקבוצת חברים משוכפלים, מכיוון שב-Jet, שכפול מסד נתונים אינו יכול ליצור סינכרון עם קובץ מאובטח בסיסמה (ראה פרק 11). כמו כן עליך להיזהר ביצירת קישור עם קבצי מסד נתונים מואבטחים בסיסמה מכיוון שכל אדם המסוג לגשת אל הקובץ המקשר אל הקובץ המוגן זוכה בגישה בלתי מוגבלת לקובץ המוגן. יתר על כן, Access שומר גירסה מוצפנת של הסיסמה לצד נתונים אחרים אודוט הקובץ הקשור. לבסוף, אם מישחו יחליפ את הסיסמה עבור קובץ הקשור, ידרשו את הסיסמה החדשה בפעם הבאה שקובץ מסד נתונים אחר יקשר אליו.

כדי להקצות ולבטל סיסמה למסד נתונים, דרושה לך גישה בלעדית לקובץ. בצע את הפעולות הבאות:

1. כדי להקצת סיסמה לקובץ, פתח את הקובץ על ידי בחירת האפשרות **פתח בלעדית** (Open Exclusive) בלחצן **פתח** (Open) שבתיבת הדו-שיח **פתחה**.
2. בתפריט **כלים**, בחר **אבטחה** (Security), **הגדרת סיסמת מסד נתונים** (Database Password).
3. בתיבת הדו שיח **קביעת סיסמת מסד נתונים**, הזן את הסיסמה שבחרת בתיבות **הטיקסט סיסמה** ו**אמת סיסמה**, ולחץ על **אישור**. בפעם הבאה שימושה יפתח את הקובץ, היישום ידרוש מיננו את הסיסמה.
4. לאחר שביצעת פתיחה בלעדית של מסד נתונים, בחר **כלים, אבטחה, בטל קביעת סיסמת מסד נתונים** (Unset Database Password). הסר את הסיסמה על ידי הקלדת הסיסמה בתיבת הדו-שיח **בטל קביעת סיסמת מסד נתונים**. פעולה זאת מבטלת את הדרישה הריאונית לסיסמה כתנאי לגישה אל מסד הנתונים.

הגדרת סיסמה למודול

ב- Access 2000 ניתן להפעיל אבטחה ברמת המשמש באמצעות סיסמת מודול. גישה חדשה זאת יוצרת אחידות בין Access לבין שאר מרכיבי Office 2000 וחלה על כל מודולי המחלקה העצמאים והמודולים הstdndrtiyms, וכן על המודולים שמאחורי טפסים ודוחות.



תרשים 10.1: השימוש בכרטיסייה **Project Protection** בתיבת הדו-שיח **Properties** כדי להגדיר סיסמת אבטחה למודולים של פרויקט

הגדרת סיסמת האבטחה לכל המודולים בפרויקט VBA נעשית בו-זמנית באמצעות עורך Visual Basic (VBE). בתפריט **Tools** (כלים) של הפרויקט, בחר בפקודה **Properties** (מאפיינים) כדי להציג את תיבת הדו-שיח **Properties** (מאפייני פרויקט). הכרטיסייה **Protection** (הנראית בתరשים 10.1) מכילה את תיבת הסימון **Lock Project For Viewing** (נעול פרויקט להציג), ותיבות טקסט להזנה ואישור של סיסמה למודול. הקצת סיסמה כתנאי להציג מודולים בפרויקט אינה מספקת כדי למנוע מהקוד לפעול כאשר אין מוגן. אם הקצת סיסמה אך לא בחרת בתיבת הסימון **Lock Project For Viewing**, כל משתמש יוכל לעורך את הקוד, וריך תיבת הדו-שיח

Project Properties Protection תהיה מוגנת. לbijוט סיסמת האבטחה של מודולים בפרויקט, מהק את כל הערכים בכרטיסיה

לאחר אבטחת המודולים בסיסמה, עליך להכנס את הסיסמה עם אחת נוספת כדי שתוכל לצפות, לעורך או להכניס קוד חדש. תוכל לאבטח טפסים ודוחות באמצעות רמת משתמש (user-level security) או סיסמות מודול (module password). אבטחת רמת משתמש מיושמת לעיצוב ושימוש בטפסים ודוחות. תוכל לחייב את המשתמש לקלב רשות Modify Design כדי להוסיף פקדים לטפסים. המשתמש גם יctrיך את הסיסמה למודולים בפרויקט, כדי לכנות פרוצדורות לאירועים עבור הפקד. ידיעת הסיסמה למודולים בפרויקט אינה מאפשרת להוסיף פקדים או להסיר פקדים מטפס. בנוסף, הרשות Modify Design (שינוי עיצוב) אינה מאפשרת למשתמש לשנות את המאפיין HasModule של טפסים ודוחות ל-NO. המשתמש צריך להכניס תחילת הסיסמה עבור המודולים של הפרויקט.

*** שימוש בקבצי mde.**

קובץ *.mde. מעניק אבטחה מוחלטת לקוד של קובץ מסד נתונים - Access. בהמרת קובץ *.mdb. לקובץ *.mde. מהדר את כל המודולים, מסלק קוד ניתן לעריכה, ודווחס את הקובץ החדש תוך שימור קובץ *.mdb. המקורי. מסד הנתונים שלך יתכווץ בגודלו בגל הסרת הקוד הניתן לעריכה. כמו כן, מכיוון שההמרה יוצרת מיטוב בינוילת הזיכרון, מהירות הפעולה של הקוד תנגד.

כדי לבצע ההמרה של קובץ *.mdb. לקובץ *.mde. עליך לקבל גישה בלעדית לקובץ עיון בענף שכותרתו "הגדרת סיסמה למסד נתונים", לעיל). בתפריט **כלים** בחיר באפשרות **עוזרי מסד נתונים** (Database Utilities), ולאחר מכן בחר באפשרות **יצירת קובץ MDE** (Make MDE File). לאחר שמירת קובץ ההמרה, הקפץ לשומר גם את הקובץ המקורי. בהגנה מסווג זה, השיטה היחידה לעורך את הקוד בקובץ מסד נתונים או להוסיף אליו, היא לעורך את השינויים בקובץ המקורי, ולאחר מכן להמיר אותו בקובץ *.mde.

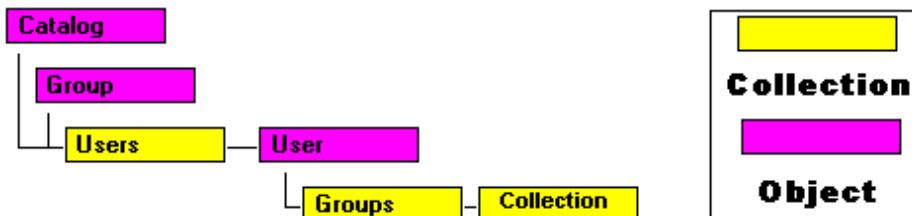
בקובץ *.mde. חלות מספר מגבלות:

- » לא ניתן להכניס שינויים בטפסים, דוחות או מודולים, או להוסיף שינויים.
- » לא ניתן ליבא או ליצא טפסים, דוחות או מודולים אל תוך קובץ *.mdb. רגיל. עם זאת, ניתן ליבא וליצא ללא קושי טבלאות, שאלות, פקודות מאקרו וKİצורי דרך אל عمודי גישה לנדרונים במסדי נתונים אחרים.
- » לא ניתן להוסיף, למחוק או לשנות הפניות בספריות אובייקטיבים או מסדי נתונים אחרים.
- » לא ניתן לבצע שינויים קוד דינמיים, הויל וקבצי *.mde. אינם מכילים קוד ניתן לעריכה (בפרק 7 תמצא דוגמה המציגת מגבלה זאת).
- » לא ניתן לבצע המרה של חבר קיים בקבוצת עותקים משוכפלים לקובץ *.mde., אולם קובץ *.mde. יכול להשתתף בקבוצת עותקים משוכפלים.

↳ קובץ *.mde יכול להפנות לקובץ מסד נתונים אחר רק אם אותו קובץ אף הוא מסוג *.mdb. עלייך להמיר קבצי *.mdb (או קבצי תוספות *.mda) בטרם תמיר את קובץ *.mdb, המכיל את הפניה אליו. על ההפניה החדשה להציג אל קובץ *.mde החדש.

בקרה על אבטחה ברמת-משתמש באמצעות קוד

באבטחה ברמת המשתמש ניתן להגדיר קבוצת משתמש המורכבת מחשבונות משתמש ומחשבונות קבוצה. ניתן ליצור באמצעות קוד חשבונות משתמש וחשבונות קבוצה, וכן להקצות הרשות לחשבונות אלה. מבנה ADOX החדש תומך בפונקציונליות מסווג זה באמצעות האובייקט Catalog, וכן אובייקטי-users ו-groups. Groups. תרשימים 10.2 מציג את היררכיה: קבוצות יכולות להשתיכ למשתמשים, וממשמשים יכולים להשתיכ לקבוצות; הן משתמשים והן קבוצות שייכים לאובייקט Catalog.



תרשימים 10.2: יחס האובייקטים User ו- Group אל האובייקט Catalog

תוכל להקצות הרשות למשתמשים, לקבוצות, או לשניהם גם יחד. כלל, השיטה העילה ביותר היא להקצות הרשות לקבוצות. עלייך למחוק את כל הרשות בירית המandal מחשבונות המשתמשים הנפרדים, ולאחר מכן להציב משתמשים בכל הקבוצות המתאימות. בסוג זה של עיצוב ברמת המשתמש, ניתן לנחל הרשות על ידי הצבת משתמשים בקבוצות, והקצתה הרשות לקבוצות, מכיוון שכל הרשות הקבוצה חלות על המשתמשים החברים בה. על ידי הגבלת הרשות המוקצחות לקבוצות, עיצוב זה יוצר נקודת מרכזית אחת לניהול הרשות.

הדוגמאות הבאות מראות כיצד ניתן באמצעות קוד מסוימות ניהול טיפוסיות ברמת המשתמש. הוואיל ומטרתן העיקרית היא להציג שגרות בסיסיות בניהול אבטחה, רק שתים מהן יכולות לוגיקה לכלcit שגיאות.

חיבור אל מסד נתונים מאובטח

השיגרה הבאה יוצרת חיבור אל מסד נתונים מאובטח. מחרוזות החיבור כוללת ארבעה משפטיים. המשפט הראשון מייעד את המאפיין Provider Connection של השיגרה. הוא מציין את Jet 4 OLE DB. המשפט השני מנסה לחבר את מאפיין מסד הנתונים של המערכת, כלומר השם והנתיב של קובץ נתונים קבוצת העבודה (הקובץ systemdemo.mdw בתיקיית Office; עלייך להעתיק את הקובץ באופן ידני אל

תיקית Office). המשפט השלישי מציין את מקור הנתונים, שמדובר בקובץ UserLevel.mdb. המשפט הרביעי מציין ערכיו זיהויים משתמש וסיסמה לצורך כניסה אל מסד הנתונים המאובטח. בדוגמה זאת, השגירה נכנסת למערכת בשם Admin. אם לא תנסה את הגדירות בריית המחדל, יוקנו למשתמש זה הרשות ניהול מיוחזות.

```
' Turn logon procedure on before running this procedure.  
' Assign password of "password" to Admin user account.  
  
Sub openUserLevel()  
Dim cnn1 As New ADODB.Connection  
Dim rst1 As New ADODB.Recordset  
  
' Open connection to target user-level secured data source; specify path for  
workgroup information file; designate logon ID and password.  
    cnn1.Provider = "Microsoft.Jet.OLEDB.4.0"  
    cnn1.Properties("Jet OLEDB:System database") = _  
        "C:\Program Files\Microsoft Office\Office\systemdemo.mdw"  
    cnn1.Open "Data Source=C:\Programming Access\Chap10\UserLevel.mdb;" & _  
        "User Id=Admin;Password=password;"  
  
' Print first field from first record to confirm connection.  
    rst1.Open "WebBasedList", cnn1, , , adCmdTable  
    Debug.Print rst1.Fields(0)  
    cnn1.Close  
End Sub
```

שתי השורות שלאחר ההערה פותחות ערכת רשומות המבוססת על החיבור ומדפיסות את ערך השדה הראשון מתוך הרשומה הראשונה. דבר זה נועד לאיימת פעולתה התקינה של הדוגמה. הטבלה WebBasedList היא אותה טבלה ששימושה בדוגמה לריבוי משתמשים, לעיל.

כדי שהשגירה OpenUserLevel תפעול, עליך להפעיל את שגרת הכניסה למערכת. דבר זה כרוך במתן סיסמה למשתמש Admin. כמו כן נדרש לשגירה גם קוביץ נתונים נתוני קבוצת עבודה. במקורה זה, על שמו להיות mdw,systemdemo.mdw, ועליך לשומר אותו בתיב שצווין בשגירה. התקليلו הנלווה כולל הן את קוביץ מסד הנתונים המאובטח, והן את קוביץ נתונים קבוצת העבודה, לביצוע קל ופיטוט של בדיקת השגירה.

הוספה ומחיקה של משתמשים

כאשר מפתחים ומנהלים ייוזם מותאם אישית עם אבטחה ברמת המשתמש, מתעורר בדרך כלל צורך בהוספה ובמחיקה של משתמשים. לפני כן כל להוסיף משתמשים, עליך להיכנס למערכת כחבר בקבוצת Admin, למשל Admin. תוכל להשתמש בשיטה Append של האוסף Users כדי להוסיף משתמשים לקטלוג או לקבוצה. עליך לציין עבור המשתמש החדש שם, ובאפשרות גם להקצות סיסמה. ADO מאפשר הקציה של

סistema בשלב מאוחר יותר בעזרת השיטה ChangePassword של האובייקט User. מרובה הוצר, איןך יכול להקצות PID, כיון שערך זה נבחר באופן אקראי על ידי OS.

שתי השגרות הבאות מציגות אחת מהגישות להפעלת השיטה Append כדי להוציאו מעתה חדש ביצועם. השגירה callMakeUser מפעילה את השגירה MakeUser ומעבירה שני ארגומנטים. הארגומנט הראשון מקצה שם משתמש חדש, השני שולח סיסמה. בדוגמה, המחרוזת "password" מכילה את הערך של ארגומנט הסיסמה.

```
' Make sure NewUser account does not exist prior to running this procedure;  
' for example, run callDeleteUser first.
```

```
Sub callMakeUser()  
    makeUser "NewUser", "password"  
End Sub
```

```
Sub makeUser(usrName As String, secureWord As String)  
Dim cat1 As New ADOX.Catalog  
cat1.ActiveConnection =  
    "Provider=Microsoft.Jet.OLEDB.4.0;" & _  
    "Data Source=C:\Programming Access\Chap10\UserLevel.mdb;" & _  
    "Jet OLEDB:System database=C:\Program Files\" & _  
    "Microsoft Office\Office\systemdemo.mdw;" & _  
    "User Id=Admin;Password=password;"  
    cat1.Users.Append usrName, secureWord  
End Sub
```

השגירה makeUser מגדירה יעד לקובץ החדשה בעזרת ההגדירה ActiveConnection. שים לב, שהוא מקצה זיהוי משתמש עם הסמכות ליצור מאובייקט Catalog. השגירה Append מושפעת מכך, ומצביע על קובץ נתוני קבוצת משתמש. השיטה Append בשגירה makeUser מוסיפה חבר חדש לאובייקט Catalog. לכן, המשתמש החדש אכן חבר אף אחת מארגוני חבר קבוצה. באפשרות גם להוסיף חבר לאובייקט Group, כך שימושה תוקנה חברות מיידית באותה קבוצה. באחת מהדוגמאות הבאות ננקת טכנית זאת.

שתי השגרות הבאות מוחקют ממקטולוג של מסד נתונים. התחבר בשייטה Delete של האוסף Users זהה לת לחבר בשיטה Delete של אובייקטי האוסף Tables, Views ו Procedures. השגירה הראשונה, callDeleteUser, מעבירה ארגומנט אחד – שם המשתמש – אל השגירה השנייה deleteUser. השגירה השנייה מוחקת את המשתמש מהקטלוג, ובמקביל מוחקת את המשתמש גם מכל שאר הקבוצות.

```
' Make sure NewUser account exists prior to running this procedure;  
' for example, run callMakeUser.
```

```
Sub callDeleteUser()  
    deleteUser "NewUser"  
End Sub
```

```

Sub deleteUser(userName As String)
Dim cat1 As New ADOX.Catalog
    cat1.ActiveConnection = _
        "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Programming Access\Chap10\UserLevel.mdb;" & _
        "Jet OLEDB:System database=C:\Program Files\" & _
        "Microsoft Office\Office\systemdemo.mdw;" & _
        "User Id=Admin;Password=password;"
    cat1.Users.Delete userName
End Sub

```

כדי למחוק משתמש, עליך להיכנס למסד הנתונים כחבר בקבוצה Admins. להפעלת השיטה Delete לא נדרש סיסמה. כל שדרוש לשיגרה השנייה הוא ארגומנט מחרוזת המכיל את שם המשתמש שיש למחוק.

הקצת קבוצות למשתמשים

אחד הטכניקות המקובלות ניהול הרשאות היא הקצת קבוצות למשתמשים, וניהול הרשאות עבור אותן קבוצות. הרשאות של משתמשים נגוראות כולם מהחברות בקבוצה. הדוגמאות בסעיף זה מוסיפות ומוחקют חברות בקבוצה מתוך חשבון משתמש. בשתי הדוגמאות נעשו שימוש בקבוצה המוכללת Users, אולם אותן טכניקות יפות גם לקבוצות מותאמות אישית.

שתי השגרות הבאות מוסיפות קבוצה לחשבון משתמש בשם NewUser. הקפד לוודא שהחשבון המשתמש אכן קיים בטרם תפעיל את השיגורה. השיגרה הראשונה, callAddGroupToUser, מעבירה שם משתמש ושם קבוצה אל השיגרה הבאה. השיגרה Append, המשמשת בשיטה AddGroupToUser כדי להוסיף את האובייקט Group לאוסף Groups עבור המשתמש. הדוגמה מעבירה ארגומנטים לשיגרה השנייה, וארגוניים אלה מורים לה להוסיף את הקבוצה Users למשתמש NewUser.

```

Sub callAddGroupToUser()
    AddGroupToUser "NewUser", "Users"
End Sub

Sub AddGroupToUser(userName As String, grpName As String)
On Error GoTo AddTrap
Dim cat1 As New ADOX.Catalog
Const acctNameAlreadyExist = -2147467259
    cat1.ActiveConnection = _
        "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Programming Access\Chap10\UserLevel.mdb;" & _
        "Jet OLEDB:System database=C:\Program Files\" & _
        "Microsoft Office\Office\systemdemo.mdw;" & _
        "User Id=Admin;Password=password;"

```

```

cat1.Groups.Append grpName
cat1.Users(usrName).Groups.Append grpName

AddExit:
    Exit Sub

AddTrap:
    If Err.Number = acctNameAlreadyExist Then
        Resume Next
    Else
        Debug.Print Err.Number; Err.Description
    End If
End Sub

```

השיגורה השנייה מפעילה את השיטה Append בניסיון ליצור קבוצה עם השם של הארגומנט השני שהועבר אליה. כshedzoer בקבוצות, שיגורה זאת פועלת בין אם הקבוצה כבר קיימת ובין אם לאו. הויל ו- Users הוא חשבון קבוצה מובנה, הוא מילא קיים תמיד. אם לא קיימת עדין קבוצה עם שם הארגומנט השני, השיטה Append תצליח; אם לא, השיגורה תיפול בתוך מלכודות שגיאות ותחולל את מספר השגיאה -2147467259. לאחר מכן היא תמשיך אל המשפט הבא. בשלב הבא השיגורה תוסיף את הקבוצה לאוסף Groups עבור האובייקט newUser. גם כאן, אם הקבוצה כבר קיימת באוסף Groups של המשתמש, השיגורה תמשיך אל המשפט הבא.

שתי השגרות הבאות מוחקют קבוצה מתיק אוסף Groups של משתמש. השיגורה הראשונה, callRemoveUserFromGroup מעבירה שם משתמש ושם קבוצה כפרמטרים אל השיגורה השנייה, removeUserFromGroup, וזה מבצעת את המשימה. הויל ואין בדוגמה זאת בדיקה לאיתור שגיאות, עליך לוודא שהקבוצה שיכת למשתמש. תוכל לעשות זאת על ידי הפעלת הדוגמה הקודמת.

' Make sure the group account exist for the user prior to running this procedure;
'for example, run callAddGroupToUser.

```

Sub callRemoveUserFromGroup()
    removeUserFromGroup "NewUser", "Users"
End Sub

Sub removeUserFromGroup(usrName As String, grpName As String)
Dim cat1 As New ADOX.Catalog
    cat1.ActiveConnection = _
        "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Programming Access\Chap10\UserLevel.mdb;" & _
        "Jet OLEDB:System database=C:\Program Files\" & _
        "Microsoft Office\Office\systemdemo.mdw;" & _
        "User Id=Admin;Password=password;"

    cat1.Users(usrName).Groups.Delete grpName
End Sub

```

הפעל את השיטה Delete כדי למחוק קבוצה מהאוסף Groups של אובייקט User. שים לב למפרט ההיררכי עבור משתמש יחיד. לאחר זיהוי המשתמש, התחריב מחיבブ ציון של האוסף Groups, ולבסוף, של השיטה Delete. התחריב מצין את שם הקבוצה כפרמטר לשיטה Delete.

יצירה, מחיקה ועקב אחר קבוצות בקטלוג

אם בכונתך לפתוח פתרונות מותאמים אישית ברמת המשתמש, בודאי תרצה ליצור קבוצות מותאמות אישית בעלות שמות שימושיים מבוסנת לkokhotick, וההרשאות המוקנות להן תואמות את הדרישות המיוחדות של היישום המותאם אישית שפיתחת. ארבע הדוגמאות הבאות מבצעות, בהתאם, את הדברים הבאים: יצירת קבוצה מותאמת אישית, מחיקת קבוצה מותאמת אישית, הכנס דוח ובו פירוט כל הקבוצות בקטלוג נתון והקבוצות המשויות לכל חשבון משתמש, והחלפת מצב החברות של קבוצה באוסף Users.

שתי השגרות הבאות מוסיפות קבוצה בשם MySecretGroup1. לאחר הפעלה אל קובץ מסד נתונים עם זיהוי משתמש המספק לביצוע החוספה, השיגרה מפעילה את השיטה Append של האוסף Groups. חובה לציין גורם מכליל עבור האוסף. כאשר מוסיפים קבוצה חדשה לאוסף Users של הפרויקט, הגורם המכיל הוא אובייקט Catalog. כאשר אתה מציב קבוצה באוסף Groups של אובייקט User, عليك לציין את המשתמש כאובייקט הבסיס עבור האוסף Groups.

' Make sure MySecretGroup1 does not exist before running this procedure;
' for example, run callDeleteGroup.

```
Sub callMakeGroup()
    makeGroup "MySecretGroup1"
End Sub

Sub makeGroup(grpName As String)
Dim cat1 As New ADOX.Catalog
    cat1.ActiveConnection = _
        "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Programming Access\Chap10\UserLevel.mdb;" & _
        "Jet OLEDB:System database=C:\Program Files\" & _
        "Microsoft Office\Office\systemdemo.mdw;" & _
        "User Id=Admin;Password=password;"

    cat1.Groups.Append grpName
End Sub
```

שתי השגרות שלහן מוחקota קבוצה מקטלוג. לפני הפעיל את השגרות, عليك לוודא שהקובוצה כבר קיימת בקטלוג. ברכז זאת על ידי הפעלת הדוגמה הקודמת. למעשה, הדוגמה הבאה מוחקota את הקבוצה שהתווסףה בדוגמה הקודמת.

```

' Make sure MySecretGroup1 exists prior to running this procedure;
' for example, run callMakeGroup.

Sub callDeleteGroup()
    deleteGroup "MySecretGroup1"
End Sub

Sub deleteGroup(grpName As String)
Dim cat1 As New ADOX.Catalog
    cat1.ActiveConnection = _
        "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=C:\Programming Access\Chap10\UserLevel.mdb;" & _
        "Jet OLEDB:System database=C:\Program Files\" & _
        "Microsoft Office\Office\systemdemo.mdw;" & _
        "User Id=Admin;Password=password;"
    cat1.Groups.Delete grpName
End Sub

```

התחביר למחיקת קבוצה דומה מאד לתחביר המשמש להוספה קבוצה. הוא מפעיל את השיטה Delete של האוסף Groups בקטלוג. לשיטה מועבר פרמטר יחיד – שם הקבוצה המיועדת למחיקה.

תוך כדי הוספה ומחיקה של קבוצות ומשתמשים, והקצאה מחדש של קבוצות למשתמשים, תוכל בקלות דוח מותאם אישית העוקב אחר חברויות קבוצות עבור האובייקט Catalog ואובייקטי User הנפרדים. השגירה הבאה מפרטת את הקבוצות באובייקט Catalog שמצוין אל מסד נתונים מסווג. לאחר מכן היא מפרטת את פרטי האוסף Groups עבור כל משתמש באוסף Users של הקטלוג.

```

Sub listGroupsInCat()
Dim cat1 As New ADOX.Catalog
Dim grp1 As New ADOX.Group, usr1 As New ADOX.User
cat1.ActiveConnection = _
    "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Programming Access\Chap10\UserLevel.mdb;" & _
    "Jet OLEDB:System database=C:\Program Files\Microsoft Office\" & _
    "Office\systemdemo.mdw;" & _
    "User Id=Admin;Password=password;"

' Groups in overall Catalog
    Debug.Print cat1.Groups.Count & " groups are in the catalog"
    For Each grp1 In cat1.Groups
        Debug.Print String(3, " ") & "* " & grp1.Name
    Next grp1
    Debug.Print

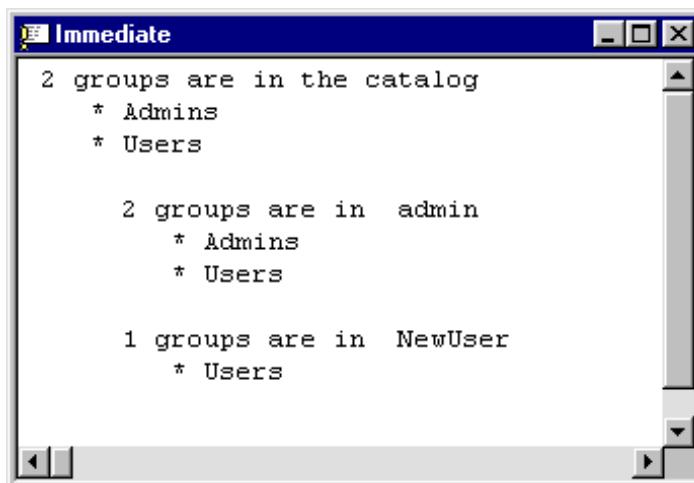
```

```

' Groups in each user
'Creator and Engine are special users that appear in the
'Users collection, but do not need to be tracked.
For Each usr1 In cat1.Users
    If usr1.Name <> "Creator" And usr1.Name <> "Engine" Then
        Debug.Print String(5, " ") & usr1.Groups.Count & _
            " groups are in " & usr1.Name
        For Each grp1 In cat1.Users(usr1.Name).Groups
            Debug.Print String(8, " ") & "* " & grp1.Name
        Next grp1
        Debug.Print
    End If
Next usr1
End Sub

```

בראש כל פירוט של חברי האוסף Groups השיגרה מדוחת על מספר החברים באוסף על ידי הפניה אל המאפיין Counts עבור האוסף. שים לב, בתרשים 10.2 להלן, שדבר זה משתנה בהתאם למשתמש. המשתמש Admin משתמש לקבוצות המוכפלות ו-Users. תוכל לנצל את הדוגמאות הקודומות ליצירה ומחיקה של משתמשים, קבוצות וחברויות של משתמשים בקבוצות.



תרשים 10.3: דוח על חברות בקבוצות מתוך השיגרה listGroupsInCat

השיגרה הבאה מדגימה את אחד השימושים האפשריים של השיגרה listGroupsInCat מהדוגמה הקודמת. השיגרה toggleNewUserInAdminsGroup עשויה לבדוק מה שמשתמע ממשמה. היא מחליפה את מצב החברות של האובייקט NewUser בקבוצות Admins. כמו כן היא מתעדת את המצב הנוכחי של האובייקט NewUser בקבוצה Admins על ידי הפעלת השיגרה.listGroupsInCat

```

Sub toggleNewUserInAdminsGroup()
On Error GoTo ToggleTrap
Dim cat1 As New ADOX.Catalog
Const notInCollection = 3265
cat1.ActiveConnection =
    "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Programming Access\Chap10\UserLevel.mdb;" & _
    "Jet OLEDB:System database=C:\Program Files\Microsoft Office\" & _
    "Office\systemdemo.mdw; User Id=Admin;Password=password;"
cat1.Users("NewUser").Groups.Delete ("Admins")

ToggleExit:
listGroupsInCat
Exit Sub

ToggleTrap:
If Err.Number = notInCollection Then
    cat1.Users("NewUser").Groups.Append "Admins"
Else
    Debug.Print Err.Number; Err.Description
End If
Resume Next
End Sub

```

שים לב שתהיליך שינוי המצביע מסתמך על לכידת שגיאות. לאחר שהיא מתחברת למסד הנתונים הרצוי ולקובץ נתוני קבוצת העבודה באמצעות הפניה לאובייקט `cat1`, השיגרהנסה למחוק את `Admins` מ奏וך האוסף `Groups` ב-`NewUser`. אם הניסיון הולח יפה, השיגרה מפעילה את השיגרה `listGroupsInCat`, ומסתיימת. אם לא, מתחוללת שגיאה. אם מתרחשת שגיאה מכיוון שהקבוצה אינה נמצאת באוסף `Groups` של המשתמש, השיגרה מוסיפה את `Admins` לוסף `Groups` של האובייקט `NewUser`. לאחר מכן השיגרה ממשיכה כאילו לא אירעה שגיאה, ומסתיימת.

הגדרת הרשות

השתמש בשיטה `SetPermissions` עבור האובייקטים `User` ו- `Group` כדי לנהל את הרשותות המוקנות לחשבון מאובטח. הפעל את השיטה `GetPermissions` עבור אובייקטים אלה, כדי להציג ערך ארוך המציין את סוג הרשותות המוקצים לקבוצה או למשתמש. שתי השיטות מאפשרות מגוון רחב של תוצאות: הן יכולות להציג ולדוח על הרשותות שונות עבור מספר סוגים אובייקטים של מסדי נתונים. בנוסף לכך, תוכל להשתמש בשיטה `SetPermission` כדי להקצות, לבטל ולמנוע הרשותות, ולבקר את אופן השימוש בהן.

שתי השגורות הבאות מעניקות לקובוצה הרשות מלאות לכל טבלה חדשה. הגדרת הרשותהה עבור טבלה חדשה אינה משפיעה על טבלאות קיימות. לכן ייתכן מצב שבו לקובוצה יש הרשות מלאות לכל הטבלאות החדשות, אך אין לה כלל הרשות לטבלאות הקיימות.

' Make sure MySecretGroup1 exists before running procedure.

```
Sub callSetAllTablePermissionsForGroup()
    setAllTablePermissionsForGroup "MySecretGroup1"
End Sub

Sub setAllTablePermissionsForGroup(grpName As String)
Dim cat1 As New ADOX.Catalog
Dim grp1 As New ADOX.Group, usr1 As New ADOX.User
cat1.ActiveConnection =
    "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Programming Access\Chap10\UserLevel.mdb;" & _
    "Jet OLEDB:System database=C:\Program Files\Microsoft Office\" & _
    "Office\systemdemo.mdw;" & _
    "User Id=Admin;Password=password;"
cat1.Groups(grpName).SetPermissions Null, adPermObjTable, _
    adAccessSet, adRightFull
End Sub
```

השיגורה הראשונה מעבירה שם קובוצה, MySecretGroup1 אל השיגורה השנייה. השיגורה השנייה מפעילה את השיטה SetPermissions עבור חבר הקובוצה הנושא שם זה. לכן עליך לוודא שהקובוצה קיימת לפני תפעול את השיגורה או להוסיף לוגיקה לכלידת שגיאות. הfrmtrטט הראשון בשיטה מכיל ערך Null מפורש. בדרך כלל, frmtrטט זה מצין שם של אובייקט מסד נתונים, כמו למשל, טבלה. ערך Null מצין שברצונך להגדיר הרשות עבור אובייקטי מסד נתונים חדשים. הfrmtrטט השני קובע את סוג האובייקט, במקרה זה Table. הfrmtrטט השלישי משמש כפועל; הוא מצין כי הפוקודה תגדייר הרשות. קבועים נוספים מציינים פעולות שונות, כמו למשל ביטול הרשות, אשר השיטה יכולה לנகוט. הfrmtrטט הרביעי מעניק למשתמש זכויות מלאות. השיטה והfrmtrטטים שלה מעניקים לקובוצה MySecretGroup1 זכויות מלאות בכל הטבלאות החדשות בקובץ מסד הנתונים newUserLevel.mdb עם קובץ נתוני קבוצת העבודה systemdemo.mdw.

עיצוב זה הוא עיצוב בסיסי וגמיש, העשוי לשמש במגוון של נסיבות שונות. לדוגמה, כדי לבטל את כל הזכויות בטבלאות חדשות, عليك להחליף בשיטה SetPermissions את הfrmtrטט השלישי, adAccessSet, ב- adAccessRevoke. כדי להגדיר זכויות עבור אובייקט מסד נתונים קיים, عليك להחליפ, בfrmtrטט הראשון, את הערך Null בשמו של אובייקט מסד הנתונים.

כיצד לחבר בין כל המרכיבים

שתי השגרות הבאות שעובדות מתחתן של הדוגמאות הקודומות ומציגות פן נוסף בשיטת SetPermissions חדשה בקובץ נתוני קבוצת העבודה systemdemo.mdw. לאחר מכן היא מפעילה את השיגרה השנייה והעבירה אליה את שם הקבוצה החדשה, וכן את שם אובייקט מסד הנתונים שעבורו היא מבקשת להקשות הרשות. שתי השורות האחרונות בשיגרה הראשונה יוצרות משתמש חדש בשם NewUser2, ומושיפות את הקבוצה MySecretGroup2 למשתמש Groups שלו. בדרך זאת חלות על המשתמש NewUser2 כל הרשותות שהוקצו לקבוצה MySecretGroup2 באמצעות השיגרה השנייה.

```
Sub callSetRIDTablePermissionsForGroupTable()
' This procedure makes a group called MySecretGroup2 and assigns
' Read/Insert/Delete Permissions for WebBasedList table to MySecretGroup2.
Then,
' it creates NewUser2 and assigns MySecretGroup2 to NewUser2. Before running
' this, delete MySecretGroup2 and NewUser2 from UserLevel.mdb if they exist.
makeGroup "MySecretGroup2"
setRIDTablePermissionsForGroupTable "MySecretGroup2", "WebBasedList"
makeUser "NewUser2"
AddGroupToUser "NewUser2", "MySecretGroup2"
End Sub

Sub setRIDTablePermissionsForGroupTable(grpName As String, tblName)
Dim cat1 As New ADOX.Catalog
Dim grp1 As New ADOX.Group, usr1 As New ADOX.User
cat1.ActiveConnection = _
"Provider=Microsoft.Jet.OLEDB.4.0;" & _
"Data Source=C:\Programming Access\Chap10\UserLevel.mdb;" & _
"Jet OLEDB:System database=C:\Program Files\Microsoft Office\" & _
"Office\systemdemo.mdw;" & _
"User Id=Admin;Password=password;"
cat1.Groups(grpName).SetPermissions tblName, adPermObjTable, adAccessSet, _
adRightRead Or adRightInsert Or adRightDelete
End Sub
```

השיגרה השנייה מקיצה לקבוצה MySecretGroup2 הרשותות קרייה, הוספה ומחיקה עבור הטבלה WebBasedList שבקובץ UserLevel.mdb. שיגרה זאת דומה לדוגמה קודמת שהتاימה להענתק זכויות לאובייקט מסד נתונים מוגדר, אולם שיגרה זאת מרשרת שלוש זכויות נפרדות כדי לקבל ערכה משולבת של הרשותות.שים לב שהתחביר מרשיר זכויות בעזרת האופרטור Or.

טרנזקציות

באפשרות לארוז באמצעות טרנזקציות שתי פעולות מסד נתונים, או יותר, כך שכולן ייצאו אל הפועל, או שאף אחת מהן לא תבוצע. טרנזקציות הן דבר שימושי במצבים שניים : כאשר מוטל על היחסם שלך לבצע סדרה של פעולות כיחידה (כמו למשל העברת כספים בין חשבון חיסכון לחשבון עובך ושב) ; וכן כאשר עליך להאיץ קבוצה של פעולות מסד נתונים באמצעות גישה בתוך טרנזקציה (מכיוון שכטיבה אל זיכרונו מטמון זמן מהירה יותר מכתיבת אל דיסק עבור כל אלמנט בקבוצת הטרנזקציה).

שלושה אובייקטי Connection - ב- ADO תומכים בטרנזקציות : BeginTrans ו- CommitTrans .Rollback .Rollback בחרם טרנזקציות מקווננות, לכל היוטר. לאחר שהפעלת טרנזקציה, באפשרות לסיים אותה באמצעות CommitTrans או בעזרת השיטה Rollback. השתמש בשיטה CommitTrans כדי להשלים את פעולות מסד הנתונים. הפעל את השיטה Rollback כדי לסיים טרנזקציה מבלי לבצע את פעולות מסד הנתונים שהופעלו מאז שיטת BeginTrans האחורונה.

השיגרה הבאה מפעילה שלוש שיטות טרנזקציה עבור האובייקט Connection . היא עברת בלולה על פני הרשומות במסד נתונים, ומחפשת התאמה לאחד משלושה קרייטריונים. כאשר השיגרה מוצאת התאמה, היא כותבת Us בשדה Sh-M-Achsen במתומו נתונים, במקום בקובץ מסד הנתונים. השיגרה מפעילה את השיטה BeginTrans ממש לפני הפעלת לולאות Do . בסוף הולאה השיגרה שואלת את המשמש אם ברצונו לבצע את השינויים. אם המשמש משיב בחוב, הדוגמה מפעילה את השיטה CommitTrans כדי לבצע את השינויים במסד הנתונים. אם המשמש משיב בשלילה, השיגרה מוחckaת את השינויים על ידי הפעלת השיטה Rollback .

```
Sub changeInTrans()
On Error GoTo changeTrap
Dim cnn1 As ADODB.Connection
Dim rst1 As New ADODB.Recordset
Dim iChanges As Long
Const conLockedByAnotherUser = -2147217887
' Open recordset based on WebBasedList in current project.
Set cnn1 = CurrentProject.Connection
rst1.Open "WebBasedList", cnn1, adOpenKeyset, adLockPessimistic, adCmdTable

' Loop through all records to find those to update to US.
cnn1.BeginTrans
Do Until rst1.EOF
    If rst1.Fields("Country") = "" _
        Or IsNull(rst1.Fields("Country")) = True _
        Or rst1.Fields("Country") = "USA" Then
```

```

rst1.Fields("Country") = "US"
iChanges = iChanges + 1
End If
rst1.MoveNext
Loop

' Commit all changes if user says so.
' Roll changes back otherwise.
If MsgBox("Are you sure that you want to update" & _
    " these " & iChanges & " records?", vbYesNo, _
    "Programming Microsoft Access 2000") = vbYes Then
    cnn1.CommitTrans
Else
    cnn1.RollbackTrans
End If
Exit Sub

changeExit:
    Exit Sub
changeTrap:
If Err.Number = conLockedByAnotherUser Then
    MsgBox "Recordset not available for update. Try again later.", vbCritical, _
        "Programming Microsoft Access 2000"
Else
    Debug.Print Err.Number; Err.Description
End If
cnn1.RollbackTrans
Resume changeExit

End Sub

```

השיגרה changeInTrans מפעילה את השיטה Rollback גם אם מתרחשת שגיאה בזמן ריצעה, כמו למשל כאשר מסד הנתונים ננעל על ידי משתמש אחר. במקרה כזה, לא מתאפשר לעורוך רשומה אחת, או יותר. במסדי נתונים מסוימים, פעולה המתבצעת בקבוצת משנה של רשומות עלולה להשחית את מסד הנתונים כולו. אם השיגרה נתקלת בשגיאה, מכל סיבה-שהיא, היא מבטלת כל פעולה תלולה ועומדת לפני שהיא מסתיימת. הפעלת השיטה Rollback שומרת על תקינותו של קובץ מסד הנתונים.

שכפול מסדי נתונים

יישומי Microsoft Access מאפשרים להציג לפני קבוצות משתמשים שונות עותקים נפרדים, או שכפול מסדי נתונים, תוך ניהול כל השכפולים בתור מסד נתונים אחד. אפשרות זאת חשובה במיוחד בישומי Access מתקדמים למשתמשים רבים, או ביישומים למשתמשים מועטים הזוקקים לגישה בלעדית למסד הנתונים, בעוד שאחרים קוראים או מעדכנים את תוכנותו. שכפל של מסדי נתוניםיעיל וחשוב בפרויקטים שבהם נדרש לסייעים או לעובדים נידים שיתופם חלק במסד נתונים, כאשר הגירסה המלאה שמורה במטה הפרויקט. ניתן לשתף נתונים מבעלי פיתוח חיבור ברציפות בין שתי נקודות.

מיכון ש- 2000 הוא הגירסה השלישית של Access הכוללת את האפשרות לשכפל מסדי נתונים, תכונה זאת כבר שוכלה ושולבה בצורה טובה. פרק זה כולל סקירה כללית על נושא השכפל ומציג את תכונות השכפל החדשות ב- Access 2000. כמו כן הוא מכיל הסברים על ניהול השכפל בעוזרת ספריית 2.1 OLE. דוגמאות הקוד ישרטו היטב את מי שרק החל לרכוש בקיימות בתכנות פתרונות שכפל, ויסייעו למי שכבר בקיא בתכונות שכפלים ב- DAO (Data Access Objects) לעبور ל- ADO (Data Objects). הדיוון בנושא OLE במסגרת פרק זה הורחב וכולג גם סוגיות אשר אין הקשורות בשימוש לשכפל.

כיצד עובד השכפל

Access מאפשר שכפל עותקים רבים של מסד נתונים באמצעות רשת תקשורת מקומית (LAN), רשת תקשורת מרחבי (WAN), רשת אינטרא-נט ארגונית, רשת אקסטרא-נט או האינטרנט. המשתמש יכול להכניס שינויים בעותק הנפרד שלו, לפי צורך. החיבור בין העותקים עשוי להיות בלתי פעיל פרקי זמן ארכויים. כמו כן ניתן לקבע שינויים המוכנסים בעותק מסוים ישוכפלו ויופיעו גם ביתר העותקים.

השם הקיבוצי של קבוצת עותקים מסד נתונים ב- Access הוא **קבוצת עותקים משוכפלים** (replica set); ניתן לעבוד בו-זמנית עם שתי קבוצות כאלה, או יותר. כל עותק בקבוצה נקרא שכפול, ועל אחד העותקים בקבוצה לשמש **תבנית בסיס** לעיצוב (design master). תבנית הבסיס לעיצוב נבדלת מיתר השכפולים בכך שהיא מסוגלת להעיבר שינויים מבניים (כמו למשל דוחות וטפסים חדשים) ושינויים בתנתונים, אל שאר חברי קבוצת העותקים המשוכפלים (מושרים אחרים המאפשרים שכפול מסדי נתונים אמנים מכילים תמייה לחילופי נתונים בין העותקים,อลם אינם מאפשרים הפעלה של שינויים מבניים בין חברי קבוצת עותקים משוכפלים).

שכפול מסדי נתונים מתרכש בין זוגות של שכפולים. השינויים מופצים בין חברי קבוצת עותקים משוכפלים תוך תהיליך הסינכרון של העותקים זה עם זה. כל עותק בזוג יכול לשלוח את השינויים שהוכנסו בו אל בן הזוג. אם אין התנגשות בין השינויים, העותקים מעדכנים את עצםם בחבילות השינויים שהם מקבלים זה מהז. אם קיימת התנגשות בשינוי מסוים מכיוון שני העותקים שינו את התכוללה באותו נקודה, אחד השינויים יינצח והשני ייפסיד'. Access שומר את השינוי 'הmpsיד' בטבלת התנגשויות. תוכל להשתמש בטבלה זאת כדי לפטור באופן יידי את התנגשויות, או שהישום יוכל להחיל כללים אוטומטיים לעיבוד התנגשויות. Access מכיל אף לפתרון התנגשויות; תוכל לבנות אף פרד שיחליף או ישלים את פעולתו של אשף זה.

שגיונות שכפול עלולים להתרכש כאשר שינוי מסוים הוא חוקי בעותק מקומי,อลם אינו חוקי באחד, או יותר, מהשכפולים האחרים בתוך קבוצה, כמו לדוגמה הזנת רשומה עם מפתח ראשי זהה בשני עותקים נפרדים בקבוצת עותקים משוכפלים. כאשר מתבצע סינכרון בין שני העותקים, מגנוון מסד הנתונים Jet דוחה את הרשומה עם המפתח הראשי ההפוך. גם בתנאים אחרים עשויות להתרכש שגיונות שכפול, כמו למשל החלטת כלל לאימות שדה בעותק מסוים, אשר כתוצאה ממנו נוצרים נתונים בלתי חוקיים בעותק אחר.

שוני עשוי להיות בין שכפולים באחת ממשתי דרכיהם. ראשית, ניתן שהם יctraco להשלים מחזור סינכרון שבמסגרתו כל השכפולים יחליפו נתונים זה עם זה. ניתן לפטור בעיה זאת על ידי שלמת פועלות הסינכרון בין חברי קבוצת העותקים המשוכפלים. שונות עלולה להיות גם כתוצאה מהתנגשויות ושגיונות שכפול – גם אם יש סינכרון מלא בין השכפולים.

ניתן לשכפל מסדי נתונים באמצעות דרכי :

- ﴿ **בעזרת הסמל המזוודה** (Briefcase)
- ﴿ **בעזרת הפקודת שכפול** (Replication)
- ﴿ **בעזרת תוכנות ב- DAO (או ב- DAO)**
- ﴿ **בעזרת Replication Manager** (מנהל השכפול)
- ﴿ **בעזרת סינכרון דרך האינטראנט**

שכפול בעזרת סמל המזودה

בעזרת סמל המזודה (Briefcase) של שולחן העבודה של Windows, יכול משתמש קצה לחתות עימו עותק של מסד נתונים מחוץ לאתר ולבצע בו עדכונים. עם החזרתו, ניתן לסנכרן את השכפול שבמחשב הנישא עם העותק שבמחשב האישית או המחלקה. מפתחים יכולים להשתמש בשכפול בעזרת סמל המזודה כדי להציג תבנית בסיס לעיצוב במחשב הנישא, ולבנותו דוחות וטפסים מותאמים אישית. לאחר מכן, הם יכולים לסנכרן אובייקטי מסד נתונים חדשים עם אלה שבעותק העבודה השוטף.

סמל המזודה של Microsoft זמין בשולחן העבודה של Microsoft Windows ב- Windows 9x ו- Windows NT 4 (ייתכן שהמזודה אינה מותקנת במחשב שlk המופעל במערכת הפעלה x9 Windows). כדי להתקין את המזודה, לחץ לחיצה כפולה על האפשרות **הוספה/הסרה של תוכניות Windows** (Add/Remove Program) שבלוח הבקרה. לחץ על הクリティיה התקנת Windows Setup (Windows Setup Properties) שבתיבת הדו-שיח מאפייני **הוספה/הסרה של תוכניות** (Add/Remove Program Properties), בחר באפשרות **עזרים** (Accessories) ולאחר מכן **פרטים** (Details). בתיבת הדו-שיח **עזרים**, בחר באפשרות **מזודה**, לחץ על אישור ופועל לפי ההנחיות. גירירה של קובץ *.mdb מהסieur (Explorer) אל סמל המזודה גורמת להמרת מסד נתונים מתבנית סטנדרטית לתבנית המצוידת בטבלאות ושדות מיוחדים, התומכים בשכפול (bijisomi Microsoft Office אחרים, המזודה רק יוצרת עותקים של קבצים שלמים מבלי לבצע סינכרון בין העותקים). המפער (reconciler) של המזודה מותיר את המקור המעודכן כתובנית בסיס לעיצוב באתר המקור, ומצביע עותק משוכפל במזודה. תוכל לשנות זאת אם דרישה לך לבנית בסיס לעיצוב במזודה.

הערה:



כאשר המפער (reconciler) של המזודה יוצר קבוצת עותקים משוכפלים, הוא שואל אם ברצונך לשמר גרסת גיבוי של מסד הנתונים המקורי. בחר באפשרות זאת, אלא אם כן אתה בטוח שלא תצרוך לחזור אל גרסת המקור, הויל ותהליך ההמרה מוסיף טבלאות ושדות חדשים רבים. אין שיטה פשוטה ואוטומטית לסלק את הטבלאות והשדות הללו בעזרת כל Access מוכלים. לפחות ספק חיצוני אחד (www.trigeminal.com) מציע תוכנית עוזר שמסירה שדות של מערכת שכפול מטבלאות המשתמש.

פקודת השכפול

בחור **כלים** (Tools), **שכפול** (Replication) כדי לקבל גישה לפקודות להפעלת מסד נתונים לבר-שכפול, לייצרת עותקים משוכפלים נוספים, לсинכרון בין עותקים, לפתורון התנשויות וליצירת גרסאות דגס ראשוני של יישומי שכפול. יצירת עותק עם התוכנה החדש **מנع מחיקות** (Prevent Deletes) אפשרית רק דרך **כלים** (אם כי ניתן לשולוט בו באמצעות קוד לאחר שנוצר).

כדי לפטור התנשויות ו.Magentaות שכפול, תיאלץ קרוב לוודאי לבצע סקירה ידנית של הח:rightים הנפרדים המופיעים בטבלאות ההtanשויות. גם אם בסופו של דבר תמכנת כליל פשרה מותאמים אישית, תעטרך לבצע הערכה ידנית של הכללים הללו בטרם תאמץ אותם בעבודה השוטפת.

תכנות JRO

ביקרו של דבר, **תכנות JRO** איננו אלא גישת ADO, הואיל והוא מסתמך על אובייקטי קישור של ADO. בעוד ש-ADO היא טכנולוגיה אוניברסלית לגישה לנוטונים, הרחבות JRO פועלות אך ורק עם מנגנון מסדי הנתונים Jet. אם **תכנתת פתרונות** שכפול מותאמים אישית ב- Access 97 או ב- Access 95, זה הזמן המתאים לעבור מתכנות ב- DAO ל- **תכנות JRO**.

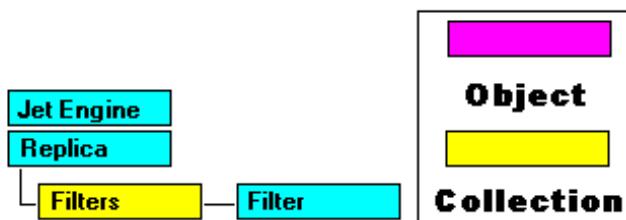
מבנה JRO תומך בשלוש שימושות כלליות:

↳ יצירה וניהול של קבוצות עותקים משוכפלים

↳ דחיסה והצפנה של מסדי נתונים

↳ רענון מטמון הזיכרון כדי לשפר את רמת הביצועים הנראית

שלושת האובייקטים המרכזיים בתבנית JRO מוצגים בתרשים 11.1. הראשון הוא האובייקט JetEngine, התומך בתכונות שהן ייחודיות למנגנון מסדי הנתונים Jet, לרבות דחיסה והצפנה של מסדי נתונים, ורענון מטמון הזיכרון.



תרשים 11.1: מבנה JRO

האובייקט המרכזי השני הוא Replica. אובייקט זה מייצג עותק משוכפל של מסדי נתונים. אובייקטי Replica הם אבני היסוד לבניית קבוצת עותקים משוכפלים. ניתן לנהל קבוצה כזאת על ידי טיפול במאפיינים ובשיטות של אובייקטי Replica. להלן חלק מהfonקציות שבאפשרות נניהל:

↳ הפיכת מסד נתונים לבר-שכפול

↳ ייצור עותקים מתבנית הבסיס לעיצוב ומעותקים אחרים

↳ סינכרון בין עותקים

↳ קריאת אפשרות השכפול של אובייקטים נפרדים

- ﴿ קביעת תבנית בסיס חדש לעיצוב
 - ﴿ שליטה באורך תקופת השמירה של היסטוריות שכפול
- מבנה OJN מכיל גם מאפיינים לניהול תוכנות חדשות וモותוקנות, כמו למשל חשיפה (visibility), סוג שכפול ופתרון התנגשויות המבוסס על קדימות.
- האובייקט המרכזי השלישי הוא Filter. השתמש באובייקט זה להגבלת התכולה בעותקים חלקים של מסדי נתונים. ניתן לבסס מסנן על טבלה או על יחס. OJN כולל אוסף Filters עבור כל המנסנים של עותק. כל המנסנים הללו יחד מגבילים את הנתונים שניתנו להזין בעותק חלקי.

מנהל השכפול

מנהל השכפול, הכלול במהדרת Office למפתחים יסייע בידך בניהול קבוצת עותקים משוכפלים בראשת, העשויה לכלול גם חיבור LAN ו-WAN. במהדרה הראשונה שצורה - Access 97, הכליל מנהל השכפול תמייה בסינכרון דרך האינטראנט באמצעות FTP. אפשרות הסינכרון החדשה באמצעות האינטראנט הכלולה ב- Access 2000 הופכת תכונה זאת לבלתי נחוצה. עם זאת, עליך להמשיך ולהשתמש במנהל השכפול עם גרסת הסינכרון דרך האינטראנט הכלולה ב- Access 2000.

הפעל את מנהל השכפול כדי להתחבר באופן מזדמן אל רשות לצורך סינכרון עקיף, תħaliħ šbu шינויים מהעותק השולח עוברים לעותק המקבל דרך תיבעה להשארת נתוניים, אם העותק המקבל סגור באותה עת. במועד מאוחר יותר, כאשר מסד הנתוניים המקבל ייפתח, סוכן הסינכרון שבשליטת מנהל השכפול יעביר אליו את העדכוניים. מנהל השכפול כולל גם ממוק גրפי למשתמש לצורך תזמון של עדכונים במרוחץ זמן קבועים. עליך להגדיר את תצורת מנהל השכפול עבור הרשת שלך.

תוכל להשתמש במנהל השכפול גם לצורך תיאום בין עותקים ברוחבי רשות תקשורת מרחבית (WAN). ניתן להציג בו באופן גרפי את הטופולוגיה של קבוצת העותקים המשוכפלים. אחד המבנים הנפוצים הוא טיפולוגיה של כוכב, שבה עותק מרכזי מחליף נתוניים עם קבוצה מסוימת של עותקי חישור (spoke replicas). כל אחד מעותקי החישור מתחבר רק אל העותק המרכזי. טיפולוגיה של חיבורים מלאים מושרת כל עותק במישרין עם כל שאר העותקים. דבר זה מאפשר העברת מהירה הרבה יותר של עדכוני נתוניים ברוחבי קבוצת העותקים המשוכפלים,อลומ הדבר מתבטאת בהגברת עומס התעבורה ברשת. טיפולוגיות מסווגים נוספים מציינות יחסיו רוחח-הפסד שונים בין מבנה לרמת ביצועים.

סינכרון דרך האינטראנט

בעזרת סינכרון דרך האינטראנט, תוכל לשכפל מסדי נתונים דרך חיבור FTP או HTTP 1.1 ב- Web. החיבור יכול להתבצע במסגרת האינטראנט, רשות ארגונית או רשות אקסטרא-נט ואין צורך בחיבור רציף בין המחשבים. שלא כמו הסינכרון הישיר

הנתמך במנהל השכפול, סינכרון דרך האינטרנט אינו מחייב הפעלה של תוכן סינכרון במחשב הלקוח. יתר על כן, סינכרון דרך האינטרנט יכול לפעול עם עותקים אוניבימיים. זהויי אחת מהמדרונות המאפיין Visibility בעותק אשר שולבה לראשונה ב- Access 2000 (עיין במאמר שכותרתו Internet Synchronization with Microsoft Jet 4.0 (סינכרון דרך האינטרנט בעזרת Microsoft Jet 4.0 (Microsoft Jet 4.0) בכתבota <http://support.microsoft.com>, שם נמצא פרטים על התקנה, ניהול ובדיקה של מערכת לсинכרון דרך האינטרנט).

כפי שהזכרנו קודם, תוכנות הסינכרון דרך האינטרנט הפכה את גרסת הסינכרון ב- Access 97 למושנת. ארבעה שיפורים מרכזיים מייחדים את תוכנות הסינכרון דרך האינטרנט ב- Access 2000 בהשוואה לגירסה הקודמת:

- ↳ תמיכה בשכפול דרך חיבור של פרוטוקול 1.1 HTTP (לרשותך תמיכה גם בשרתி Netscape באמצעות פרוטוקול זה).
- ↳ רמת ביצועים משופרת בהקשר עם הצפנה.
- ↳ תמיכה בעותק יקל חדש המותאם במיוחד לשכפול מבוסס-Web.
- ↳ מפתחות רישום חדשים לכיוונו עדין של פסקי הזמן של המסתنصرן.

כאשר אתה בוחר שירות Web שעמו תבצע סינכרון, Access קובע באופן אוטומטי אם להשתמש בפרוטוקול HTTP או FTP. מלבד יכולת אפשרות להפעלה על גבי שרתיך Netscape, פרוטוקול HTTP מאפשר לבצע לבצע סינכרון עם מסנכרון באינטרנט תחת המטריה של שירות proxy בעל תצורה מתאימה. Access אינו תומך במפורש בתצורה ההפוכה. אם העותק המקורי שלך אינו מוצפן, סינכרון דרך האינטרנט אינו מ暢ין באופן אוטומטי את העדכוניים שהוא אל עותק היעד לsnsyncron. הגירסה הראשונית לשכפול דרך האינטרנט הצפינה באופן אוטומטי את כל העדכוניים שנעוודו להעברה דרך ה- Web. העותקים החדשניים, האוניבימיים והקלים, יכולים ליצור שכפול רק עם עותק האב שלהם בשרת ה- Web. את עותק האב בשרת ה- Web, عليك לנחל בעזרה מנהל השכפול.

שינויים בעיצוב השכפול

כאשר אתה מסוגל לשכפול של מסד נתונים אחד האמצעים המותאים לעיל, בדרך כלל אתה מוסיף לכל טבלה נוספת של טבלאות מערכתי, וכן קבוצה של שדות. הטבלאות והשדות הללו מסוימים בניהול פרויקט השכפול. עם זאת, הם עלולים להגדיל באופן ניכר את נפח מסד הנתונים. מעבר לכך, טבלאות ושדות השכפול המיוחדים גוזלים ממשאבי יישום מסד הנתונים, דבר שמחזית בו במקצת את מספר הטבלאות המותאמות אישית המירבי לכל מסד נתונים, את מספר השדות המותאמים אישית המירבי לכל טבלה, ואת מספר הבטים הזמינים לכל רשותה לשימושים מותאמים אישית ביישום מסד הנתונים. הבנה של שינוי העיצוב הללו תסייע בידך בניהול פרויקטים של שכפול ב- Access.



כדי להציג את רוב טבלאות ושדות השכפול המאוחדים, בחר את הפריט **אפשרויות (Options)** מתוך תפריט **כלים (Tools)**. בחר בתיבת הסימון **ובייקטים של המערכת (System Objects)**, ולאחר מכן על אישור.

שדות של מערכת שכפול

בדרך כלל מתווספים לטבלאות של יישום שכפול ארבעה שדות חדשים: `s_GUID`, `s_Collineage`, `s_Generation` ו-`s_Lineage`. השדה `s_GUID` מזוהה באופן ייחודי כל שורה בכל טבלה בקבוצת העותקים המשוכפלים. בשני עותקים שונים, יינתנו לשורות זהות בטבלאות זהות ערכי GUID (מזוהה גלובלי ייחודי) שונים. Access בונה את מחרוזות GUID ב-16 בתים באופן המעניק להן ייחוד גלובלי.

בתכנון טבלאות לישום של שכפול, אפשרותך להפעיל מזוהה GUID המופק באופן אוטומטי, כמפתח ראשי עבור מסד נתונים. אם מזוהה GUID משמש כשדה מפתח ראשי של טבלה, Access אינו יוצר את השדה `s_GUID` כאשר אתה הופך טבלה מסוימת לנינטנת לשכפול. במקרה זה הוא משתמש במפתח הראשי לאוთה מטרה. כדי להשתמש במזוהה GUID כמפתח ראשי, בחר בקידוד העתק משוכפל (ID Replication ID) למאפיין של גודל שדה (FieldSize) עבור שדה בעל סוג הנתונים **מספר** (Number) או **מספר אוטומטי** (AutoNumber).

השדה `s_Generation` עוקב אחר הדור של שינויים שהוכנס בטבלה. הנתונים בשדה זה הם מסוג **מספר שלם אורך** (Long Integer). הערך 0 מייצג שינויים חדש המחייב שכפול ללא תנאי. לאחר ההפעזה הראשונית של שינויים אל עותק אחר, Access מעדכן את ערך השדה, כך שהוא מייצג את הדור החדש ביותר של שינויים. תהליך השכפול מנהל מעקב אחר הדור האחרון שנשלח לכל עותק מכל עותק אחר. כאשר נפתח תהליך חילופין חדש, ממשיך בדור החדש הבא של שינויים מאז הסינכרון האחרון שבוצע עם העותק.

השדה `s_Lineage` עוקב אחר ההיסטוריה השינויים שנעשו בכל שורה בטבלה. הנתונים בשדה זה הם מסוג **ובייקט OLE**. השדה מצין متى שורה מסוימת נשלחה לעותק אחר. הוא מונע אפשרות של שינוי חזרה של שינויים שכבר נשלח לעותק אחר כלשהו.

השדה `s_Collineage`, המכיל נתונים מסוג **ובייקט OLE** (OLE Object), תומך בשכפול ברמת-עמודה. תוכונה זאת חדשה ב-Access 2000. בגרסאות קודמות של Access נעשה שימוש בשכפול ברמת-שורה (בסעיף הבא, שכותרתו "חדשושים בתחום השכפול ב-Office" נעמיק את הדיוון בנושא זה). השדה `s_Collineage` עוקב אחר מידע המאפשר לגלוות שינויים ברמת-עמודה במהלך סינכרון.

כל שדה המכיל נתונים מסוג **תזכיר** (Memo) או מסוג **ובייקט OLE** מקבל גם שדה דור נפרד. מכיוון ששדות אלה עשויים להיות גדולים במיוחד, אין הם מופצים, בהכרח, מעותק אחד למשנהו כאשר מתחלף ערך בשדה של שורה כלשהו. שדות אלה מופצים בין העותקים ורק כאשר ערכם משתנה בפועל. שדות הדור הנפרדים שלהם עוקבים אחר

נתונים אלה לצורך תהליכי השכפול. בגירסה הניתנת לשכפול של טבלת Category של Northwind יש שדה דור מייחד אחד עבור הקטגוריה Pictures, אולם בטבלה Photo יש שני שדות דור מיוחדים – אחד עבור השדה Note, והשני עבור השדה Photo.

טבלאות של מערכות שכפול

מספר טבלאות מערכות תומכות בהתקנות של קבוצת עותקים משוכפלים, כפי שמצווג בטבלה שלහן. חלק מהטבלאות, כמו למשל MSysTableGuids, עשוי להיות מוקוצר ובסיסי. בטבלה MSysTableGuids שמוררים מזהה GUID עבור שמות כל הטבלאות בעותק נתון (להוציא טבלאות השכפול וטבלאות ההתקנויות המיחזדות, המוסתרות). חלק מהטבלאות משתמשות במשהו GUID השמורים בטבלה זאת כדי לזהות טבלאות מוגדרות בקבוצת עותקים משוכפלים. בטבלאות אחרות נשמרים פרטים על פעולות היסטוריות או פעולות מתניות. בטבלה MSysTombstone נשמרים פרטיים אודות רשותות שנמחקו. הלוגיקה המובללת לשכפול משתמש בטבלה זאת כדי למחוק רשומות בעותקים המקוריים, במהלך הסינכרון.

טבלה	תיאור
MSysConflicts	ניהול מעקב אחר כל ההתקנויות. טבלה זאת מופצת בין כל החברים בקבוצת עותקים משוכפלים.
MSysExchangeLog	ניהול מעקב אחר נתוני סינכרון בין עותק מסוים לבין כל שאר החברים בקבוצת עותקים משוכפלים.
MSysGenHistory	שמירת נתונים אודות כל דור של סינכרון. טבלה זאת מונעת משלוח של שינויים מודדורות יישנים לחברים פעילים בקבוצת עותקים משוכפלים, וכן עדכון של עותקים ששוחזרו מתוך עותק גיבוי.
MSysOthersHistory	שמירת נתונים אודות דורות של עדכונים מחברים אחרים בקבוצת עותקים משוכפלים.
MSysRepInfo	טבלה זאת מכילה רשומה אחת, ובها פרטים הרלוונטיים לקבוצת העותקים המשוכפלים כולה. טבלה זאת מופצת בין כל החברים בקבוצת עותקים משוכפלים.
MSysReplicas	טבלה זאת שומרים נתונים על כל העותקים בקבוצת עותקים משוכפלים.
MSysSchChange	שמירת נתונים אודות שינויים בעותק המשמש לבניית הבסיס לעיצוב, לצורך הפרצה בין שאר החברים בקבוצת העותקים המשוכפלים.
MSysSchemaProb	שמירת נתונים אודות ההתקנויות בין החברים בקבוצת עותקים משוכפלים. אם אין ההתקנויות בלתי פטורות, טבלה זאת לא תופיע.
MSysSchedule	טבלה זאת משמשת את סוכן הסינכרון המקומי לניהול התזמון של פעולות סינכרון עם עותקים אחרים.

טבלה	תיאור
MSysSideTable	טבלה זאת מכילה נתונים מפורטים על התנגשויות.
MSysTableGuids	טבלה זאת משיכת שמות טבלה למזהה GUID. גם מערכות שכפול אחרות משתמשות במזהה GUID אלה.
MSysTombstone	שמירת נתונים היסטוריים אודוט רשות שנקהקו, ותמייה בעדכוני מחיקה בכל רחבי קבוצת העותקים המשוכפלים.
MSysTranspAddress	שמירת נתונים אודוט מסנכרנים המנהלים עותקים בקבוצת עותקים משוכפלים.
MSysContents	שמירת נתונים הקשורות עבור עותקים חלקיים. טבלה זאת תופיע רק בעותקים חלקיים.
MSysFilters	שמירת נתונים מסננים עבור עותקים חלקיים. טבלה זאת תופיע רק בעותקים חלקיים.

הטבלה MSysConflicts מפנה אל קבוצה של טבלאות שאמנם אין טבלאות מערכתי, אולם הן מוסתרות. בטבלאות אלה שמורים פרטיים אודוט התנגשויות ו.cgiות נפרדות עבור כל טבלה. לפי כללי מתן השמות,ימה של טבלה מוסתרת כזאת יהיה TableName_Conflicts. לדוגמה, אם יש התנגשויות אחת או יותר בין נתונים בשני עותקי הטבלה Employees, נוצרת עבור העותק ש'מפסיד' בהתנגשויות טבלה בשם Employees_Conflict. השורות בטבלה מתעדות את הרשותה של הפסידה' ומגילות המלצה להמשך. כאשר משתמשים פתרים התנגשויות באמצעות לוגיקה מוכללת, Access תלוי בטבלאות אלה ומנהל אותן. אם תבנה לוגיקה מותאמת לפתרון התנגשויות, גם הפתרונות המותאמים שתבנה יצטרכו לנחל טבלאות אלה. לדוגמה, לאחר שנפטרו כל ההתנגשויות בטבלה מסוימת, על היישום שלך למחוק את הטבלה TableName_Conflict. פעולה זאת תמחק את השורה המקבילה מהטבלה MSysConflicts.

טבלאות ושדות השכפול מטילים מגבלות נוספות על העיצוב של יישומי Access. לדוגמה, Access מותיר הצבה של 255 שדות בטבלה, לכל היותר. בשכפולים, בדרך כלל נוספים בכל טבלה ארבעה שדות (s_GUID,s_Lineage,s_Generation ו-s). כאשר שדה נוסף עבור כל שדה תזכיר או שדה אובייקט OLE. שיקולים דומים השdots של טבלה, עלייך להוtier מקום עבור שדות השכפול המיוחדים. שיקולים דומים חלים על מנין הבטים המירבי לכל רשומה. עיין במאמר שכותרתו Microsoft Jet 4.0 Replication in Microsoft Jet 4.0 (<http://support.microsoft.com>), שם תמצא הנחיות בנושאים אלה ובנושאים אחרים בתחום העיצוב המתמקד של שכפולים.

גיבוי ו恢存 מסדי הנתונים המקוריים

עליך לש拷ול ביסודות אם רצונך להפוך קובץ כלשהו לבר-שכפול, מכיוון שלא קיימת תוכנה מוכללת לשחזר מסד הנתונים המקורי. מסיבה זאת, אולי רצוי שתיצור גיבוי של קובץ מסד הנתונים לפני שתאפשר לשכפל אותו.

אם ידועים לך השמות של טבלאות ושדות השכפול, תוכל לשחזר מトוך אחד העותקים של קבוצת עותקים משוכפלים קובץ מסד חדש שאינו בר-שכפול ואשר יוכל את הנתונים הקיימים של אותו עותק. עליך לצרף טבלאות מהעותק הנבחר אל העותק החדש של מסד הנתונים. צרף רק את שדות המשמש, ולא את שדות השכפול המאוחדים. תוכל ליבא את כל הטבלאות במקומות זאת (לחוציא, כמובן, טבלאות השכפול המיוחדות), ולבצע שאלות מחיקה כדי להסיר את שדות השכפול המיוחדים. הוסף את קשרי הגומלין בין הטבלאות, ולאחר מכן ייצא את שאר אובייקטי מסד הנתונים.

חידושים בתחום השכפול ב- Access 2000

אחד מגורמי השדרוג החשובים ביותר ב- Access 2000 הוא השקת מבנה ODBC לשיטה שכפול באמצעות קוד. הויל וחלק מהתכונות חדשות, כגון המאפיין Visibility, איננו זמין בממשק תכנות DAO המקורי – גם לא בגירסה 3.6 העדכנית ביותר – אין ספק שהיה عليك ללמוד את השיטה החדשה לתכנות שכפולים.

שכפול דו-כיווני ב- Jet-SQL Server

Access 2000 מאפשר שכפול דו-双向性的 בין עותקי Jet ו- Microsoft SQL Server. העברה מסוג זה מחייבת עבודה עם Jet 4 ו- SQL Server. העברה דו-כיוונית פירושה ש-SQL Server יכול לשמש כמחסן מרכזי עבור קבוצת יישומי Access שאינם מחוברים. עובדים שעבודתם מחייבת ניירות, והפעילים יישומי Access, יכולים להעביר עדכונים למסד נתונים מרכזי, ולטעתו את השינויים העדכניים ביותר מסד הנתונים המרכזי. עליך להתחיל בעותק משוכפל של SQL Server, או להגדיל עותק משוכפל ב- Access לימיidi SQL Server.

כאשר אתה מגדיר קבוצת עותקים משוכפלים של Jet ו- SQL Server, חובה להציג עותק של SQL Server במרכז, ואילו עותקי Jet יכולים לפעול כחישורים. עותקי Jet יכולים לבצע חילופי תכנים דו-双向性的 עם עותק SQL Server מרכזי, אולם לא יכולים לבצע זאת בין עצמם. על עותק SQL Server המרכזי לשמש תמיד כמתוך בו כל זוג נתונים של עותקי Jet המוצבים סביבו. הויל ו- SQL Server אינם אלא מנגנון של מסד נתונים, שאינו מכיל סביבת פיתוח יישומים מלאה כמו Access, לא תוכל לשכפל אובייקטי יישומיים של Access, כמו טפסים ודוחות, אל תוך SQL Server מרכזי. עם זאת, ניתן מסווג זה מפיצה על חיסרונו הנזכר לעיל בכך שהוא מאפשר שימוש בשאר היכולות המיוחדים של מערכות SQL Server, כמו למשל עיבוד שרת-לקוח, ובאפשרויות דירוג של ריבוי מעבדים (עלין בנושא Implementing Merge Replication to Access Subscribers SQL Server Books Online ב-).

עדכוניים ברמת-עמודה

אחת השיטות החשובות להגברת התפקוה בקרב עובדים המשתמשים במסד נתונים בר-SCPOL היא לצמצם את מספר ההתנשויות. Access 2000 מאפשר לראשונה עדכוניים ברמת-עמודה במטרה למצער את מספר ההתנשויות בין שני עותקים נתונים. בגרסאות קודמות של Access, עדכוניים משני עותקים שונים התנגדו, אם נערך בהם שינויי באותה רשומה – גם אם שינויו שני שדות שונים באותו רשותה – מכיוון שרמת המקבב הנמוכה ביותר אחר עדכוניים הייתה רמת השורה. ב- 2000 Access תוכל ליצור עותקים המזהים שינויים עד לרמת השדות הנפרדים. לכן, משתמש אחד יוכל לשנות את מספר הפקס שלו בוטק אחד, בעודו משתמש השני משנה את מענו של איש הקשר עבור אותו לקוח, בוטק אחר. כאשר נערך סינכרון בין שני העותקים, לא מתרחשת התנגשות.

מעקב ברמת-עמודה הוא בירורת המבדל עבור כל העותקים החדשניים. אם תעדיף זאת, תוכל לבחור במעקב המקורי, ברמת-שורה. כאשר מעכוניים מסד נתונים בר-SCPOL מתוך גירסה קודמת של Access, הוא משמר את הגדרת המקבב ברמת-שורה. מכיוון שמחирו של מעקב ברמת-עמודה גבוהה במנוחים של רמת הביצועים והגודל, כדאי שלא לשימוש בו אם הסבירות להתנשויות נמוכה. התוכנה מעקב ברמת-עמודה פועלת גם עם עותקי SQL Server.

רמת החשיפה של עותק משוכפל

Access 2000 מכיל לראשונה שלוש רמות חשיפה של עותקים משוכפלים: גלובלי, מקומי וארגוני. אפשרותך לשלוט ברמת החשיפה של עותק משוכפל כמאפיין ב-RJO. לא ניתן לשנות את רמת החשיפה של עותק משוכפל לאחר שנוצר.

עותק משוכפל עם רמת חשיפה גלובלית זהה בתפקידו לעותקים המשוכפלים המקומיים ב- Access. הם יכולים לבצע שכפול מול כל עותק משוכפל אחר, ונראים לעין בכל רחבי קבוצת העותקים המשוכפלים. לעותקים משוכפלים עם רמת חשיפה מקומית או אণונית תפקידים מיוחדים המאפשרים צמצום בגודלם בהשוואה לעותקים הגלובליים הרגילים. עותקים משוכפלים שנוצרו מעותקים מקומיים ואणוניים יורשים את הגדרת מאפיין החשיפה של עותק האב, אולם יש להם מאפיין Replicated ייחודי. לא ניתן ליצור תבנית בסיס לעיצוב עותק משוכפל מקומי או אණוני.

עותקים מקומיים חשופים רק לעין עותק האב, והם עורכים חילופי תכנים אך ורק עותק האב. היוזמה לחילופי תכנים יכולה להיות הן של עותק האב והן של העותק המקומי. נסף לכך, עותק האב המרכזי של עותק מקומי יכול לתזמון אירועי סינכרון חזוריים מול עותק מקומי. עותקים מקומיים אינם יכולים לעזר בחילופי נתונים במשרין עם עותקים אחרים בקבוצת עותקים משוכפלים. עם זאת, ניתן להפיץ שינויים בעותק מקומי, דרך עותק האב. התנשויות בין עותק מקומי לעותק האב מסוימים תמיד בנצחון האב.

עותקים אונימיים מיועדים להפצה דרך חיבור מבוסס-Web (FTP או 1.1 HTTP). כדי לבצע בהצלחה סינכרון בין עותק ל��ח אונימי לבין עותק מרכזי ב- Web, המקור של העותק האונימי המקורי צריך להיות עותק גלובלי המוחלט באמצעות מנהל השכפול בשרת ה- Web. תוכל להפעיל עותקים בכל אמצעי מתאימים (למשל, דרך ה- Web או באמצעות תקליטור). בדומה לעותק המקומי, העותק האונימי יכול לעזרך סינכרון רק באמצעות תקליטור. עם עותק האב, אולם האב אינו יכול לתזמון שכפולים עם עותקי הצאצא האונימיים שלו. חילופי הנתונים נערכים תמיד ביוזמת העותק האונימי. התגשויות במהלך הסינכרון מסתיימות תמיד בנצחון עותק האב.

פתרונות התגשויות המבוסס על קידימות

ב- 2000 חל כל חדש כברירת מחדל לפתרון התגשויות. במהדרות הקודמות של Access ההכרעה בהתגשויות בין עותקים נפלה לטובת העותק שביצע את מס' השינויים הגדול ביותר ברשותה נתונה. אם שני עותקים משוכפלים ביצעו מס' שינה זהה של שינויים ברשותה, ניצח העותק בעל הערך הנמוך ביותר במאפיין ReplicaId. בurveה החדש לפתרון התגשויות, בירית המחדל מחלוקת את הכלל: העותק בעל הקידימות הגבוהה יותר הוא זה שניצח. הערכיהם בהגדירות מאפיין הקידימות של עותקים שעווים לנوع בין 0 ל-100. אם לשני עותקים קידימות זהה, העותק בעל הערך הנמוך ביותר במאפיין ReplicaId הוא זה שניצח. יתרונו של הכלל החדש בכך שהוא עולה בקנה אחד עם הכלל החל בשכפול ב- 7 SQL Server.

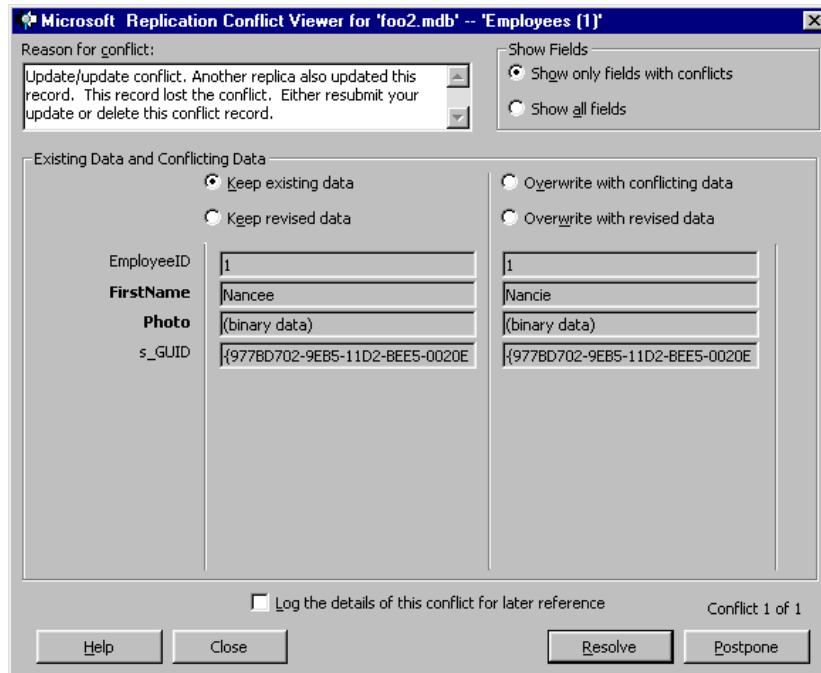
הגדרת המאפיין Priority (קידימות) בעותק משוכפל ניתנת לקריאה בלבד מרוגע שהעותק נוצר. לפי בירית המחדל, ניתנת לעותק הראשוני של מס' נתונים הגדרת קידימות של 90. הקידימות שמקבל עותק גלובלי המבוסס על עותק אחר היא 90 אחוזים מן העותק הראשוני. בעותקים אונימיים ומוקומיים נקבע ערך 0 במאפיין Priority. עותקים משוכפלים שהועתקו באמצעות MS-DOS CompactDatabase מקבלים ערך קידימות התואם את המקור. מסדי נתונים השיטה שעברו המرة מקבלים ערך קידימות של 90. בכל מקרה, תוכל להציג במאפיין Priority כל ערך בטוחה החוקי בעת שאתה מאפשר שכפול של מס' נתונים. חובה להציג בהגדירת המאפיין Priority של כל העותקים הגלובליים שייווצרו בהמשך ערך שווה לזה של עותק האב, או נמוך ממנו (אלא אם כן האדם שיווצר את העותק המשוכפל חבר בקבוצה Admins, או שהינו הבעלים של מס' הנתונים).

שכלולים שונים

סדרה של התאמות שונות משלימה את פונקציונליות השכפול החדשה ב-2000 Access. שתאים מהן מפשיות מסוימות שגורתיות כמו הגבלת התגשויות של עותק משוכפל, או פתרון שגיאות סינכרון. במקרה אחר, שינוי בעיצוב עשוי להשפיע על השיטה המיטיבית להפצת שינויים בעיצוב היישום. התוכונה החדשה **מנע מחיקות** (Prevent Deletes) שתוארה קודם לכן מאפשר למשתמש למחוק רשומות. באמצעות תוכנה זאת ניתן בנקל למנוע משתמשים בלתי מנוסים מחיקה בשוגג של תכנים חשובים. תוכנה זאת

איינה מונעת הפעלה של מחיקות אל עותק זה מעותק אחר, למשל עותק המנוהל בידי מנהל מסד נתונים.

כפי שכבר הזכרנו, ניתן כעת לטפל בהתנגשויות ובשינויים סינכרון באותו משך האשף פתרון התנגשויות (Conflict Resolution) מציג הן התנגשויות והן שינויים (ראה תרשימים 11.2). דבר זה מבטל את הצורך בעיבוד נפרד באמצעות שני ממשקים. נוסף לכך, האשף פועל הן בשכפולי 4 Jet והן בשכפולי 7 SQL Server.



תרשים 11.2 : Microsoft Conflict Viewer של אשף פתרון התנגשויות מראה הן התנגשויות והן שינויים

בשעת סינכרון, מחיקות מקובלות תמיד קדימות גבוהה יותר מכל שינוי אחר הקשור ברשומה. דבר זה נכון בכל גרסאות Access, אולם ב- Access 2000, רשותה ש'הפסיד' מול פעולה מחיקה, מוסיפה ערך בטבלת ההתנגשויות של טבלת המשתמש. הגרסאות הקודומות הتعلמו מעדכנים ש'הפסיד' מול פעולה מחיקה.

Access 2000 כולל לראשונה תמייה בהיררכיה של רשומות מתנגשות. אם יש להתנגשות בין המפתחות הראשיים בשני עותקים, האחד י'נצח' והשני 'יפסיד'. הרשותה בטבלאות האחרות הקשורות לאוֹתָה טבלה יפסידו אף הוא. כאשר אתה מתכוון את המפתח הראשי בעותק המפסיד, התיכון ב- Access 2000 עבר בהיררכיה הרשותה הקשורות לטבלה זאת בטבלאות אחרות, כך שאין צורך בהתקומות נפרדות. בגרסהות קודומות של Access נדרש תיקונים נפרדים בטבלה הראשית ובטבלאות הקשורות אליה.

Access 2000 מכיל תבנית אחסון חדשה להפעה על החלטתך כיצד להפעיל שינויים תוכנה בפרויקט משוכפל. בגרסאות קודמות ניתן היה לשנכרן שינויים, Access, אובייקטים פרטיים של Access, כמו טפסים ודוחות. ב-2000 אובייקטים של Access כמו טפסים, דוחות, מודולים ועמודים נמצאים באובייקט בינארי גדול ייחד, או בקובץ "project.adp" הנפרד. שיטת האחסון החדשה כופף לשכל כל האובייקטים אם מתעורר צורך לעדכן אובייקט מסוים. אם פתרו זה אינו אטרקטיבי, תוכל להפוך את פרויקט Access שבtabנית הבסיס לעיצוב לבלי NiTן לשכפול. במקרה זה תוכל להפעיל ללא הגבלה שינוי עיצוב, באמצעות אחרים (למשל, התקיטור).

טכניקות פיתוח ב- R0N

סעיף זה מציג טכניקות פיתוח ב-R0N לעובדה עם שכפולים, ובין היתר הפיכת מסד נתונים לבר-שכפול, יצירת עותק משוכפל מלא או חלק, דחיסת מסדי נתונים, שנכרז מסדי נתונים, וтиיעוד מאפיינים של עותק משוכפל.

הפיכת מסד נתונים לבר-שכפול

השתמש בשיטה MakeReplicable של אובייקט Replica כדי להפוך מסד נתונים לבר-שכפול. תהליך זה גורם לעותק המשוכפל החדש להפוך לתבנית בסיס לעיצוב, עם הגדרת חסיפה גלובלית במאפיין Visibility. הפיכת מסד נתונים רגיל של Access למסד נתונים בר-שכפול מוסיפה את השדות והטבלאות המיוחדים אשר תוארו קודם. שדות וטבלאות אלה עשויים להגדיל באופן ניכר את נפח מסד הנתונים, لكن ראוי להזכיר עותק גיבוי של מסד הנתונים לפני הפעוק אותו לבר-שכפול.

להלן תחביר השיטה : MakeReplicable

`Replica.MakeReplicable ConnectionString, ColumnTracking`

מחוזות החיבור מחייבת אל מסד הנתונים שברצונך להמיר לתבנית ניתנת לשכפול. באפשרותך להגדיר את המאפיין ActiveConnection של האובייקט Replica בטרם תפעיל את השיטה MakeReplicable. עם זאת, ארגומנט מחוזות החיבור של השיטה עוקף את המאפיין ActiveConnection של העותק המשוכפל. הארגומנט ColumnTracking של השיטה הוא משתנה בוליאני. ערך ברירת המחדל שלו הוא True. זכור, שדבר זה עשוי, להלכה, לסייע במצומצם ההתנסויות בסינכרון. אם אין זה סביר שתתרחשו התננסויות (לדוגמה, מכיוון שכל העריכה תתבצע בעותק משוכפל ייחיד), עלייך לשקל הצבת ערך False במשתנה. במקרה כזה, המעקב אחר התננסויות בסינכרון יבוצע בשיטה הרגילה של מעקב ברמת-שורה. דבר זה מונע את הנפילה ברמת הביצועים הצפופה במקרה של מעקב ברמת-עמודה.

השגרות הבאות מחייבות את השיטה MakeReplicable על מסד הנתונים של Northwind בעעה שמסד הנתונים מבצע פונקציות של גיבוי ולכידת שגיאות. השיגורה McN היא מציבה ערכים בארגומנטים שלה replicaName ו path, ולאחר McN היא מפעילה את השיגורה makeDesignMaster. בדוגמה מבוצע שרשור של משני path ו McNreplicaName. השיגורה השנייה מפעילה את shiite MakeReplicable עבור הערכים שהועברו אליה.

```

Sub callMakeDesignMaster()
    path = "C:\Program Files\Microsoft Office\Office\Samples\"
    replicaName = "Northwind.mdb"
    ' Set the second parameter to True to invoke
    ' column-level tracking of resolution conflicts.
    makeDesignMaster path & replicaName, True
End Sub

Sub makeDesignMaster(newReplica As String, _
    Optional ColumnTracking As Boolean)
On Error GoTo DMTrap
Dim repMaster As New JRO.Replica
' Offer to copy database for restoring it after making the database replicable.
If MsgBox("Do you want to make a backup copy", _
    vbYesNo, "Programming Microsoft Access 2000") = vbYes Then
    Set fs = CreateObject("Scripting.FileSystemObject")
    fs.Copyfile newReplica, "c:\My Documents\DMBackup.mdb"
End If

' Optionally make the newReplica database replicable.
If ColumnTracking = True Then
    repMaster.MakeReplicable newReplica
Else
    repMaster.MakeReplicable newReplica, False
End If
' Clear reference to Design master.
Set repMaster = Nothing
DMExit:
    Exit Sub

DMTrap:
If Err.Number = -2147467259 And _
    Left(Err.Description, 5) = "Could" Then
    MsgBox "Can not create replica because file does " & _
        "not exist. Fix path/file name and try again.", _
        vbCritical, "Programming Microsoft Access 2000"
Resume DMExit

```

```

ElseIf Err.Number = -2147467259 And _
    Left(Err.Description, 8) = "Database" Then
    MsgBox "Database is already replicable. Use the " & _
        "CreateReplica method to base a new replica " & _
        "on it.", vbCritical, "Programming Microsoft Access 2000"
    Resume DMExit
ElseIf Err.Number = 53 Then
    MsgBox "Original file not found for backup copy. " & _
        "Correct file name and try again.", vbCritical, _
        "Programming Microsoft Access 2000"
    Resume DMExit
Else
    Debug.Print Err.Number; Err.Description
End If

End Sub

```

לפני הפעלת השיטה makeDesignMaster, השירה MakeReplicable שואלת את המשמש אם ברצונו ליצור עותק גיבוי. אם המשמש משיב **פנ**, השירה יוצרת מופיע של האובייקט FileSystemObject, ולאחר מכן מפעילה את השיטה Copyfile כדי ליצור גיבוי לקובץ. דבר זה מאפשר בנקל לחזור לגרסת המקור של מסד הנתונים, ללא שדות וטבלאות השכפול המיעודים.

השירה makeDesignMaster מקבלת ארגומנט אחד, או שניים. הראשון הוא שרשור של path ו-replicaName. תבנית הבסיס לעיצוב של קבוצת העותקים המשוכפלים החדש היא קובץ שמו נקבע לפי ערך הארגומנט replicaName. השירה השנייה יכולה לקבל גם ארגומנט נוסף. אם אפשרות זאת מתמשחת, מדובר במשתנה בוליאני המציין אם יש להפעיל מעקב ברמת-עמודה אחר התגובהיות בסינכרון. דוגמה זאת מעבירה ערך True. עיצוב הדוגמה מחייב את המשמש לציין True בארגומנט השני כדי למשמע מעקב ברמת-עמודה. אי-значו הארגומנט השני יגורום למשתנה הבוליאני להניח כי ערך ברירת המחדל שלו, False. הואיל והשיטה MakeReplicable יוצרת לפי ברירת המחדל עותקים משוכפלים עם מעקב ברמת-עמודה, למעשה השירה אינה צריכה לציין בפועל True כדי ליצור עותק משוכפל עם תכונה זאת.

הערה:

אין בנמצא מאפיין למעקב ברמת-עמודה. לכן, עליך לעקוב ידנית אחר המצב של משתנה זה בכל העותקים המשוכפלים שלך.



השירה לוכדת במפורש שלוש שגיאות שונות. האחת היא בהפעלת האובייקט FileSystemObject, ושתי האחרות מקורן ב-Jet.שים לב, שמרכיב השכפול ב-Jet מעביר בחזרה מספר Err זהה (-2147467259) עבור שתי שגיאות שונות במבוקח. לרובו המזול, לשגיאות אלה תיאורים שונים. הדוגמה בעמוד הקודם משתמשת

בתוכנה זאת כדי להבדיל בין השתיים (במערכת עובדת יש לשתמש במקטע תיאור ארוך יותר כדי לזהות בוודאות את סוג השגיאה, או לשתמש בטכנית מוקדמת יותר לניתוח שגיאות).

הערה:



ניתן להבדיל בין שגיאה מקורית לשגיאות Jet מביי להסתמך על המאפיין ErrDescription. עם זאת, למאפיין זה נודעת חשיבות פוטנציאלית עבור מפתחים רבים. הטכנית השנייה כרוכה בספירת האוסף Errors באובייקט Connection. פעולה זאת מחייבת מנגנון מסד הנתונים המקורי מספרי שגיאה מובחנים.

יצירת עותקים מלאים נוספים

החל את השיטה CreateReplica על מופע חדש של אובייקט Replica כדי להפוך את המופיע לחבר חדש בקבוצת עותקים משוכפלים. לפני הפעלת השיטה, הגדר עברו המופיע החדש למאפיין ActiveConnection, כך שיציביע אל תבנית בסיס לעיצוב, אוjal עותק משוכפל אחר מתוך קבוצת המטרה של עותקים משוכפלים. שיטה זאת תיכשל אם ההגדירה ActiveConnection מצינית בשגגה מסד נתונים אשר במאפיין ReplicaType שלו הוכח ערך jrRepTypeNotReplicable. ככל, שיטה זאת מחייבת עותק משוכפל בעל סוג וחשיפה זהים למקור. עם זאת, מכיוון שאמורה להתקיים רק תבנית בסיס אחת לעיצוב, יצירת עותק משוכפל לפי הדגם של תבנית הבסיס לעיצוב מהזירה עותק משוכפל גלובלי נוסף, גם הוא כפוף לפרמטרים עבור השיטה. התchapיר הכללי להחלה השיטה מובא להלן:

```
Replica.CreateReplica ReplicaName, Description, ReplicaType, _  
    Visibility, Priority, Updatability
```

הפרמטר ReplicaName מצין את הניב ושם הקובץ של העותק המשוכפל החדש, אמצעיות 255 תווים לכל היותר. Description הוא שדה אופציוני המסייע בזיהוי חברים בקבוצת עותקים משוכפלים. ערך ברירת המחדל לפרמטר jrRepTypePartial הוא ReplicaName, jrRepTypeFull, jrRepTypePartial, jrRepVisibilityGlobal, jrRepVisibilityLocal, jrRepVisibilityAnon (מקומי) או jrRepVisibilityAnon (אנונימי). אם לא תציין המחדל, jrRepVisibilityLocal (מקומי) יינקטו כלבי ברירת המחדל; התווחה המירבי הוא מ-0 ועד 100, כולל. לפי ברירת המחדל, עותק משוכפל מלא מקבל 90 אחוזים מערך הפרמטר Priority של עותק האב. הפרמטר Updatability קובע אם העותק החדש יהיה לקריאה בלבד (הערך jrRepUpdReadOnly), או לקריאה וכ蒂יה כאחת (הערך jrRepUpdFull).

הדוגמה הבאה יוצרת עותק משוכפל המבוסס על תבנית הבסיס לעיצוב של Northwind מן הדוגמה הקודמת. ההגדלה ActiveConnection יוצרת את העותק המשוכפל. העותק המשוכפל החדש הוא עותק מלא, עם רמת חשיפה גלובלית. הנתיב אל העותק המשוכפל הוא `My Documents\foo.mdb`. בהתאם להגדרות הפרמטרים של העותק המשוכפל החדש, תיאורו הוא `"foo full replica"`.

```
Sub makeFullReplica()
Dim repMaster As New JRO.Replica
' Point repMaster at a design master mdb.
    repMaster.ActiveConnection = _
    "C:\Program Files\Microsoft Office\Office\Samples\Northwind.mdb"

' Make sure foo.mdb is deleted before running the next line.
    repMaster.CreateReplica "c:\My Documents\foo.mdb", "foo full replica", _
        jrRepTypeFull, jrRepVisibilityGlobal, , jrRepUpdFull
End Sub
```

אין צורך ליצור גיבוי במקורה זה, הואיל והפקודה יוצרת עותק משוכפל חדש שחייב להתבסס על עותק קיים. עם זאת, השגירה עלולה להיכשל אם העותק המשוכפל כבר קיים, או אם התבנית האמורה לשמש לייצרת העותק אינה קיימת (כלומר, הקובץ חסר), אולם מדובר בסוגיות פשוטות של לכידת שגיאות. הדוגמה הראשונית נקתה גישה מסויימת למשימה זאת, תוך יצירת עותק משוכפל. יתכן שתרצה להניח, שהגירסה החדשה הופכת כל גירסה קיימת בעלת שם זהה למינשנת. הדוגמה השנייה מימושת את הלוגיקה הזאת בכך שהיא מוחקת את העותק המשוכפל הישן, פוליה המבטלת את הגורם האפשרי לאחת השגיאות עוד לפני שהשגיאה צעה.

יצירת עותק משוכפל חלק, ומסננים

עותק משוכפל חלק הוא עותק משוכפל שאילו מכיל את כל הנתונים שאמור להכיל עותק מלא. עותק משוכפל מסווג זה שימושי בעבודת סניפים או עובדים ניידים, הזוקרים לגישה אל קבוצת משנה מתוך הנתונים המוחזקים במטה. עותק חלק מאפשר למשתמש לעיין רק בחלק מהנתונים המלאים, והדבר מפחית את היקף העדכון שנדרש כדי לسانכרן את העותק המשוכפל.

לאחר יצירת עותק משוכפל חלקו וריך בעזרת השיטה `CreateReplica`, عليك לאכלא אותו בתונונים. לכל עותק חלקו אוסף `Filters` (מסננים). במסגרת אוסף זה, כל אובייקט `Filter` מצין נתח נתוניים שונה שייכלל בעותק המשוכפל החלקי. כדי שעותק משוכפל חלק יוכל נתונים כלשהם, عليك לציין עבורו מסנן אחד או יותר, לצרף את המסנן או המסננים אל אוסף `Filters` של העותק המשוכפל, ולאחר מכן להפעיל את השיטה `.PopulatePartial`.

אפשר לבסס מסנן על הפסוקית WHERE מתוך משפט SQL (ללא מילת המפתח WHERE), או על יחס. הוסף וצין מסננים עבור עותק משוכפל חלקי באמצעות השיטה Append של האוסף Filters. שיטה זאת מקבלת שלושה ארגומנטים: המאפיין TableName קובע את הטבלה שהמסנן יחול עליה, המאפיין FilterType קובע באמצעות FilterCriteria מי יסנן את הערכים עבור הטבלה, משפט קריטריוניים ב-SQL (הערך jrFilterTypeTable), או יחס (הערך jrFilterTypeRelationship). המאפיין jrFilterTypeTable כולל את שם היחס או את הפסוקית WHERE מתוך משפט SQL המגביל את הרשומות של טבלה.

השיטה PopulatePartial עברור עותק משוכפל מנקה את כל הרשומות בעותק המשוכפל החלקי ומאלסת את העותק בהתאם למסננים החלים עליו. היא עושה זאת באמצעות סינכרון של העותק החלקי עם העותק המלא. השיטה מקבלת שני ארגומנטים. הראשון הוא משתנה אובייקט המצביע אל העותק החלקי שיש לאכלס מחדש. השני הוא משתנה מחזוזת, המגדיר את הנתיב ואת שם הקובץ של העותק המלא, שמולו מתבצע הסינכרון של העותק המשוכפל החלקי. בכלל, עליק להשתמש בשיטה PopulatePartial עבור עותק החלקי כאשר אתה יוצר את העותק החלקי, או אם שינית את המסננים שלו. כדי להפעיל את השיטה עבור עותק משוכפל חלקי, עליק לפותח תחילת את העותק הרצוי עם גישה בלעדית, הואיל ובשלב הראשוני, השיטה מנקה את כל הרשומות מן העותק החלקי כדי לאכלסו מחדש.

ארבע השגרות הבאות מדגימות את אחת הגישות להגדרת שני עותקים חלקיים באמצעות האוסף Filters והשיטה PopulatePartial. השיגרה PopulatePartial מפעילה את השיגרה makePartialFilter פעמיים, כל פעם עם קבוצת ארגומנטים שונה. תחילתה היא מפעילה את השיגרה כדי ליצור עותק משוכפל חלקי בשם "Partial of Northwind.mdb". לאחר מכן היא חוזרת על התהיליך ויוצרת עותק משוכפל נוסף בשם ."Partial of foo.mdb"

```
Sub callMakePartialFilter()
    makePartialFilter "Partial of Northwind.mdb", _
        "Northwind.mdb", "C:\Program Files\Microsoft Office\Office\Samples\"
    makePartialFilter "Partial of foo.mdb", "foo.mdb", "C:\My Documents\"
End Sub

Sub makePartialFilter(replicaName As String, _
    sourceName As String, path As String)
    Dim rep As New JRO.Replica
    Dim filt1 As JRO.Filter
    ' Delete old partial.
    strfile = path & replicaName
    deleteFile (strfile)

    ' Make partial.
    makePartial path, replicaName, sourceName
End Sub
```

```

' Open connection to partial and append filter.
rep.ActiveConnection = _
    "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & _
    path & replicaName & ";Mode=Share Exclusive"
rep.Filters.Append "Employees", jrFilterTypeTable, "Title='Sales Representative'"
rep.Filters.Append "Customers", jrFilterTypeTable, _
    "Country='Spain' AND City='Madrid"

' Populate partial from source.
rep.PopulatePartial path & sourceName
End Sub

Sub deleteFile(strfile)
On Error GoTo deleteTrap
Dim cnn1 As New ADODB.Connection
' Prepare to delete file.
Set fs = CreateObject("Scripting.FileSystemObject")
fs.deleteFile strfile

deleteExit:
    Exit Sub
deleteTrap:
If Err.Number = 70 Or Err.Number = 75 Then
    MsgBox "Partial is unavailable to system. " & _
        "Close it so that the system can create a " & _
        "new one.", vbCritical, "Programming Microsoft Access 2000"
ElseIf Err.Number = 53 Then
    Resume Next
Else
    Debug.Print Err.Number; Err.Description
End If
Resume deleteExit
End Sub

Sub makePartial(path As String, replicaName As String, sourceName As String)
Dim rep As New JRO.Replica

    rep.ActiveConnection = path & sourceName
    rep.CreateReplica path & replicaName, replicaName, jrRepTypePartial, _
        jrRepVisibilityGlobal, , jrRepUpdFull
End Sub

```

השיגרה makePartialFilter מקבלת ארגומנטים מהשיגרה callMakePartialFilter. אליה מציינים את שם העותק המשוכפל החלקי שיוצרים, ואת העותק המלא שהוא מקור הנתונים עבור העותק החלקי. השיגרה makePartialFilter גם מדירה ומצרפת את אובייקטי Filter עבור העותק החלקי, ומפעילה את השיטה PopulatePartial כדי להחיל את המנסנים.

תחילה, השיגרה מוחקת את כל העותקים החלקיים המקוריים, ששם והמקום שלהם זחים לאלו של העותק החלקי שהיא אמורה ליצור. פולה זאת נעשית על ידי הגדרת משתנה מחוץ למובוס על המשתנים path ו-replicaName. השיגרה זאת מקבלת מהשיגרה הקוראת. לאחר מכן מוחקת המשנה החדש מועבר אל השיגרה deleteFile. אלא אם מתרחשת שגיאה בלתי צפואה, השיגרה deleteFile מבצעת אחת משלוש שימושות: היא מוחקת את הקובץ הישן של העותק המשוכפל, או שהיא מזקירה למשתמש לסגור עותק משוכפל פתוח כדי שהיישום יוכל למחוק אותו, או שהיא מתעלמת משגיאה שהקובץ אינו קיים (קוד שגיאה 53).

לאחר שהשיגרה מוחקת, או מנסה למחוק, את הקובץ המקורי, היא יוצרת עותק חלקו חדש על ידי הפעלת השיגרה makePartial. הקריאה מעבירה שלושה ארגומנטים: path, replicaName ו-sourceName. השיגרה makePartial זהה כמעט לגמרי ביצובה לשיגרה makeFullReplica. מלבד השימוש במשתנים ציוו הנטיב ושם הקובץ, ההבדל המרכזי הוא שהשיגרה makePartial מציבה ערך jrRepTypePartial במאפיין jrRepTypeFull, בעוד שהשיגרה makeFullReplica מציבה במאפיין זה את הערך jrRepTypeFull. שים לב, שהשיגרה makePartial את שם העותק החדש כשרשור של המשתנים path ו-replicaName. המאפיין ActiveConnection של מופע העותק החדש מצין את העותק המלא המשמש מקור לעותק החלקי. השיגרה מצינית מ庫ר זה כשרטור של המשתנים path ו-sourceName. לשם כך חובה לשמור את העותק המלא ואת העותק החלקי באופן נתיב, אולם אין קושי בהסרת מגבלה זאת. הדוגמה الأخيرة בפרק זה מראה כיצד לעשות זאת.

לאחר שהשיגרה makePartial מוחירה את השלייה לשיגרה makePartialFilter, העותק החלקי החדש קיים, אולם אינו מכיל נתונים. השיגרה makePartialFilter מאלצת אותו בתנאים. תחילתה היא מציבה את המאפיין ActiveConnection של מופע העותק בעותק החלקי החדש, וпотחת את העותק החדש במצב גישה בלעדית. זכור, שדבר זה נדרש כדי להפעיל את השיטה PopulatePartial. בשלב הבא השיגרה מדירה ומצרפת לעותק שני אובייקטי Filter. המסן הראשון מחלץ נתונים 'לקחוות' ממדריך שטאקים מוגדר כ'יציגי מכירות'. המסן השני מחלץ נתונים 'נתוני' לבסוף. רק שתי טבלאות קולוטות רשותם path ו-sourceName בין העותק החלקי החדש. (באפשרות להוציא מסננים כדי לאכלס טבלאות נוספות בעותק החלקי).

סינכרון עותקים משוכפלים

שתי השגרות הבאות יוצרות סינכרון בין עותקים בתרחישי שכפול טיפוסיים. שתיהן מפעילות שגרות ADO בסיסיות. השגרה Add שבעותק synchNorthwindFooToAdd מוסיפה רשומה חדשה בטבלה Employees שבუותק synchNorthwind.mdb. לאחר מכן היא יוצרת סינכרון בין foo.mdb ל- Northwind כדי להפיץ את הרשותה החדשה אל foo.mdb, ולאחר מכן מוחקת את רשותה העובדת החדשה מ- Northwind. השגרה השניה, synchFooNorthwindToDelete, יוצרת סינכרון בין foo.mdb ל- Northwind כדי למחוק את הרשותה גם מהუותק synchNorthwind.

```
Sub synchNorthwindFooToAdd()
Dim rep1 As JRO.Replica
Dim cnn1 As New ADODB.Connection
Dim rst1 As ADODB.Recordset
' Open connection to Northwind and set reference for Northwind as a replica.
cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=c:\Program Files\Microsoft Office\" & _
    "Office\Samples\Northwind.mdb"
Set rep1 = New JRO.Replica
rep1.ActiveConnection = cnn1

' Add a new employee to Northwind.
Set rst1 = New ADODB.Recordset
rst1.Open "Employees", cnn1, adOpenKeyset, adLockOptimistic, adCmdTable
rst1.AddNew
    rst1.Fields("FirstName") = "Rick"
    rst1.Fields("LastName") = "Dobson"
    rst1.Fields("BirthDate") = Date - 1
' When it comes to learning about computers and my faith
' in the Lord, I am always newly born.
    rst1.Update
' Synchronize Northwind with its full replica (foo.mdb).
    rep1.Synchronize "c:\My Documents\foo.mdb", _
        jrSyncTypeImpExp, jrSyncModeDirect

End Sub

Sub synchFooNorthwindToDelete()
Dim rep1 As JRO.Replica
Dim cnn1 As New ADODB.Connection, cmd1 As ADODB.Command
' Open connection to foo and set reference to foo as a replica.
cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=c:\My Documents\foo.mdb"
Set rep1 = New JRO.Replica
rep1.ActiveConnection = cnn1
```

```

' Execute command to remove an employee from foo.mdb.
Set cmd1 = New ADODB.Command
With cmd1
    .ActiveConnection = cnn1
    .CommandText = "DELETE Employees.* FROM Employees" & _
        " WHERE LastName='Dobson'"
    . CommandType = adCmdText
    .Execute
End With
' Synchronize foo with its design master (Northwind.mdb).
rep1.Synchronize "c:\Program Files\Microsoft Office\" & _
    "Office\Samples\Northwind.mdb", jrSyncTypeImpExp, jrSyncModeDirect
End Sub

```

השיגרה synchNorthwindFooToAdd משתמשת ב- ADO כדי להוסיף רשומה לטבלה אחד העותקים, ולהפיצה אל טבלה מקבילה בעותק אחר. השיגרה פותחת בהצהרה על האובייקטים Connection, Replica ו- Recordset. לאחר מכן יוצרת חיבור עם מסד הנתונים Northwind ומזכיבה את המאפיין ActiveConnection של העותק בחיבור. היא יוצרת מופיע של אובייקט Recordset באותו חיבור שבו נמצא העותק. לאחר מכן מוסיפה לטבלה רשומה של עובד בשם Rick Dobson. השיגרה מסתiya מעתה שעמם יש להחליף השיטה Synchronize על העותק Northwind, ומציינת את foo כעותק שמאנו מוכנים. בשלב האחרון, רשומה העובד החדשה עוברת מ- foo ל- Northwind.

השיגרה synchFooNorthwindToDelete מוחקת את הרשימה החדשה מ- foo. כמו כן היא מחזירה את Northwind במצבו הקודם על ידי פועלות סינכרון בין foo ל- Northwind. שיגרה זאת משתמשת באובייקט Command כדי להשמש עובד שם משפחתו Dobson מהטבלה Employees בעותק foo. לאחר ביצוע הפקודה, היא מчиילה את השיטה Synchronize על העותק foo כדי להפיץ את המחיקה לעותק Northwind.

עבודה עם עותקים משוכפלים הנמנעים מחיקות

אין קושי ליצור עותק משוכפל עם מניעת מחיקות, אולם עליך ליצור אותו דרך משך המשמש. בחר **כליים** (Tools), **שכפול** (Replication), **יצירת עותק משוכפל** (Create) (Replica), ובחר בתיבת הסימון **מניעת מחיקות** (Prevent Deletes) בתיבת הדו-שיח שתיפתח על המסך. עותק משוכפל מסווג זה מאפשר לישום להפץ עותק שאיןואפשר מחיקה ישירה של פריטים מתוכו. אולם ניתן להשיג תוצאה זהה באמצעות הגדרות האבטחה של Access, אולם הרבה יותר פשוט לבחור את תיבת הסימון הנזכרת לעיל. בעוד שהמשתמש בעותק עם מניעת מחיקות איןו יכול למחוק רשומות במישרין, העותק המשוכפל יכול לקבל עדכוני מחיקה מעותקים אחרים. אחד השימושים לעותק משוכפל מסווג זה הוא לסייע בידי מנהלים למנוע מחיקת רשומות לפני שמירtan הנאותה בארכיון.

שלוש השגורות הבאות מטפלות בעותקים עם מניעת מחיקות באמצעות ADO. השגירה SynchAddToFoo2 מוסיפה מתוך העותק foo.mdb רשותה חדשה לעותק TryToDeleteFromFoo2. שם העותק הוא Rick Dobson, foo2.mdb. השגירה השניה, foo2.mdb, מנסה למחוק את הרשומה הזאת במישרין מהעותק Execute. הודעת השגיאה המוצגת בתרשימים 11.3 מראה כיצד פקודה הכלולה בשגירה. אם מתעורר צורך בהגבלת הופסות פרט להגבלת מחיקות, תוכל להציג בארגומנט Updatability של השיטה CreateReplica את הערך jrRepUpdReadOnly. אין צורך במשק המשמש כדי לבצע פעולה זאת. השגירה השלישייה, synchFoo2ToDelete, מוחקת את העותק Rick Dobson foo.mdb, ולאחר מכן מפיצה את המיקחה אל העותק foo2.mdb. למראות שהמשתמש אינו יכול למחוק רשומה במישרין מתוך עותק עם מניעת מחיקות, מנהל מערכת יוכל להפיץ מתקפה בעוררת פוליה של סיינכרון.

```

Sub SynchAddToFoo2()
Dim rep1 As JRO.Replica
Dim cnn1 As New ADODB.Connection
Dim rst1 As ADODB.Recordset
' Open connection to foo and set reference to foo as a replica.
    cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=c:\My Documents\foo.mdb"
    Set rep1 = New JRO.Replica
    rep1.ActiveConnection = cnn1

' Add a new employee to foo.
    Set rst1 = New ADODB.Recordset
    rst1.Open "Employees", cnn1, adOpenKeyset, adLockOptimistic, adCmdTable
    rst1.AddNew
        rst1.Fields("FirstName") = "Rick"
        rst1.Fields("LastName") = "Dobson"
        rst1.Fields("BirthDate") = Date - 1
    rst1.Update

' Synchronize foo with foo2.mdb.
    rep1.Synchronize "c:\My Documents\foo2.mdb", _
        jrSyncTypeImpExp, jrSyncModeDirect
End Sub

Sub TryToDeleteFromFoo2()
Dim rep As JRO.Replica
Dim cnn1 As New ADODB.Connection, cmd1 As ADODB.Command
' Open connection to foo2 and set a reference to it as a replica.
    cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=c:\My Documents\foo2.mdb"
    Set rep1 = New JRO.Replica
    rep1.ActiveConnection = cnn1

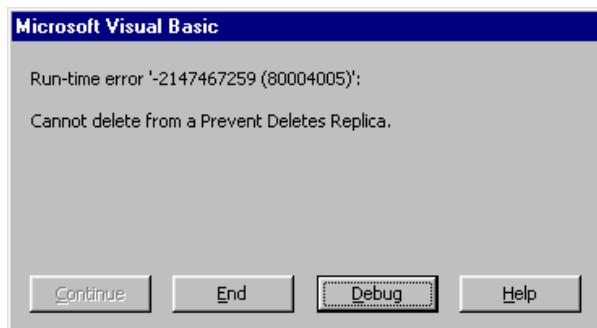
```

```

' Execute command to remove employee from foo2.mdb;
' it fails because Foo2 is a Prevent Deletes replica.
    Set cmd1 = New ADODB.Command
    With cmd1
        .ActiveConnection = cnn1
        .CommandText = "DELETE Employees.* FROM Employees" & _
            " WHERE LastName='Dobson'"
        .CommandType = adCmdText
        .Execute
    End With
End Sub

Sub synchFooFoo2ToDelete()
Dim rep1 As JRO.Replica
Dim cnn1 As New ADODB.Connection, cmd1 As ADODB.Command
' Open connection to foo and set reference to foo as a replica.
    cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=c:\My Documents\foo.mdb"
    Set rep1 = New JRO.Replica
    rep1.ActiveConnection = cnn1
' Execute command to remove an employee from foo.mdb.
    Set cmd1 = New ADODB.Command
    With cmd1
        .ActiveConnection = cnn1
        .CommandText = "DELETE Employees.* FROM Employees" & _
            " WHERE LastName='Dobson'"
        .CommandType = adCmdText
        .Execute
    End With
' Synchronize foo with its design master (Northwind.mdb).
    rep1.Synchronize "c:\My Documents\foo2.mdb", _
        jrSyncTypeImpExp, jrSyncModeDirect
End Sub

```



תרשים 11.3: הודעת השגיאה המוצגת בעקבות ניסיון למחוק רשומה מתוך עותק המוגדר עם התוכנה **מנע מחיקות**

עבודה עם מאפייני עותק משוכפל

בדיקת המאפיינים של עותק משוכפל泰山 בידך להבין את התנהוגותה של קבוצת עותקים משוכפלים. לדוגמה, אם ידוע לך סוג העותק המשוכפל, תדע גם אם אותו עותק ישתו שינווי סכמה עם עותקים משוכפלים אחרים. המאפיין Priority (קדימות) של עותק משוכפל קבוע אליו משני עותקים "ינצח" כאשר יש התנגשות בין שני עותקים. בכלל, העותק בעל הקדים מוגבהת יותר ינצח. שלוש השגורות הבאות מדפיסות את המאפיינים של עותק משוכפל וחושפות תוכן כדי כך ערבי מאפיינים.

הויל ולא קיים אוטס Replicas, קרובה לוודאי שתציגו לפי סדר רישימה ובה מספר עותקים משוכפלים שברצונך לעבד, במיוחד כאשר העותקים המשוכפלים משתמשים לקבוצות עותקים משוכפלים שונות. השירה callPrintTypePriority מגדרה משתמש path ו-replicaName עבור כל עותק משוכפל שהוא מבקשת לבדוק, ולאחר מכן היא מפעילה שתי שגורות משנהו. אחת מהן (printReplicaType), מחזירה את ערך המאפיין Priority של העותק המשוכפל, ואילו השנייה (printPriority) מחזירה את ערך המאפיין Path של העותק המשוכפל. בנוסף לארגומנטים Path ו-replicaName, קבוצת השגורות מחליפה ערכיהם עם ארגומנטו ששמו Exist. זהו משתנה בוליאני המציין אם קובץ מסוים קיים בנתיב נתון. אם הקובץ לא קיים, ברור מליו שאין לו ערך קדים.

```
Sub callPrintTypePriority()
On Error GoTo TypeTrap
Dim repMaster As New JRO.Replica
Dim path As String, replicaName As String
Dim Exist As Boolean
' Assign Boolean for file existing.
    Exist = True

' Assign path and replica names, then
' call procedures for printing type and priority.
    path = "C:\Program Files\Microsoft Office\Office\Samples\"
    replicaName = "Northwind.mdb"
' If file does not exist, the next line catches it.
    repMaster.ActiveConnection = path & replicaName
    printReplicaType replicaName, repMaster.ReplicaType, Exist
    printPriority replicaName, path, Exist
    replicaName = "Copy of Northwind.mdb"
    repMaster.ActiveConnection = path & replicaName
    printReplicaType replicaName, repMaster.ReplicaType, Exist
    printPriority replicaName, path, Exist
    replicaName = "Northwind2.mdb"
    repMaster.ActiveConnection = path & replicaName
    printReplicaType replicaName, repMaster.ReplicaType, Exist
    printPriority replicaName, path, Exist
```

```

path = "C:\My Documents\"  

replicaName = "foo.mdb"  

repMaster.ActiveConnection = path & replicaName  

printReplicaType replicaName, repMaster.ReplicaType, Exist  

printPriority replicaName, path, Exist  
  

path = "C:\Program Files\Microsoft Office\Office\Samples\"  

replicaName = "Partial of Northwind.mdb"  

repMaster.ActiveConnection = path & replicaName  

printReplicaType replicaName, repMaster.ReplicaType, Exist  

printPriority replicaName, path, Exist  
  

path = "C:\My Documents\"  

replicaName = "Partial of foo.mdb"  

repMaster.ActiveConnection = path & replicaName  

printReplicaType replicaName, repMaster.ReplicaType, Exist  

printPriority replicaName, path, Exist  

replicaName = "DMBackup.mdb"  

repMaster.ActiveConnection = path & replicaName  

printReplicaType replicaName, repMaster.ReplicaType, Exist  

printPriority replicaName, path, Exist  
  

path = "C:\My Documents\"  

replicaName = "foo2.mdb"  

repMaster.ActiveConnection = path & replicaName  

printReplicaType replicaName, repMaster.ReplicaType, Exist  

printPriority replicaName, path, Exist  
  

TypeExit:  

    Exit Sub  
  

TypeTrap:  

    If Err.Number = -2147467259 And _  

        Left(Err.Description, 19) = _  

        "Could not find file" Then  

            Exist = False  

            Resume Next  

    Else  

        Debug.Print Err.Number, Err.Description  

        Resume TypeExit  

    End If  

End Sub

```

```

Sub printReplicaType(repName As String, _
    typeNumber As Integer, Exist As Boolean)

    If Exist Then
        ' Decode replica type enumeration constants or...
        Select Case typeNumber
            Case jrRepTypeNotReplicable
                Debug.Print repName & " is not replicable."
            Case jrRepTypeDesignMaster
                Debug.Print repName & " is a Design Master."
            Case jrRepTypeFull
                Debug.Print repName & " is a Full Replica."
            Case jrRepTypePartial
                Debug.Print repName & " is a Partial Replica."
        End Select
    Else
        ' print that file does not exist.
        Debug.Print repName & " does not exist."
    End If

End Sub

Sub printPriority(replicaName As String, path As String, Exist)
    Dim repMasterP As New JRO.Replica
    ' Print priority and reset Exist.
    If Exist = True Then
        ' Assign connection for replica.
        repMasterP.ActiveConnection = path & replicaName

        If repMasterP.ReplicaType <> jrRepTypeNotReplicable Then
            ' Print priority for replicas.
            Debug.Print "It's priority is " & repMasterP.Priority & "."
        Else
            ' Print message for no replica.
            Debug.Print "Therefore, it has no priority."
        End If
    Else
        ' Print message for file does not exist.
        Debug.Print "Therefore, it has no priority."
    End If
    Debug.Print

    Exist = True
End Sub

```

השיגרה callPrintTypePriority משתמשת בלבידת שגיאות כדי לקבוע אם קובץ נתון קיים, וכי להגביל יכולות אס מתרבר שאינו קיים. כאשר היא מנסה להציג את המאפיין ActiveConnection של מופיע עותק משוכפל לקובץ שאינו קיים, רכיב השכפל של Jet מוחזר את קוד Err המקבול (2147467259-), בתוספת משפט תיאורי. מלבדה השגיאות משווה את המשפט עם קוד Err של השכפל ב- Jet. אם הלוגיקה ללכידת שגיאות קובעת שהקובץ אכן אינו קיים, המלכוות מצבה Urk False ברגומנט, printReplicaType ו- printPriority מפרשות את הערך False שהוצע ב- Exist ומגיבות בהתאם.

השיגרה callPrintTypePriority שולחת את ערך המאפיין ReplicaType עבור עותק משוכפל כאשר היא מפעילה את השיגרה printReplicaType. השיגרה בודקת את הערך של Exist לפני שהיא מנסה לפענחו את ערך המאפיין ReplicaType. אם Exist מכיל ערך False, השיגרה רק תציג בחלון Immediate (מיידי) הودעה על כך שהקובץ אינו קיים. אם Exist מכיל True, משפט SelectCase יפענחו את המאפיין ReplicaType.

הערה:



השיגרה printReplicaType מייצגת ערכי ReplicaType בעדרת שמות הקבוע ReplicaTypeEnum ב- JRO. תוכל לעיין בשמות אלה באמצעות **Object Browser** (סורק האובייקטים). הקלד ReplicaType בתיבה **Search text**, ולאחר על הלחוץ **Search**. הבחירה המוחזרת, בחר את המחלקה ReplicaTypeEnum עבור ספריית JRO (זהירות: בספריית DAO קיימת מחלוקת ReplicaEnum שונה). בחירה זאת תגרום להצגת רשימה של שמות הקבוע וערכים ReplicaType שלהם.

יעצוב השיגרה printPriority אחרת במקצת מזוה של השיגרה printReplicaType. השיגרה printPriority יוצרת בתוך עצמה מופיע של עותק משוכפל. לאחר מכן היא מצאה ערך מאפיין עבור העותק המשוכפל. בנוסף לבזיקת הערך המוצב במשתנה Exist, היא בודקת גם את המאפיין ReplicaType במשתנה jrlRepTypeNotReplicable. אם מתרבר שהעותק אינו ניתן לשכפל (לפי הערך המוחזר jrlRepTypeNotReplicable), השיגרה מדפסה הודעה שלפיה אין ערך קדימות עבור הקובץ. אחרת, השיגרה מדפסה את הערך המוחזר של המאפיין Priority במשתנה העותק המשוכפל. בטרם תחזיר את השיליטה לשיגרה לעבד את העותק המשוכפל הבא.

דחיסה והצפנה של עותקים משוכפלים

ספריית JRO תומכת ביוטר מעותק משוכפל אחד. לדוגמה, ניתן לדחוס ולהצפין קבצים, וכן ניתן לרענן את מטמון הזיכרון. שתי השגורות הבאות יוצרות גיבוי של עותק משוכפל על ידי דחיסתו והצפנה העותק הדחוס. גישה זאת מתאימה במיוחד

כאשר שולחים באינטרנט קובץ המכיל מידע רגיש. הדוגמה דוחשת את הקובץ Northwind2.mdb אל תוך Northwind.mdb. אם רצונך בכך, תוכל לציין נתיבים שונים .Northwind2 ו-Northwind עבור

```
Sub callCompactADB()
    compactADB "Northwind.mdb", "Northwind2.mdb", _
        "C:\Program Files\Microsoft Office\Office\Samples\"
End Sub

Sub compactADB(oName As String, cName As String, _
    opath As String, Optional cpath As String)
    Dim je As New JRO.JetEngine
    Dim strIn As String, strOut As String
    ' Is optional path specified?
    If cpath = "" Then
        cpath = opath
    End If

    strIn = opath & oName
    strOut = cpath & cName
    deleteFile strOut

    je.CompactDatabase _
        "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=" & strIn, _
        "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=" & strOut & ";" & _
        "Jet OLEDB:Encrypt Database=True"
End Sub
```

בנייה פתרונות עם MSDE ופרויקט' Access

Microsoft Access 2000 Microsoft SQL מציג שני חידושים בעבודה עם מסדי נתונים של Server. החידוש הראשון הוא MSDE (Microsoft Data Engine), מנגנון הנתונים של MSDE. Microsoft Server מאפשר אחסון של נתונים מקומיים בתבנית מסדי נתונים של SQL Server. בדומה למסדי נתונים של SQL Server, מסדי נתונים של MSDE יכולים לשמש כמערכות עורפיות בפתרונות העוסקים בקבצים מסוג *.adp. הפיתוח עם MSDE יקל עליך את המעבר מפתרונות המשרתים מרכזיים מקומיים, מחלקיים, לפתרונות מורחבים המשרתים את צורכיהם של ארגונים שלמים ובהם אלף משתמשים. החידוש השני הוא קבצים מסוג *.adp. (אותיות הסיומת adp מייצגות את המילים Microsoft Project). ב- Access Data Project פרויקטים של Access נעשו שימוש בטפסים, דוחות, מודולים ואפייו פקודות מאקרו, מקובל, אולם עם מסד נתונים של SQL Server במקום מסד נתונים Jet. קובץ מסד הנתונים מהסוג החדש אינו מכיל טבלאות או שאלות של Microsoft Access, אולם הוא מאפשר גישה אל טבלאות, תצוגות ושגרות מאוחסנות של SQL Server.

ברוק זה תלמד כיצד להתקין את מנגנון הנתונים של Microsoft ולנהל אותו, וכייזד לפתח יישומים מותאמים אישיות בעזרת פרויקטים של Access. כמו כן תלמד כיצד לעבד עם טבלאות, תצוגות, שגרות מאוחסנות, דוחות וטפסים. דוגמאות הקוד יבהירו כיצד ליצור אוטומציה של היישומים בפרויקטים של Access בעזרת ADO OLE DB (ActiveX Data Objects) ו- Microsoft VBA (הפרקים 3 ו-4 עוסקים בנתיבי TABS לתבניות נוספות של מסדי נתונים מרוחקים פרט ל- SQL Server).

מנגנון הנתונים של Microsoft (MSDE)

תוכל לנצל את מנגנון הנתונים של Microsoft לבניית פתרונות על גבי מחשבים אישיים בעלי מעבד יחיד, המופעלים במערכות הפעלה x9 Microsoft Windows או Microsoft Windows NT. פתרונות המופעלים על גבי MSDE מתאימים לקבוצות משתמשים, אולם ניתן בקלות להעביר פתרונות MSDE 7 גם ל- Microsoft SQL Server 7, Microsoft SQL Server Enterprise Windows NT Server Windows Edition, ומערכות הפעלה מתקדמות יותר מסדרת 2000.

פתרונות MSDE תואמים לחולtin את אלה שפותחו עבור SQL Server 7. תוכל ליצור דגם אב של פתרון בעזרת MSDE וקובצי *.adp. במחשב המקומי, ולאחר מכן להעביר את הטבלאות, התצוגות והשגרות המאוישנות של מסד הנתונים ל- SQL Server לצורך בדיקה, ליטוש ופרסום. ניתן להפץ בין הבוחנים והמשתמשים את קובצי *.adp. ולשנות את קישור הנתונים שלהם כך שייצביו אל מסד הנתונים של SQL Server במקום אל מערכת האב המבוססת על MSDE. יישומים מסוים זה מתאימים לצרכיו של ארגון, ולא רק לצרכים מחלקיים.

לעומת Jet MSDE

בunikom, פתרונות Access המבוססים על Jet קלילים יותר לניהול, מידת התאמות שלהם עם פתרונות שנבנו בעזרת גרסאות קודמות של Access גדולה יותר, והם גוזלים פחות מאשר MSDE. בניית פתרונות Access המבוססים על MSDE מחייבת בקיאות מסוימת בסוגי הנתונים החדשניים, בכלים השונים לפיתוח שאלות, ובטכניות של ניהול מסדי נתונים. לעומת זאת MSDE מציע אפשרות שחזור נתונים מעולות, רישום טרנזקציות מוככל, אבטחת Windows NT משולבת ופוטנציאלי לקיבולת מסד נתונים אדירים (בעת המرة ל- SQL Server). תוכל לנצל את משק המשמש של Access לבניית טפסים ודוחות עבור פתרונות המבוססים על כל אחד משני המנגנונים הללו. בשל העיצוב בתוכנות שרת-לקוח הטבוע בפתרונות MSDE, יהיה לךsciiti למספר כללים בהפעלת MSDE והמשק למשתמש. תוכל לתכנן פתרונות לשני מנגנוני מסדי הנתונים גם בעזרת VBA ו- ADO.

הן MSDE והן Jet תומכים בשילומיות קשרים מודרנת, דבר שמאפשר הצהרה על יחסים בין טבלאות באמצעות גрафים. עם זאת, בעוד Jet תומך בעדכונים ומיחיקות בהתאם להיררכיה קשרים, ברמת המנגנון, עם MSDE חובה למשתמש תכונות אלה ב- SQL באמצעות גורמים (triggers) טבועים בקוד. ניתן ליצור תבנית גורם לטבלה בלחיצת עכבר ימנית על הטבלה הרצויה בחולון **מסד נתונים** (Database), ובבחירה באפשרות **גורמים** (Triggers). לאחר מכן יש ללחוץ על **חדש** (New) כדי לפתח את התבנית. ב- SQL Server Books Online תמצא כללי תחביר, דוגמאות ורקע כללי על פיתוח גורמים).

אם נדרש רק שחזור מלא מתוך קובץ גיבוי, בחר **כלים (Tools)**, **עזרי מסד נתונים (Database Accessories)**, **שחזור (Restore)** בפרויקט Access. פועלה זאת דומה לשימרת עותקי גיבוי של קבצי מסד נתונים - Jet ו恢復 הקבצים מתוך העותקים. רק MSDE תומך בשחזור עד לנקודת זמן מסוימת מתוך קובץ יומן. עליך לתוכנת שחזור מסווג זה בעורת SQL, Transact-SQL, שיש לו תמייהה בעורחה המקורי של 2000 Access. העזרה המקוונת כוללת דוגמאות script ותחביר לשחזור נתונים בטבלה עד לנקודת זמן מסווג מותק קובץ היומן של הטבלה.

הערה:



אם בנסיבות מסוימות מוטל عليك ניהול אוסף של שימושות מהחייבות שימוש ב- Transact-SQL, רצוי להשיג עותק מלא של SQL Server (במקום חבילת MSDE החופשית המכילה עיקרי קוד זהה של מגנון מסד נתונים). Enterprise Manager מכיל אוסף גדול של אשפים וכן משק גרפי למשתמש, המבטל את הצורך בקוד של Transact-SQL לביצוע שימושים כמו גיבוי ו恢復.

קובצי מסד נתונים ב-Jet הם ככל קטנים מקבצים שוווי-ערך המבוססים על MSDE. גודלו של קובץ Northwind.mdb עומד על כ-2MB. היקף זה כולל את כל טבלאות מסד הנתונים, שאר האובייקטים של מסד הנתונים, והקוד. גירסת Northwind המבוססת על MSDE כוללת קובץ מסד נתונים (*.mdf) וקובץ של פרויקט Access (*.adp). גודלם של קבצים אלה עומד על כ-3.7MB. פתרונות MSDE מנהלים באופן אוטומטי קבצי יומן המשיעים במקורי שחזור. עם זאת, על מסדי נתונים ב-Jet להעביר את מסדי הנתונים שלהם ברחבי הרשות כשמדבר בפתרונות מרובי-משתמשים. לעומת זאת פתרונות המבוססים על MSDE מבצעים את כל שימושים בלבד הנוטניים בשרת, ומעבירים רק את הקבוצות המוחזרות ברחבי הרשות. דבר זה עשוי להפחית את נפח התעבורה ברשת, ולהאיץ את מהירות הביצוע. הן Jet והן MSDE מאפשרים מסדי נתונים בנפח של עד 2GB, אולם פתרונות MSDE ניתנים להמרה בקלות ב-MSDE 7 SQL Server. רמת יכולם לתמוך במסדי נתונים בנפח העולה על 1 מיליון טרה-בייט (ט'יב). היביצועים של SQL Server 7 עולה ככל שעולה מספר המעבדים במחשבים שבהם הוא מופעל, אולם רמת היביצועים של Jet אינה משתנה באופן משמעותי כתוצאה נוספת של מעבדים.

הערה:



בדומה ל-7 SQL Server הוותיק יותר, MSDE מיועד לעבודה עם קבוצות עבודה קטנות.

Jet ו-MSDE נועדו, כל אחד, למטרות מיוחדות. MSDE הוא הכליל האידיאלי עבורך אם בכוונתך להשתמש ב-SQL Server בגרסאות עתידיות של היישום. כאשר MSDE פועל במערכות Windows NT, הוא מעניק אבטחה משולבת של מערכת ההפעלה ומסד הנתונים, דבר העשויה להפחית את העומס בתחום ניהול האבטחה. כמו כן MSDE

מאפשר שחזור לנקודת זמן נתונה. לעומת Jet מותאם יותר לישומים שאיןם יישומי משהימה קרייטיים, אשר השיקול המכריע בהם הוא פשוטות הפיתוח. פתרונות Jet גוזלים פחות מקום ובמקרים מסוימים הם עדיפים, אם אתה כפוף למוגבלות של זיכרונו או מקום פנוי בדיסק. הוואיל ומנגנון Jet המשווק עם Access 2000 עודכן במידה שולית בלבד, רמת התאמיות הגבוהה ביותר שלו היא מול Access 97 וגרסאות קודמות.

התקנה והגדרת התצורה של MSDE

MSDE הוא מנגנון מסד נתונים אופציונלי למפתחי ומשתמשי Access. תוכל להתקין אותו על ידי הפעלת הקובץ SetupSQL.exe מתוך תקליטורי Office 2000. אין צורך בהכנות מיוחדות לפני התקנת MSDE במערכות הפעלה 9x Windows. לעומת זאת, על משתמשי Windows NT 4 לבצע שתי פעולות כהכנה להתקנת MSDE. תחילת עליהם להתקין את Service Pack 3, ולאחר מכן להפעיל את הקובץ hotfix.exe מתוך תקליטורי Office 2000 ב- Windows NT 4 Service Pack 4. לאחר מכן נדרש בפועלות מכינות לקראת התקינה.

לאחר שהתקנת את MSDE, עליך להפעיל את המנגנון לפני השימוש בו. במחשבים עם מערכות הפעלה 9x Windows, בצע את הפעולות הבאות:

1. בתפריט **התחלת** (Start) בחר באפשרות **תוכנות** (Programs).
2. בחר באפשרות **Msde** ולאחר מכן בפרק **Service Manager** (מנהל השירותים).
3. בתיבת הרשيمة הנפתחת **Services** (שירותים), בחר באפשרות **MSSQLServer**, ולאחר על **Start/Continue** (התחל/המשך).

תוכל לבחור את תיבת הסימון **Auto-Start Service** אם אתה מעוניין בהפעלה אוטומטית של MSDE בכל אתחול של מערכת ההפעלה. אם לא בחרת באפשרות זאת, תיאלץ לחזור על הפעולות המתוארות לעיל כל פעם שתשתמש ב- MSDE.

משתמשי NT יכולים להפעיל את MSDE כמו כל שירות אחר של Windows NT. בלוח הבודקה, לחץ לחיצה כפולה על הסמל **Services**, ולאחר מכן בחר באפשרות **MSDE Services**. לחץ על **Start** בתיבת הדיו-שייה **Startup Type** כדי להפעיל את השירות. בעת לחץ על **Startup Type** ובחר אחת מהאפשרויות בפרק **Startup Type** **(Automatic)**, **(Disabled)**, או **Manual**.

לאחר ההתקנה וההפעלה, יופיע בשורת המשימות של Windows סמל המציג את ניהול השירותים ב- SQL Server. השתמש בסמל זה כדי להפעיל, לה歇ה או לעצור את MSDE. השתמש בו גם לפיתוח **Service Manager** כדי שתוכל לבצע פונקציות דומות Microsoft SQL Server וסוכן Microsoft Distributed Transaction Coordinator עבור

מקובל לדרוש מהמשתמש להזין זיהוי וסיסמה בטרם יורשה לפתח קובץ עבור מסד נתונים שרת-לקוח. MSDE מאפשר לך לנחל את אבטחת הכנינה בעזרת חשבונות פרטיים וקובוצתיים עבור מסדי נתונים מוגדרים, או בעזרת אבטחת pass-through של Windows NT. ערכי בירית המהידל של MSDE הם sa כזיהוי משולבת הלוקחה מממשקת Windows. כדי ליצור חשבונות אבטחה לגישה אל מסדי נתונים פרטיים, בחר **כלים** (Tools), **בטחה** (Security), **אבטחת מסד נתונים** (Database). משתמשי Windows 9x יכולים להשתמש בחשבונות אבטחה של MSDE רק למסדי נתונים פרטיים. התפקידים של מסדי נתונים ב- MSDE דומים לתפקידי הקבוצות ב- Access. מסד נתונים ייחד יכול לשמש תמייך לשינויים שונים, או יותר, והוא יורש את רמת האבטחה הנמוכה ביותר מלאה החלות בתפקידים השונים שליהם הוא משתיך.

כאשר מפעילים את MSDE מתוך Windows, הוא תומך בפרוטוקול TCP/IP וב- Multi-Protocol, אך לא ב- Named Pipes. עם זאת, רוב הלוקוחות מנסים להתחבר לשרת MSDE באמצעות Named Pipes כברירת מחדל. אם אתה מעוניין שמחשב-לקוח ייצור כניסה עבור שרת MSDE באמצעות TCP/IP או באמצעות协议 Multi-Protocol, בחר בפקודה Client Network Utility Msde מתוך תפריט המשנה msd, לאחר לחיצה על **התחל**.

פרויקט Access

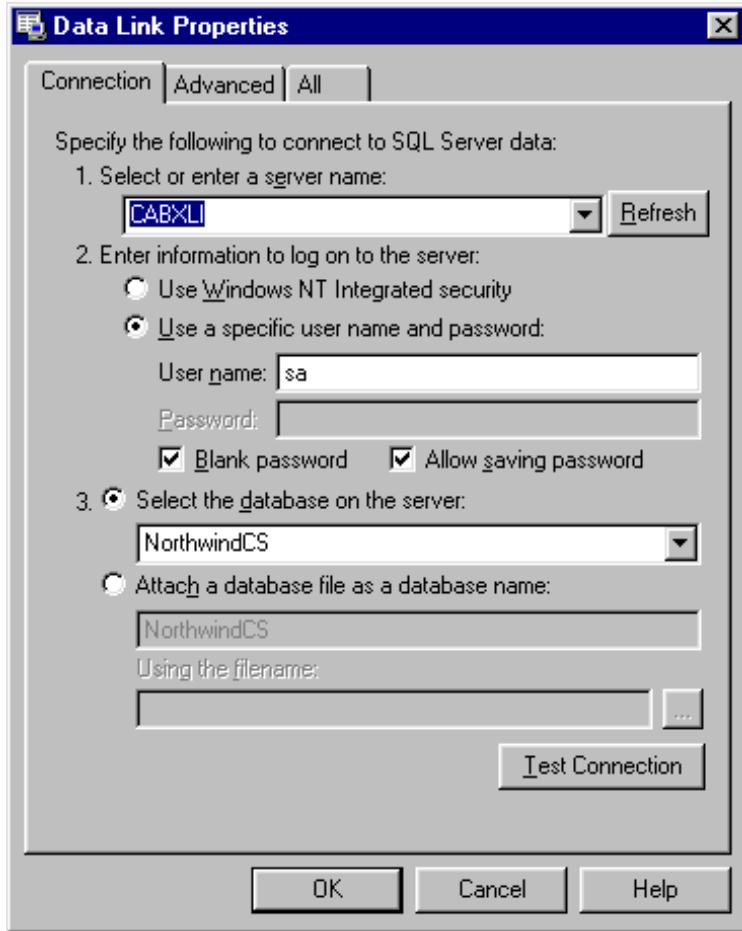
בעזרת פרויקטים של MSDE עם SQL Server, תוכל לפתח יישומי שרת-לקוח באותה קלות שאתה מפתח יישומי שרת קבצים. המשך של פרויקטי Access חושף תשעה סוגים בסיסיים של אובייקטי יישומי. טבלאות, צצגות, שגרות מאוחסנות ודיagramות של מסדי נתונים מייעדים לקטולוג של מסד נתונים MSDE או SQL Server; אלה נשמרים בקובץ מסד הנתונים. טפסים, דוחות, דפי גישה לנתונים (לפיזור תכני Web), פקודות מאקרו ומודולים נשמרים בקובץ *.adp*. פרויקט Access מבצע תיאום עם מסד הנתונים שרת-לקוח באמצעות חיבור OLE DB. קובץ מסד הנתונים שהוא בדרך כלל קובץ מסוג *.mdf*, וקובץ *.adp* – הם שני החלקים המרכיבים את פתרונו השרת-לקוח.

חיבור פרויקט Access עם מסד נתונים

תוכל לחבר קובץ *.adp עם מסד נתונים אחד הסוגים הבאים:

- ↳ קובץ מסד נתונים 7 SQL Server במערכת 9x Windows NT או Windows 2000 (עם Service Pack 4 ומעלה).
- ↳ קובץ MSDE במערכת 9x Windows NT או Windows 2000 (עם Service Pack 5) Windows NT SQL Server 6.5 ומעלה).

כדי לפקח על החיבור עם מסד הנתונים שרת-לקוח, או כדי לאפס אותו, בחר באפשרות **התקשרות** (Connection) מתוך התפריט **קובץ** (File). בתיבת הדו-שיח **Data Link Properties** (המוצגת בתרשים 12.1, להלן), בחר בשרת של מסד נתונים, סוג אבטחה ושם מסד הנתונים. בדרך כלל, שם השרת של מסד הנתונים הוא שם המחשב שבו מופעל השירות.



תרשים 12.1: תיבת הדו-שיח **Data Link Properties** (מאפייני קישור הנתונים)

התרשימים מציג חיבור אל מסד נתונים MSDE הפועל במחשב בשם CABXLI.CABXLI. שם פרט בתיבת הדו-שיח איןנו מצין במפורש שמדובר במסד נתונים MSDE, ולא במסד נתונים SQL Server. הואיל והמחשב CABXLI מבוסס על מערכת ההפעלה Windows 9x, אין אפשרותתו להפעיל אבטחה מושלבת של Windows NT. תיבת הדו-שיח **Data Link Properties** משמשת בערכיו בירית המחדל של נתוני הקישוריות הייחודיים למשתמש. לאחר שהוקם הקשר עם שרת מסד הנתונים, תיבת הרשימה הנפתחת מציגה את כל מסדי הנתונים שנמצאים בשרת. אחרי שתבחר בפריטים

הרצויים, קובץ adp*. יאכלס באובייקטים את הטבלאות שלו ואת התצוגות, השגרות המאוחסנות ואוסף התרשיים של מסדי הנתונים. הגדרות אלה נשארות בתוקפן בין מופעים שונים של פרויקט Access נתון.

לימוד ממדי הנתונים ו-Pubs - NorthwindCS

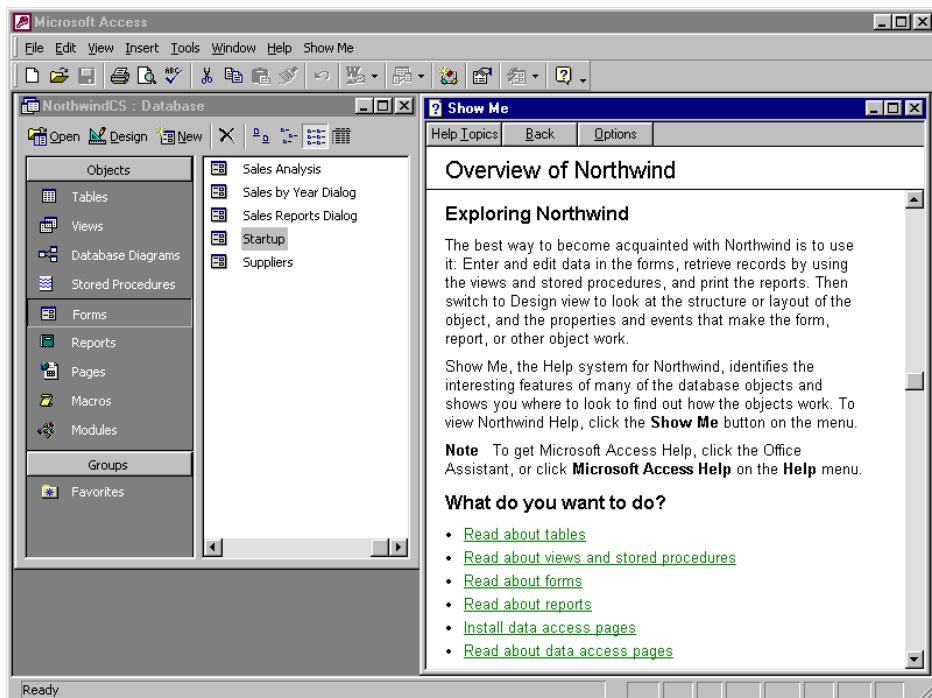
משווק הן עם שרת הקבצים והן עם גרסת שרת-לקוח של יישום ההדגמה Northwind. גירסת השרת-לקוח מוגדרת כפרויקט Pubs. מסד הנתונים Pubs אינו משוק עם Access 2000, אולם מזה זמן רב הוא נחשב למסד הדוגמה הסטנדרטי של SQL Server. אם אתה מפעיל פרויקטים של Access עם ניהול מסד נתונים SQL Server. דוגמאות ADO ודוגמאות מסדי נתונים רבות אחרות בקבץ העורקה המקוונת של Access 2000 מכילות הפניות למסד נתונים זה. לכן, תוכל להרחיב את היכרותך עם התכונות החדשנות של מסדי נתונים ב- Access 2000 באמצעות ההיכרות עם מסד הנתונים Pubs.

שימוש בפרויקט ובמסד הנתונים של NorthwindCS

הקובץ NorthwindCS.adp מועתק אל המחשב שלך בעט ההתקנה של Access Samples (מסדי נתונים לדוגמה של Access). זהו פרויקט Access בעל רכיבים תפקודיים דומים ונתונים זהים לאלו שבקובץ Northwind.mdb הקלאסי. MSDE מעמיד לרשותך את שני הקבצים. כדי להשתמש ב-NorthwindCS, عليك להתקין תחילה את MSDE במחשב, או ליצור חיבור עם ניהול מסד נתונים 7 SQL Server, מכיוון שקבצי מסד הנתונים של Northwind מושוקים עם שני מוצרים אלה.

כאשר תפעיל את NorthwindCS.adp לראשונה, קטע script מתאים יברר אם MSDE מותקן במחשב שלך. אם כן, תתבקש להורוות l-script אם ברצונך לטעון את מסד הנתונים ולהברר אליו את פרויקט NorthwindCS. אם לא, תוכל להציג את פרויקט NorthwindCS אל מסד הנתונים של Northwind במנהל מסד נתונים SQL Server.

פרויקט NorthwindCS מוסיף פריט **Show Me** (הראה לי) מיוחד בשורת התפריטים. אם תבחר בפריט זה, תוצג לפניך תיבת דו-שיח (הנראית בתרשים 12.2), המעמידה לרשותך את הדוגמה של פרויקט Access. במסגרת הדוגמה יוסברו התכונות המייחודות של פרויקט Access וסוגי הנתונים השונים ב-SQL Server. על אף קיום התאמאה מסוימת בין סוגי הנתונים ב-Jet וב- SQL Server, יש ביניהם גם הבדלים משמעותיים. לדוגמה, סוג הנתונים **Timestamp** הוא ייחודי ל-SQL Server. כמו כן, ב-SQL Server ניתן ליצג ערכי **Currency** (מטבע) וערבי **Date/Time** (תאריך/שעה) באופן שונה מאשר ב-Jet. נוסף לכך, ב-SQL Server קיימים במפורש סוגי נתונים Unicode וכolumbia שאינם ב- Unicode.



תרשים 12.2: תיבת הדו-שייח Show Me בפרויקט NorthwindCS

הבדל עקרוני נוסף בין פרויקטים של Access וקבצי *.mdb. מוקבלים טמון בשימוש בתצוגות, בשגרות מאוחסנות ובשאליות. תיבת הדו-שייח Show Me מסבירה כיצד התצוגות והשגרות המאוחסנות מחליפות את השאליות בישומים מותאמים אישית. זכור, שבאזור קבצי *.mdb, תצוגות ושגרות מאוחסנות הופכים לזמן נון מול מנגןון מסד הנתונים Jet, אולם יש הבדל בין כללי התחריר החלים על תצוגות ב-Jet לעומת SQL Server. לדוגמה, קבצי *.mdb. משפרים מיוון רשומות בתצוגה באמצעות מונח ORDER BY SQL. המשפט ORDER BY ב-SQL עברור תצוגה אסור הוא ORDER BY SQL וhaven't ORDER BY MSDE. במנחי מסדי נתונים אלה מסוג שרת-לקוח, משפט ORDER BY שמור לשגרות מאוחסנות בלבד.

אם תיבת הדו-שייח Show Me אינהעונה על צרכיך באופן מלא, נסה להיעזר בעזרה המקוונת של Access. פתח את העזרה המקוונת וחפש עזרה בנושא "עובדת עם פרויקט Microsoft Access". תמצא עזרה מקוונת נוספת בתחום פניות פרויקטים של Access אם תפתח טבלה של פרויקט Access בתצוגת **עיצוב** (Design) ותלחץ על F1. סגור את חלון העזרה של Microsoft Access לפני הלחיצה על F1 כדי להיכנס לחلك אחר של מערכת העזרה. בתחום זה של העזרה המקוונת תמצא גם פרטיהם אודוט SQL Transact-SQL וחוודעות שגיאת ב-SQL Server. לבסוף, תוכל להסתיע גם ב-SQL Server Books Online (כדי להיעזר במשאב זה נדרש לך מחשב שמותקנת בו הגירסה המלאה של 7.(SQL Server 2000).

שימוש בmseד הנתונים Pubs עם פרויקטים של Access

אם עבדת בעבר עם ODBC Direct, או שפיתחת פתרון מותאם לmseד נתונים מרוחק בעזרת Access, קרוב לוודאי כבר פגشت mseד הנתונים Pubs. תוכל להסתמך על היכרות זאת ולהרchieבה בעובדה עם פרויקטים של Access. זכור שדרוש לך עותק של SQL Server כדי לעבוד עם mseד הנתונים Pubs. תוכל להפעיל את mseד הנתונים עם פרויקט Access ב- SQL Server 6.5 או ב- SQL Server 7.

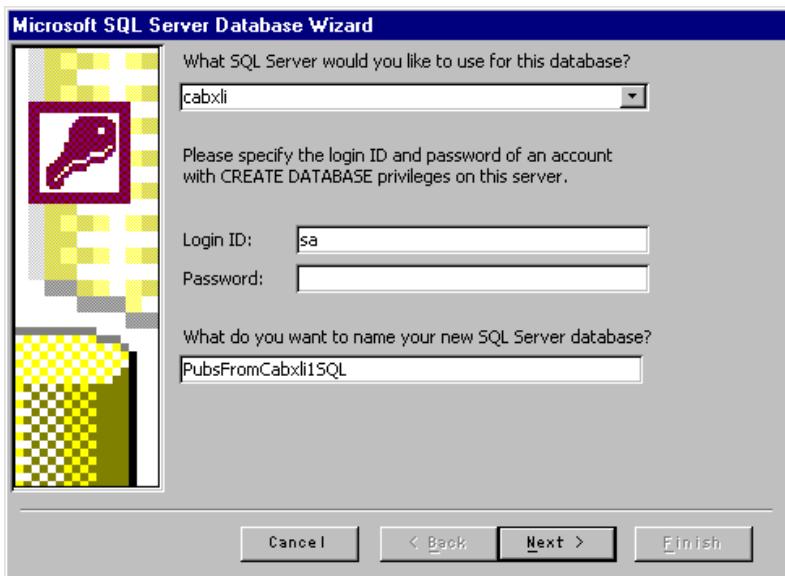
כדי לעבוד עם mseד הנתונים Pubs מותוך פרויקט Access דרוש לך קובץ `adp.*` שיצבייע אל Pubs. אם לא עומד לרשותך קובץ `adp.*` מנותק, תוכל ליצור קובץ זהה על ידי יצירת פרויקט חדש ולאחר מכן שחרור mseד הנתונים שלו. בשלב הבא عليك לפתוח את תיבת הדיו-שיך **Data Link Properties** ולכוון את פרויקט המנותק כך שיצבייע אל mseד הנתונים Pubs.

כדי ליצור קובץ `adp.*` חדש, הפעל את Access וצור mseד נתונים חדש. לשם כך בחר בלחצן האפשרות **ASFי mseד נתונים, דפי ופרויקטים של Access Database** (**New**). פעולה זאת תפתח את תיבת הדיו-שיך חדש (**Wizard, Page and Projects** (לחלופין, תוכל להזוץ על הפריט חדש בסרגל הכלים **mseד נתונים**). בחר ב录制ית **כללי** (**General**), ולאחר מכן לחץ להחיה כפולה על הפריט **פרויקט (mseד נתונים חדש)** (**Project (New Database)**). בתיבת הטקסט **שם הקובץ** (**Name**) הקלד שם עבור קובץ `adp.*` החדש. לחיצה על **צור** (**Create**) מפעילה את **ASF mseד נתונים** (**Microsoft** **SQL Server Database** (המודגש בתרשים 12.3). האשף מכיל את שם השרת המקומי CABXLI (בדוגמה שלנו), וכן הצעה של שם mseד הנתונים, המבוססת על שם הקובץ שהקלדת. עליק להזין את נתונים חשבון המשמשו (זיהוי משתמש וסיסמה – sa כזיהוי ותו ריק כסיסמה בדרך כלל מספיקים). כעת לחץ על **הבא** (**Next**) ולאחר מכן על **סיום** (**Finish**) כדי להשלים את התהליך. מרגע שפרויקט Access זמין, נתק אוטומטית **ASF mseד נתונים** (**Drop-SQL-Database** (**Database Accessories**)), **שחרור mseד נתונים של SQL** (**Tools**), **עוזרי mseד נתונים** (**Tools**). לבסוף, פתח את תיבת הדיו-שיך **Data Link Properties** של פרויקט Access המנותק. הzon את הנתונים שיכוונו את קובץ הפרויקט אל mseד הנתונים Pubs ב- SQL Server.

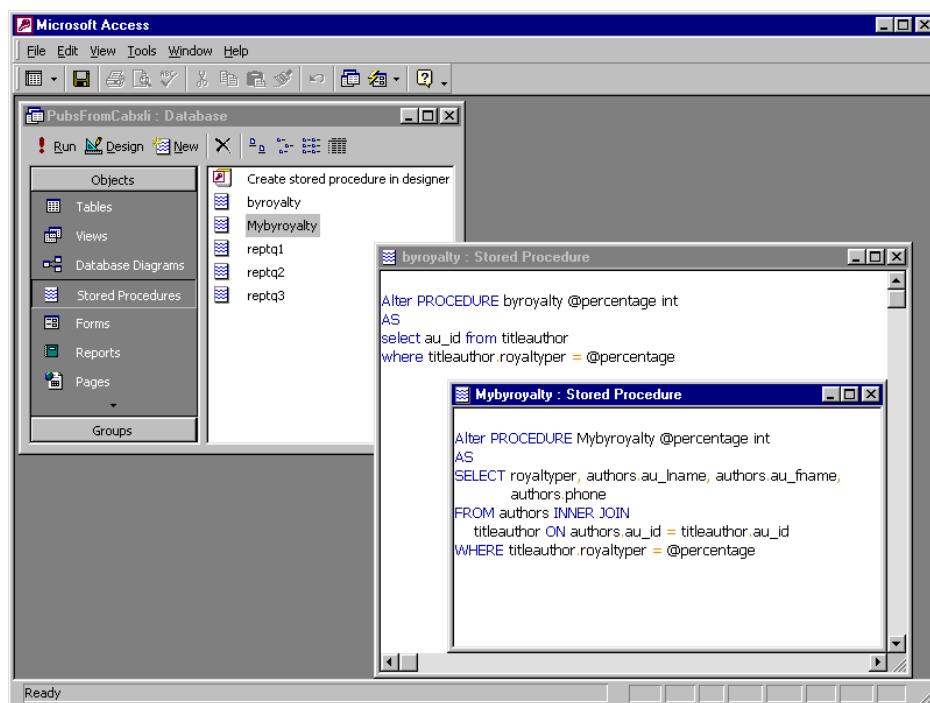
הערה:



אם כל שדרוש לך הוא קובץ `adp.*` מנותק (כדי לכוון אותו אל mseד נתונים Pubs או mseד נתונים אחר כלשהו), תוכל לkcטר את התהליך. הקש על המקש Esc כאשר מופיע מסך הפתיחה של **ASF mseד נתונים Microsoft SQL Server Database**. לאחר מכן פתח את תיבת הדיו-שיך **Data Link Properties** של קובץ `adp.*` המנותק שייזכר, וכoon אותו כך שיצבייע אל mseד הנתונים הרצוי.



תרשים 12.3: מסך הפתיחה של אשייף מוד נתונים Database



תרשים 12.4: השגרות המאוחסנות במסד הנתונים Pubs

לאחר יצירת פרויקט Access המקשר אל מסד הנתונים Pubs, תוכל לשאוב ממנו דוגמאות מעניינות של הטכנולוגיה SQL Server שיאפשרו לך ללטש את הכלים שלך. תרשימים 12.4 מציג את ארבע השגורות המוחשנות ב-Pubs בסגירה byroyalty פטואה. דוגמה זאת מוכיחת את זיהויי המחבר של כל חבר בעל הסכם על אחזוי תמלוגים, אשר توأم את הקטל היחיד בתגובה לשאלת הפרמטרים. השגירה המוחשנת Mybyroyalty היא הרחבה של השגירה הראשונה, והיא מוכיחת את שמו הפרטי, שם המשפחה ומספר הטלפון של המחבר.

שחזור מסדי נתונים של SQL Server

SQL Server הוא מסד נתונים בעל עוצמה תעשייתית, אולם ניתן לפגוע במערכת באופן שגורם לאובדן יכולת לעבוד עם הנתונים. לדוגמה, ככל במדיה עלול להשחת את העותק הראשי של מסד הנתונים. כאשר הנזק חמור מאוד, יתכן שלא תצליח כל להפעיל את SQL Server. במקרה זה תיאלץ להסיר את ההתקנה של SQL Server או MSDE. להסרת התקנה פגומה, בחר באפשרות **Uninstall MSDE** מתפריט התחלה (Start) או מلوוח הבדיקה. פעולה זאת תסיר את כל מסדי הנתונים של המערכת, כמו העותק הראשי של מסד הנתונים, אולם תשאיר על כמה מסדי נתונים של משתמשים, כמו למשל NorthwindCS.

כasher תחליל לעבוד עם MSDE או עם SQL Server, בודאי תחזירשוב ושוב על התקנתה מנהל מסד נתונים שרת-локוט כדי להעניק את היכרות עם התהילה. לאחר התקנתה מחדש של SQL Server או MSDE, העתק אליהם את העותק הראשי החדש מותך ותוך גיבוי עדכני של העותק הראשי של מסד הנתונים, או בהנה מחדש העותק הראשי כך שיכיר במסדי הנתונים הנוכחיים. כל אחת משתי השיטות הללו מאפשר לך להתחבר עם מסדי הנתונים הנוכחיים.

רצוי בהחלט לשמר עותק שוטף של העותק הראשי של מסד הנתונים, הואיל ודבר זה עשוי לפחות פועלות החיבור של התקנה חדשה של SQL Server עם מסדי נתונים קיימים. צור גיבוי לעותק הראשי של מסד הנתונים לאחר יצירה או מחיקה של מסד נתונים חדש של משתמש, או חשבונות כניסה למערכת. הוספה משתמש חדש למסד נתונים אינה משנה את העותק הראשי, מכיוון שננתוני אבטחה, כמו חשבונות משתמשים, נכנסים אל תוך קובץ מסד הנתונים.

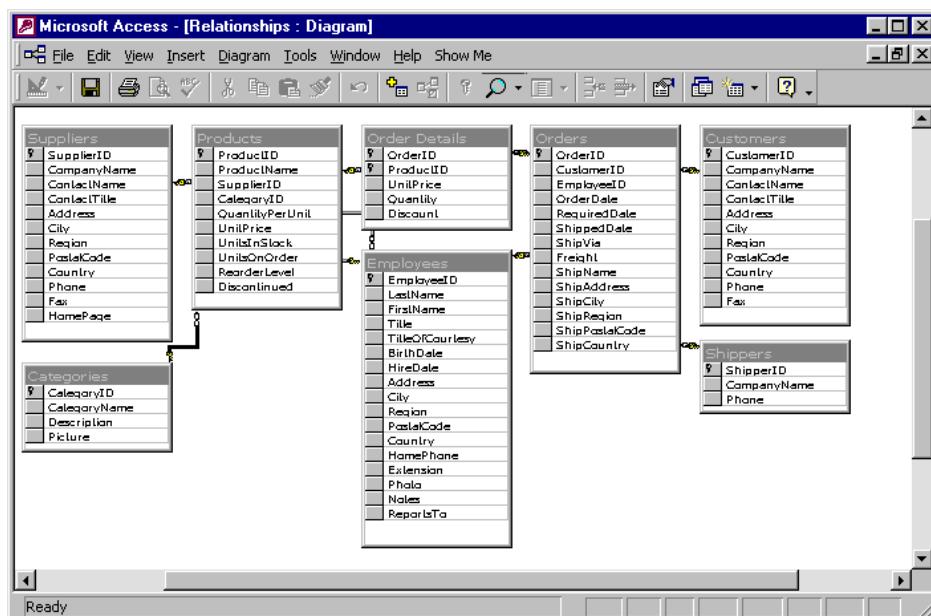
שחזור העותק הראשי כרוך בשינוי השם של קבצי מסדי הנתונים היישנים של המשתמשים, ויצירת מסדי נתונים של מצינני מקום עם שמות מסדי הנתונים הקודמים. عليك לצורך פרויקט Access עבור כל קובץ מסד נתונים שכונתך לשוחרר. כאשר אתה יוצר פרויקט Access חדש, ערוץ את השם בתיבות הדיו-שים שלו באשף מסדי הנתונים Microsoft SQL Server Database, והקצת לו את שמו של אחד מקובצי מסד הנתונים שאתה מבקש לכלול בעותק הראשי החדש. לאחר בניית פרויקט Access חדש עבור כל אחד מקובצי מסדי הנתונים שאתה מבקש לשוחרר, העתק את קבצי מסדי הנתונים היישנים (אלה שאთ שמות שנייה) על גבי הקבצים החדשניים שיוצרים. פעולה זאת מעדכנת את העותק הראשי כך שיזהה את קבצי מסדי הנתונים של המשתמשים מההתקנה הקודמת. כל קובץ ישן מועתק אל שם שהעותק הראשי מכיר.

דיאגרמות וטבלאות של מסדי נתונים

דיאגרמה של מסד נתונים היא אוסף של אובייקטי מסד נתונים, בדומה לטלואות, תצוגות ושרורות מאוחסנות. Access 2000 שומר אותן יחד עם קובץ מסד הנתונים של SQL Server או MSDE, שלא כמו פרויקט Access. בעוד שדיאגרמה של מסד נתונים נפרד עשויית לדמויות לתצוגה המופיעה בחלוון **קשרי גומלין** (Relationships) של קובץ *.mdb. קלאסי ב-Access, דיאגרמות של מסדי נתונים שרת-לקוח גמישות הרבה יותר, ומאפשרות שימוש הדזקה יותר בעיצוב מסד הנתונים. לדוגמה, תוכל ליזום עיצוב של טבלאות חדשות מתוך חלון של דיאגרמת מסד נתונים.

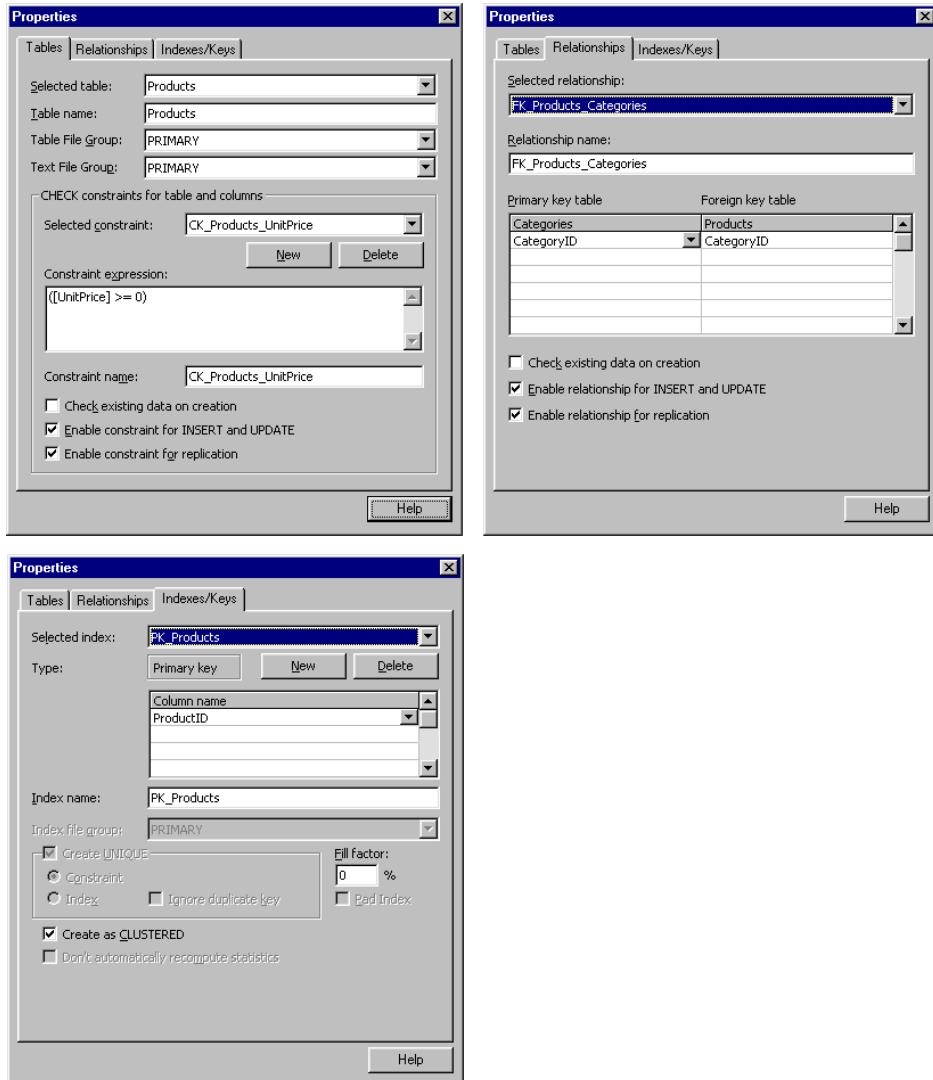
מיפוי קשרי גומלין באמצעות דיאגרמות של מסדי נתונים

תרשים 12.5 מציג את דיאגרמת מסד הנתונים של מסד הנתונים NorthwindCS. בדומה לחלוון **קשרי גומלין** בקבצי Access רגילים למסדי נתונים Jet, הדיאגרמה מתארת את הטעלאות שבשימוש במסד הנתונים ואת קשרי הגומליין ביניהם. תוכל לבצע ציווון עדין של קשרי הגומליין בין שתי טבלאות על ידי לחיצה על הקו המחבר ביניהן, ובחרה באפשרות **מאפייניות** (Properties) מຕפריט הקיזור. בתרשימים 12.5 מוצג קשר הגומליין בין הטעלאות **Products** ו-**Categories**, **Employees** ו-**Order Details**, **Products** ו-**Order Details**.



תרשים 12.5: דיאגרמת מסד הנתונים מתוך פרויקט NorthwindCS

תרשים 12.6 מציג את תיבת הדו-שייח Properties (מאפיינים) עבור קשר הגומלין בין שתי הטבלאות שנבחר בתרשים 12.5. הכרטיסיות מכילות הרבה יותר מידע אודוות קשיי הגומלין בין הטבלאות מאשר התצוגה בחלון **קשרי גומליין** בקובץ מסד נתונים קלאסי של Jet או Access.



תרשים 12.6: תיבת הדו-שייח Properties עבור קשר הגומلين בין Products ל- Categories במסד הנתונים NorthwindCS

השתמש בכרטיסייה **Tables** (טבלאות) כדי לשנות בתנאי האימוט שיכולו על נתונים המזומנים אל תוך הטבלאות. למשימה זאת, השתמש במוגבלות. כדי לבחור הגבלה, עירוך אותה או למחוק אותה, בחר בהגבלה הרצוייה מתוך תיבת הרשימה הנפתחת

חדש על טבלה. לחץ על **Delete** (מחק) כדי למחוק את הגבלה הנבחרת באותו רגע. שלוש תיבות הסימן שבתחתית הכרטיסיה **Tables** קובעות متى הגבלה מסוימת תיכנס לתוקפה. שתי הבחירה בפינה השמאלית העליונה בתרשים 12.6 מחלות את הגבלה כאשר משתמש מוסף או מעדכן נתונים בטבלה, או כאשר הטבלה מועתקת אל מסד נתונים אחר. השתמש ברשימה הנפתחת **Selected Table** (טבלה נבחרת) בראש הכרטיסיה **Tables** כדי לבחור טבלה אחרת ולהחיל עליה הגבלות נוספות.

במצב הראשוני הכרטיסיה **Relationships** (קשרי גומלין) מראה את קשר הגומלין בין הטעלות **Products** ו-**Categories**. באפשרות לבחון גם את קשרי הגומלין של הטבלה **Products** עם טבלאות אחרות, כמו **Suppliers** או **Order Details**, וזאת באמצעות תיבת הרשימה הנפתחת **Selected Relationships** (קשרי גומלין נבחרים). כדי להגיע לתוצאה זהה בקובץ **.mdb**.*. עילך לבחור במפורש את קשר הגומלין הרצוי בחלון **קשרי גומלין**.

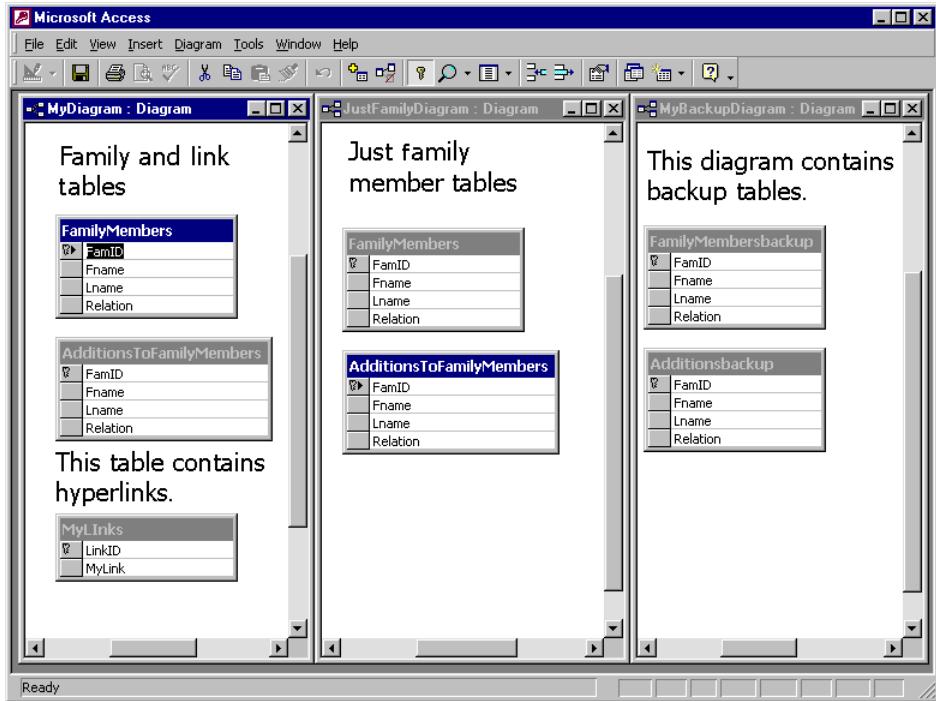
הכרטיסיה **Indexes/Keys** (אינדקסים/מפתחות) מציגה הגדרות עבור המפתחות והאינדקסיםקיימים של הטבלה, ומאפשרת הוספת פריטים חדשים או מחיקת פריטיםקיימים. השדה **ProductID** מגדיר מפתח ראשי עבור טבלה **Products**. האינדקס מאורגן באשכולות, ופירוש הדבר זהות בין הסדר הפיסי והסדר הלוגי של השורות.

ניהול דיאגרמות של מסדי נתונים מרובים

תרשים 12.7 מציג שימוש אפשרי נוסף בדיאגרמות של מסדי נתונים. בתרשים נראה שלוש דיאגרמות שונות, שכלן מותיחסות אל אותו מסד נתונים. קבצי **.mdb***. קלאסיים אינם מאפשרים הצגה של דיאגרמות מרובות המתארות את קשרי הגומלין בתוך מסד נתונים; לכל קובץ חלון **קשרי גומלין** יחיד. לעומת זאת, ב-**SQL Server** ניתן להציג חלונות מרובים למסדי נתונים. חלונות של דיאגרמות מסדי נתונים עשויים להכיל תכנים בלבדים או חופפים. לדוגמה, שתי הטעלות – **FamilyMembers** ו-**AdditionsToFamilyMembers** – מופיעות בו בחלון **MyDiagram** והן בחלון **JustFamilyDiagram**. תוכל להעתיק חלון אל תוך הלוח, ולהדביק אותו בחלון אחר. הואיל ונitin לאבטח חלונות של דיאגרמות מסדי נתונים, תוכל לשנות בתצוגות מסד הנתונים שיעמדו לרשות קבוצות שונות של משתמשים.

כפי שנitin נראה בתרשים 12.7, אפשר להוסף תוויות לדיאגרמות כדי לזהות את אחת הטעלות, או את כולן. כדי לשנות סגנון, גודל, צבע או מאפיין אחר של גופו בתוויות, לחץ לחיצה ימנית על תיבת תוויות כדי לפתוח את תיבת הדיו-שים **גופן** (Font).

תוכל לעורוך את רישימת החלונות של דיאגרמות מסד נתונים בתחום מסויים על ידי ספירת חברי האוסף **AllDatabaseDiagrams**. כמו כן התוכניות שלך תוכלנה לקרוא את המאפיין **IsLoaded** של חברי האוסף ולדוח אם חלון מסוים פתוח או לא. עם זאת, לא ניתן להגדיר את המאפיין **IsLoaded** כך שיפתח ויסגור חלונות של דיאגרמות מסד נתונים. האוסף **AllDatabaseDiagrams** שיק לאחד משני האובייקטים **CodeData** או **CurrentData** – בתוך היישום.



תרשים 12.7: שלושה חלונות של דיאגרמות עבור אותו מסד נתונים

השיגרה הבאה יוצרת את רשימת כל החלונות של דיאגרמות מסד נתונים, ומצינית ליד כל חלון אם הוא פתוח, או לא :

```

Sub listDatabaseDiagrams()
Dim obj As AccessObject, dbs As Object

' Set the hierarchical container for AllDatabaseDiagrams.
Set dbs = Application.CurrentData

' List database diagrams by whether they are loaded.
For Each obj In dbs.AllDatabaseDiagrams
    If obj.IsLoaded = True Then
        Debug.Print obj.Name; " is loaded."
    Else
        Debug.Print obj.Name & " is not loaded."
    End If
Next obj
End Sub

```

עריכת טבלה בחלון של דיאגרמת מסד נתונים

באפשרותך לעורך טבלאות או להוסף טבלאות חדשות דרך חלון של דיאגרמת מסד נתונים או באמצעות האוסף **אובייקטים (Objects)**, **טבלאות (Tables)** שבחלון **מסד נתונים (Database)**. לכל גישה יתרונות משלה לביצוע משימות שונות.

חלונות של דיאגרמות מסד נתונים שימושיים ביותר ליצירת משימות של הגדרת נתונים באופן גרפי או בתיבת הדו-שיח **מאפיינים (Properties)** של טבלה נתונים. כדי להקצות מפתח ראשי, בחר בבורר השורות של השדה בתוך דיאגרמת מסד הנתונים. אם המפתח הראשי מבוסס על יותר משדה אחד, לחץ לחיצה רצופה על מקש Ctrl ובחור את שאר השדות המשתתפים בהגדרת המפתח הראשי. להשלמת המשימה, לחץ לחיצה ימנית על השורה או השורות שנבחרו, ובחור באפשרות **מפתח ראשי (Primary Key)** (Primary Key) מתרפיט הקיצור.

חלונות של דיאגרמות מסד נתונים שימושיים גם להגדרת קשרי גומלין. התחל בבחירה בורר השורות עבור המפתח הראשי בטבלה הראשונה. אין צורך להשתמש במפתח ראשי, אולםעדיף לחביב להציג בשדה ערכים ייחודיים. כתע, גורר את השדה הנבחר מהטבלה הראשונה וחרר אותו בפס הכותרת של הטבלה השנייה. תיבת הדו-שיח **Create Relationship** (שדומה דמיון רב לחלונית הימנית העליונה בתרשים 12.6) תופיע על המסך. שנה את קשרי הגומלין כנדרש, ולאחר מכן על **OK**.

תוכל לכפות שלמות קשרים באמצעות תיבות הסימון שบทחתייה תיבת הדו-שיח **Properties** ובכרטיסייה **Create Relationship** שבຕיבת הדו-שיח **Enable Relationship For Insert And Update** כדי לכפות שלמות קשרים על כל הנתונים החדשניים במסד הנתונים הנוכחיים. אם שלמות הקשרים כבר חלה על הנתונים המקוריים, או אם אין צורך בכך, הימנע מבחירת תיבת הסימון **Check Existing Data On Creation**. אם בשל החלט שלמות קשרים במסד נתונים האמור לקבל שורות מממד הנתונים שלך, ההערכה של רשותם רבות תיכשל (בגיל הפרט כללי מפתחות), השאר את תיבת הסימון **Enable Relationship For Replication** ללא בחירה.

עריכה ויצירה של טבלאות באמצעות גיליון עבודה

תוכל להוסף למסד נתונים טבלה חדשה על ידי לחיצה ימנית בחלון של דיאגרמת מסד נתונים ובחירה באפשרות **טבלה חדשה (New Table)**. לאחר שקיבלת את השם שהפיקה המערכת או שהקצתית שם חדש, יופיע על המסך גיליון עבודה זה משמש למטען שמות לשדות, להקצתת סוג נתונים ולפירוט הגדרות נתונים נוספות. גיליון עבודה זהה נפתח מתוך האוסף **טבלאות (Tables)** שבחלון **מסד נתונים** כאשר לחצים על האפשרות **חדש (New)** או **עיצוב (Design)**. גיליון העבודה חושף מספר מאפייני טבלה ושדה ידידותיים למשתמש.

תרשים 12.8 מציג שני גליונות עבור הטבלאות MyLinks ו-FamilyMembers שהופיעו גם בתרשים 12.7. למעשה, קשרי הgomlin עודכנו בהשווה לתרשים 12.7, כך שמתאפשרים בין הטבלאות קשרי gomlin מסווג יחיד לרבים. השדה FamID בטבלה MyLinks קשור אל השדה הראשי FamID בטבלה FamID.FamilyMembers. תאפשר ליצור באמצעות מפתח זר שיחיה תואם לבדוק למפתח הראשי בטבלה אחרת, ממש כמו שנהוג בקובץ mdb *.accdb. העתק את המפתח הראשי מהטבלה הראשונה אל הלחט, והדבק אותו בטבלה השניה.

The screenshot displays two table designs in Microsoft SQL Server Management Studio:

- MyLinks : Table**

Column Name	Datatype	Length	Precision	Scale	Allow Nulls	Default Value	Identity	Identity Seed	Identity Increment	Is RowGuid
LinkID	int	4	10	0	<input type="checkbox"/>		<input checked="" type="checkbox"/>	1	1	<input type="checkbox"/>
MyLink	char	125	0	0	<input checked="" type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
FamID	int	4	10	0	<input checked="" type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
- FamilyMembers : Table**

Column Name	Datatype	Length	Precision	Scale	Allow Nulls	Default Value	Identity	Identity Seed	Identity Increment	Is RowGuid
FamID	int	4	10	0	<input type="checkbox"/>		<input checked="" type="checkbox"/>	1	1	<input type="checkbox"/>
Fname	char	20	0	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
Lname	char	25	0	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
Relation	char	30	0	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>

תרשים 12.8: שני גליונות עבור המראים את האפשרויות להגדרת שדות בטבלאות של MSDE נתונים או MSDE SQL Server

המפתח הראשי FamID בטבלה FamilyMembers מכיל נתונים מסוג Integer של SQL Server. סוג נתונים זה מקביל לסוג Long בMSDE נתונים של Jet או Access. המפתח הראשי LinkID בטבלה MyLink משתמש בסוג נתונים Integer עם הגדרת Identity. שים לב, שימוש זה, שלא כמו ב- Access, מכיל משקל גרפי לעמודה ערכיו זיהוי תחיליים וערכי צעד. סוג הנתונים varchar מיוחד לשדות להגדרת ערכי זיהוי תחיליים וערביים. סוג הנתונים makibil לנוטונים שאינם ב-UTF-8Unicode הוא varchar. גם במקורה זה, Access עם אינו חושף את הבדיקה בין סוגי הנתונים.

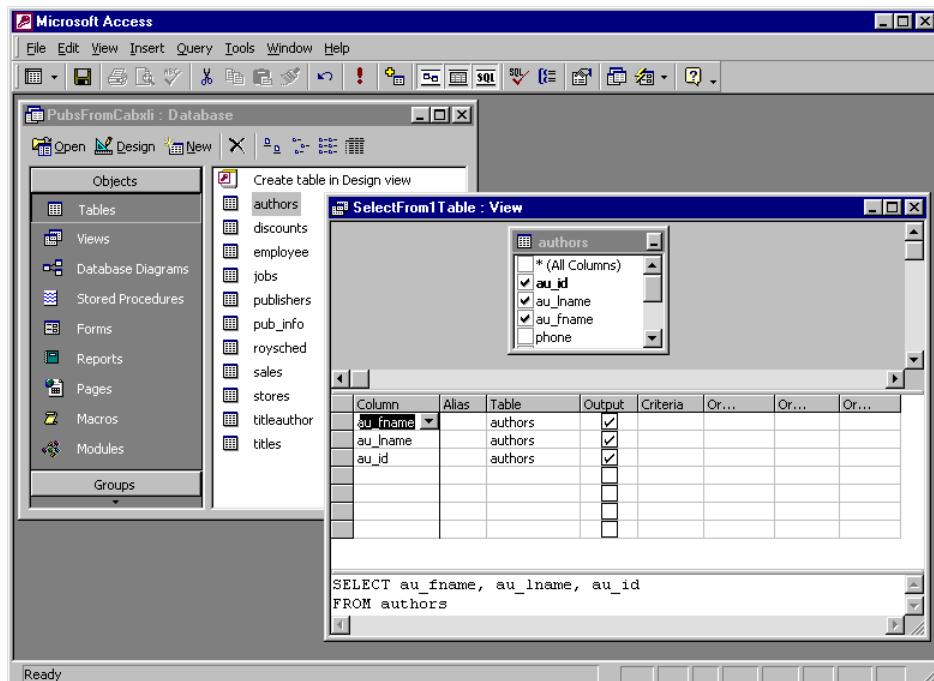
תצוגות וסדרות מאוחסנות

שלא כמו SQL Server, Access אינו מכיל שאילות לבניית יישומים מותאמים אישית, אולם יש בו רכיבים פונקציונליים דומים, המשגים באמצעות תצוגות וסדרות מאוחסנות. תצוגה היא משפט SQL המציג שורות מבלי להשתמש בפרמטרים או במשפט ORDER BY. לעומת זאת שירה מאוחסנת יכולה להשתמש הון בפרמטרים והן במשפט ORDER BY כדי לציין קבועות שורות להחזרה. סדרות מאוחסנות יכולות להוציא לפועל גם משפטים SQL שאינם מחזירים שורות, כמו INSERT, UPDATE ו- DELETE.

שימוש בתצוגות

כדי להקל עליך את הנטיגלות לשינוי כללי כתיבת השאלות במעבר מ-Jet ל-MSDE או ל-SQL Server, הסטייע במעכבי השאלות החדש של Access, הדומה דמיון רב למעכבי השאלות המבוססות על Jet. המשמש במעכבי השאלות לבניית תצוגות (את השגורות המאוחסנות תיאלץ לכתוב ללא עזרתו של אשף בניה). מעכבי השאלות בניו משלווה מסכים, שניתן להפעיל או לכבות. בניית התצוגה יכולה להיעשות בכל אחד משלושת המסכים, וכל ערך המוזן בלוח אחד מעודכן אוטומטית גם את שני הלוחות האחרים.

תרשים 12.9 מציג את מעכבי השאלות בבניה של שאלות בחירה המתאפיינת אל הטבלה authors שבבסיס הנתונים Pubs. המסקعلון הוא חלונית הדיאגרמה. לחץ לחיצה ימנית על חלונית זאת כדי להוציא כל טבלה או תצוגה קיימת כמקור קלט ל视窗. בחר את תיבת הסימון הסמוכה לשדה שאתה מבקש להוציא לחלונית הבחירה החדשה. בחר את תיבת הסימון בעמודות הקלט כדי לציין אם שדה מסוים נדרש שבמרכזו. השתמש בתיבות הבחירה בעמודות הקלט כדי לציין אם שדה מסוים-Amor להופיע בקבוצה המוחזרת של תצוגה נתונה. כמו כן תוכל להוציא קרייטריוונים בעמודות שמיינן כדי להגביל את מספר השורות שתצוגה יכולה להחזיר. אם אתה מדיף כתיבת SQL על פני הצבעה ולחיצת, השתמש בחלונית SQL (החלונית התחתונה) לכתיבת משפטי SQL.



תרשים 12.9: מעכבי השאלות

בסרגל הכלים **הציג עיצוב** (View Design) ובשורות התפריטים תמצא עזרה נוספת לכתיבת תצוגות. הלחוץ SQL מפעיל ומשבית את חלונית ה-SQL. לפי ברירת המחדל, הלחוץ שימושי ללחוץ SQL מבע פונקציה דומה עבור חלונית הרשות וחלונית הדיאגרמה. הלחוץ שמיין, המסומן בסימן ✓ (לחוץ **בדוק תחביר SQL** (Verify SQL)) מבצע בדיקת תחביר למשפטי SQL שאיתה מזין היישר אל תוך חלונית הרשות ה-SQL. הלחוץ **קיבוע לפי** (Group By) הסמוך ללחוץ **בדוק תחביר SQL** משנה את חלונית הרשות כדי להתחילה בעיצוב של שאלת צבירה. התנהגותו דומה לו של הלחוץ **Totals** (Σ) במבנה השאלות הקלاسي של Jet. לאחר שיעציבת את התצוגה כרצונך, לחץ על הלחוץ **אובייקט חדש** כדי לבצע אוטומציה של הנקנת טופס או דוח המבוססים על התצוגה שיעציבת (אם טרם שמרת את התצוגה, תתבקש לשמר אותה תחילה).

הערה:



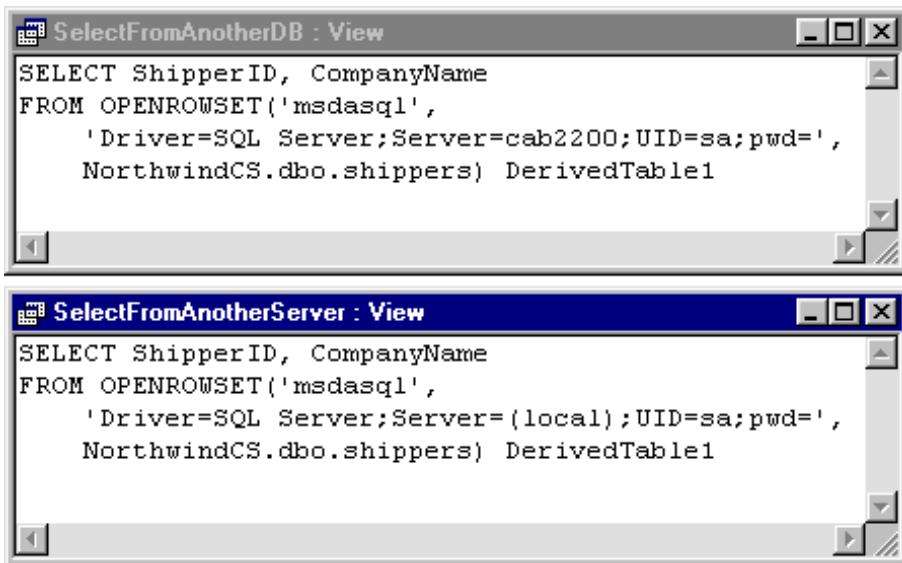
מעצב השאלות כולל את היתרונות של עיצוב שאלות גרפיות לבניית תצוגות ללא הטרחה הנוסף של האגדת טבלאות מרוחקות (כנדרש עם קבצי *.mdb, *.accdb). משפטי SQL עבור התצוגות מתבצעים בשורת מסד הנתונים, ואילו פרויקט Access רק מציג את הקבוצה המוחזרת.

אין מוגבל לבניית תצוגות השואבות נתונים מהחיבור הפעיל של פרויקט Access. באפשרותך להשתמש במילת המפתח OPENROWSET כדי לכוון תצוגה אל מסד נתונים אחר בשרתו, או אפילו בשרת אחר. דבר זה יפה גם לשרטים מסווגים שונים, כמו למשל Oracle או Jet. מילת המפתח OPENROWSET מקבלת שלושה ארגומנטים מופרדים זה מזה בפסיקים. הוסף גרש בתחילת ובסוף הארגומנט הראשון והשני כדי לציין שמדובר במחוזות. הארגומנט השלישי הוא שם של אובייקט מסד נתונים. הארגומנט הראשון מכיל את שם ספק OLE DB עבור מקור הנתונים החלופי.

בשתי הדוגמאות הבאות נעשה שימוש בספק OLE DB עבור מקורות נתונים תואמי-ODBC. הארגומנט השני מכיל את מחוזות החיבור. הספק קובע את התחביר הנכון לארגומנט זה. הארגומנט השלישי הוא אובייקט מסד נתונים בתבנית SQL Server. התבנית הכללית לצוין האובייקט היא linksServer.catalog.schema.object. linksServercatalog.schema.object. המאלה שולב SQL Server שמצויב אל מקור הנתונים ההטרוגני המרוחק. אם אין מוציאה לפועל שאלתה עבור מקור נתונים שאינו SQL Server, אין זכות לfrmater זה. הוא שם מסד הנתונים, ו-schema מצין את הבעלים של מסד הנתונים. Object הוא שם הטבלה.

תרשים 12.10 מראה ייצוגים של שתי תצוגות בחלונות SQL. שנייהם לקוחות מקובץ PubsFromCabxli.adp. לצורך סביבת הבדיקה, קובץ זה שוכן במחשב שבו CABXLI, אולם בתיבת הדיו-שיכון שלו Data Link Properties מצויין קשר עם מסד הנתונים Pubs שהוכן במחשב שבו cab2200.cab. כל תצוגה משתמשת במילת המפתח OPENROWSET כדי להתחבר אל מקור נתונים שאינו מסד הנתונים Pubs במחשב Shippers. התצוגה SelectFromAnotherDB מתחברת את הטבלה Shippers.cab2200.

שבסיד הנתונים NorthwindCS.cab2200, במחשב CABXLI. מוצב הפעלה Shippers בשרת המקומי – SQL Server. מוצב השאלות מושך SQL DerivedTable1 בסופם של שני משפטים SQL.



The image shows two separate windows of the Microsoft Query application. Both windows have a title bar labeled 'SelectFromAnotherDB : View' and 'SelectFromAnotherServer : View'. The top window contains the following T-SQL code:

```
SELECT ShipperID, CompanyName
FROM OPENROWSET('msdasql',
    'Driver=SQL Server;Server=cab2200;UID=sa;pwd=',
    NorthwindCS.dbo.shippers) DerivedTable1
```

The bottom window contains the same T-SQL code:

```
SELECT ShipperID, CompanyName
FROM OPENROWSET('msdasql',
    'Driver=SQL Server;Server=(local);UID=sa;pwd=',
    NorthwindCS.dbo.shippers) DerivedTable1
```

תרשים 12.10: תחביר SQL עבור מילת המפתח OPENROWSET, המשמשת ליצירת חיבור עם מקורות נתונים שמחוץ לחיבור הפעיל.

שימוש בשגרות מאוחסנות

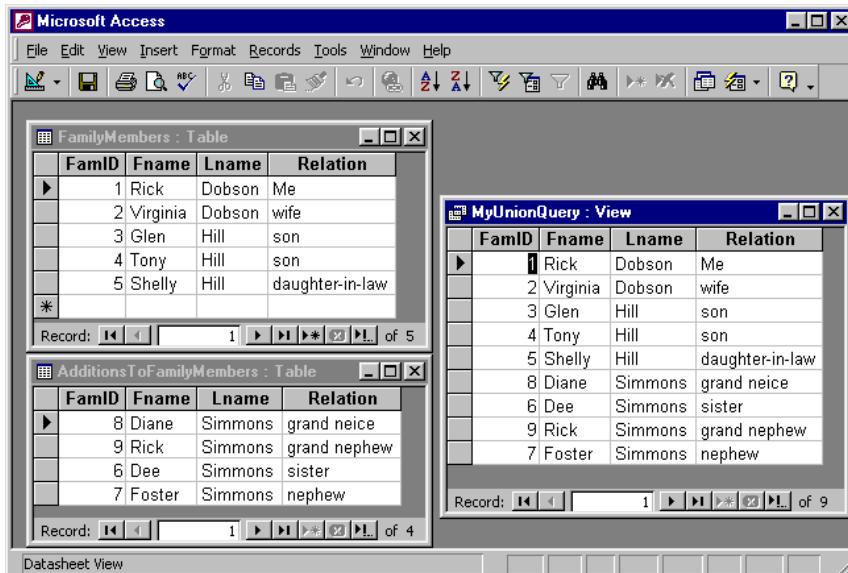
באפשרות להשתמש בשגרות מאוחסנות לבנייה נוספת והגדלת העוצמה של תצוגות ביישומי SQL Server מותאמים אישית. לא קיים מוצב שאילותות לשגרות מאוחסנות. טכניקת השגרות המאווחסנות מפיצה על חיסרונו זה על ידי שימוש באוצר מיילים משופר המבוסס על Transact-SQL, שהיא גירסה מיוחדת של שפת SQL עבור SQL Server. הגמישות של Transact-SQL בשילוב עם העוצמה של השגרות המאווחסנות שלה מעניקות לגירסה זו דירוג שני או שלישי כשפת תכנות מודעדפת, אחרי VBA ו- ADO, וזאת במיוחד אם בכונתך לפתח בעזרת פרויקטים של Access.

תרשים 12.11 מציג את התוצאה של שאילתת איחוד בפרויקט של Access. זכור, שאילתת איחוד משרשת קבוצות של רשומות בזו אחר זו. במקרה זה, הקוד (הMOVIA להלן) משרשר את הטבלה FamilyMembers עם הטבלה AdditionsToFamilyMembers. הקוד והנתונים לדוגמה עבור שאילתת האיחוד נמצאים בקובץ adp1.adp, בתיקטורי המצורף. הטבלה AdditionsToFamilyMembers מפנה אל מסד הנתונים adp1SQL.

```

SELECT FamID, Fname, Lname, Relation
FROM AdditionsToFamilyMembers
UNION
SELECT FamID, Fname, Lname, Relation
FROM FamilyMembers

```



תרשים 12.11: שאלת איחוד ב- SQL Server (וב- MSDE) יכולה לשרש שתי טבלאות, או יותר, מבליaina יכולה למיין אותן

הקוד נמצא בתצוגה ששםה MyUnionQuery. שים לב, שהשורות אינן ממוניות לפי השדה FamID מכיוון שאין ב- SQL משפט ORDER BY עבור TZוגה זאת. זכור שתצוגות ב- SQL Server אין מאפשרות שימוש במשפט ORDER BY. אם ברצונך למיין את הרשומות לפי FamID או לפי כל שדה אחר, عليك להפעיל לשם כך שיגרה מואחסנת.

הדוגמה הבאה מתאימה את ה- SQL לתצוגה ובונה שיגרה מאוחסנת גמישה יותר. במקרה זה, השגירה שמה StoredProcedure1 מבצעת את השרשור וממיינת לפי .FamID

```

ALTER PROCEDURE StoredProcedure1 AS
SELECT FamID, Fname, Lname, Relation
FROM AdditionsToFamilyMembers
UNION
SELECT FamID, Fname, Lname, Relation
FROM FamilyMembers
-- Here's the ORDER BY phrase
ORDER BY FamID

```

שםו של המשפט ALTER נוצר מיקולתו לשנות שירה שנוצרה לראשונה בעורת המשפט CREATE PROCEDURE. עם זאת, ניתן ליצור שירה מתחילה באמצעות ALTER PROCEDURE. שים לב, שהתחביר מחיב להוסיף את מילת המפתח AS לאחר המשפט ALTER PROCEDURE. יתרת השירה המואחסנת זהה לتزוגה MyUnionQuery, להוציא שתי השורות האחרונות. השורה הלאני-אחרונה מראה את התחביר להערה שנדרשת לה שורה אחת בלבד. התחל את השורה בשני תווי מקף (--). השורה الأخيرة מכילה את הרשומות לפי FamID.

כאשר משתמשים בפרויקטים של Access, רצוי לרכוש בקיאות בשגרות מאוחסנות מטיענים רבים. מגנוני MSDE ו-SQL Server תומכים בסדרה של פונקציות מינימליות במערכות שגורות מאוחסנות של המערכת. שגורות אלה שוכנות במסד הנתונים הראשי. כל שגורות המערכת הללו נפתחות ב- sp. מסיבה זאת, עליך להימנע מהשימוש בצירוף sp בשמות השגורות המואחסנות המותאמות שלך.

שירה מאוחסנת בשם sp_server_info משוב אודות כל סוגי הפונקציונליות בשרת SQL Server Books Online (SQL Server Books Online) מכיל מידע שירה זאת. תוכל להפעיל אותה ביצועם שלך כשירה מאוחסנת כדי לקובע למנגנון SQL Server הגדרות מאפייניות שיסייעו לך להחליט כיצד לתוכנת אותו. לדוגמה, השורה ה-18 של הקבוצה מוחזרת רגילה עשויה לציין אם השרת ימיין תוך התחשבות באוטיות רישיות, אם לאו. הדוגמה הבאה בונה ביצוע שלך כשירה מאוחסנת פשוטה המפעילה את שרת המערכת המואחסנת sp_server_info כדי לבדוק את כללי האיסוף בשרת שאתה מחובר אליו כרגע. הדוגמה מראה גם את התכנית להערה המשתרעת על פני יותר מושואה אחת.

```
ALTER PROCEDURE DetermineSortOrder
/*If row 18 shows sort_order - nocase,
then sorts are case insensitive.
*/
AS
EXEC sp_server_info 18
```

סיבה נוספת להרחבת הידע אודות שגורות מאוחסנות קשורה בኒזול הפרמטרים שלhn. תוכל להעביר אליהן פרמטרים כדי לקבל ערכי החזקה באמצעות פרמטר הפלט. השירה DetermineSortOrder שולפת את השורה ה-18 מתוך הקבוצה מוחזרת שהפיקה השירה המואחסנת sp_server_info. המספר 18 הוא פרמטר המציין לשגרת המערכת איזו שורה עליה להציג מתוך הקבוצה המוחזרת הרגילה שלה.

שתי השגורות המותאמות אישית הבות מעכבות שגורות מאוחסנות שמקובלות פרמטרים, ומעבירות אליהן ערכים באמצעות קוד. שגורות מאוחסנות מותאמות אישית עם פרמטרים יכולות להציג באופן אוטומטי דרישת ערכים אם איןך מעביר להן ערכים באמצעות קוד. השירה הראשונה מבצעת שתי פעולות: היא מפעילה את השירה שמייקה דוח מכירות, ולאחר מכן מעבירה ערכים אל השירה השנייה, שקובעת לדוח תאריכי תחילת וסיום, וכן שם של עובד. תוכל לעורך את שלושת הפרמטרים בשירה הראשונה כדי לשנות את הפלט של השירה השנייה.

```
ALTER PROCEDURE RunEmployeeSalesByDate
AS
EXEC "Employee Sales by Date" '1/1/95', '1/1/98', 'Buchanan'
```

הSIGRAה השניה, לעיל, מקבלת את הפרמטרים כחלק מהמשפט ALTER PROCEDURE. כל ציון פרמטר מכיל שני ארגומנטים: השם, כמו למשל @Beginning_Date, וכן סוג נתונים, כמו varchar(20) לדוגמה datetime או. עלי ציון סוג הנתונים המתייחס לשם של עובד בשירה המאוחסנת להתקאים להגדרת סוג הנתונים עבור שם העובד בטבלה Employees. משפט SQL לחבר שלו טבלאות, ופסוקית WHERE מפעילה את הפרמטרים.

```
/*This procedure accepts three parameters to determine the
content for a sales report.

@Beginning_Date is the start date period.
@Ending_Date is the end period.
@lastname specifies the employee for which
the report lists sales.

*/
ALTER PROCEDURE [Employee Sales by Date]
@Beginning_Date datetime, @Ending_Date datetime,
@lastname varchar(20)
AS
SELECT Employees.Country, Employees.LastName, Employees.FirstName,
       Orders.ShippedDate, Orders.OrderID,
       "Order Subtotals".Subtotal AS SaleAmount
FROM Employees INNER JOIN (Orders INNER JOIN "Order Subtotals" ON
                           Orders.OrderID = "Order Subtotals".OrderID)
                           ON Employees.EmployeeID = Orders.EmployeeID
-- This format is for comments on a single line.
WHERE Orders.ShippedDate Between @Beginning_Date And @Ending_Date And
      Employees.LastName = @lastname
```

דוחות וטפסים

לאחר שהתחברת אל מקור נתונים מרוחק אחד, או יותר, ולאחר שביצעת סינון, צבירה או שילוב בין מקורות אחרים, תוכל להציג באמצעות Access 2000 את התוצאות שקיבלת. הוויל וקיים שילוב הדוק בין טבלאות, תצוגות ושורות מאוחסנות בין טפסים, דוחות ומודולים, תוכל להגיש בנקל נתונים שרת-לקוח בדרך המקובלת ב-Access להגשת נתונים שרת-קובץ מאז המהדורה הראשונה. מהזרות קודמות של Access אמנים אפשרו גישה לנכוני שרת-לקוח, אולם עיבוד שרת-לקוח של ממש מעולם לא היו קלים כבמהדורה זאת. האמנים יכולים קבצי adp.* לשרת מקורות

נתונים של שרת-לקוט באוֹתָה דְּרַךְ שֶׁקְבַּצִּי mdb.*. שִׁירְטוּ מִקְוּרוֹת נְטוּנִים שֶׁרְתַּ-קּוֹבֵץ? התשׁוֹבָה לְשָׁאַלָּה זוֹ אֲתָא תְּלוּיָה בְּמִזְדַּת הָעֲנֵין וּבְגָבוֹלּוֹת הַדְּמִינוֹ שֶׁלְךָ. הַדּוֹגְמָאוֹת הַבָּאוֹת עֲשָׂוִות לְהַצִּית בָּךְ אֶת נִיצּוֹן הַיְצִירָתִיָּת.

מיון ועיצוב באמצעות דוחות, ועוד

בעזרת מעצב השאלות, קל למדי לسان, לצרף ולכבר נטוּנים, אולם לא ניתן לעשות בו הרבה במונחים של מצגות. צורך שלא ניתן אפילו למיין את הרשומות בתוך הצוגה. לעומת זאת בדוחות, כשר העיבוד מוגבל למדוי, אולם אפשרויות המיוון והעיצוב של נתוניים לדפסות מצוינות (ואפילו ל-Web, אם מביאים בחשבון תМОנות – ראה פרק 6).

תרשים 12.12 מראה כיצד ניתן להשתמש בתוכנות של פרויקט Access עם דוחות כך שיישלימו זה את זה. הדוח ממיין רשומות לפי FamID. כמו כן הוא מעצב באופן מותנה את הצבע לתצוגת שורה לפי הערך של FamID. ערכאים נמכרים מ-6 מופיעים בשחור, וערכים השווים או גדולים מ-6 מופיעים באדום. בנוסף לכך, התצוגה מראה את האבזרים המקבילים של דוח, לרבות כתורתה, קו מפריד בין כתורת העמודות לבין הערכאים בעמודות, וכן עיצוב לכותרות עמודות וכותרת הדוח, המבדיל בין לבן העיצוב של גוף הדוח.

FamID	Fname	Lname	Relation
1	Rick	Dobson	Me
2	Virginia	Dobson	wife
3	Glen	Hill	son
4	Tony	Hill	son
5	Shelly	Hill	daughter-in-law
6	Dee	Simmons	sister
7	Foster	Simmons	nep hew
8	Diane	Simmons	grand neice
9	Rick	Simmons	grand nephew

תרשים 12.12: דוח המבוסס על התצוגה MyUnionQuery המופיעה בתרשים 12.11

מלבד הקוד לשאיילת האיחוד עבור התצוגה, לא נדרש תכנות נוספת בדוח זה. לאחר שבנית את התצוגה, הצב את שם התצוגה בהגדרת המאפיין Record Source של הדוח. להוציאו הצעים השונים המבוססים על ערכיו FamID, שאר תכונות הדוח מקבלות את הגדרות ברירת המחדל לדוחות טבלאים. הפוקודה החדשה **עיצוב מותנה** (Conditional Formatting) בתפריט **עיצוב** (Format) מפשטת באורה דרמטי את משימת הchallenge המותנית של צבעים על תכולתן של תיבות טקסט (ראה פרטם בפרק 5). אפשרו אלה מחייבות שימוש בשגרות אירוע Format. על ידי שילוב התצוגה והדוח תוכל השתמש בכל אחד מהם לביצוע המשימה שהוא מותאם לה ביותר. הואיל ודיוח הוא אחד מה יתרונות הבולטים של Access, יכולתו של קובץ adp.* לעבד מקרים נטוניים שרת-ליך יכול לצמצם באופן דרמטי את העלות של הפקה והפצת דוחות אינפורטטיביים וקלים לקריאה בכל רחבי הארגון.

הוסף היפר-קישורים

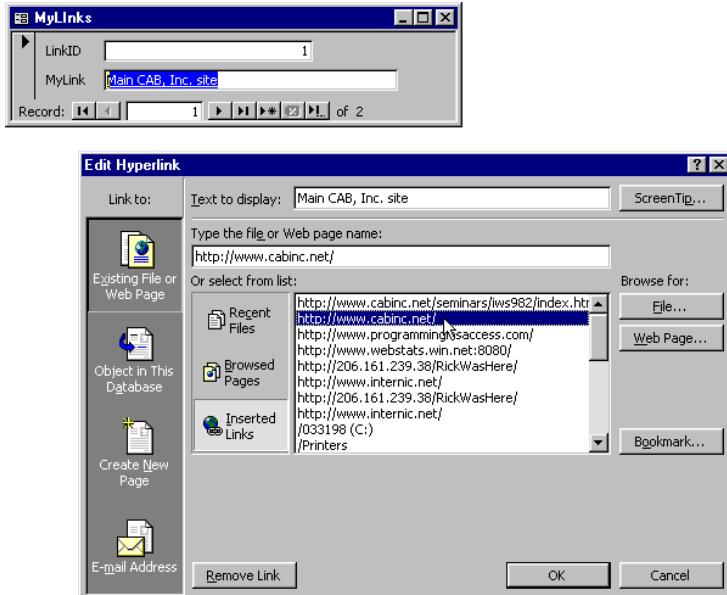
במסדי נתונים של SQL לא קיימים נתונים מסווג **היפר- קישור** (Hyperlink) (בזכותו, Jet תומך בסוג נתונים זה כחרחבה של סוג הנתונים **תזכיר** (Memo) – ראה פרק 1). עם זאת, עדין תוכל להוסיף ולעקוב אחר היפר-קישורים בתוך טופסי פרויקטים של Access. לשם כך عليك לבצע ארבע פעולות :

1. הקצה לעמודה בטבלה אחד מסוגי הנתונים הבאים : nvarchar ,char ,nchar . אלה הם שמות של סוגי נתונים קבועים ומחרוזות בעלות אורך משתנה, ב- Unicode או בתבנית שאינה .
- 2.פתח טופס בתצוגת **עיצוב** (Design) וփוך את הטבלה למקור הרשומות של הטופס .
3. הצב את השדה בעל סוג הנתונים מחרוזת באחד משדות הטופס .
4. הגדר למאפיין **הו היפר- קישור** (IsHyperlink) של השדה ערך כן (Yes). זהו מאפיין חדש ב- Access 2000, המיעוד במפורש להיפר-קישורים בטופסי פרויקטים של Access .

לאחר יצירת את הטבלה והטופס, תוכל להוסיף לטבלה היפר-קישורים באמצעות הטופס, בעזרת הפוקודה **הוסף** (Insert), **היפר- קישור** (Hyperlink). פוקודה זאת פותחת תיבת דו-שיח להגדרת או ערכת ה- URL של היפר- קישור ואת הטקסט המוצג שלו. לאחר שההיפר-קישורים עוצבו, יכול המשתמש לעקוב אחריהם עד לאתרי Web המתאימים על ידי לחיצה על הערך שבתוכו שדה היפר- קישור. דפדף בሪית המחדל יוביל אותו לאתר הנכון. לאחר מכן יוכל המשתמש לחזור ל- Access על ידי לחיצה על החץ Back (הקודם). פונקציונליות זאת התווספה ב- Access 97 ; בעת היא קיימת גם עבור מסדי נתונים MSDE ו- SQL Server 2000 במאיצעות Access .

תרשים 12.13 מציג את מסך MyLinks ב- Access, תוך כדי תהליך ערכת היפר- קישור. תוכל לדפס אל אתר ה- Web הרצוי כך שלא תיאlez להקליד את ה- URL. נוסף לכך,

תוכל להשתמש בתיבת הטקסט **tekst Shirley** (Text To Display) כדי להזין את הטקסט שיופיע במקומות ה-URL. שים לב, שניתן להשתמש בתיבת קבצים מוקומי, ואפשרו להתחבר לכתובות דואר אלקטרוני. האפשרות השניה מפעילה את תוכנת ברירת המחדל לדואר אלקטרוני בתחנת עבודה של דפדפן, עם השם וכתובת הדואר האלקטרוני שציינית בקישור.



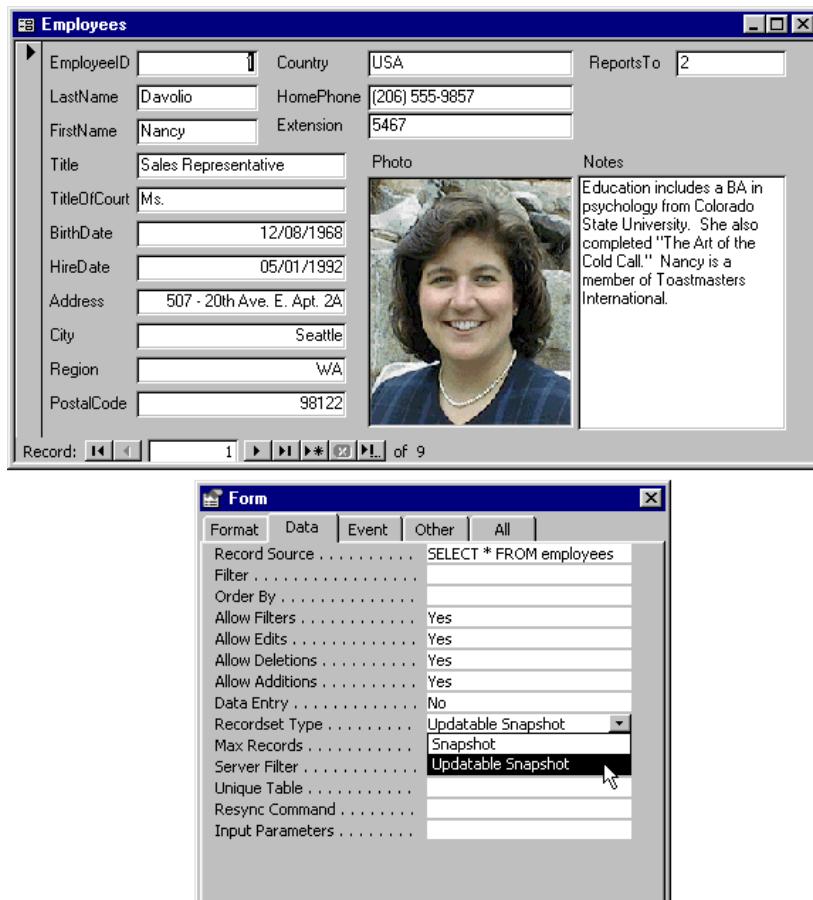
תרשים 12.13: השימוש בתיבת הדו-שיח **עריכת היפר- קישור** (Edit Hyperlink) כדי להזין ולערוך היפר-קישורים בסיד נטונים משותף לשורת-לקווח של היפר-קישורים.

עריכה והציגה של נתונים באמצעות טפסים

בפרויקטים של Access יש שתי הגדרות אפשריות למאפיין **סוג ערכת רשות** (Recordset Type) לצורכי עבודה עם טפסים ומוקורות נתונים מוקובלים, כמו למשל טבלאות של עובדים או לקוחות. הגדרות אלה ייחודיות לקבצי *.mdb ו-*.adp. הריגילים הגדרה שונה במאפיין **סוג ערכת רשות**.

כאשר אתה מעצב יישום משתמש בטפסים עם נתונים ממסד נתונים SQL או MSDE, יישום הלקוח תמיד עובד עם תמונה של הנתונים המקוריים בשורת. למרות שהנתונים המקומיים הם תמונה של הנתונים המקוריים בשורת, ניתן לעדכן אותם בשורת. כדי לאפשר פונקציונליות מסווג זה, קיימות ב- Access 2000 ההגדרות **תמונה ניתנת לעדכון** (Updateable Snapshot) ו**תמונה** (Snapshot) עבור המאפיין **סוג ערכת רשות** בפרויקטים של Access. תוכל לקבוע הגדרות אלה בכרטיסיה נתונים (Data) שבתיבת הדו-שיח **מאפיינים** (Properties) של הטופס. כמו כן תוכל לטפל במאפיין **סוג ערכת רשות** באמצעות VBA או מאקרו של Access.

כאשר אתה מציב במאפיין **סוג ערכת רשומות** של טופס את ההגדה **תמונה ניתנת** לעדכון, המשמשו יכול לשנות את מקור הנתונים שביבוסדו של הטופס כאילו אותו קובץ נמצא בשרת קבצים מקומי. בעזרת המאפיין Lock של פקדים נפרדים, תוכל לאפשר עריכה ברונית של קבוצת משנה של פקדים בטופס. נוסף לכך, תוכל לשלוט בסוג השינויים המותרים לביצוע בוחנות עבודה של לקוחות. ישנו הגדרות כנ"ל נפרדות לאפשרויות **אפשר עריכה** (Allow Edits), **אפשר מחיקות** (Allow Deletions) ו**אפשר תוספות** (Allow Additions). כל שינוי שתערך מתוך קובץ של פרויקט Access יופץ אל השרת. לכל משתמש בפרויקט צריך להיות עותק נפרד של קובץ ה-*.adp, למרות שככל משתמש פונה אליו מSED נתונים שרת-לקות. למרות שינויים מופצים מתחנות העבודה של לקוחות אל מSED הנתונים של השרת, כדי לצפות בתוצאות מופצות מתחנות העבודה של לקוחות אל מSED הנתונים של השרת, כדי לצפות בשינויים שערכו אחרים על המשתמש בחזור באפשרות **רענן** (Refresh) מתפרק רשותם (Records) כדי לראות שינויים שנערכו בידי משתמשים אחרים.



תרשים 12.14: השימוש בטפסים במסגרת פרויקטים של Access כדי להכניס שינויים בתוצאות

תרשים 12.14 מציג דוגמת טופס בפרויקט Access. הטופס מבוסס על נתונים משרת מרוחק. מתחת לטופס מוצגת קטע מגילון המאפיינים של הטופס, החושף את שתי הגדירות האפשריות למאפיין **סוג ערכות רשות** (Recordset Type). (Recordset Type). הוויל והגדרה שבתוקף היא **תמונה ניתנת לעדכון** (Updateable Snapshot) (Updatable Snapshot), המשמש יכול להחליף את מקור הנתונים שביבטו של הטופס. למעשה, בגין הגדרות הכון לאפשרויות אפשר **עריכה**, **אפשר מחיקות** ו**אפשר תוספות**, הוא יכול לבצע את השינויים בשלושת הסוגים הללו בסיס נתונים. באפשרות למנגנון שימוש באפשרויות אלה על ידי הצבת ההגדרה **תמונה** (Snapshot) (במאפיין **סוג ערכות רשות**).

الطائف הוכן בעזרת אשף הטעסים האוטומטיים (בתוספת עERICA מיינימלית). המעבר לשומה הבאה גורם להעברת כל פעולה עERICA שהבצע המשתמש בשודות בהזורה לשורת. אם הגדרת המאפיין **סוג ערכות רשות** היא **תמונה**, מגיב בהודעה שקבוצת הרשות אינה ניתנת לעדכון. הלחוץ(msomn) ב-X מאפשר למשתמש לבטל פעולה טעונה ארוכה של רשותות רבות אל תחנת העבודה מקומית.

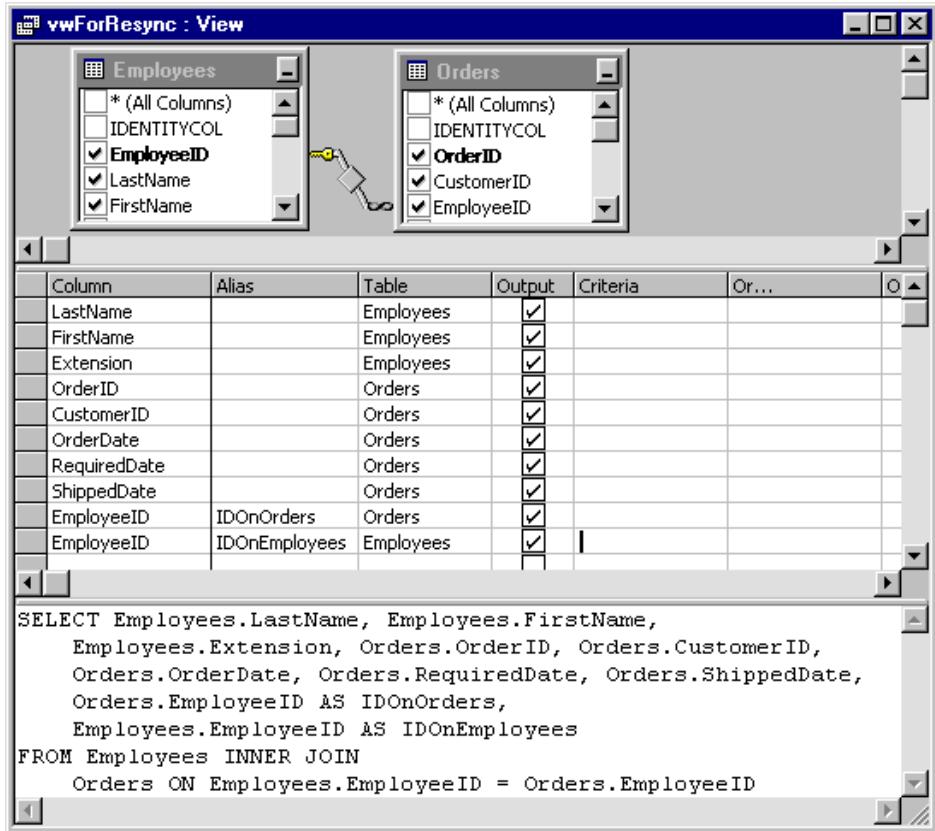
סינכרון חדש של טופס באמצעות נתוני יחיד לרבים

כאשר המקור של טופס בפרויקט Access מבוסס על מערכת רשומות שמקיימים בה קשר גומלי של יחיד לרבים בין שתי טבלאות, ניתן לאכלס באופן אוטומטי את כל השודות בצד היחיד על ידי הגדרת המפתח הזור בצד הרבים. תוכונה זאת שימושית במיוחד כאשר מוסיפים רשומות חדשות. לדוגמה, אם עתה עוסק בעיצוב טופס המייצג נתונים עובד ומכירות, היחס היחיד לרבים נגור מהעובדת לכל עובד יכול לבצע פעולות מכירה רבות. עלייך רק להזין את זההו העובד עבור טבלת המכירות, ולהשלים את הרשות. פועלה זאת תגרום לאכלוס אוטומטי של כל שודות העובד בצד היחיד של מקור הרשותות שברקע. לאחר מכן להזין את נתונים המכירות בצד הרבים של מקור הרשותות. באותה קלות תוכל לבדוק את נתונים העובד גם ברשותה קיימת.

כדי להפוך תוכונה זאת לזמיןה, יש להגדיר את המאפיין **סוג ערכות רשות** בתוור **תמונה ניתנת לעדכון**. הצב את שם הטבלה בצד הרבים של קשר הגומלי בתוור המאפיין **טבלה ייחודית** (Unique Table). כמו כן עליך להציב כהגדרת המאפיין **פקודת Resync** (Resync Command) מחרוזת המייצגת משפט SQL שמאפשר את הסינכרון מחדש. להלן נציג תחבורת ליצירת משפט זה בשיטה מקוצרת.

תרשים 12.15 מראה את תצוגת העיצוב של יחיד לרבים עבור התצוגהvwForResync.

החולונית העליונה מראה את קשר הגומלי היחיד לרבים בצורת קו המחבר בין הטבלאות Orders ו-Employees. שים לב שתצוגה זאת מקשרת בין הפעולות באמצעות המפתח הראשי בטבלה Employees ובузורת המפתח הזור המקביל (EmployeeID) בטבלה Orders. החולונית התחתונה מציגה את קוד SQL של התצוגה. המאפיין **פקודת Resync** שלالطائف מחייב ערכת שינוי פשוט במשפט זה.



תרשים 12.15: תצוגה זו מייצגת קשר גומלין של יחיד לרבים כמו זה המשמש לשינכרון חדש של טופס, כאשר משפט SQL הוא הבסיס להגדרת המאפיין **פקודת Resync**

תרשים 12.16 שלහן מציג טופס מייד לאחר הזנת ערך 9 לATABט הטקסט EmployeeID. מדובר בטופס אוטומטי גגיל (AutoForm), עם שינויי מזעריים. לחיצה על בורר הרשומות (record selector) מעלה נתוניים אל תוך כל תוך כל בורר הרשומות העובדים בטופס. הזנת ערך חדש בתיבת הטקסט EmployeeID ולחיצה על בורר הרשומות תעליה באופן אוטומטי נתונים עובדים המתאימים לערך ID החדש. זהו יתרונה של פונקציית השינכרון מחדש.

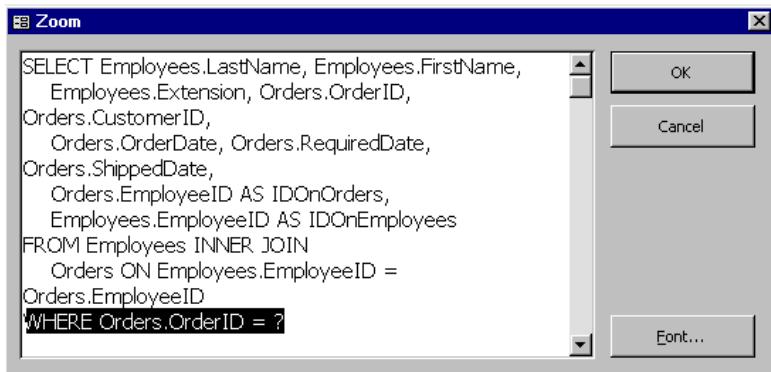
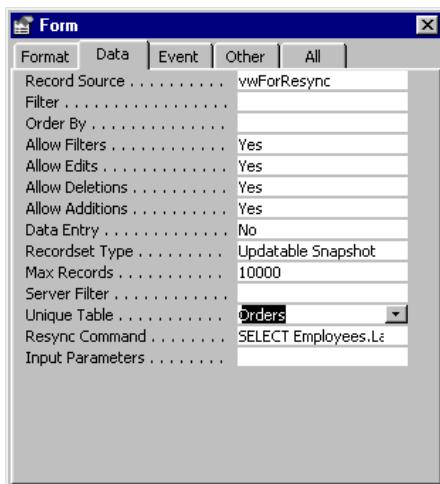
תרשים 12.17 מציג את הגדרות הטופס החופכות את פעולת השינכרון האוטומטית לויינה. הגדרת המאפיין **סוג ערבת רשומות** (Recordset Type) של הטופס היא **תמונה ניתנת לעדכון** (Updateable Snapshot), והמאפיין **טבלה ייחודה** (Unique Table) שלו מצביע אל צד הרבים של מערכת הרשומות שברקע, כלומר, הטבלה Orders. ההגדרה **פקודת Resync** (Resync Command) המוגדרת זהה בדיק SQL לקוד SQL של הטופס שבבסיס, בתוספת שורה אחת: `WHERE Orders.OrderID=?`. זהו השלב הסופי המשלים את תהליך הפיכתו של טופס לזמן בעזרת פונקציית השינכרון מחדש.

vwForResync

EmployeeID	9
LastName	Dodsworth
FirstName	Anne
Extension	452
OrderID	11083
CustomerID	
OrderDate	
RequiredDate	
ShippedDate	

Record: [Navigation Buttons] 832 [Navigation Buttons] of 832

תרשים 12.16: טופס המציג סינכרון מחדש מול מסד נתונים מרוחק



תרשים 12.17: הכרטיסיה נתונים (Data) ב吉利ון המאפיינים של הטופס; ההגדרות
תמונה ניתנת לעדכון ופקודת Resync נחוצות לסינכרון מחדש

סוגיות של תכנות

אחד השיטות להרחבת היכולות של פרויקט Access היא להיעזר ב-SQL. שיטה זאת הולמת במיוחד כשמדבר בתצוגות ובסגרות מאוחסנות. כמו מהדוגמאות הקודומות כבר ראית מה ניתן לעשות בעזרת SQL, אולם תוכל גם את ב- VBA וב- ADO. שאר הדוגמאות בפרק זה עוסקות ביישום הידע זהה בבדיקה עם פרויקטים של Access, ואפילו בבדיקה עם מסדי נתונים SQL Server ו- MSDE ללא משך המשמש הנוח של פרויקטים של Access.

עבודה עם טפסים

למרות הפונקציונליות יוצאת הדופן של משך המשמש בפרויקטים של Access מול מקורות נתונים מרוחקים, ניתן לבצע אוטומציה ופישוט של תהליכין על ידי תכונות של פיתוח פתרונות מותאמים אישית. אנו נבחן שלושה תחומים הנוגעים לפרויקטים של Access: פתיחת טופס, איתור רשומה והציג שינויים שביצעו משתמשים אחרים.

פתיחה טופס

כאשר פותחים טופס בעזרת משך הפרויקטים של Access, הטופס מכיל עותק מקומי של הנתונים המרוחקים בתחנת הלכה. העותק המקומי הוא תמונה, נconaה למועד זמן מסוימת, של הטופס עם הנתונים המרוחקים. כאשר פותחים טופס באמצעות קוד, יש ליזור את העותק המקומי של הנתונים המרוחקים. אחד היתרונות בפתיחת טופס באמצעות קוד היא שניתן להקטות ערכיהם באופן דינמי למטרון המקומי המכיל את הנתונים המרוחקים. היחסים שלך יכול לעשות זאת מכיוון שרכת הרשומות שהקצתה לטופס בעזרת VBA גוברת על ההגדלה **מקור רשומה** (Record Source) שבגילוון המאפיינים של הטופס.

השיגרה הבאה בונה מערכת רשומות עבור טופס לפני פתיחתו. תחילתה היא מדירה הפניה למופיע חדש של מערכת רשומות: היא מציבה את ההגדלה adUseClient במאפיין CursorLocation (מיקום הסמן) של מערכת הרשומות כדי לקבוע את מיקום הנתונים בטופס (זכור, הטופס מקבל את הנתונים מהמטרון המקומי שבתחנת העבודה, ולא מהשרת המרוחק). כתע היא פותחת את מערכת הרשומות בעזרת משפט SQL המחלץ נתונים מהמקור המרוחק ומצביע אותם בעותק המקומי. הדוגמה משתמש בטופס המופיע בתרשים 12.14 לעיל. שורת הערא מראה משפט SQL המسلح לדרישת בירית המחדל של מקור הרשומה של הטופס. לאחר יצירת העותק המקומי של הנתונים המרוחקים, השיגרה פותחת את הטופס ומציבת את העותק המקומי כהגדלה המאפיין Recordset (מערכת רשומות) של הטופס. מאפיין חדש זה מכיל את הפונקציונליות של המאפיין RecordsetClone; בנוסף, שינויים שנעשו בערכת הרשומות יופיעו באופן אוטומטי בטופס – המאפיין RecordsetClone יוצר עותק לקרה-בלבד של מערכת הרשומות של הטופס. בדומה למאפיין RecordsetClone, המאפיין Recordset של הטופס זמין רק באמצעות הקוד.

```

Sub openForm()
Dim rst1 As ADODB.Recordset
' Establish a local recordset and populate it with values;
' can override property sheet settings.
    Set rst1 = New ADODB.Recordset
    rst1.CursorLocation = adUseClient
    rst1.Open "SELECT * FROM employees", _
        CurrentProject.Connection, adKeySet, _
        adLockPessimistic
' Optionally run with WHERE clause to demo override effect.
'   "SELECT * FROM employees WHERE employeeid>3"
' Open the form.
    DoCmd.openForm "frmemployees2"
' Assign recordset to the open form; can override a setting on the property sheet.
    Set Forms("frmemployees2").Recordset = rst1
End Sub

```

הערה:



ערכת הרשומות שנוצרת עבור מופע של ערכת רשומות תישאר זמינה כל עוד הטופס פתוח. סגירה ופתיחה מחדש ידנית של הטופס תגרום לטופס לחזור להגדרת **מקור רשומה** (Record Source) שבגלאיון המאפיינים שלו.

איתור רשומה

בעובדה עם נתונים בטופס, אחת המשימות השכיחות היא איתור רשומה מסוימת. שתי השגרות הבאות מבצעות משימה זאת. השגרה locateEmployee מנהה את השימוש להזין ערך של זיהוי עובד, ומעבירה ערך זה אל השגרה השניה,.findByID. השגרה השנייה מפעילה את השיטה FindRecord של האובייקט DoCmd כדי לאתר את הרשומה ולמקם את הטופס ברשומה החדשה.

```

Sub locateEmployee()
' Ask for employee ID and pass it on to findByID.
    employeeNumber = InputBox("Type the ID for the employee you want", _
        "Programming Microsoft Access 2000")
    findByID CLng(employeeNumber)
End Sub

Sub findById(eid As Long)
On Error GoTo findByIdTrap
' Set focus to employee ID field and launch find.
    Forms("frmemployees2").EmployeeID.SetFocus
    DoCmd.FindRecord eid

```

```

findByIdExit:
' Report mismatch before exiting.
If Forms("frmemployees2").EmployeeID <> eid Then
    MsgBox "No employee with ID " & eid & ".", _
        vbExclamation, "Programming Microsoft Access 2000"
End If
Exit Sub

findByIdTrap:
If Err.Number = 2450 Then
' Open form if it is closed and start find again.
    openForm
    Resume
Else
    Debug.Print Err.Number, Err.Description
End If
End Sub

```

השיגרה השנייה מזזה שתי בעיות אפשריות. ראשית, אם לא נמצאת רשומה תואמת, השיגרה מציגה הודעה ברוח זאת. שנית, אף אחת משתי השגרות אינה שגרת אירוע, לכן ניתן להפעיל גם מחוץ לטופס. אם הטופס סגור, השיגרה מזזה את הבעה ופותחת את הטופס כדי לנסות פעם נוספת לרשום את רשותם היעד.

הציג שינויים שביצעו משתמשים אחרים

Access הוא ביסודות סביבת פיתוח מרובה-משתמשים, ופרויקטים של Access מօפעלים תכופות כדי לשרת מטרות מרובות-משתמשים. לכן דרושה לך שיטה לרענון המטמון המקומיי, כדי שתוכל להציג עדכונים, תוספות ומחיקות שבוצעו בידי אחרים. השיגרה הבאה מבצעת משימה זאת על ידי אייכלוס חדש של ערכת הרשומות המקומית של הטופס בתנאיו מקור הנתונים המרווח. בדומה לדוגמאות הקודמות, דוגמה זאת מבוססת על הטופס המופיע בתרשים 12.14.

```

Sub requeryRemoteRestoreID()
Dim int1 As Integer
' Turn off screen updates and save employee ID.
    DoCmd.Echo False
    int1 = Forms("frmemployees2").EmployeeID
' Requery local recordset from the server.
    openForm
' Reposition to employee ID before requery and
' restore screen updating.
    Forms("frmemployees2").EmployeeID.SetFocus
    DoCmd.FindRecord int1
    DoCmd.Echo True
End Sub

```

השיגרה מאכלסת מחדש את מטמון הנתונים המקומי על ידי הפעלת השיגרה openForm (שעסקנו בה קודם). אם הטופס פתוח, הקריאה רק מאכלסת מחדש את המטמון. הויל וaiclus מחדש של ערכות הרשותות של טופס מקם באופן אוטומטי את הטופס ברשותה הראשונה, השיגרה שומרת את מיקום הרשותה הנוכחי לפני שהיא מאכלסת מחדש את המטמון המקורי בנתוני מהן הרשותה המרווח. לאחר מכן מאכלסת מחדש את המטמון מחדש מחפש בין הרשותות כדי לאתר את הרשותה המקורי המרווח. פועלה זאת משוחררת את מיקום הרשותה הישנה, אם היא עדיין זמינה (כלומר, אם לא נמחקה). אם משתמש אחר מחק את הרשותה שהיתה הרשותה הנוכחייה הקודמת. השיגרה יציג את הרשותה הראשונה במטמון המקומי.

עבודה עם מודולים עצמאיים

ההדגשה החזרת ונשנית על DAO לאורך ספר זה תשרת אותך היטב כאשר תעסק בפיתוח פתרונות מותאמים אישית מול מקורות נתונים MSDE ו-SQL Server. פרק 4 מכיל מספר דוגמאות לטיפול במקרים נתונים מרוחקים. בסעיף זה נבחן מחדש סוגיות תכנות מסוימות בהקשר עם פרויקטים של Access ו-MSDE.

פתיחה טבלה

הדוגמה הבאה משלבת בין פתיחת טבלה במקור נתונים מרוחק לבין כתיבת התוצאות בחלון Immediate (מיידי), המשמש לצורך העניין כמדריך טלפוניים. לשם כך הדוגמה משתמשת באובייקטים Connection (חיבור) ו- Recordset (ערכת רשומות) של DAO. השיגרה פותחת ביצירת מוףע חדש של האובייקט Connection של DAO. לאחר מכן מגדרה מחרוזת חיבור ומשתמשת בה כדי ליצור חיבור עם מסד הנתונים NorthwindCS בשרת cab2200. בשלב הבא היא פותחת ערכת רשומות קדימה-בלבד לкриאה- בלבד, המבוססת על הטבלה Employees במסד הנתונים NorthwindCS. ערכת רשומות מסווג זה קבילה בדוח המדף פעמי אחת ויחידה ברשותה של ערכת רשומות. השיגרה מדפסה את מדריך הטלפוניים של עובדים בחלון Immediate בעוזרת LOLAAT Do כדי לעבור על פני הרשותות העוקבות, לפי הסדר.

```
Sub openTableOnRemoteServer()
Dim cnn1 As ADODB.Connection
Dim rst1 As ADODB.Recordset
' Open connection to NorthwindCS database on cab2200 server.
Set cnn1 = New ADODB.Connection
strCnn = "Provider=sqloledb;" & _
    "Data Source=cab2200;" & _
    "Initial Catalog=NorthwindCS;" & _
    "User Id=sa;Password=;"
cnn1.Open strCnn
```

```

' Open employee table with a forward-only, read-only recordset;
' this type is OK for a single pass through the data.
    Set rst1 = New ADODB.Recordset
    rst1.CursorType = adOpenForwardOnly
    rst1.LockType = adLockReadOnly
    rst1.Open "employees", cnn1, , , adCmdTable

' Print an employee telephone directory.
    Do Until rst1.EOF
        Debug.Print rst1.Fields("FirstName") & " " & _
            rst1.Fields("LastName") & " has extension " & _
            rst1.Fields("Extension") & "."
        rst1.MoveNext
    Loop

' Close the connection and recover the resource.
    cnn1.Close
    Set cnn1 = Nothing
End Sub

```

סבירת הבדיקות של פרק זה כוללת שרת CABXLI, מוחשב מקומי בשם SQL Server השוכן במחשב שבו נמצאת הוראות SQL. התchapir של הפניה לשרתים מקומיים שונה במקצת מהתחapir להפניה אל שרתים מרוחקים. במקומות שבהם שרת מוגדר בשם, תומך לצין פשוט (local). מכיוון שגם במחשב המקומי שלו מותקן מסד הנתונים NorthwindCS, אין צורך בשינויים נוספים בשיגורה. שיגורה זאת זהה לשיגורה הקודמת, למעט הקטע האחד המופיע להלן. שים לב לשם השרת החדש. בפועל, אולי רצוי שהמשתמשים יקבעו לעצם שם חשבונם כנימה, או שישתמשו בחשבון מוגבל בעל פחוות זכויות מאשר sa.

```

' Open connection to NorthwindCS database on the local server.
    Set cnn1 = New ADODB.Connection
    strCnn = "Provider=sqloledb;" & _
        "Data Source=(local);;" & _
        "Initial Catalog=NorthwindCS;" & _
        "User Id=sa;Password=;"
    cnn1.Open strCnn

```

תומך לعباد תצוגות באמצעות אותו תchapir המשמש אותו לטבלאות. עלייך רק לתיחס את שם התצוגה במרכאות כפولات, כפי שנהוג בטבלאות. עדיין תידרש להשתמש בהגדירה Option.adCmdTable לפרטיטורutz. תתקליטורutz המצויר מכיל דוגמה בשם openViewOnRemoteServer.

פתחת שירה מאוחסנת

שגרות מאוחסנות יכולות להחזיר ערכות רשומות. החזרת ערכות רשומות באמצעות שגרות מאוחסנות במקומותعزירת טבלאות תעניק תוספת גמישות ליישום מכיוון שהשימוש בשגרות מאוחסנות ישחרר אותו מהתלות בטבלאות במקור הנתונים הקיימים, או אפילו עותקים מדויקים של הטבלאות במקור הנתונים הנוכחי. באמצעות שגרות מאוחסנות תוכל לשנן רשומות, לחשב ערכים חדשים ולצרכו ערכי שדה לאורץ הרשומות בטבלה.

השיגרה הבאה מפעילה את השיגרה המאוחסנת ששמה Ten Most Expensive Products וمدפסה בחלון Immediate את שמות עשרת המוצרים היקרים ביותר ומחיריהם. מכיוון שם השיגרה מכיל תווי רווח, חובה לתחום אותו בסוגרים מרובעים.שים לב גם לכך שהשיגרה משתמשת ב- AdCmdStoredOProc כArgument Options. דבר זה מורה לicode ADO לצפות לשם-שיגרה המכיל SQL, אך לא למשפט SQL של ממש. הויל והשיגרה תמיד מעבדת 10 רשומות, השיגרה להדפסת הרשומות משתמשת בלולאת For החוזרת על עצמה מ-1 עד 10. בכל שאר המובנים, השיגרה להדפסת הקבוצה המוחזרת של שיגרה מאוחסנת זהה לשיגרה להדפסת טבלה שלמה.

```
Sub openProcedureOnRemoteServer()
Dim cnn1 As ADODB.Connection
Dim rst1 As ADODB.Recordset
Dim int1 As Integer
' Open connection to NorthwindCS database on cab2200 server.
    Set cnn1 = New ADODB.Connection
    strCnn = "Provider=sqloledb;" & _
        "Data Source=cab2200;" & _
        "Initial Catalog=NorthwindCS;" & _
        "User Id=sa;Password=;"
    cnn1.Open strCnn
' Open employee table.
    Set rst1 = New ADODB.Recordset
    rst1.CursorType = adOpenForwardOnly
    rst1.LockType = adLockReadOnly
    rst1.Open "[Ten Most Expensive Products]", cnn1, , , adCmdStoredProc
' Print prices for 10 products.
    For int1 = 1 To 10
        Debug.Print rst1.Fields(0) & " has a unit price of $" & rst1.Fields(1) & "."
        rst1.MoveNext
    Next int1
' Close the connection and recover the resource.
    cnn1.Close
    Set cnn1 = Nothing
End Sub
```

הצבת ערכים במאפיין CursorLocation

הגדרות שתציב ב-CursorLocation עשויות להשפיע באורך מהותי על רמת הביצועים, אfilו כאשר מדובר במספר יינוי של מקורות טבלאים. הדוגמה הבאה מראה את הדבר בעזרת מקור רשומה המכיל מעט מעל 19,000 רשומות. השירה יוצרת את ערכת הרשומות בעזרת תצוגה המבוססת על המכפלת הקרטזית של הטבלאות Employees ו- .vwLargeView. שם התצוגה הוא Order Details

הדוגמה פותחת את מקור ערך הרשומות בעזרת CursorLocation או adUseServer adUseClient כהגדרת המאפיין CursorLocation שלה. הדוגמה adUseServer גורמת לשירה להתקדם דרך הרשומות שברשת, בזו אחר זו. הדוגמה adUseClient מעבירה את הרשומות אל תחנת הבדיקה המקומית, כך שהשגורות יכולו לגשת אל הרשומות דרך תחנת עבודה מקומית מבלי לחזור אל השורת עברו כל רשומה.

השירה הראשונה בדוגמה מנהה את המשמש להזין את הגדרת CursorLocation הרצוייה. השירה השנייה מכינה דוח ומדפיסה אותו בחלון **Immediate**. היא מדפיסה גם את מיקום וסוג הסמן, וכן את שעת ההתחלה, שעת הסיום ומשך הזמן של המשימה. תוכל להתאים את התבנית הכללית הזאת לבחינות צירופים שונים בין הגדרות של מסדי נתונים לבין מקורות הנתונים שלך.

```
Sub openRemoteWithCursorLocation()
Dim cnn1 As ADODB.Connection
Dim rst1 As ADODB.Recordset
Dim start As Date, done As Date
' Open connection to NorthwindCS database on cab2200 server.
strCnn = "Provider=sqloledb;" & _
    "Data Source=cab2200;" & _
    "Initial Catalog=NorthwindCS;" & _
    "User Id=sa;Password=;"
Set cnn1 = New ADODB.Connection
If MsgBox("Use local cursor?", vbYesNo, _
    "Programming Microsoft Access 2000") = vbYes Then
    cnn1.CursorLocation = adUseClient
Else
    cnn1.CursorLocation = adUseServer
End If
cnn1.Open strCnn

' Open vwLargeView view; create the view in the remote
' database server before running the procedure.
' The sample uses the Cartesian product of the
' Employees and Order Details tables.
' Notice that the connection setting for CursorType silently
' overrides the recordset property setting.
```

```

Set rst1 = New ADODB.Recordset
rst1.CursorType = adOpenKeyset
rst1.LockType = adLockOptimistic
rst1.Open "vwLargeView", cnn1, , , adCmdTable
start = Now
Do Until rst1.EOF
    temp = rst1.Fields(1)
    rst1.MoveNext
Loop
done = Now
reportResults cnn1.CursorLocation, rst1.CursorType, start, done
End Sub

Sub reportResults(cloc As Integer, ctype As Integer, _
startedAt As Date, endedAt As Date)
Debug.Print "Results for: "
Select Case cloc
    Case adUseServer
        Debug.Print "    Cursor Location: adUseServer"
    Case adUseClient
        Debug.Print "    Cursor Location: adUseClient"
    Case Else
        Debug.Print "    Faulty Cursor Location setting"
End Select
Select Case ctype
    Case adOpenForwardOnly
        Debug.Print "    Cursor Type: adOpenForwardOnly"
    Case adOpenKeyset
        Debug.Print "    Cursor Type: adOpenKeyset"
    Case adOpenDynamic
        Debug.Print "    Cursor Type: adOpenDynamic"
    Case adOpenStatic
        Debug.Print "    Cursor Type: adStatic"
End Select
Debug.Print "Start time to nearest second: " & startedAt
Debug.Print "End time to nearest second: " & endedAt
Debug.Print "Difference in seconds: " & DateDiff("s", & startedAt, endedAt)
End Sub

```

תרשים 12.18 מציג את התוצאות של הפעלת השגרות, תחילת עם ההגדירה `adUseClient` ולאחר מכן עם ההגדירה `.asUseServer`. מעבר הלולאה בין הרשומות נמשך שנייה אחת בלבד עם סמן מקומי, אולם נדרשו 37 שניות למעבר הלולאה דרכן הרשומות בעורף סמן בצד-השרת. התוצאות עשויות להשנותו, בהתאם למגוון רחב של גורמים, לכן רצוי לבדוק את העניין עם הגדרות נוספות מול מסדי הנתונים המשמשים אותו.

The screenshot shows the Microsoft Visual Studio Immediate window titled "Immediate". It displays the output of a VBA script comparing cursor locations and times between two different ADO cursor types: adUseClient and adUseServer.

```
Results for:  
    Cursor Location: adUseClient  
    Cursor Type: adStatic  
Start time to nearest second: 01/28/1999 7:28:26 AM  
End time to nearest second: 01/28/1999 7:28:27 AM  
Difference in seconds: 1  
Results for:  
    Cursor Location: adUseServer  
    Cursor Type: adOpenKeyset  
Start time to nearest second: 01/28/1999 7:57:36 AM  
End time to nearest second: 01/28/1999 7:58:13 AM  
Difference in seconds: 37
```

תרשים 12.18: דוגמת פלט המראה את תוצאות הבדיקה של אותו מקור נתונים, פעם עם ההגדירה adUseServer ופעם עם ההגדירה adUseClient

דוגמה זאת חשובה מכמה טעמים. ראשית, רמת הביצוע משתנה מאוד בהתאם להגדירה. שנית, ניתן לראות באמצעותה ש-ADO משנה את ההגדירות בלבד אם קיימת ביניהן התנגשות. לדוגמה, התוכנית מציבה adOpenKeyset כהגדרה של סוג הסמן, אולם הגדרה זאת מתנששת עם הגדרה adUseClient, שכן מתרגמים ADO משנה בשקט (לא התראה) את הגדרת סוג הסמן ל-`adStatic`. דבר דומה מתרחש גם במקרים נוספים. שלישיית, דוגמה זאת מציגה תבנית פשוטה שתוכל להתאים בנקל לבדיקת הגדרות של מסדי נתונים ביישומייך המותאימים אישית.

Web-ה Access 2000

Microsoft Access 2000 ממשיך במסורת הcapsולה של טכנולוגיית אינטרנט חדשנית ושיטות שימושיות ופשיות לפיתוח פתרונות בתחום מסדי הנתונים ל-Web. Access 2000 מכיל דפי גישה לעמודים, פקדי ActiveX חדשים, קישוריות מסד נתונים משופרת, ו-HTML דינמי (DHTML). יכולות חדשות אלו מציינות את קיצו של אשף פרסום התכנים ב-Web מתוך Access 97, שהכיל משך לפחות של מתאר HTML. פקד זה לא היה אלא טכנולוגיית ביןיהם, עד שישלם פיתוחה של שפת DHTML מבוססת על סטנדרטים.

פרק זה סוקר את הטכנולוגיות המסורתניות של Access ל-Web, ולאחר מכן יעסוק בטכנולוגיות ההיפר-קישורים, לרבות האובייקט Hyperlink וסוג הנתונים היפר- קישור (Hyperlink). הפרק מוקדש ברובו לדפי גישה לנוטונים, המפשטים את תהליכי הדיווח וקשרי הגומלין עם מסדי נתונים דרך ה-Web. סביבת הפיתוח לדפי גישה לנוטונים עשויה גם בתפקיד מארח לרכיבי Web ב- Microsoft Office 2000.

גישה מסורתיות

טכנולוגיות Access המסורתיות ל-Web מתאימות במיוחד לפיתוח פתרונות בנהפה פעילות נמוך (כלומר, עד 10,000 מבקרים ביום) ל-Web ולמסדי נתונים. מכיוון שבמרקמים מסוימים יש אי-התאמה בין גרסאות דפדףים שונות של יצרנים שונים, דוחoka הגישות המסורתיות הן הבוטחות, וגם קל למדי להשתמש בהן. Access 2000 יכול לשולש גישה מסוג זה:

- ﴿ פרסום גליונות נתונים בתבנית IDC/HTX, HTML ו-ASP.
- ﴿ שימוש בטפסי HTML. אין ב- Access סביבת פיתוח מוכلالת לטפסי HTML, אולם תוכל להתאים את טפסי HTML הישנים שלק לטכנולוגיית OLE DB/ADO (Data Objects).
- ﴿ פרסום דוחות בתבנית תמונה (snapshot) בתווך תיקיית FTP. שיטה זאת מפשטה את הגישה לדוחות Access באמצעות דפפני Netscape.

פרסום גליונות נתונים

כדי ליציא גליון נתונים של Access, בחר בפקודה **יצא** (Export) מתוך **תפריט קובץ** (File) בחולון מסד הנתונים. פקודזה זאת (המפרסמת גם גליונות נתונים שבבסיס טפסים ודוחות) מכילה חלק מהfonקציונליות שהיתה בעבר לאשף פרסום התכנים ב-Web המיוושן. הפקודזה מפרסמת גליונות נתונים בתבנית HTML, HTML IDC/HTX ו-ASP. התבנית HTML אמנים סטטיות, אולם קל לעורך אותה באמצעות עורך HTML הרגילים. שתי התבניות האחירות דינמיות. הם משתמשות בשרת כדי לכתוב תכנים HTML כך שיישקפו את התוצאות העדכניות ביותר בגליון הנתונים.

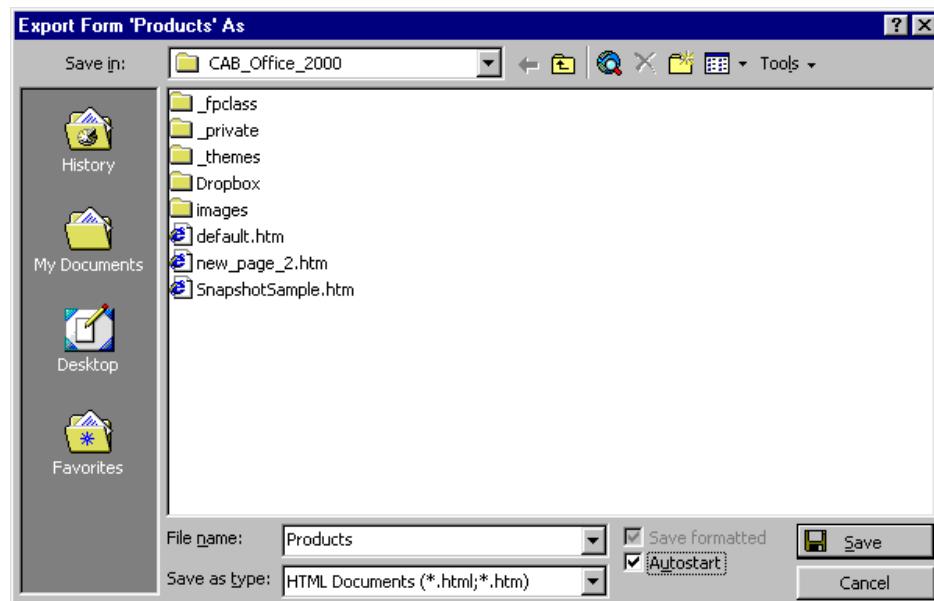
תבנית IDC/HTX מיועדת למעשה לשתי המהדרות הראשונות של Microsoft Internet Server (IIS), אולם היא מסוגלת לפעול גם עם מהדרות חדשות יותר. בניית ASP פועלת רק עם המהדרות החדשנות יותר, אולם מאפשרת שילוב של Microsoft Visual Basic Scripting Edition Microsoft Jscript ו-HTML בקוד. לא כדאי לבחור באופן אוטומטי באחת התבניות הדינמיות. אם אתה עובד עם גליונות נתונים גדולים או בסביבה מרובת-משתמשים, ואם התכנים בגליונות הנתונים אינם משתמשים תדירים, או אם הזמן אינו גורם חשוב, ניתן בהחלט שתבנית HTML הסטטית תתאים לך יותר, הואיל ופרסום דפים סטטיים מהיר יותר.

פרסום בתבנית HTML

כדי לפרסם גליון נתונים בתבנית HTML, בחר טבלה, שאלתה, טופס או זוז בחולון מסד הנתונים, ולאחר מכן בחר באפשרות **יצא** מתפריט **קובץ**. בתיבת הדו-שייך, השתמש ברשימה הנפתחת **שמור ב** (In Save) לבחירת מקום לקובץ. אם אתה מפרסם בראשת ארגונית (אינטרא-נט), המיקום יכול להיות תיקיה באתר. אם אתה מפרסם

בשרת Web מרוחק, תוכל להשתמש בכל תייקה מקומית לאיסוף דפי HTML, ולאחר מכן להעביר את דפים לשרת המרוחק באמצעות הפקודה **Import**, **File** – ב- **.FTP**. FrontPage 2000 מוכן להשתמש גם בתיבים אחרים, כמו למשל פרוטוקול CAB.

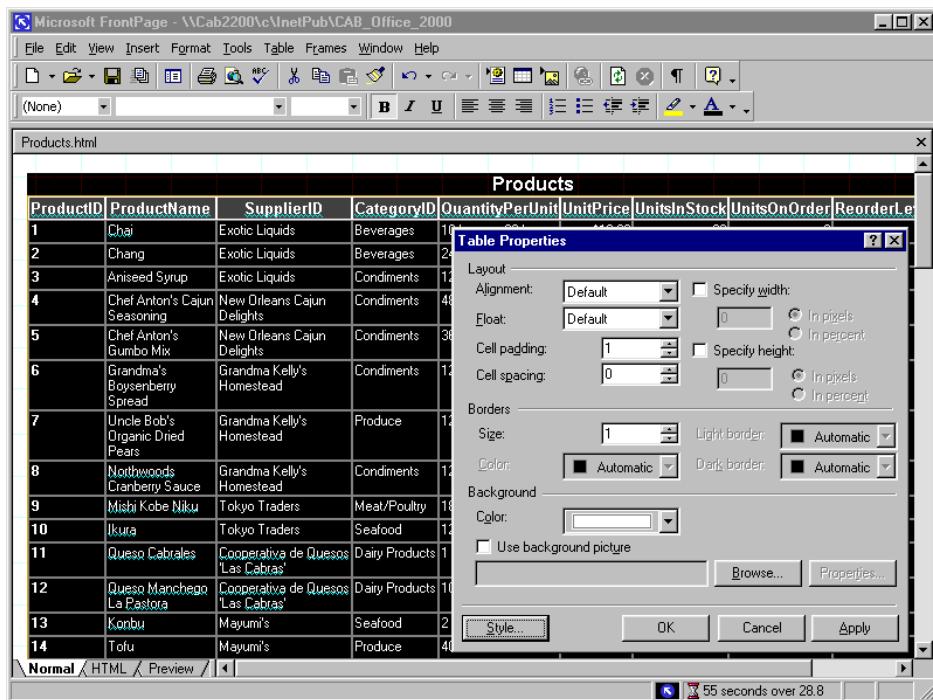
תרשים 13.1 מציג את תיבת הדו-שיח **יצא** לפרסום גיליוון הנתונים שבבסיס התופס Products מתוך מסד הנתונים Northwind. תיבת הרשימה הנפתחת **שמור כסוג** (Save As) מראה את תבנית HTML. התיבה **שמור** ב- מגדירה את התיקיה (As Type) כתיקיית הבסיס של אתר ברשות ארגונית מקומית בעל שם זהה. תיבת הסימון **אוטומטי** (Autostart) נבחרה, שכן הדף יופיע ברגע שייפרסם. דבר זה יכול לגרום להפעלת דפדף להיפתח כאשר גיליוון הנתונים פתוח.



תרשים 13.1: תיבת הדו-שיח **יצא**

כאשר לוחצים על **שמור** (Save), מופיעה תיבת הדו-שיח **אפשרויות פלט HTML** (HTML Output Options). כאן תוכל לציין תבנית לדף המכיל את גיליוון הנתונים. אם הגדרת ערכת נושא לדף default.htm, תוכל להפנות אל דף זה. אם לא, תוכל להפנות אל קובץ מקומי עם העיצוב שישמש אותו באתר המרוחק.

אחד היתרונות של תבנית HTML הוא שהפעולה שעלייך לבצע היא עיברת טבלת HTML. תרשימים 13.2 מציג את גיליוון הנתונים Products שפורסם ב- FrontPage 2000, ואת תיבת הדו-שיח **Properties**, **Table Properties**, המכילה חלק מאפשרויות העיצוב, לרבות הגנת תאים, מרוחך בין תאים וצבוע או תמונה רקע. כמו כן תוכל להגדיר מאפיינים לתאים, שורות ועמודות כך שיציינו אפשרות עיצוב אחרות. באפשרות לבטל עיצובים מותאמים אישית ב- FrontPage, ואפשרו לבחור עמודות או שורות נפרדות, ולמחוק אותן באופן ברור.



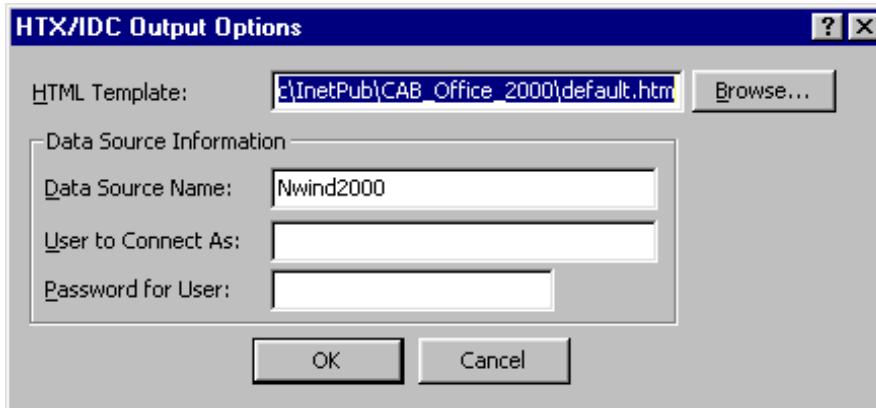
תרשים 13.2: לאחר טעינת גליונות הנתונים אל תוך FrontPage, תוכל לעורר את טבלת HTML שתופיע

פרסום בתבנית IDC/HTX

כדי לפרסם גליון נתוניים באמצעות התבניות הדינמיות, דרוש לך שם מקור נתונים (DSN) בקשריות מסד נתונים פתוחה (ODBC). על שם מקור הנתונים לשכון בשרת Web, לצד מסד הנתונים שאליו הוא מפנה. גישת DSN של קבצים היא לכורה גמישה יותר, הואיל ונitin לשЛОוח את שם מקור הנתונים לשרת, אולם במקרים רבים, ספקי שירות אינטרנט מחילים כללים מיוחדים שהופכים את הפעלת הקבצים הללו למסובכת יותר. אצל ספקי השירותים נהוג להפיק דרכ קבע DSN אחד או יותר לכל חשבון של אתר Web עם זכויות שימוש במסדי נתונים. לכן פעמים רבות קל יותר להשתמש בשמות מקור נתונים של המערכת ולא בהתאם של קבצים. ככל מקורה, שמות מקור נתונים של המערכת עדיפים מבחינה רמת הביצועים.

לאחר שבחרת מקור גליון נתונים בחלון מסד הנתונים, ולאחר שבחרת קובץ, יצא, בחר באפשרות *.ht;*.idc (IIS 1-2). בתייבת הרשימה **שמור כסוג**. בחר באפשרויות המתאימות לך, באופן שנהוג עבור קובץ HTML. לחיצה על **שמור** תגרום לפיתוח תיבת הדו-שיך **אפשרויות פלט IDC/HTX** (IDC/HTX Output Options) (המודצת בתרשימים 13.3). השתמש בלחצן **עיוון** (Browse) לבחירת התבנית שתבוצע את הדף שיזיר את התוצאות מקובץ IDC. כדי לפעול, קובץ IDC זוקק לשם DSN בשרת.

בתיבת הטקסט **שם מקור הנתונים** (Data Source Name), הzon את שם מקור נתונים שמנהל האתר שלך הקצה למשימה זאת. תיבת הדו-שיח כוללת גם שדות זיהוי משתמש וסיסמה, למקורה שמסד הנתונים בשרת ה- Web פועל עם אבטחה ברמת המשתמש (מקרים כאלה נדרים, במיוחד בישומי אינטרנט).



תרשים 13.3: תיבת הדו-שיח אפשרויות פלט HTX/IDC

קובצי IDC ו-HTX משלימים זה את זה. קובץ IDC מפעיל שאילתת מול מסד נתונים של Access בשרת ה- Web. קובץ HTX משתמש בהרחבות HTML כדי לעצב את הקבוצה המוחזרת בתבנית HTML, לצורך הצגה באמצעות דף-דף. אמנים ניתן כתוב את הקוד של קובץ HTX בהרחבת HTML, ואולם הקובץ מוחזר אל הדף-דף HTML טהור, כדי שכל דף-דף יוכל לקרוא את התוצאות. מכיוון שקובץ IDC יכול לחושף את הקבוצה המוחזרת שלו רק באמצעות קובץ HTML, תבנית זאת מתאימה לעיבוד גליונות נתונים. עם זאת, אין היא מתאימה לעיבוד טפסים, מכיוון שתבנית החזרה של HTML אינה קלט מתאים עבור מרבית טפסי HTML.

כאשר מפעילים זוג של HTX/IDC מתוך שרת Web, הדפים פונים אל קובץ IDC, שחייב לשכון בתיקיית שרת Web המסוגלת להפעיל scripts. קובץ IDC מפעיל באופן אוטומטי את קובץ HTX, אשר בתורו מעבד את הקבוצה המוחזרת של קובץ IDC ומsegג את התוצאות, בתבנית HTML, בחזרה לדף-דף.

פרסום בתבנית ASP

בדומה לקובצי IDC ו-HTX, קובצי ASP זוקקים להפניה אל DSN. בתיבת הדו-שיח יצא, בחר באפשרות (*.asp) Microsoft Active Server Pages (Microsoft Active Server Pages (*.asp) מתייבת הרשימה שמור בסוג. בתיבת הדו-שיח אפשרויות פלט עמודים של שירותי Microsoft Active Server Pages Output Options (Microsoft Active Server Pages Output Options) הzon שם של DSN. שאר השדות בתיבת הדו-שיח אופציונליים, ותלויים בהעדפות העיצוב שלך ובהגדרות האבטחה של מסד הנתונים (שדות אלה אופציונליים גם בתיבות הדו-שיח IDC ו-HTX).)

תבנית ASP שונה מtabנית IDC/HTX בכך שיש רק קובץ אחד. קובץ ASP מבצע את השאלה ומעצב את התוצאה לצפייה באמצעות הדפסן. הקובץ מפעיל את השאלתה שלו ויוצר את ה-HTML בשורת, ולאחר מכן שרת ה-Web מעביר את דפי ה-HTML בחזרה אל הדפסן. קוד ASP שלහן כותב את גילוון הנתונים שבבסיס הטופס Products שבסיד הנתונים אל תוך דף Northwind אל תוך דף :

```
<HTML><HEAD>
<META HTTP-EQUIV="Content-Type"
      CONTENT="text/html; charset=windows-1252">
<TITLE>Products</TITLE>
</HEAD><BODY>
<%
If IsObject(Session("Nwind2000_conn")) Then
    Set conn = Session("Nwind2000_conn")
Else
    Set conn = Server.CreateObject("ADODB.Connection")
    conn.open "Nwind2000","",""
    Set Session("Nwind2000_conn") = conn
End If
%>
<%
If IsObject(Session("Products_rs")) Then
    Set rs = Session("Products_rs")
Else
    sql = "SELECT * FROM [Products]"
    Set rs = Server.CreateObject("ADODB.Recordset")
    rs.Open sql, conn, 3, 3
    If rs.eof Then
        rs.AddNew
    End If
    Set Session("Products_rs") = rs
End If
%>
<TABLE BORDER=1 BGCOLOR=#ffffff CELLSPACING=0 RULES=none>
<FONT FACE="Arial" COLOR=#000000><CAPTION><B>Products</B>
</CAPTION></FONT>
<THEAD><TR>
<TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
<FONT SIZE=2 FACE="Arial" COLOR=#000000>ProductID</FONT></TH>
<TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
<FONT SIZE=2 FACE="Arial" COLOR=#000000>ProductName</FONT></TH>
<TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
<FONT SIZE=2 FACE="Arial" COLOR=#000000>SupplierID</FONT></TH>
<TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
```

```

<FONT SIZE=2 FACE="Arial" COLOR="#000000>CategoryID</FONT></TH>
<TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
<FONT SIZE=2 FACE="Arial" COLOR="#000000>QuantityPerUnit</FONT></TH>
<TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
<FONT SIZE=2 FACE="Arial" COLOR="#000000>UnitPrice</FONT></TH>
<TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
<FONT SIZE=2 FACE="Arial" COLOR="#000000>UnitsInStock</FONT></TH>
<TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
<FONT SIZE=2 FACE="Arial" COLOR="#000000>UnitsOnOrder</FONT></TH>
<TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
<FONT SIZE=2 FACE="Arial" COLOR="#000000>ReorderLevel</FONT></TH>
<TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
<FONT SIZE=2 FACE="Arial" COLOR="#000000>Discontinued</FONT></TH>
</TR></THEAD>
<TBODY>
<%
On Error Resume Next
rs.MoveFirst
do while Not rs.eof
%>
<TR VALIGN=TOP>
<TD BORDERCOLOR=#808080 ><B>
<FONT SIZE=1 FACE="MS Sans Serif" COLOR="#000000>
<%=Server.HTMLEncode(rs.Fields("ProductID").Value)%><BR></FONT>
</B></TD>
<TD BORDERCOLOR=#808080 >
<FONT SIZE=1 FACE="MS Sans Serif" COLOR="#000000>
<%=Server.HTMLEncode(rs.Fields("ProductName").Value)%><BR>
</FONT></TD>
<TD BORDERCOLOR=#808080 >
<FONT SIZE=1 FACE="MS Sans Serif" COLOR="#000000>
<%=Server.HTMLEncode(rs.Fields("SupplierID").Value)%><BR>
</FONT></TD>
<TD BORDERCOLOR=#808080 >
<FONT SIZE=1 FACE="MS Sans Serif" COLOR="#000000>
<%=Server.HTMLEncode(rs.Fields("CategoryID").Value)%><BR>
</FONT></TD>
<TD BORDERCOLOR=#808080 >
<FONT SIZE=1 FACE="MS Sans Serif" COLOR="#000000>
<%=Server.HTMLEncode(rs.Fields("QuantityPerUnit").Value)%><BR>
</FONT></TD>
<TD BORDERCOLOR=#808080 ALIGN=RIGHT>
<FONT SIZE=1 FACE="MS Sans Serif" COLOR="#000000>
<%=Server.HTMLEncode(rs.Fields("UnitPrice").Value)%><BR>

```

```

</FONT></TD>
<TD BORDERCOLOR="#808080" ALIGN=RIGHT>
<FONT SIZE=1 FACE="MS Sans Serif" COLOR="#000000>
<%=Server.HTMLEncode(rs.Fields("UnitsInStock").Value)%><BR>
</FONT></TD>
<TD BORDERCOLOR="#808080" ALIGN=RIGHT>
<FONT SIZE=1 FACE="MS Sans Serif" COLOR="#000000>
<%=Server.HTMLEncode(rs.Fields("UnitsOnOrder").Value)%><BR>
</FONT></TD>
<TD BORDERCOLOR="#808080" ALIGN=RIGHT>
<FONT SIZE=1 FACE="MS Sans Serif" COLOR="#000000>
<%=Server.HTMLEncode(rs.Fields("ReorderLevel").Value)%><BR>
</FONT></TD>
<TD BORDERCOLOR="#000000" ALIGN=RIGHT><B>
<FONT SIZE=2 FACE="System" COLOR="#000000>
<%=Server.HTMLEncode(rs.Fields("Discontinued").Value)%><BR>
</FONT></B></TD></TR>
<%>
rs.MoveNext
loop
%>
</TBODY><TFOOT></TFOOT>
</TABLE></BODY></HTML>

```

שים לב, שיש בקוד תערובת של HTML ו-VBScript. כמו כן נעשה בו שימוש ב-ADO כדי ליצור את החיבור אל מקור נתונים, לשולף ערכת רשומות, ולצrrף את התוצאות לטבלת HTML. לולאת Do מעבירה את תוכן ערכת הרשומות אל תוך טבלת HTML. שום חלק מהקוד ב-VBScript אינו נשאר בגרישה שהשתרת משרג אל הדפדפן; הדפדפן מקבל HTML טהור. תכונה זאת מתאימה את ASP לסוגים רבים ושוניים של דפדףים. השימוש שבכתיבת HTML במסגרת העובודה השוטפת היא הגורם להשניה כאשר דפדפן מפעיל קובץ ASP. אם דפים מסוימים מכילים יותרChr'ות ושדות לתרגומים, נדרש להם יותר זמן חיבור מאשר לדפים המכילים מספר שדות ושורות בלבד. מומלץ לתכנן את תוכן דפי ASP כך שיכילו כמהות מספקת של מידע בדף ייחיד, מבלי לגרום לעיכובים מיוחדים בעיבוד.

שימוש בטפסי HTML

אחד היתרונות של טכנולוגיית ASP הוא התמיכה שלה בטפסי HTML. נקודה זאת חשובה, הואיל וכל הדפדףים תומכים בטפסי HTML. תוכל להשתמש בטפסים אלה לאיסוף נתונים ממבקרים באתר, או להחזיר תוצאות רשומות מסד נתונים במבנה רשומה (לעומת מבנה גיליון נתונים). טופס HTML כולל פקד אחד, או יותר, להציג או לשמור של נתונים. ככל, הטופס מכיל לפחות לחץ אחד למשלו שדות הטופס אל שרת Web. אם מדובר בטופס המחייב מענה, אחת התוכניות בשרת תוכל לעבד את שדות הטופס, ולהפיק דף תגובה.

במידה רבה, טפסי HTML פועלים כמשחק קריאה-ותגובה בין דפדפן ושרת Web. המשמש בצד הדפדפן מלא את השדות ולוחץ על הלחצן **שלח** (Submit) בטופס. כאשר הדפדפן שולח את הטופס לשרת, הוא מעביר את ערכיו השdots בטופס, וגם שם של תוכנית בשרת אשר יודעת מה לעשות באוטם ערכיהם. זהו חלק ה'קריאה' במשחק. שרת Web-הו שולח את ערכיו השdots אל תוכנית העיבוד המתאימה בשרת. תוכנית זאת יכולה ליזור הדלע ערכיה הקלט, לבדוק את חוקיותם, לצרף אותם למקור רשומות, לבצע בדיקת מידע לנתחים מתוך מקור רשומות, ועוד. בדרך כלל היא מכינה דף החזר כלשהו לדפדפן. זהו חלק ה'תגובה'. קבצי ASP יכולים לשמש הן כטופס הראשוני המקבל את הקריאה, והן כתוכנית התגובה המשיבה לקריאה.

תרשים 13.4 מציג זוג טפסי HTML המדגימים את אופי הקריאה-ותגובה המאפיין את העיבוד של טפסי HTML. הטופס העליון כולל תיבת רשימה נפתחת עם שמות של עובדים במסד הנתונים Northwind. המשמש יכול לבחור שם, לדוגמה Buchanan, ולחוץ על הלחצן **Get Extension** כדי לשגר את הטופס לשרת. זהה פעולה ה'קריאה'. שרת Web-הו מעביר את טופס הקלט לקובץ telereturn.asp, הקורא את הערך בתיבת הרשימה, ממחפש את מספר הטלפון של העובד המבוקש, וכותב עבורה הדפדפן דף המכיל את הנתונים המבוקשים. זהה פעולה התגובה.

Access 2000 אינו כולל אשפים מוכללים לבניית דפים מסוג זה עם טפסי HTML, אולם ההתלהיך אינו מסובך כלל. האתגר המרוכז הוא למידת התחביר של מספר פקדים לטפסי HTML, וכי怎ד ליזור בתוך קובץ ASP עירוב בין HTML ו-script (למשל VBScript). דוגמת הקוד הבאה היא כל שנדרש לדף העליון בתרשים 13.4. שם הקובץ של קוד זה הוא telelookup.asp. שם זה מופיע בתיבת הרשימה **Address** שבדף ה

- עלינו בתרשים.

```
<%
set cnn1 = server.createobject("ADODB.Connection")
cnn1.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=C:\Program Files\Microsoft Office\Office\" & _
    "Samples\Northwind.mdb;"
set rs = Server.CreateObject("ADODB.Recordset")
rs.open "select * from employees", cnn1
%>
<html>
<head><title>Extension Lookup Window</title></head>
<body>
<form name=MyForm method=Post action=telereturn.asp>
Last name: <select name=DCombo>
<%
do while not rs.eof
%>
    <option value=<%=rs.fields("EmployeeID")%>>
        <%=rs.Fields("LastName")%>
<%
rs.MoveNext
Loop
%>
```

```
</select>
<input type="Submit" value="Get Extension"><br><br>
</form></body></html>
```

The image consists of two vertically stacked screenshots of Microsoft Internet Explorer windows.

Top Window (Extension Lookup Window):

- Address bar: http://cab2200/cab_office_2000/telelookup.asp
- Form fields:
 - Last name:
 - Get Extension
- Dropdown menu:
 - Davolio
 - Fuller
 - Leverling
 - Peacock
 - Buchanan **(highlighted)**
 - Suyama
 - King
 - Callahan
 - Dodsworth

Bottom Window (Extension Result Window):

- Address bar: http://cab2200/cab_office_2000/telereturn.asp
- Form fields:
 - First Name:
- Text:

The extension for Steven Buchanan is 3453. [Go back to the combo box form for another extension lookup.](#)

תרשים 13.4: טופס HTML בתמונה העילונה קורא לשרת ה-Web וגורם ליצירת דף המuna שבתמונה התחתונה

שים לב שה-`script` תחום בסימנים `<% - %>`. מקטע ה-`script` הראשון יוצר חיבור אל מסד הנתונים Northwind באמצעות מנהלי DB OLE שאינם זוקקים ל-DSN. המקטע הראשון אף יוצר ערכת רשומות עם הנתונים שיאכלסו את תיבת הרשימה הנפתחת. בעת, חלק מקוד HTML פותח דף, ועליו טופס. בהצחה הטופס, הקוד מגדר את `So`, `telereturn.asp` כתוכנית שתעבד את הטופס. ממש לפि תחילתה של לולאת `HTML`. התוכנית מצהירה על תיבת רשימה נפתחת עם מילת המפתח `select` של שפת `HTML`. היא מעניקה לפקד את השם `dcombo` (שמות בשפת HTML אינם תלויי-רישיות). לולאת `- Do` מaculaת את תיבת הרשימה בערכיהם מtower ערכות הרשומות שיצר מקטע ה-

Get Extention Submit בעל תווית `script`

הקובץ פותח אף הוא ביצירת חיבור וערכת רשומות, כפי שמצוג להלן. אולם פעולת פתיחה זאת נבדلت בשני מובנים חשובים מזו של הקובץ `telelookup.asp`. ראשית, קובץ זה משתמש ב- DSN כדי לעזור ביצירת החיבור. דבר זה אינו הכרחי, אלא שקבצי ASP רבים מעוצבים באופן זה. שנית, הקוד קורא את הערך של `so`, `dcombo` ומשתמש בו כדי ליזור את משפט SQL לשאלתה שתחפש את מספר השולחה של העובד.

```
<%
set rs = Server.CreateObject("ADODB.Recordset")
sql = "select * from employees"
sql = sql & " where employees.employeeid = "
sql = sql & request.form("dcombo")
rs.open sql,"DSN=Nwind2000"
%>
<html>
<head><title>Extension Result Window</title></head>
<body>
<form>
First Name: <input TYPE="TEXT" VALUE="
    <%=rs.fields("FirstName")%>">
</form>
The extension for <%=rs.fields("FirstName")%>&nbsp;
    <%=rs.fields("LastName")%>
is <%=rs.fields("Extension")%>.
<a href="telelookup.asp">
    Go back to the combo box form for another extension lookup.
</a>
</body></html>
```

לאחר שיצר ערכת רשומות עם הנתונים המבוקשים עבור העובד שנבחר בקובץ `telereturn.asp`, `telelookup.asp` מנצח דף. הדף מכיל את שמו הפרטיו של העובד בתיבת טקסט HTML, מחרוזת HTML פשוטה עם השם ומספר השולחה, והיפר- קישור להחזרת השליטה לקובץ `asp`. כאשר שרת ה- Web מעביר את הדף בחזרה לדפסון, מסתיים מחזור הקרייה-ותגובה.

שימוש בתמונות FTP עם דפפני Netscape

פרק 6 עסוק בתבנית התמונה (snapshot) לדוחות, והציג דוגמאות של יכולות התבנית, אולם לא הזכיר שימוש בתמונות עם FTP. כאשר מציירים תמונות בתקינות FTP המיעודת לשרת Web, דפפני Netscape יכולים לפתחו אוטומטית מרוחק בשרת ה-Web. המשמש אינו חייב לטעון את הקובץ אל המחשב שלו ולהפעיל מציג נפרד, כפי שנדרש בפרוטוקול HTTP.

כאשר דפפני Netscape פותח תמונה של דוח מתוך תיקיות FTP, הוא יוצר באופן אוטומטי חלון חדש עבור אותו דוח. תרחיש זה מניח ש- **Snapshot Viewer** (מציג התמונות) מותקן במחשב שלך, וכי יש בתוכנת העבודה שיזק בין קבצים מסג.snp.*. בין המציג. תרשימים 13.5 מציג מקטע מתוך תמונה של הדוח Catalog במאזן הנתונים Northwind. התרשימים מציג ציור, גופנים מורביים וסמנני עיצוב נוספים, כמו למשל קווים מפרידים. הטופס להזנת הזמינות שבסוף הדוחCatalog אף הוא העתק נאמן.



תרשים 13.5: מקטע מתוך תמונה דוח, שנפתחה בעזרת דפפני Netscape

בעת שמירת קובץ התמונה, הצב אותו בתיקיה FTPROOT עבור שרת ה-Web (או ftp://webserver/**filename** ב-Netscape **Location** ב-webserver). הצלם את שם שרת ה-Web שלך במקום **filename**. הצב את שם שרת ה-Web נפוץ ומקובל מאוד, תוכן זה, שרת ה-Web הוא cab2200). מכיוון שפרוטוקול FTP נפוץ ומקובל מאוד, תומך להיעזר במנגנון פרטום זה לubahוד עם סוגים רבים של שרתני Web ודפדפני Web.

שימוש בהיפר-קישורים

פרק 5 למדנו כיצד להשתמש בהיפר-קישורים כדי לנוטט בטפסים בתוך יישום מותאם אישית. זהו רק חלק קטן מהfonקציונליות התמונה בהיפר-קישורים. תוכל לנוטט גם אל מסדי נתונים אחרים, Access, אל סימניות במסמך Word, או אל כתובות URL בראשת ארגונית ובאינטראקטיבית. תוכל להשתמש בהיפר-קישורים גם להפעלת תוכנת הדואר האלקטרוני שלך (כמו למשל Outlook) עם כתובות של פרט מוגדר ונושא נתון.

סוגים של היפר-קישורים

היפר- קישור יכול להופיע כתווית, כלחצן פקודה, או כפקדי תמונה לא מאוגדים בטפסים, בדוחות ובדפי גישה לנוטנים. כמו כן ניתן ליצור היפר-קישורים מאוגדים, כך שתבליה תוכל להכיל סדרה של היפר-קישורים. אפשרוות להקצות היפר- קישור ייחודי לכל רשותה בטבלה. אחד השימושים הבורורים מאלהם אפשרוות זאת הוא שיווק כתובות URL לכל רשות רשותה (כמו למשל, חברות). לאחר שעקבת אחר היפר- קישור מאוגד או בלתי מאוגד, תוכל להשתמש בפקד **Back** (הקודם) בסרגל הכלים של הדפדפן או בסרגל הכלים אינטרנט (Web) של Office, כדי לנוטט בחזרה אל Access מכתובות יעד של היפר- קישור.

סוג הנתונים **Hyperlink**

כאשר משתמשים בהיפר-קישורים מאוגדים, היישום מצין שסוג הנתונים הוא **היפר- קישור** (Hyperlink). סוג נתונים זה עשוי להכיל עד ארבעה חלקים, כולל אחד מהם מופרד באמצעות הסימן #. החלק של טקסט התוצאה יופיע ראשון אם תציג זאת. טקסט זה מופיע במקומות הכתובת שאליה מוביל היפר- קישור. לדוגמה, במקומות היפר- קישור המראה את הכתובת של אתר Microsoft ב- Web, ניתן להציג את הכתובת Microsoft.

החלק השני מכיל את הכתובות – כתובות URL או UNC של קובץ היעד. הכתובות עשויה להיות מוחלטת או יחסית. באתרי Web, מקובל לציין היפר-קישורים באופן יחסית. ב-Access, כתובות יחסית נגורת מההגדרה **בסיס היפר- קישור** (Hyperlink Base) בקובץ Access. תוכל לבדוק הגדרה זאת על ידי בחירה בפקודה **מאפייני מסד נתונים**. Microsoft מתרירט **קובץ** (File) שבחלון מסד הנתונים.

החלק השלישי הוא הכתובת המשנית, המכילה את שם החלק של הקובץ שיש לנוט אליו. החלקים שנייתם לנוטו אליהם תלוים ברכיב Office. ב- Access ניתן לנוטות אל אובייקטים של מסד נתונים, כמו למשל טפסים, דוחות, טבלאות ושאלות. תוכל לציין סוג של אובייקט מסד נתונים, כמו למשל טבלה, ואת שמו של אובייקט מוגדר, כמו למשל Customers. אין צורך לציין את סוג האובייקט, אולם אם מסד נתונים מכיל מספר אובייקטים באותו שם, Access ינותו אליהם בסדר אקראי. אם היפר- קישור מכיל הן כתובת והן כתובת משנה, הקישור ינות אל קובץ אחר. אם חסר החלק המכיל את הכתובת, הקישור ינות אל אובייקט במסד הנתונים הנוכחי.

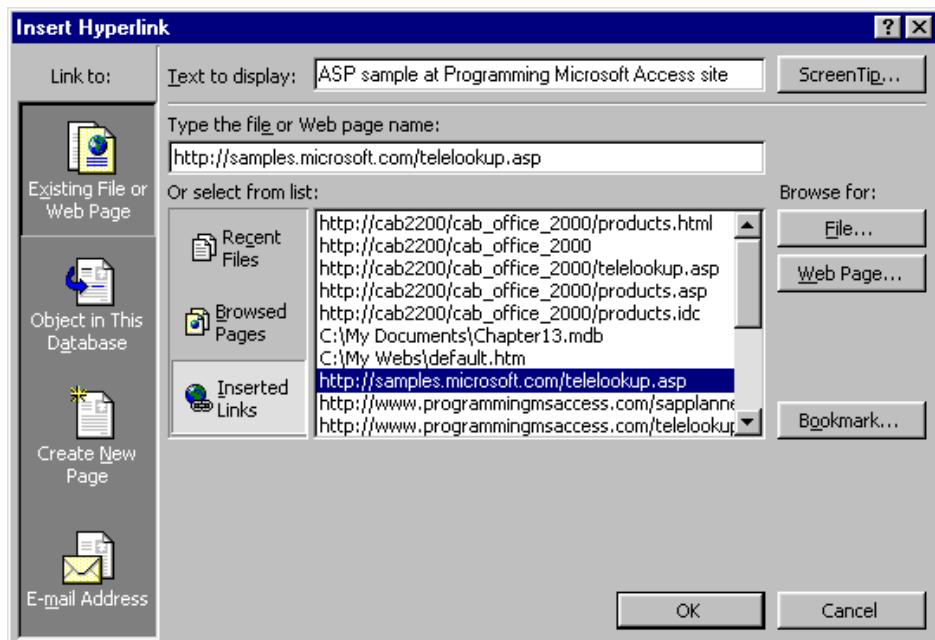
החלק הרביעי, תיאור המסך, חדש ב- Access 2000. הטקסט מופיע כאשר מניחים את הסמן על היפר- קישור. הוא עשוי להכיל תוכורת אודוט מטרתו של היפר- קישור. תיאורים אלה מופיעים עבור היפר- קישורים בדף מוצגים בעזרת Internet Explorer 4 ומעלה. כמו כן הם מופיעים ביישומים מותאמים אישית ב- Access 2000.

הוספה ועריכה של היפר- קישורים

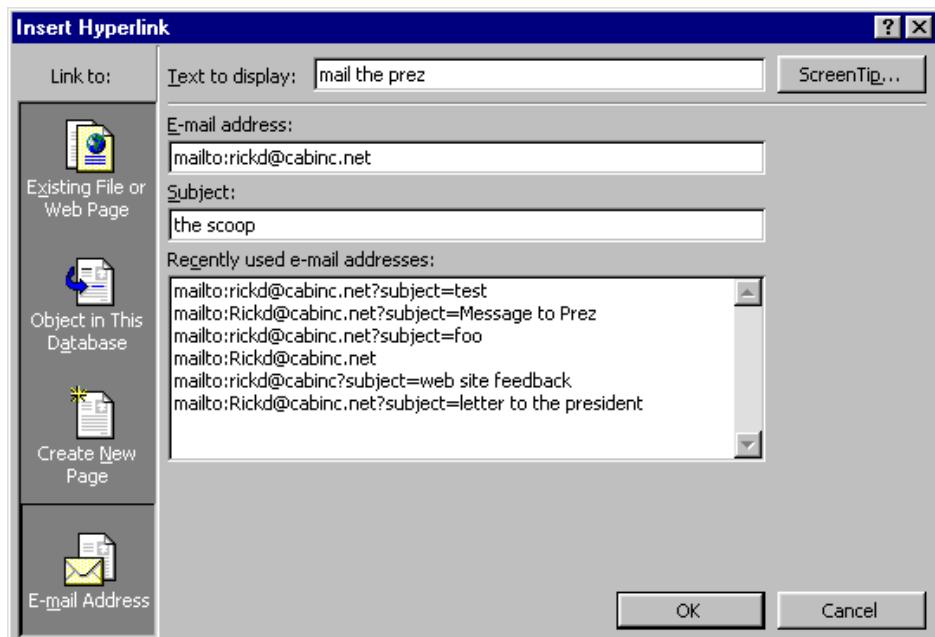
כאשר מקלידים כתובת היפר- קישור חדשה, פשוט ובוטוח יותר להשתמש בכל הוסף היפר- קישור (Insert Hyperlink), או בפקודת התפריט המקבילה. תוכל להפעיל את הפקודה **הוסף היפר- קישור** מכל נקודה שבה אפשר ליצור היפר- קישור במסמך. לדוגמה, תוכל לבחור פקודה זאת מתוך הרצוגה **עיצוב** (Design) של טופס, אם תרצה להוסיף היפר- קישור כתווית. תוכל להעניק להחצן פקודה יכולות ניווט על ידי לחיצה על הלחצן **בניה** (Build) הסמוך למאפיין **כתובת היפר- קישור** (Hyperlink Address) או **כתובת משנה של היפר- קישור** (Hyperlink SubAddress) של לחצן הפקודה. בדרך דומה, תוכל להעניק יכולות ניווט גם לפקץ ציר.

כל הפעולות הללו פותחות את תיבת הדו- שיח **הוספה היפר- קישור** בטופס. ערך היכיון של התווית הוא דוגמת ASP שבאתר הדמיוני Programming Microsoft Access שכתובה ה- URL שלו היא <http://samples.microsoft.com/telelookup.com>. לחץ על **תיאור מסך** (ScreenTip) כדי לפתוח תיבת דו- שיח שנייה להזנת טקסט עבור תיאור המסך המשויך לאותו היפר- קישור.

תוכל להציג על הלחצן **כתובת דואר אלקטרוני** (E-Mail Address) בפיס האפשרויות כדי לפתח את תיבת הדו- שיח בגירסה מתאימה לדואר אלקטרוני (ראה תרשימים 13.7). תיבת הדו- שיח מוסיפה באופן אוטומטי כתובת פרוטוקול mailto כדי להפעיל מהדפן את תוכנת ברירת המחדל לדואר אלקטרוני בתחנת העבודה. כמו כן תוכל לציין את הנמען ואת הנושא. בדומה לבחירה מתוך אתרים שכבר ביקרת בהם בעברaziין את הנמען ואת הנושא. בועת שאתה מקצה כתובת היפר- קישור, תוכל לבחור מבין כתובות דואר אלקטרוני ונוסאים שכבר הציגו. היפר- קישור שתרשימים 13.7 יוצר מציג את המשפט **mail the prez**. כאשר משתמש לווח על משפט זה, הוא מפעיל את תוכנת הדואר האלקטרוני של המשתמש, כאשר הנמען הוא rickd@cabininc.net, והנושא הוא **.the scoop**.



תרשים 13.6: תיבת הדו-שיח הוסף היפר- קישור של כתובת URL



תרשים 13.7: תיבת הדו-שיח הוסף היפר- קישור להודעה בדואר אלקטרוני

לאחר הוספת היפר- קישור ביצועם, תוכל ללחוץ עליו לחיצה ימנית בעבר ולבחר באפשרות **היפר- קישור** מתפריט הקיצור כדי לראות את האפשרויות לעיבוד היפר- קישורים קיימים. הפוקודה **עריכת היפר- קישור** (Edit Hyperlink) היא הבחירה הטבעית לצורך הכנסת שינויים בהיפר- קישור קיים. הפוקודה פותחת את תיבת הדו- שיח **עריכת היפר- קישור**, שיעצבוה זהה לתיבת הדו- שיח **הוספת היפר- קישור**. השתמש בתיבה זאת לעדכון השדות הרצויים בהיפר- קישור. לאורך זמן, ניתן שתצריך ללטש את הטקסט המוצג או את כתובות ה- URL של היפר- קישור. בדומה לכך, תוכל להוסיף ערכאים בראשיות הנמענים של קישורי המשגרים דואר אלקטרוני, כדי שהודעות ינותבו אל יותר נמענים באופן ישיר.

דוגמאות של היפר- קישורים

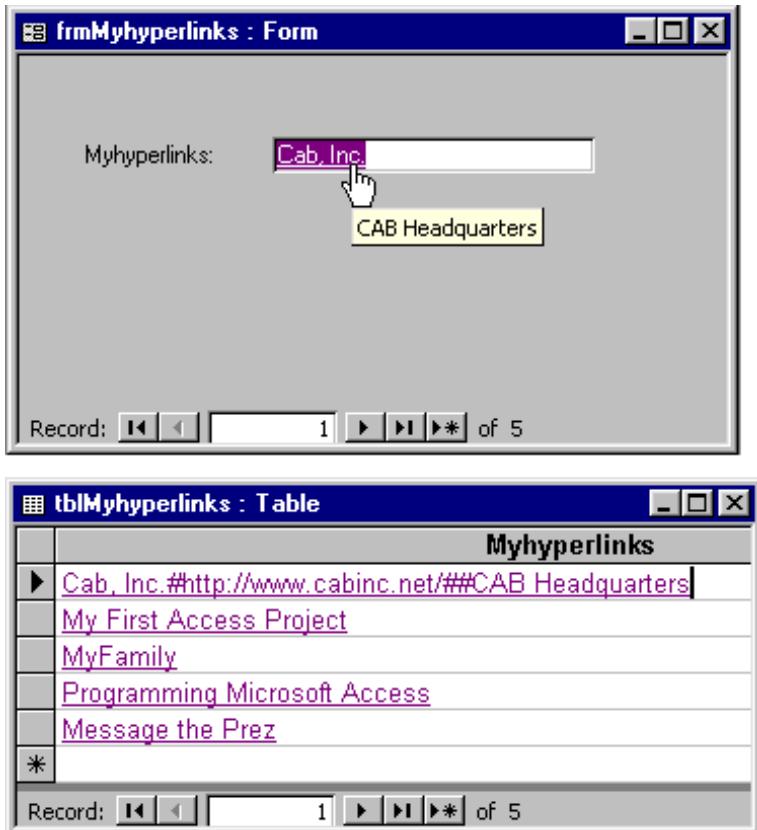
אחד הדפים הפופולריים ביותר באתר Web רבים הוא דף מועדפים (Favorites) שהוא טופס Access המציג אל אתרים בעלי ערך או מעניינים בעיני המחבר. באפשרות להוסיף דפים דומים בפתרונות Access מותאמים אישית. דף מועדפים יכול להכיל אתרים כמו קישורי דואר אלקטרוני או קישורי Web אל בית העסק שלך, וכן כתובות של אתרים המכילים תמיה טכנית במחirk נמוך או בחינוך.

תרשים 13.8 מציג דף מועדפים במתכונת של טופס Access, עם שני צירום ושתי תוויות. כל הארבעה משמשים כהיפר- קישורים. הציר הראשון הוא לוגו החברה של CAB. כאשר משתמש עם חיבור פעיל לאינטרנט לחוץ על הציר, Access מוביל אותו לאתר Web של החברה. הציר השני הוא הלוגו של Microsoft Access, שוביל את המשתמש אל אתר Microsoft Access בתוך האתר CAB. שני היפר- קישורים במתכונת תווית פותחים קישורי דואר אלקטרוני אל עבודות בחברה.



תרשים 13.8: דף מועדפים המכיל שני צירום ושתי תוויות, המשמשים כולם כהיפר- קישורים

תרשים 13.9 מציג טופס עם היפר- קישור מאוגד המבוסס על טבלה של היפר- קישורים. הטבלה מופיעה בתמונה שmotחאת לטופס בתרשים. תיאור המסקן, CAB Headquarters, מופיע עבור היפר- קישור הראשון. התצוגה התchapונה מראה את היפר- קישור האמור במצב עריכה, החושף את חלקו היפר- קישור. שם לב שיש טקסט תצוגה וכן חלק המכיל כתובות, אך אין כתובות משנה. היפר- קישור מסוימים בטקסט תיאור המסקן.



תרשים 13.9: טופס עם היפר- קישור מאוגד המציג תיאור מסך עבור היפר- קישור הראשון בטבלה; הטבלה חושפת את התchapון המלא של היפר- קישור הראשון

יצירת דפי גישה לנוטונים והשימוש בהם

דף גישה לנוטונים הם קבצי HTML מאוגדי-נתוניים. במודלים רבים, הם נראים ומתנהגים כמו הכלאה בין דוח Access לטופס Access, מכיוון שניתנו להשתמש בהם לדודר בין הרשומות במקור נתוניים. החיבור של הדף למסד נתונים מאפשר למשתמש להוסף, לעדכן או למחוק נתונים. בדרך כלל, אולם לא בהכרח, הנתונים באים מתוך אובייקט מסד נתונים בקובץ מסד הנתוניים הנוכחי. תוכל להשתמש בכלים **מיון** ו**קיבוץ**

(Sorting and Grouping) להציג נתונים באופן היררכי – לדוגמה, בדוחות עם כותרות קבוצות או מקטעים מרובים ומקוונים. **הסמל עמודיות** (Pages) שבחלון מסד הנתונים מייצג את אובייקטי דפי הגישה לנ נתונים המשויכים למסד נתונים. אולם Access אינו שומר דפים במסד הנתונים, אלא כקובץ DHTML במערכת הקבצים, לדוגמה, בתיקה בשרת הרשת הארגונית. **האוסף עמודיות** (Pages) שנמצא בקובץ מסד נתונים הוא אוסף של היפר-קישורים אל קבצי ה-DHTML. יצירת דפים נעשית במסגרת Access, ואילו המשתמשים יכולים לפתח אותם ב- Internet Explorer 5 או ב- Access 2000.

הערה:

אם משתמשים מרובים יפעלו דף באמצעות דפדף בתוכנות העבודה שוננות, יתכן שתיאלץ לשנות את ההגדירה **Data Source** בקבצי ה-JET, DHTML, Uniform (Uniform Naming Convention) במקום זאת.



יצירת דף גישה לנ נתונים

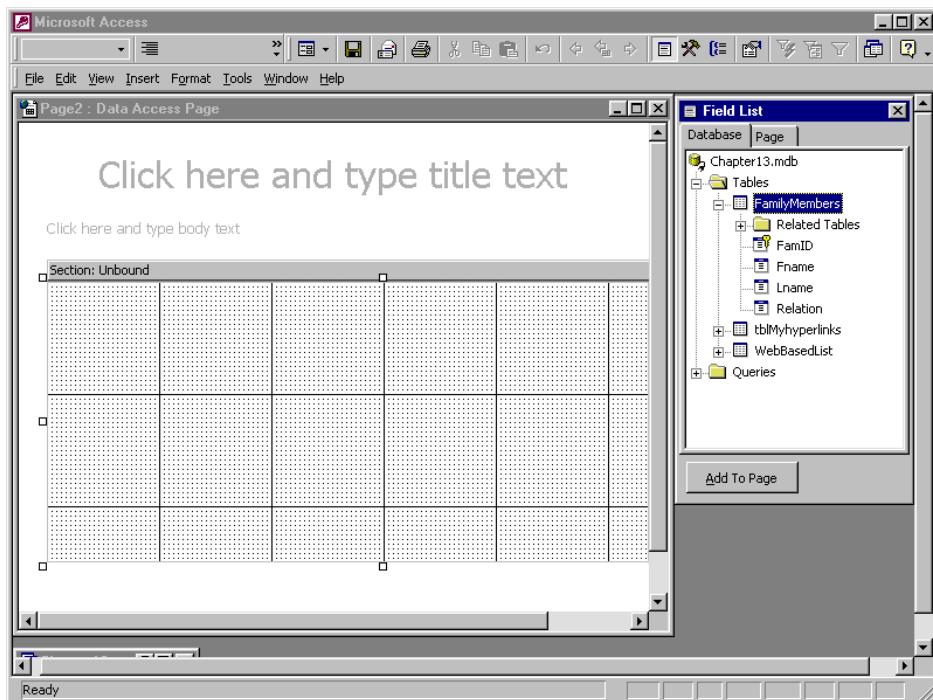
Access 2000 מכיל אשפים לייצרת דפי גישה לנ נתונים על בסיס טבלי או במתכונת של עמודות. אפשר גם לבסס דף חדש על דף Web קיים. עם זאת, רבים יעדיפו ליצור דף גישה לנ נתונים על ידי פיתחת דף ריק בתצוגת **עיצוב** (Design) ואכלולו בפקדים מתוך תיבת הדיו-שיכון **רשימת שדות** (Field List) (המודגת בתרשימים 13.10).

כדי לאכלא את הדף, תוכל לפתח את הצומת **טבלאות** (Tables) או **שאלילות** (Queries) ולהשוו את שמות הטבלאות בחיבור הפעיל. אם תיתקל בטבלה שתרצה לבחור ממנה ערכים לדף החדש, תוכל להרחיב את הטבלה ולסקור את השדות הנפרדים. בשלב זה תוכל לבחור שדות על ידי גירתם ובחירהם בדף החדש. לחילוףין, תוכל לאכלא את הדף בכל השדות של טבלה נתונה על ידי גיררת שם הטבלה אל הדף. לאחר מכן תוכל לסדר את השדות בעיצובה של טבלת עמודות או טבלת ציר. אותן פעולות תוכל לבצע גם בשאלילות.

הערה:

העיצוב של דוח טבלת ציר (Pivot Table) שמקורו ב- Excel, הוא העיצוב טבלי מיוחד שמאפשר ייצור של סיכומים מקובצים וניתוחים של הנתונים במקור רשומות. נראה שטבליות ציר כמשמעות חזית פופולרי לעובדה עם קבויות נתונים רב-ממדיות, מובטח.





תרשים 13.10: השתמש בתיבת הדו-שיח **רשימת שדות** כדי לאכלס דף גישה לנוטונים ריק בפקדים המאוגדים אל שדות של מסד נתונים

כדי לשנות את החיבור הפעיל, לחץ לחיצת ימנית על מחבר מסד הנתונים בפינה השמאלית-علילונה של תיבת הדו-שיח **רשימת שדות**, ובחר באפשרות **התקשרות** (Connection) מתוך תפריט הקיצור. בתיבת הדו-שיח **Data Link Properties** (מאפייני קישור נתונים) שתופיע על המסך, בחר במנהל חיבור OLE DB ובמסד נתונים המתאים למנהל התקן זה. כאשר תסגור את תיבת הדו-שיח, תציג תיבת הדו-שיח **רשימת שדות** את הטבלאות והשאליות מסד הנתונים החדש.

יצירת דף טורי פשוט והשימוש בו

תרשים 13.11 מציג דף גישה לנוטונים פשוט המשמש כטופס עבור הטבלה FamilyMembers שנמצאת בקובץ מסד נתונים. הטופס מציג ארבעה שדות. הפקד שמתוחת לשדות תומך בניווט ובפונקציות נוספות. לחיצה על שני לחצני הניווט בשני צידי חלון מצין הרשומות תגרום למעבר קדימה ואחוריה בין הרשומות בערכת הרשומות שעלייה מבוסס הדף.

Page1

Form for Family Members

FamID	<input type="text" value="1"/>
Fname	<input type="text" value="Rick"/>
Lname	<input type="text" value="Dobson"/>
Relation	<input type="text" value="Me"/>

FamilyMembers 1 of 5

תרשים 13.11: דף גישה לנוטרים המשמש כתופס; המשתמש יכול לדפסף, להוסיף, לערוך, למיין ולסנן רשומות בעזרת פקד הניות שמתוח לשדות ששת החלצנים הראשונים מימיין למציאן מקור ומספר הרשומה הם :

- ↳ לחץ Next Record (הרשומה הבאה). מעבר לרשומה הבאה.
- ↳ לחץ Last Record (הרשומה האخונה). מעבר לרשומה האחונה.
- ↳ לחץ New Record (רשומה חדשה). הוספה רשותה חדשה.
- ↳ לחץ Delete Record (מחיקת רשותה). מחיקת הרשותה הנוכחיות.
- ↳ לחץ Save Record (שמירת רשותה). שמירת השינויים שנערכו ברשותה הנוכחיות.
- ↳ לחץ Undo Record (ביטול שינויים). ביטול כל השינויים שבוצעו ברשותה הנוכחיות ולא נשמרו.

אם הזנת רשותה שאין זוקק לה עוד, מחק אותה בעזרת החלוץ Delete Record. כמו כן תוכל לעורך רשומות על ידי הקלדה על גבי התוכן שהן מכילות, הוספת תכנים או מחיקת תכנים קיימים. כדי לשמר שינויים שביצעת, השתמש בלחץ Save Record.

אם אין מועוני לשומר אותם, לחץ על Undo Record.

שאר הלחצנים בפקד הניווט (להוציא לחץ העזרה שבקצת הימני) מיועדים למילוי וסינון:

↳ לחצני **Sort Ascending** (מיון בסדר עולה) ו-**Sort Descending** (מיון בסדר יורד). ציינו את כיוון המילוי. כדי למיין לפי ערכים בשדה, לחץ בשדה זה ברשומה כלשהי, ולאחר מכן לחץ על אחד מלחצני המילוי.

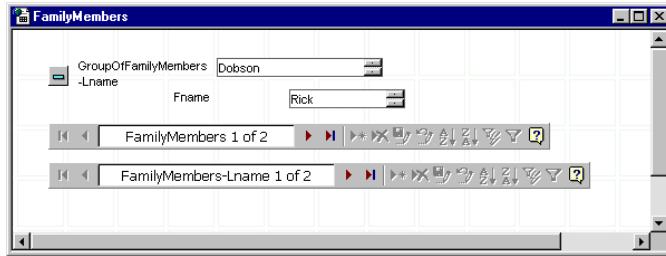
↳ לחצן **Filter by Selection** (סינון לפי בחירה). מhil מסנן המבוסס על הערך הנוכחי בשדה. לחץ בתוך השדה כדי לבחור ערך לסינון (לדוגמא, השדה Lname עם הערך Dobson), ולאחר מכן לחץ על הלחצן **Filter by Selection**. בעקבות פעולה זאת התצוגה תשתנה ותציג רק את הרשומות המתאימות לערך הסינון שנבחר. לדוגמא, אם בחרת בערך Dobson עבור השדה Lname בתרשימים 13.11, מצינו הרשומות ישתנה מ- FamilyMembers 1 of 5 ל- FamilyMembers 1 of 2.

↳ לחצן **Toggle Filter** (סינון דו-מצבי). מhil או מבטל את החלת המASN. כל התפקודיות הזאת עומדת לרשות המשתמש מתוך Access עצמו, כמו גם מתוך דףדף שנמצא בראשת ארגונית (אינטריא-נט). התצוגה המוצלחת ביותר של דפי גישה לנתונים, כולל תפקודיות של כל התכונות, היא באמצעות Internet Explorer 5. Internet Explorer, רידידה בתפקודיות בדרגות שונות מלואה את השימוש ב- Internet Explorer. משתמשים המפעילים דפדף אחרים בלבד Internet Explorer ייאלצו להציג גם פקדי ActiveX כדי לקרוא דפי גישה לנ נתונים ולטפל בהם.

קיבוץ רשומות

אחת התכונות החשובות של דפי גישה לנ נתונים היא יכולת לקבץ רשומות ולהרחיב קבוצה באופן מותנה, כך שהערכים הנפרדים שהיא מכילה ייראו. תרשימים 13.12 מציג יכולת זו. הדף הנראה בתרשימים מקבץ רשומות מהטבלה Lname לפי השדה Lname. כל רשומות הטבלה ששם המשפחה בהן זהה, מתקבצות לקבוצה אחת. כאשר המשתמש לוחץ על לחץ ההרחבה האפור (הוא משתנה מ- [-] ל- [+]), מופיע פקד ניוט שני, מוקדם, ומאפשר דפדף בין הרשומות בעלות אותו שם משפחה. לחיצה על פקד הניווט החיצוני תגרום לסגירת התצוגה המורחבת ותעביר אל הרשומה הראשונה עם שם המשפחה הבא.

ניתן להציג יותר מרשותה אחת בדף. השתמש בהגדלה **גודל דף נתונים** (Data Page Size) שבחולון **מיון וקיבוץ** (Sorting and Grouping) כדי לציין כמה רשומות להציג בכל פעם (ראה חלון תחתון בתרשימים 13.13). החולון מאפשר לקבוע את גודל הדף באופן בלתי תלוי עבור שדה הקיבוץ כמו גם עבור השדות שבתוך קבוצה. השתמש בחולון **מיון וקיבוץ** לדפי גישה לנ נתונים כדי לקבוע אם יהיו קבוצות מקטיע כותרתعلינה וכותרת תחתונה, כפי שאתה משתמש בחולון המקביל כשםדובר בדוחות. השתמש בהגדלה **קיבוץ לפי** (Group By) כדי לציין מרוחקים, טווחי ערכים וקידומות לשדות קיבוץ.



תרשים 13.12: ניתן לקבץ ולבצע הרחבה מותנית של רשומות בקבוצות, בדף גישה נתוניים

תרשים 13.13: בתצוגת **עיצוב**, תוכל להוסיף פקדי הרחבה כדי לאפשר למשתמשים לפעול עם הדפים, וביצע בהם הרחבה מותנית כדי לראות את הרשומות בתוך קבוצה נתונה

החלון העליון בתרשימים 13.13 מראה את התצוגה **עיצוב** (Design) של הדף מתרשים 13.12. שים לב, שהוא מכיל שני פקדי ניווט. באפשרות להציג פונקציית קיבוץ לשדה על ידי בחירת השדה ולהציגו על הלחצן **קדם** (Promote) (החץ המצביע שמאליה בסרגל הכלים **עיצוב עמוד** (Page Design)). כאשר מקור הרשומות של דף נגורז מקשר

gomelin shel yichid-lrbim, nitnu lehshemsh b'kli myon v'kiyavz cdi lekvez at cel hashdotot mazd ha'yichidi shel ksher hogomelin be'makom lepi hashdotot b'zad hirbim'. B'makraha zeh, Access mimin v'mekvez at rishomot lepi ha'mpatch ha'rashi shel zed ha'yichidi b'ksher hogomelin.

רכibi Web shel Dafi Office 2000 gisha l'natonim

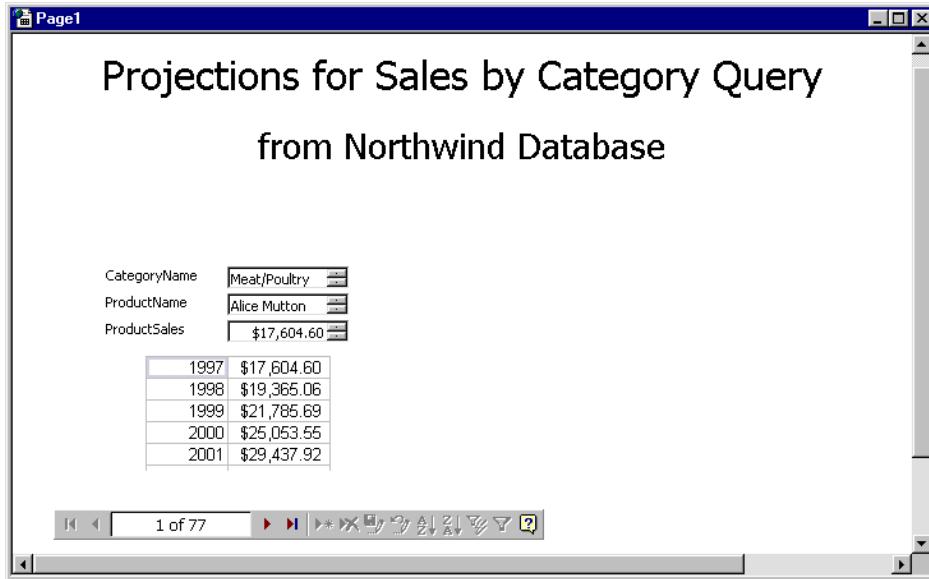
Office 2000 mesuok belooyit shelosha rcbi Web (b'mahdorot Standard, Professional v'Developer): Gilyon alktroni, Tbhla v'resimat Tbhlat zir. Hashemsh bercbim ala lehshlma v'herchba shel tpekodiot masd ha'natonim ha'kollola b'dafi gisha l'natonim. Am ba'atir masoisim ish resiun ha'mtir ha'faza br'shat aragonit, manhal ha'atir v'kol lag'dir at tzorot ha'dpdafim b'masgrot ha'rshonim kz shiutenu ba'open automati v'igdir at tzorot rcbi-h Web shel Office b'pum ha'rashona sh'mozman lhem letuon dz b'uzrat rcbi Microsoft Office v'nosaf ul kabiyut ha'tzora shel rcbi Web shel Office ba'-(2000 Resource Kit).

Dogmat Gilyon alktroni shel Office

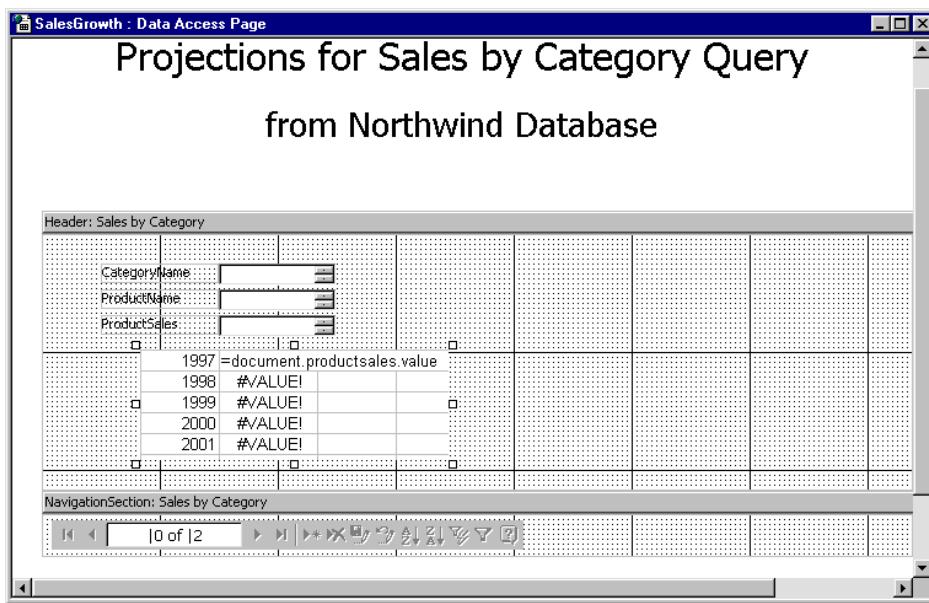
Tershim 13.14 matig at achd ha'simiosim shel rcbi gilyon alktroni b'daf gisha l'natonim. Ha'pkdim b'daf matigim at ha'shotot ProductName, CategoryName v-ProductSales shel sha'ila'ata Sales By Category b'masd ha'natonim Northwind. Sha'ila'ata zot machshabat at s'k-kol ha'mciorot lifi mo'azar b'shנת 1997. Sha'ila'ata matiga at zihovi ha'katgoria v'ha'sm ubor kl mo'azar, l'zad ha'natoni ha'mciorot.

Ha'dogma batreshim 13.14 merachiba at sha'ila'ata ha'basiyah ul ydi yizirat thzitot ha'mciorot m'shat 1998 v'ud s'nat 2001, kol. Tachila, rrcib mutaki' at ha'uruk ha'noch'i shel ha'pkd ProductSales m'hadz'al gilyon alktroni. La'achr mcn hoa mohash s'draha ha'dragtiet shel shiurri ha'gadol b'mciorot ha'chal m'shat 1997. Shiur zot magdil at ha'mciorot menha le'shna. L'mrotot shiurri ha'gadol ha'chal b'-1997 ainah zeha la'khol ha'mciorim. Le'sof, cdi ha'g'on ul ha'noschot m'pni zak, nitnu la'nuol galionot alktroniim n'barim kz she'mastamim la'yo'el le'shotot b'shogeg.

Tershim 13.15 meracha at ha'tzoga **uyzob** shel dz ha'gisha l'natonim batreshim 13.14. Shim lab, shmatchah le'pkd ProductSales mo'piu gilyon alktroni. Ba'afshorotk l'hosif rcbi gilyon alktroni shel 2000 Office b'uzrat ha'pkodot **ha'osfa** (Insert), **gilyon alktroni shel Office Spreadsheet**, v'lehata'aim at ha'rcib le'shimosh batuk ha'iyusom sh'l ydi la'chiza y'mnit ul ha'rcib v'bachira ba'afshrot **arozz' klimim shel ma'piyinim** (Property). B'gilyon alktroni batreshim 13.15 mosterim sergal ha'klimim, ps ha'cotarta, cotrotot ha'umodot v'cotrotot ha'shorot. B'nosaf, nitnu urk False lag'drotot ps ha'gillah ha'opki v'ha'anci shel ha'rcib.



תרשים 13.14: רכיב הgiיון האלקטרוני בדף הגישה לנוטונים משתמש בנוטוני מכירה לשנת 1997 מתוך ממד הנתונים כדי לגבות תחזית מכירות עד 2001, כולל



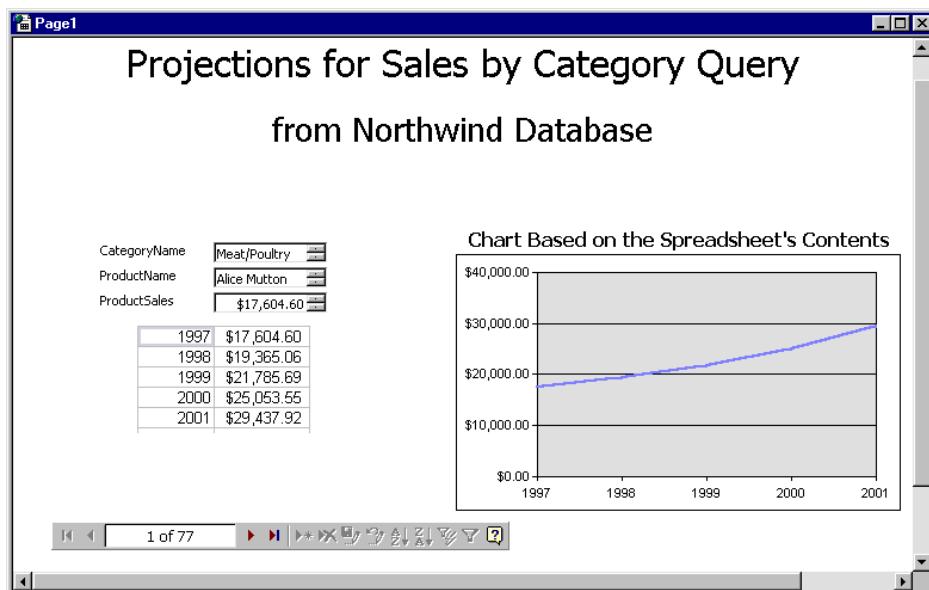
תרשים 13.15: תצוגת עיצוב של דף הגישה לנוטונים מתרשימים 13.14

תא נתוני המכירות העליון בגלילון האלקטרוני מכיל נוסחה המתעדכנת עם כל רשותה חדשה בדף הגישה לנוטונים : =document.productsales.value . האיבר Productsales מפנה לדף גישה לנוטונים. האיבר Productsales מפנה לפקץ מסויים בדף. ולבסוף, המאפיין value מפנה אל הערך הנוכחי של הפקץ בדף הגישה לנוטונים. ערך זה משתנה רק אם המשמש עבור לרשותה חדשה במקור הרשומות שברקע הדף.

רישומי המכירות לשנים 1998 עד 2001, כולל, מחושבים - #-#VALUE! בתצוגה **יצוב**. זכור, שלפי תרשימים 13.14 יש לתאים אלה ערכים חוקיים בתצוגת **עמוד** (Page). דבר זה נובע מהנוסחה שבתא העליוןון האלקטרוני משתמשת בערך הנוכחי של productsales בדף הגישה לנוטונים.

דוגמת תרשימים של Office

הדוגמה המסיימת פרק זה בנזיה על הדוגמה הקודמת, ועיקרא תוספת רכיב תרשימים של Office 2000. רכיב זה מציג באמצעות תרשימים את הערכים שבגילוון האלקטרוני. התרשימים מציג באופן גרפי את הגידול במכירות לאורך זמן לכל מוצר. תרשימים 13.16 מציג מוצר מסוים, בלווית התוצאות של הגלילון האלקטרוני ותיאור גרפי של הנוטונים באותו הימני של הדף. התרשימים מתעדכנים באופן דינמי כל פעם שהערכים בגילוון האלקטרוני משתנים.



תרשימים 13.16: רכיב התרשימים של Office 2000 בדף גישה לנוטונים מקבל ערכים מרכיב גילוון אלקטרוני, שתחזית המכירות בו משתנה כאשר המשתמש עbor לרשותה אחרת כדי להוסיף תרשימים בדף גישה לנוטונים, בחר את האזור הרצוי בדף, ולאחר מכן בחר הוסףה (Insert), **תרשימים של Office Chart** (Office Chart). הפקודה מפעילה אשף שמנחה את המשתמש לאורך תהליך העיצוב של תרשימים. תוכל לבחור את סוג התרשימים ומוקור

הרשומות שלו, ולהגדיר אילו ערכים להציג בו. לאחר סיוםת את השימוש באשף, **Property** לחץ לחייב ימנית בתרשים ובחר באפשרות **ארוג כלים של מאפיינים** (Toolbox) כדי לעורוך את הפריטים שבחורת או לציין אפשרות אחרות שלא הוצגו לבחירתך באופן מפורש בעת היצירה.

דוגמת רשימה של טבלת ציר

הפקד של רשימה טבלת ציר מאפשר למשתמש למילוי, לפחות, חלק לرمות ולטפל בתונונים. כמו כן הוא מסוגל לפעול עם נתונים של יותר ספקיים בהשוואה לדף גישה לנוטונים רגיל, או רכיבי Web אחרים של Office 2000. מקורות הנתונים של פקץ זה יכולים להיות גליאון עבודה, מסד נתונים, וקוביות נתונים רב-מידית.

תרשים 13.17 מציג פקץ של רשימה טבלת ציר בדף גישה לנוטונים. הדף מבוסס על הטבלה Orders שבמסד הנתונים NorthwindCS. הפקץ מונה הזמינות לפי ShipCountry, EmployeeID ו- CustomerID. לחץ על הפקץ הרחב (Expand) הסמוך לשדות CustomerID ו- EmployeeID כדי להרחיב את הטבלה במסגרת אותה עמודה או שורה. בנוסף, תוכל להציג באופן קבוצה משנה של הטבלה על ידיפתיחה רשימת האלמנטים המכילה את כל המדינות, הלקוחות והעובדים, ובחירה באחד או בכמה מהם מכל קבוצה. תיבת כלים המופיעה בראש רשימה טבלת הציר מכילה עוד אפשרויות ניתוח, כמו למשל סינון ומילוי.

The screenshot shows a Microsoft Access window titled "Page2". Inside, a PivotTable is displayed with the heading "This PivotTable is based on NorthwindCS". The table structure is as follows:

ShipCountry	EmployeeID	CustomerID					
		Count of OrderID	Count of OrderID	Count of OrderID	Count of OrderID	Count of OrderID	Count of OrderID
ALFKI		2		1	2		1
ANATR				2		1	
ANTON		1		3		1	
AROUT		3		2	4		1
BERGS		4	1	6	1	2	
BLAUS				1	1		1
BLONP			1	2	3	1	2
BOLID						2	
BONAP		3	1	3	4	1	
BOTTM		2	2	4	2		2
BSBEV		1	1	2	2		2

תרשים 13.17: רשימה טבלת ציר בדף גישה לנוטונים המציג נתונים מתוך מסד הנתונים NorthwindCS

נושאי תכונות הקשרים בדף גישה לנוטונים

בעזרת דפי גישה לנוטונים, תוכל לתוכנת פתרונות במספר רמות שונות. דפי גישה לנוטונים יכולים גם כמארכחים לרכיבי Web של Office. הן דפי גישה לנוטונים והן רכיבי Web של Office מכילים TABיות אובייקט שתוכל לנצל לצורך פיתוח פתרונות באטען קוד. עוד ניתן לתוכנת פתרונות ב- VBA או בשפת scripting ל- Web, כמו למשל .VBScript.

סעיף זה בודק פתרונות המבוססים על VBA, על האוסף AllDataAccessPages ועל האובייקט AllForms. האוסף AllDataAccessPage פועל כמו האוספים AllReports ו- AccessObject. אובייקטים אלה זמינים בין אם יש דף גישה לנוטונים פתוח ובין אם לאו. האוסף מאפשר לך לאתר במלואה את כל קבוצת דפי הגישה לנוטונים הקשורות בפרויקט של ממד נתונים. השירה הקרה הבא ב- VBA רושמת את כל דפי הגישה לנוטונים בפרויקט נתון, ומציינת ליד כל אחד אם הוא פתוח או לא.

```
Sub listPages()
Dim myPage As AccessObject
    For Each myPage In Application.CurrentProject.AllDataAccessPages
        Debug.Print myPage.Name & IIf(myPage.IsLoaded, _
            " is loaded.", " is not loaded.")
    Next myPage
End Sub
```

לماפיין FullName באובייקט AccessObject שבאוסף AllDataAccessPages נודעת משמעות מיוחדת. זכור, שדפים אינם נשמרים כאובייקטים בקובץ מסד הנתונים, אלא הם קבצי DHTML נפרדים. מיקומם של הדפים יכול להיות בכל נקודה-שהיא בראש המיקומית. המאפיין FullName מציין את הנטייה ואת שם הקובץ עבור דף גישה לנוטונים. השירה הקרה רושמת את כל הדפים בפרויקט, ופרטת אותם לפי המאפיינים Name ו- FullName. ערך המאפיין Name הוא שם מקוצר לדף הגישה לנוטונים המופיע בחלון מסד הנתונים.

```
Sub whereArePages()
Dim myPage As AccessObject
Dim obj As Object
    Set obj = Application.CurrentProject
    For Each myPage In obj.AllDataAccessPages
        Debug.Print "The link "; myPage.Name & _
            " points at " & myPage.FullName & "."
    Next myPage
End Sub
```

השיטה היחידה של האובייקט `DataAccessPage` היא `ApplyTheme`. השתמש בשיטה זאת כדי ליצור אוטומציה בהחלפת ערכת נושא של Office על הדפים שאיברי האוסף `AllDataAccessPages` מצביעים אליהם. שיטה זאת תפעל באופן תקין רק אם הדף פתוח במצב עיצוב. שתי השגרות הבאות מחללות ערכת נושא על כל דפי הגישה לנוטונים בפרויקט, בין אם הדפים פתוחים ובין אם לאו, ובין אם הם פתוחים במצב עיצוב, ובין אם לאו. השגרות גם משוחזרות כל דף גישה לנוטונים ומחזירות אותו למצב הפתיחה והתצוגה שהיו לו לפני החלת ערכת הנושא. אם אין מרווחה מערכת הנושא שנבחרה, הפעל מחדש את השירה `callSetTheme` עם הארגומנט `.Blank`.

```

Sub callSetTheme()
' Test with Artsy or Blends.
' Clear with Blank.
    setTheme "Artsy"
End Sub

Sub setTheme(ThemeName As String)
Dim myPage As AccessObject
Dim obj As Object
Dim blnCloseit As Boolean
Dim blnMakePageView As Boolean
'Loop through all DataAccessPages.
    Set obj = Application.CurrentProject
    For Each myPage In obj.AllDataAccessPages

        ' Get Page open in Design view.
        If myPage.IsLoaded = False Then
            DoCmd.OpenDataAccessPage myPage.Name, acDataAccessPageDesign
            blnCloseit = True
        Else
            If DataAccessPages(myPage.Name).CurrentView <> _
                acDataAccessPageDesign Then
                DoCmd.Close acDataAccessPage, myPage.Name
                DoCmd.OpenDataAccessPage myPage.Name, acDataAccessPageDesign
                blnMakePageView = True
            End If
        End If

        ' Apply Theme.
        DataAccessPages(myPage.Name).ApplyTheme ThemeName
        DoCmd.Save acDataAccessPage, myPage.Name
    
```

```
' If necessary, restore page.  
If bInCloseit = True Then  
    DoCmd.Close acDataAccessPage, myPage.Name  
    bInCloseit = False  
ElseIf bInMakePageView = True Then  
    DoCmd.Close acDataAccessPage, myPage.Name  
    DoCmd.OpenDataAccessPage myPage.Name, acDataAccessPageBrowse  
End If  
Next myPage  
End Sub
```


מהדורת Office 2000 לפתחים (ODE)

מהדורת Office 2000 לפתחים (ODE) כוללת רכיבים חשובים רבים לפיתוח יישומים בסביבת Microsoft Access, כולל אפשרויות התקנה והטמעה רב-תכליתיות. מהדורת ODE שוקה במקור עם Office 97 בתור התפתחות טבעית של Access SDK (ערכת Microsoft Access 97). מפתחים מנוטים ב-Access יכולו ערכו של ODE בתור משאבי הפיתוח של Access. יכולות יכירות בערךו של ODE עשויה כברת דרך אמצעי לגישה אל מנגנון זמן ריצה ואשף ההתקנה של Access. ODE מושפעים ממנהן זמן ריצה וריצתם יאפשרו פיתוח יישומים או מושפרים במידה רבה:

- ﴿ גרסת זמן ריצה של Microsoft Data Engine (MSDE)﴾
 - ﴿ אשף הארזזה וההטמעה﴾
 - ﴿ כלים המאפשרים אינטגרציה נתונים (data binding) בין Microsoft Forms המשמשים יחד עם יישומי Office נוספים (במקום Access Forms)﴾
 - ﴿ ספרן הקוד (Code Librarian) לאחסון וניהול של תגזיiri קוד ושגרות מלאות תיעוד מודפס ומקוון בהיקף מלא לפתחי יישומים לסביבת Office 2000﴾
- פרק זה מציג סקירה קצרה של קבוצת הרכיבים של ODE ובוחן נושאים בעלי עניין מיוחד לפתחי Access (החתמה דיגיטלית היא נושא שגם מפתחים אחרים ימצאו בו עניין).

סקירה כללית על ODE

ה יתרונות של ODE באים לידי ביטוי בשלושה תחומיים : תפוקת Microsoft Visual Basic for Applications, גישה לנתונים ופיתוח וניהול. ODE מיועדת למילוני מפתחים מקצועיים שיצרים פתרונות בעורת Office, VBA וכלי גישה לנוטנים. חלק גדול מפתרונות אלה מנצלים את יכולות הגישה לנוטנים רבות העוצמה של Access מתוך יכולות של יישום אחר של Office.

ODE כוללת את מוצרי Microsoft הבאים :

- Microsoft Word 2000
- Microsoft Excel 2000
- Microsoft PowerPoint 2000
- Microsoft Access 2000
- Microsoft Outlook 2000
- Microsoft Publisher 2000
- Microsoft FrontPage 2000
- Microsoft PhotoDraw 2000
- Microsoft Small Business Tools

ODE כוללת תיעוד ודוגמאות שיסייעו לך לבניית פתרונות בתוך יישומים אלה וביניהם. VBA זמין בכל משפחת מוצרי Office, אך הקשי העיקרי בשימוש בישום חדש הוא לימוד מודל האובייקט. התיעוד המודפס של ODE כולל ע론 ובו תיאורים גרפיים של מודלי האובייקט של היישומים שנמנו לעיל (למעט PhotoDraw). ODE כוללת תיעוד של 24 מודלי אובייקט המיעדים לשימושים העיקריים, משותפים להם, מותאמים במיוחד לשירותי גישה לנוטנים או מיועדים לטכנולוגיות Web שימושיות בישומי Office.

כלי VBA לפরיון עבודה

כלי VBA לפריוון עבודה מסייעים להציג את התוכני כתיבת הקוד בכל יישומי Office שתומכים ב-VBA. מדובר במיגון כלים רחב, החל בערך מחרוזות פשוט לבניית משבטי SQL, וכלה במעצב התוספות (Add-In Designer) של COM Component Object (COM), שמאפשר את הקמת ספריות הדינמי (DLL) בצורת תוספות Model, בשני יישומי Office או יותר.

מעצב התוספות של COM (COM Add-In Designer) מספק גישה מושלבת לבנייה ושימוש בתוספות בין כל יישומי Office. גרסאות קודמות של Office חייבו גישות שונות לבנייה, אחסון והפעלה של תוספות.

ספרן הקוד (Code Librarian) הוא מסד נתונים המאפשר לשתף קוד, פונקציות ומודולים בין חברי צוות הפיתוח. איש פיתוח בוודא יכול אף הוא לנצל אותו לאחזר קוד שכתב בעבר. באפשרות להוסיף בעצמך קוד מותאם אישית אל מסד הנתונים ולהפץ אותו באמצעות מילת מפתח וקריטריונים נוספים. ספרן הקוד מספק ערכת קוד בסיסית עבור מטלות תכנות מקובלות.

פרשן הקוד של VBA (VBA Code Commentator) ו**מטפל השגיאות של VBA** (VBA Error Handler) מעדכנים בעצם את המודולים כדי לבצע משימות רגילות חינניות בפעולות פיתוח מותאמות אישית רבות.פרשן הקוד מוסיף העורנות, כמו שם איש הפיתוח, התאריך והפרמטרים שאיש הפיתוח צריך לשלוח לשגורות. הוא מרכיב מידע זה מתוך תבנית הניתנת להטאה אישית. התבנית מאפשרת לצוות פיתוח הפועל בארגון גדול לשנות את התהילה הבסיסי כך שייענה על צורכייהם המיוחדים. מטפל השגיאות בונה קוד מותוקן (standardized) לטיפול בשגיאות בכל שגרות היישום. ניתן להתאים גם את התבנית שלו באופן אישי. מעבר לכך, מטפל השגיאות מספק סביבת עבודה להוספת קוד טיפול בשגיאות שתכתב בעצמך.

עורץ המחרוזות של VBA (VBA string Editor) הוא תיבת דו-שיח לכתיבה, עריכה, העתקה והדבקה של משפטים SQL מותאמים אישית בקוד VBA. אורץ זה הוא כלי פשוט לביצוע שימוש ייחידה שלוללה להיות מוכבתת למדוי.

ODE מפשט גם את עבודות הפיתוח בצוות באמצעות Microsoft Visual SourceSafe, SRCIBI המסירה והקבלת שלו מאפשרים לחבר הוצאות לטפל במודול VBA אחד או יותר, בעוד שעמינתיו מטפלים בהיבטי אבטחת איכויות או ברכיבי שילוב המערכת של שאר חלקים היישום. Visual SourceSafe מאפשר גם להשוות בין גרסאות קוד VBA ולשזור גרסאות קודמות, כל זאת בצורה קלה ופושטה.

התוסף יבוא/יצוא קוד מרובה של VBA (VBA Multi-Code Import/Export) מאפשר ליבא קוד אל מודול ווגם ליצא אותו ממנו. טוונים רכיב זה בתור תוספת בוודדת, אך הוא כולל שני ממשקי משתמש: אחד לייבוא של קוד והآخر ליצוא של קוד.

התיעוד המודפס והמכוון של ODE מאפשר לשפר את מיומנויות הפיתוח ומתראר טכניקות לפיתוח יישומים המשלבות בין רכיבים. נוסף למודולי האובייקט המודפסים, התיעוד כולל את המדריך למתכנת VBA. במצורף לתיעוד המכוון תמצא את האוסף המינוחי של תקליטורי רשות הפיתוח של Microsoft MSDN – Microsoft Developer Network (Developer Network) ובו כל החומר המודפס ומסמכים נוספים; החומר כולל תמייה במנגנון חיפוש.

גישה לנוטונים

ODE כוללת אוסף כלים התומכים בפיתוח גישה לנוטונים. הדובדבן שעל הקצתה הוא, קרוב לוודאי, מנגנון הנתונים של Microsoft המוצע להפצה חזורת – MSDE (Microsoft Data Engine). באפשרות לשולח גירסה עצמאית של יישום מותאם אישית שכתבת, שתומכת ב-MSDE, מבלי לשלם תמלוגים על כך. הדבר מתאים במיוחד לגרסאות

ابتיפוס של יישומים שעשויים להפתח לישומי SQL רחבי היקף. חתימת המשאבים של MSDE קטנה בהרבה מזו של SQL Server, ולכן מומלץ להשתמש בו במקרים שדרישת המשאבים של SQL תובעניית מדי. ניתן לבנות גם טפסים ודוחות מותאמים אישית כנגד SQL Server בעזרת הרכיב החדש פרויקט Access (עיין בפרק 12 לקבלת מידע נוסף אודוות MSDE ופרויקטים של Access).

להבטחת תאימות מירבית לאחר, עליך לבנות פתרונות באמצעות Microsoft Access Run Time. חvíלה זו מזקמת את Access 2000 ואת Jet ללא מסך המשמש. היחסום המותאם אישית שככבותה הוא המשק החשוב היחיד. חvíלה זו, הפטורה מתשלים תמלוגים, מאפשרת לשולח פתרונות הפעילים ברמת היחידה הארגונית, כאשר דרישות המשאבים שלהם איןן גדולות במשך הזמן. לדוגמה, ניתן לנצל את Access Run Time כדי ליצור פתרון עבור מפעל המספק שירותי ציבוריים (כגון מים וחשמל) מפעיל שירותי ציבוריים מתקיימים במשך תקופה ארוכה יחסית ולעתים קרובות הם מתפקדים ברמה מיטבית או ברמה קרובה לה. דרישות המערכת בסביבה מסווג זה אינן נוטות לגדול משמעותית לאורך חיי היחסום.

פקד הנתונים של ADO (ADO Data Control) ומעצב סביבת הנתונים (Data Environment Designer) מאפשרים כליל גישה גרפיים למקורות נתונים ADO ו-ODBC. פקד הנתונים של ADO מוכר למי שהשתמש במעצב חיבור משתמש ActiveX (ActiveX User Connection Designer) כדי ליצור אובייקטי נתונים מרווחים בזמן העיצוב. פקד הנתונים של ADO מאפשר יצירת חיבור למקור נתונים אחד בזמן נתון ובאפשרותו לנוטר לרשומה הבאה, הקודמת, האחונה והראשונה בערכת הרשומות של מקור הנתונים. מעצב סביבת הנתונים מאפשר ליצור אובייקטים רבים מהסוג Connection, שכל אחד מהם כולל אובייקטים רבים מהסוג Command עם ערכות רשומות מתאימות. המעצב מאפשר לגורור ולשחרר שדות מאובייקט מסווג ב- Microsoft Forms וגם במעצב דוחות נתונים (Data Report Designer).

כליל גישה גרפיים אלה אינם מבטלים כמעט השימוש בטכניקות התכניות הידניות של ADO. תוכל עדיין להשתמש בקוד של ADO כדי לבנות חיבור גישה לנתונים אל מקורות נתונים מוכרים ולטפל בהם מותוך יישומי Office אחרים (במהשך הפרק נציג דוגמה לימוש מושלמת זו באמצעות הנתונים Excel).

מעצב דוחות הנתונים ב-VBA ומעצב סביבת הנתונים משמשים יחד לבניית דוחות מותאמים אישית ללא כתיבת קוד כלשהו. ניתן לייצא דוחות שנוצרו בדרך זו בתור מסמכי HTML (Hypertext Markup Language). תוכל ליצור דוחות באמצעות גرافים, אך השימוש בקוד מעניק לך שליטה גדולה יותר בעיצוב, הדפסה, הצגה לפני הדפסה ושמירה של הדוחות ההיררכיים.

הטמעה וניהול

אשר האריזה וההטמעה (Package and Deployment Wizard) של Office 2000 מהווה שיפור משמעותי לעומת אשפי ההתקנה וההפעלה בגרסאות קודמות. כל פעולות האשף מונעות באמצעות תפריטים. באפשרות נצלו להפקת חבילת התקינה מקצועית

שפועלת בצורה דומה לזו של 2000 Office. אחד הרכיבים המלהיבים ביותר שלו היא האפשרות לפרוס פתרונות מותאמים אישית באמצעות האינטרנט, כך שלקוחות מרוחקים יכולים להתקין את הפתרונות המותאמים אישית שתשלח אליהם ללא צורך בדיסקט או בתקליטור. ניתן אף להפוך את פעולות האזיה וההטמעה לאוטומטיות באמצעות scripts מותאמים אישית שמבלטים את האפשרות לטעת בבחירה מסci האשף.

באפשרות נצל את האשף כדי להפיץ פתרונות המבוססים על MSDE או Run Time Environment לשימוש חוץ. יחד עם זאת, אליך להתעלם ממתן פתרונות המנצלים את מלאו רכיבי הפעולה של גרסה 2000 Access. אחת הסיבות לכך הן נוטים להשתמש בפתרונות מותאמים אישית הרבה יותר מאשר הבחירה של Access. ל��וחות אחדים רוצים לקחת חלק בתחזוקת הממערכות שלהם, כדי להקטין את עלות הבעלות הכלולית.

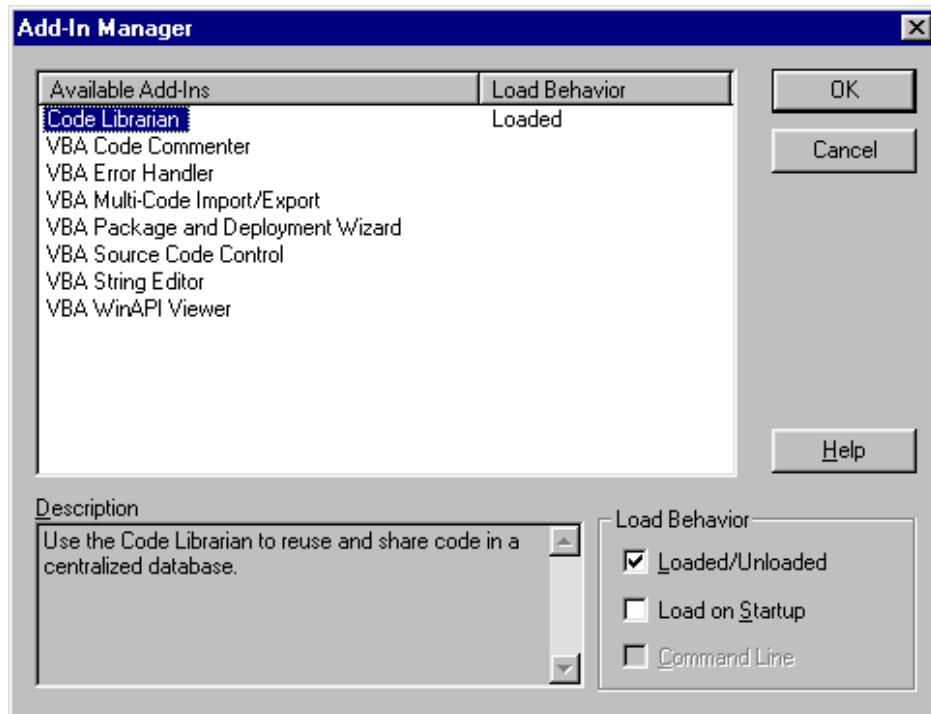
מנהל השכפול של Microsoft Replication Manager, משיק משתמש גרפי שמשווק עם ODE, מאפשר להציג ולנהל מסדי נתונים Jet משוכפלים בראש או באינטרנט. מנהל השכפול מmir מסד נתונים לתבנית בסיס לעיצוב, יוצר עותקים נוספים ומנהל קבוצת עותקים. באפשרות נצלו ניהול חילופי נתונים ומבנה נתונים בין חברי קבוצת העותקים המשוכפלים, באמצעות גרפים. מנהל השכפול מאפשר סינכרון בשלושה סגנונות: לפי בקשה, בחיבור הבא ובלוח זמינים וגיל. בנוסף, הוא תומך בחילופי נתונים בין מסדי הנתונים Jet ו-SQL Server (יעיין פרק 11 לקבלת מידע נוסף אודוות שכפול מסד נתונים והציג דוגמאות שימושיות כיצד להנלה שכפול).

כל פתרון מותאם אישית יכול לצאת נשר ממערכת עזרה או מסמכים כלשהם. משתמשים זוקרים לסמך שմסביר כיצד עליהם להגיב למערכת, ומפתחים צרכיים תרשימי זרימה וקיטוע קוד שמחישים את הקשר בין קוד ואובייקט מסדי נתונים חלק מהתמייה בפיתוח המערכת. ODE משווקת עם HTML Help Workshop, HTML, HTML Help Image Editor ו-HTML Help Image Editor. ככל זה מאפשר בשאר חלקי הספר,ナルכו באמצעות הכליל להציג את הסמן (cursor) – רכיב שנעדך מחייבות לכידת מסכים רבות. הוא תומך בלבידת מסך מלא וגם בלבידת חלונות שונים בתוך המסך, בהתאם למועד זמן (timer), לחיצה בעבר או הקשה במקלדת.

ספרן הקוד

כדי לעשות שימוש חוזר בקוד, יש צורך במקום כדי לאחסן אותו, אמצעי לחפש אותו ומנגנון שיאחזו אותו. **ספרן הקוד** (Code Librarian) המשווק עם ODE מאפשר לעשות את כל הפעולות האלו, ועוד יותר מהן. הפעלת **ספרן הקוד** היא פעולה דו-שלבית (בכך הוא דומה לתוספות מוכפלות אחרות של ODE). בפעם הראשונה בהפעלה (session) שתרצה להפעיל את ספרן הקוד, בחר באפשרות **Add-Ins** (תוספות) בחלון עורך VB של פרויקט ולאחר מכן לחץ לחיצה כפולה על **Code Librarian** בתיבת הדו-שיח

Load Behavior וnbsp;ונבחרת תיבת הסימון **Loaded/Unloaded** של הקבוצה **Loaded/Unloaded** לחץ לחיצה כפולה על תוספות כלשהן שברצונך לנצל במהלך הפעלה. באפשרות לבחור בתיבת הסימון **Load On Startup** כדי לגרום לתוספת להתחיל לפעול עצמה בעת פתיחת החלון **פרויקט** (Project). לאחר טיענת התוסף, באפשרות להפעיל מהתרפיט **Add-Ins**.

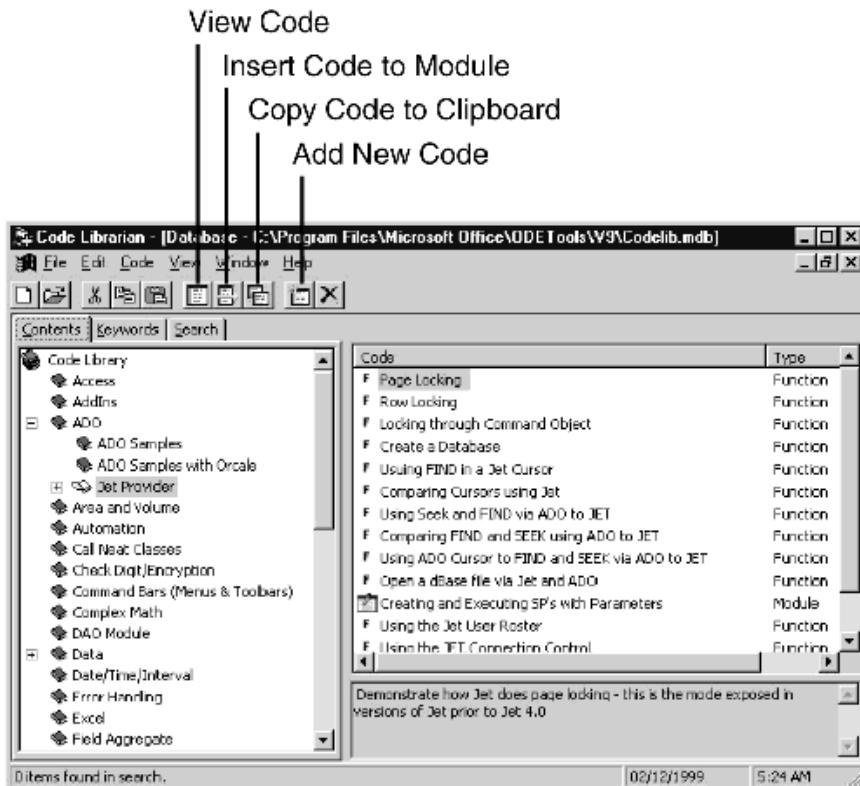


תרשים 14.1: תיבת הדו-שיח **Add-In Manager**

תרשים 14.2 מציג את חלון ספרון הקוד שמארגן את דוגמאות הקוד לפי קטגוריות. ניתן להרחיב את תחום הקטגוריות. באפשרותך לקשר תיאור עם מקטע קוד כדי שתוכל לzechoto. הכרטיסיה **Keywords** (מילות מפתח) מאפשרת לאחזר רשימת דוגמאות קוד הקשורות למילת מפתח. הכרטיסיה **Search** (חיפוש) מאפשרת לאחזר דוגמאות המכילות מילת מפתח הקשורה לדוגמה או המכילות מילת מפתח בטקסט המלא של הקוד.

סדרת הכלים **Code Librarian** כולל כלי ניהול טיפוסים. לדוגמה, באפשרותך להציג את הקוד שברקע פריט נבחר על ידי לחיצה על לחוץ **View Code** (הציג קוד) (ראה תרשים 14.2). לחוץ **Insert Code To Module** (הוסף קוד למודול) שמיימין מוסיף את הקוד לחלון הקוד במקומות הנוכחי שבו נמצא הסמן לחלון, וכך יש לוודא שהסמן אינו נמצא בשגירה כלשהי אחרת (אלא אם ברצונך להוסיף את הקטע דוקא אליו). באפשרותך גם לגורור ולשחרר דוגמת קוד מתוך הספרון אל קוד שאתה יוצר. לחוץ

העתק קוד אל הלוח) מאפשר להעתיק קוד ללוח. נוח לעשות זאת כשרוצים להעתיק קוד אל הפעלה (session) אחרת שאינה מוצגת במסך באותו רגע. כל שעליך לעשות הוא לעבור אל הפעלת היעד ולהדביק את תוכן הלוח בחלון הקוד שלו.



תרשים 14.2 :Code Librarian : מארגן את דוגמאות הקוד לפי קטגוריות, אך ניתן לחפש אותן גם לפי מילת מפתח, או בטקסט המלא

באפשרותך גם לשנות קוד קיים או להוסיף קוד חדש לספרון הקוד. כדי לשנות קוד, פתח את הספרון כדי להציגו, עורך את השינויים הדורשים ולחץ על **Apply** (החל). סגור את חלון ספרון הקוד כדי להחיל את השינויים. להוספת קוד חדש, לחץ על לחצן סרגל הכלים **Insert New Code** (הוסף קוד חדש); תיפתח תיבת דו-שיח ריקה של ספרון הקוד. הדבק את דוגמתה הקוד מהלוח אל תיבת הדו-שיח ותנו שם לדוגמה בתיבת הטקסט **Name** (שם). תיבת הטקסט **Name** נמצאת בדיקון מעל למקום בו הדבקת את הקוד. הקצה את הסוג (כגון קטע קוד, פונקציה, מודול או מודול מחלוקת) באמצעות הכרטistica **Description** (תיאור). הקלד תיאור של הדוגמה בתיבת הטקסט מתחתי לתיבת הרשימה הנפתחת **Sample Type** (סוג דוגמה). באפשרותך גם להקצות דוגמה לדוגמה ומילת מפתח אחת או יותר. לחץ על **OK** כדי לשמר את הדוגמה.

תרשים 14.3 מציג סדרת דוגמאות של ספק Jet של ADO. כפי שניתן לראות, הדוגמאות כוללות שני אוסףים נוספים. ספרן הקוד משוק כשהוא כולל אוסףים אלה ודוגמאות רבות נוספות בתור ערכה בסיסית. חלק מהדוגמאות עשויות להתיאים לשימוש בİŞושומים המותאמים אישית שתיצור, בתנאי שתעוררן בהם שינויים בהיקף כזה או אחר. אחת הדוגמאות שבנהרשים מחשבת את גודל הקובץ הנוכחי. הדוגמה השנייה מחשבת את גילו של אדם, בהתאם על תאריך הולדתו והתאריך הנוכחי.

```

Microsoft Visual Basic - Chapter 14 - [Module1 (Code)]
File Edit View Insert Debug Run Tools Add-Ins Window Help
[General] printAge
Function GetMDBSize() As Long
    ' Returns the current size of the current .mdb file in bytes.
    ' The .mdb file must be opened for shared access for the function to work.
    Dim Db As Database, f As Integer
    Set Db = CurrentDb()
    f = FreeFile
    Open Db.Name For Binary Shared As #f
    GetMDBSize = LOF(f)
    Close f
    Db.Close
End Function
Function Age(Bdate, DateToday) As Integer
    ' Returns the age in years between 2 dates.
    ' Doesn't handle negative date ranges i.e. Bdate > DateToday.
    If Month(DateToday) < Month(Bdate) Or
        (Month(DateToday) = Month(Bdate) And Day(DateToday) < Day(Bdate)) Then
        Age = Year(DateToday) - Year(Bdate) - 1
    Else
        Age = Year(DateToday) - Year(Bdate)
    End If
End Function

```

תרשים 14.3: הדוגמאות המוצגות כוללות ב- **Code Librarian**

פתרונות אריזה והטמעה

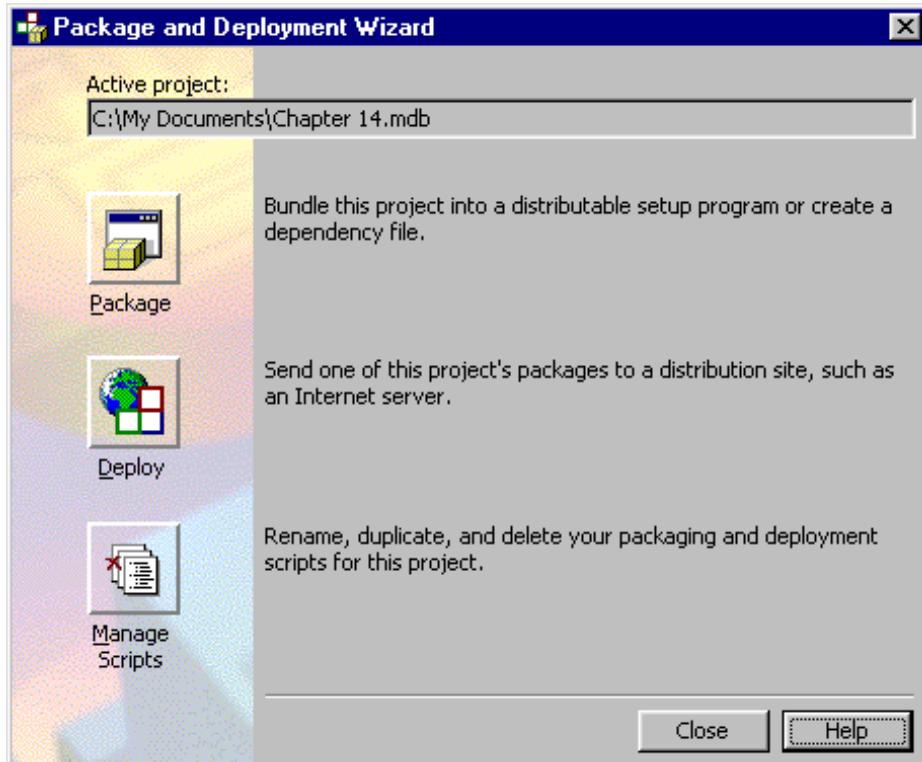
אשף האריזה וההטמעה מבצע שלוש פעולות:

◀ אריזה – יצירת קובץ דחוס אחד או יותר מסוג *.cab להפצה בדיסקט, בתקליטור, ברשת או באינטרנט.

◀ הטמעה – ניהול הפצת הקבצים הדחוסים אל אמצעי היעד. קיימים 23 מסכים שתומכים בפעולות האריזה וההטמעה. האשף יוצר בעצמו script שרשום את בחירותיך בעת ביצוע פעולה זו או אחרת.

↳ ניהול scripts – שמירה, שינוי שם, הסרת ושימוש חוזר ב-scripts. לאחר שמסמך המנסכים בעלי אפשרות בחירה רבות יכול להיות גדול מאוד, החזרת ה-scripts יכולת לחסוך זמן ניכר ולצמצם את הסיכון לשגיאות.

מפעילים את האשף בצוරה זהה לזו שפעילים את ספרן הקוד. טען את התוספת ולאחר מכן בחר אותה מתוך התפריט **Add-Ins** בחילון VBE. בטרם תפעיל את האשף, עליך לטעון את היישום המתאים אישית שברצונך לאירוע ולפROSS. תרשימים 14.4 מציג את המסך הראשון של האשף.



תרשים 14.4: המסך הראשון של אשף האריזה וההתמעה

הערה:



טעינת התוספת עלולה להימשך שניות אחדות. מד התקדמות שיוצג ידועה לך על מהלך הטעינה. אם תנסה להפעיל את התוספת טרם סיום טעינה, עלולה להתרחש שגיאת זמן ריצה.

באפשרותך להפץ מחדש פתרונות מותאמים אישית או רכיבים שתיצור באמצעות **Office**, למעט אלה שתצר עוזרת FrontPage ו-**OutLook**. התמיכה בפרויקט, שני יישומים אלה, מוגבלת לuprofile המשמש ולא לפרויקטים עצמאיים.

באפשרות להפץ פרויקטים עצמאיים עבור יישומי Office הנוטרים. תוכל גם לכלול קבצים נוספים החינויים לביצוע פרויקט, כגון תמונות מפתח סיביות, קבצי DLL, פקדי ActiveX וקבצי מסמכים אחרים של Office.

בין הפתרונות הפלוטלים יחד עם מסדי הנתונים Jet או MSDE Run Time או MSDE הניתן להפעלה חוזרת, מה שמאפשר לפתרונות המותאמים אישיתشتירות לביצוע פונקציות תחזקה של מסד נתונים במחשבים שלא התקינו בהם את Office 2000. אפשרותם לכלול בקבצי *.cab. גם את הקובץ Graph9.exe שתומך בתוצאות הנתונים הגרפיות.

הערה:



ישומים המנצלים את רכבי האינטרנט של Microsoft Office 2000 (Office Web Components) מחייבים רשיון תחנת עבודה להפעלת 2000 Office. יחד עם זאת, אפשרותם להפץ בצורה חופשית יישומים העשויים שימוש ברכבים אלה. כמו כן, לא קיים מנגנון זמן ריצה עבור יישומי Office הרגילים, מלבד זה שנזכר לעיל, וכן יש להתקין את Excel במחשבים שימושיים להפעיל פרויקט Excel שמוסף מחדש.

אם תבחר להפץ את היישום המותאם אישית באמצעות Access Run Time, ייתכן שתרצה לבדוק את היישום באמצעות אפשרות שורת הפקודה /runtime/, כדי לדמות את סביבת המחשב שראeo משתמשים שיפעלו את היישום ללא שימוש ב-Access. כשהטפסים המותאמים אישית מפעלים את Access Run Time, הם חינויים יותר מתמיד, מכיוון שאין למשתמשים גישה אל רכיבי ממשק משתמש מוכלל של Access, כגון חלון מסד נתונים או אפילו תיבות דו-שיח של הודעת שגיאה. סיבה זוTZDKK לטופס פתיחה המשמש כממשק ניוטו (ראה פרק 5) ומנגנון לכידת שגיאות קפדי (ראה פרק 1). ללא מנגנון מעין זה, השגיאות תגרומו לישום להסתיים בצורה לא מסודרת, דבר שיבלב את המשתמשים.

באפשרות ליצור שני סוגים אריזה בעורת פונקציית האrizה של האשף. אם בדעתך להפץ את היישום באמצעות דיסקט, תקליטור או משאב רשות משותף, צור חבילה סטנדרטיבית. אם ברצונך לפרסום את היישום באמצעות דיסקטים בלבד, בחר באפשרות Multiple Cabs, והאשף יתאים את גודל קבצי *.cab. כך שאף אחד מהם לא יהיה גדול יותר מקיבולת של דיסקט וגיל. להפצת היישום באמצעות האינטרנט, عليك ליצור חבילת הפעלה לאינטרנט.

גישה לנוטנים ביישומים שאינם Access

כזכור, ODE מאפשר גישה לנוטנים בשלושה נתיבים: מעצב סביבת הנתונים, פקדים נתונים ADO ומודל האובייקט של ADO. ייתכן שההתלה תחווש בנוח להשתמש בתנאי מודל האובייקט של ADO (פרק 2 מציג מבוא למודל האובייקט של ADO; פרטיהם

נוספים ניתן למצוא בפרקאים אחרים). הדוגמה שלפנינו מאכלהת גיליון עבודה של Excel באובייקטים Connection (חיבור) ו-Recordset (ערכת רשומות) של ODO. הקוד שבודגמה מעתיק את שמות השדות מהטבלה Products של מסד הנתונים Northwind אל השורה הראשונה של גיליון עבודה של Excel. לאחר מכן, הוא מעתיק ערכי שדות אל השורות הבאות מתוך ערכת הרשומות בהמשך גיליון הנתונים. הדוגמה מופעלת מתוך פרויקט VBA הקשור אל קובץ חוברת עבודה.

```

Sub northwind2XL()
Dim cnn1 As ADODB.Connection
Dim rst1 As ADODB.Recordset
Dim RowCnt, FieldCnt As Integer
' Create instances of the connection and recordset objects.
    Set cnn1 = New ADODB.Connection
    Set rst1 = New ADODB.Recordset
' Set cnn1.ConnectionString to Northwind DSN.
    cnn1.ConnectionString = "DSN=Northwind"
' Open connection and recordset.
    cnn1.Open
    Set rst1.ActiveConnection = cnn1
    rst1.Source = "Select * FROM Products"
    rst1.Open , , adOpenStatic, adLockOptimistic
' Write in column headings in first row.
    RowCnt = 1
    For FieldCnt = 0 To rst1.Fields.Count - 1
        Cells(RowCnt, FieldCnt + 1).Value = rst1.Fields(FieldCnt).Name
        Rows(1).Font.Bold = True
    Next FieldCnt
' Fill rows with records, starting at row 2.
    RowCnt = 2
    While Not rst1.EOF
        For FieldCnt = 0 To rst1.Fields.Count - 1
            Cells(RowCnt, FieldCnt + 1).Value = rst1.Fields(FieldCnt).Value
        Next FieldCnt
        rst1.MoveNext
        RowCnt = RowCnt + 1
    Wend
End Sub

```

מחרוזות החיבור של האובייקט Connection מtabסת על אובייקט DSN. ה-DSN-ה שבודגמה מצביע אל מסד הנתונים Northwind, אך מאפשרתו לציין מסד נתונים כלשהו אחר. השימוש ב-DSN מפשט אתعيיצוב תחביר מחרוזות החיבור, מכיוון שהוא

מאפשר להגדיר את החיבור אל מסד הנתונים בצורה גרפית דרך הסמל ODBC של לוח הבקרה. שים לב גם לתחביר הסטנדרטי המשמש להגדרת חיבור וביסוס ערכות רשות על החיבור.

השגרה מיישמת את המאפיין Cells שפינה באופן משתמש לאובייקט ActiveWorksheet במודל האובייקט של Excel. מציינים תאים בודדים בגילוון העבודה על ידי ציון אינדקס השורה ואחריו אינדקס העמודה. קביעת ערך האינדקס בקטע הקוד המקwon בלולאות מאפשר לך לעבור על עמודות לרוחב השורה ועל שורות לאורך הגילוון.

חתימה דיגיטלית של פרויקטי VBA

Microsoft Office 2000/Visual Basic Programmer's Guide הוא מקור עשיר לתיעוד היבטי פועלה קרייטיים רבים, כולל חתימה דיגיטלית של פרויקט. אם בידך תעודת Authenticode, תוכל להחותם פרויקטי VBA שתיצור באמצעות Excel, Word, PowerPoint ו-Outlook כך שלא יגרמו להצגת אזהרת המאקרו, גם כשהרchip האבטחה במחשב המשתמש הוגדר לרמת אבטחה גבוהה.

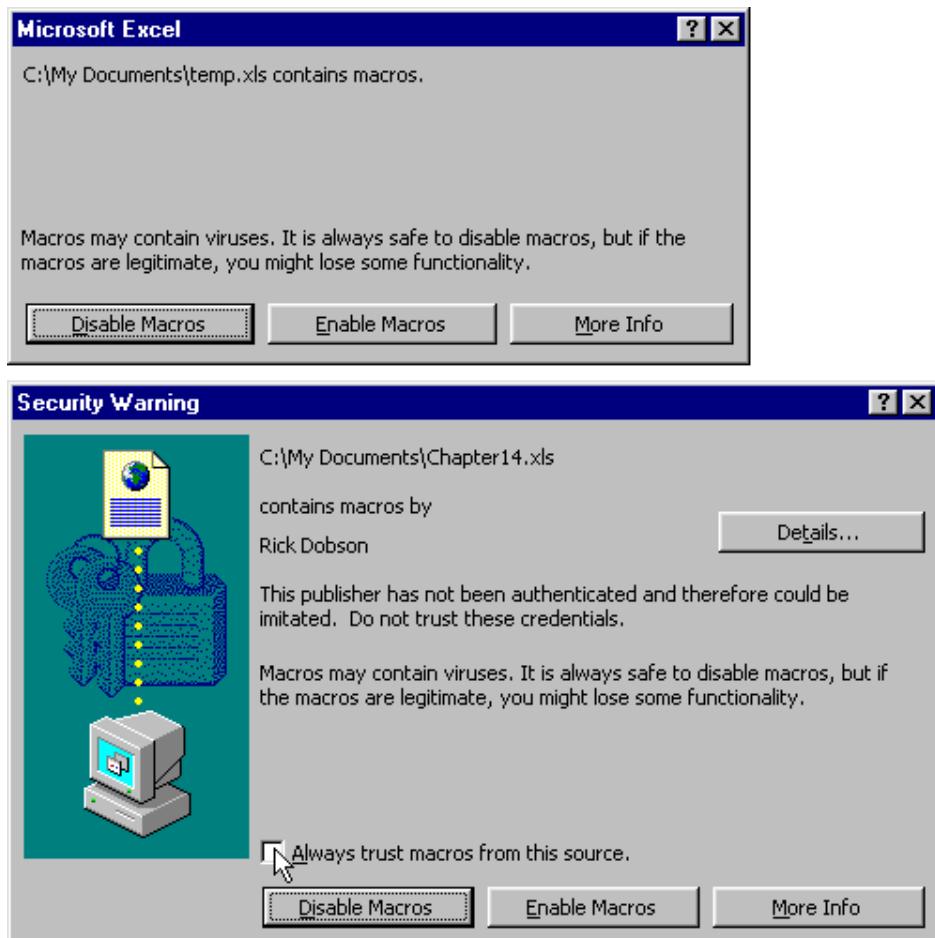
תרשים 14.5 מציג זוג הודעות אזהרה. ההודעה העליונה מוצגת בתגובה למסמכי פרויקט VBA בלא חתומים. ההודעה התחתונה מוצגת כמשמעות לפתח מסמכים המכילים פרויקטים חתוםים שעדיין לא הוגדרו כאמינים. לא ניתן להוסיף חתימה דיגיטלית ישירות לרשימת המקורות האמינים – עליך לבצע תחילת את תיבת הסימון Always trust macros from this source (בטעת תמיד במאקרו ממוקור זה) ולהפוך את המאקרו שלו לזמניים. כתוצאה לכך, ניתן יהיה לבוטוח בכל המסמכים הנושאים את החתימה הדיגיטלית של מסמך זה, ולא תוכנה אזהרות מאקרו כלשהן בפתחת פרויקטי VBA שהגיעו מהמקור הנדון.

הערה:



אימות באמצעות Authenticode פועל רק בבדיקות עבודה שמעילות את Internet Explorer בגרסה 4 ואילך. לא ניתן להתקין את תוכנת האימות Authenticode בבדיקות עבודה שבהן פועלת גירסה מוקדמת יותר של הדפדפן Internet Explorer, או שmotikon בהן דף מסוג אחר.

טכנולוגיית Authenticode מותבססת על הצפנה באמצעות מפתח ציבורי להחتمת מסמכים תוכנה, כגון פרויקטי VBA. כשמפתח תוכנה שמור קודץ, תוכנת Authenticode מבצעת פעולה hashing ליצירת "טביעה אצבע דיגיטלית" במסמך ומצפינה את טبיעה האצבע באמצעות מפתח פרטי. כאשר הנמען מקבל את המסמך, תוכנת Authenticode מנסה לשחרר את טביעה האצבע ולפנעה אותה. ניתן לפותח מסמכים שעוברים תהליך זה ללא הצגת הודעות אזהרה. מסמכים שלא עברו את התהליך, יציגו את ההודעה. אם משתמש כלשהו אחר עורך שינויים בפרויקט VBA ושומר אותו, החתימה הדיגיטלית מסולקת מהמסמך, ולכן פתיחתו בפעם הבאה תגרום להצגת הודעה אזהרה.



תרשים 14.5: הودעת אזהרה טיפוסית מפני מאקרו, שמוצגת בפתיחת מסמך בלתי חתום, והודעת אזהרה במסמך ראשון שהגיאו ממוקר חדש חתום דיגיטלי.

להחותמה דיגיטלית של פרויקט VBA עליך להשיג תחיליה אישור דיגיטלי חוקי. ניתן לעשות זאת בשלוש דרכיים: הדרך הראשונה היא ליצור עצמאך חתימה באמצעות תוכנית השירות Selfcert.exe הכלולה ב- Office 2000. דרך זו מתאימה למטרות בדיקות אישיות או לקבוצת עבודה. הדרך השנייה היא לרכוש אישור באמצעות האישורים הפנימית של הארגון. דרך זו מתאימה להפצת פרויקטים בארגון או בגודל וברכיב ספקיו. הדרך השלישית היא השגת אישור דיגיטלי מרשות אישורים מסחרית. זהו הפתרון הכללי הטוב ביותר.

החותימה הדיגיטלית ישימה לפרוייקט VBA בלבד, ולא למסמכי Office הקשורים אליו, כגון חוברת עבודה. הדבר מאפשר לשנות את המסמך הבזק (חוברת עבודה), אך לא את פרויקט VBA שלו היא שייכת. בחלון VBE, בחר את הפרוייקט

בחלון **Digital Signature Project Explorer** (סිיר הפרויקט). לאחר מכן בחר **Tools** (כליים). בפעם הראשונה שתשתמש באישור, בחר אותו ולחץ פעמיים על **OK**. אחרת, לחץ על **OK** בלבד.

זוג השגרות שלפניך מודגים כיצד ניתן לקבוע אם קובץ חוברת העבודה הוחתם דיגיטלי. השירה callIsDSigned קוראת לפונקציה isDSigned פעמיים. בקריאה הראשונה, callIsDSigned מעבירה את שם הקובץ הראשון. בקריאה השנייה, callIsDSigned מציגה דרך אפשרית להשגת שם הקובץ מהמשתמש והברתו אל isDSigned. חוברת העבודה צריכה להיות פתוחה כדי שניתן יהיה לקבוע אם הוחתמה דיגיטלי, ולכן callIsDSigned פותחת את הקובץ בשם סיפק המשתמש.

```
Sub callIsDSigned()
Dim myName As String
' Check whether current workbook is digitally signed.
    myName = Application.Workbooks(1).Name
    isDSigned (myName)
' Prompt user for workbook name and
' check whether that workbook is digitally signed.
    myName = InputBox("Type name as filename.xls", _
                      "Programming Microsoft Access 2000", myName)
    Workbooks.Open myName
    isDSigned (myName)
End Sub

Public Function isDSigned(fileName As String)
    isDSigned = Application.Workbooks(fileName).VBASigned
    If isDSigned Then
        Debug.Print fileName & " is digitally signed."
    Else
        Debug.Print fileName & " is not digitally signed."
    End If
End Function
```

הפונקציה isDSigned מאחסנת את ערך ההחזרה של המאפיין VBASigned ולאחר מכן משתמש אחד מתוך זוג משפטי הדפסה בהתאם לערך המוחזר. אם חוברת העבודה מוחתמת דיגיטלי, המאפיין מקבל ערך True. באפשרות נצל דומה לזה כדי לקבוע אם מסמכי Word או מצגות PowerPoint הוחתמו דיגיטלי (נוסח התכibir המדוקיק לפתח הקובץ משנה מיישום ליישום, אך PowerPoint ו-Word תומכים אף הם במאפיין VBASigned).

אינדקס

האינדקס הינו בשפה אנגלית.

~~לשם נוחות הקראיה התחילו אותו מסוף הספר. כלומר, סופו בעמוד הבא ותחלתו~~
~~כמו בספר באנגלית.~~

Index

A

AccessObject object
enumerating, 330
properties of, 357

AccessObject Type property, 360-361

Access Projects, 208, 501-507
connecting to a database, 501-503
forms and, 522-524
hyperlinks and, 521-522
learning from the NorthwindCS and Pubs databases, 503-507
programmatic issues and, 527-535
finding a record, 528
opening a form, 527
viewing changes made by other users, 529-530

querying in, 263-265

reports and, 519-521

resynchronizing a form with its one-to-many data in, 524-526

standalone modules and, 530-535

stored procedures and, 516-519

Access Run Time, 570

acForm constant, 24

action queries, 202-203, 234-243
append queries, 237-239
delete queries, 239-242
hiding, 239

action queries, continued
make-table queries, 242-243
update queries, 234-237

ActionQueryDemo procedure, 241

Activate event, 323

ActiveConnection property
Catalog object's, 134

Connection object's, 102
Recordset object's, 104, 111
Replica object's, 480-481, 483-484, 487, 489, 495
ActiveDocument object, 432
ActiveX Data Objects. See ADO (ActiveX Data Objects)
ActualSize property, Field object's, 116
AddARecord procedure, 110-111
addCbrBtns procedure, 403-404
addContacts procedure, 427
AddFromFile method, 353
AddGroupToUser procedure, 457-458
AddIdx procedure, 187-188
Add-In Designer, Component Object Model (COM), 568
AddIndex procedure, 143
Add-In Manager dialog box, 571-572
AddNew method, 92, 108, 111, 188-189
addOneContact procedure, 425-426
AddPKErr procedure, 185-186
AddPK procedure, 183
Add Procedure dialog box, 28, 30
AddReference procedure, 355-357
addShowAssistantsAndRocky procedure, 401-403
AddValuesSQL procedure, 191-192
Add Watch dialog box, 40
ADO (ActiveX Data Objects), 42, 85, 98-154

- data access models, 98-99 (see also ADODB library; ADOX library; JRO library)
- event model, 99
- OLE DB providers and, 99
- tables, techniques for creating and
 - managing, 178-197
 - avoiding replacing an existing table, 180-181
 - creating a table, 178-183
 - data in other formats, 192-197
 - dynamically populating a table, 188-192
 - indexes, 183-187
 - MSDASQL provider, 194-195
 - primary keys, 183-187
 - recordsets, using, 188-189
 - replacing a table, 182-183
 - saving recordsets, 190-191

SQL, using, 191-192
TransferDatabase and TransferSpreadsheet methods, 195-197
ADO Data Control, 570
ADODB library, 98-99, 100-133
 Command and Parameter objects, 121-129
 Errors collection, 129-133
 Field object. (see Field object)
 Recordset object. (see Recordset object)
ADOX library, 98, 134-154
 adding an index with, 143-145
 autoincrement field values and, 145-146
 Catalog object, 134-137
 Column object, 138
 creating tables with, 142-143
 data types and subtypes, 179-180
 enumerating fields with, 140-142
 enumerating tables with, 139-140
 Index object, 138-139
 Key object, 139
 overview of, 134-135
 Procedure object, 150-154
 Table object, 137
 View object, 146-149
.adp files, 208, 358, 497
aggregate functions, 201
 SQL, 69
 query design and, 223-224
aggregations, Simple Query wizard and, 209-211
All collections, 357-362
 enumerating members of, 359-360
AllDataAccessPages collection, 563-565
AllDatabaseDiagrams collection, 358
AllForms collection, 23, 291-293, 330, 359-362, 365-366
AllModules collection, 357-362
Allow Zero Length property, 172
AllQueries collection, 358
AllReports collection, 305, 330-332
AllStoredProcedures collection, 358
AllTables collection, 359-360
AllViews collection, 358

ALTER COLUMN keywords, 166, 252
ALTER TABLE keywords, 45, 166, 252
AnimateActor procedure, 387
Animation property, 381
animation(s), Assistant
 displaying searching animations, 382-383
 previewing, 385-387
 selecting animation types, 383-384
answerHelp procedure, 393-394
AnswerWizard object, 370
AppendChunk method, 116
Append dialog box, 237
Append method, 93
 adding an index and, 143
 creating a table and, 178
append queries, 203, 237-239
ApplyTheme method, 564
arguments, 24-25
 for procedures, 27
arithmetic operators, query design and, 220-221
arrays, 49-50
As keyword, 49
ASP format, publishing datasheets in, 541-544
Assistant characters, 383-384
AssistantIdleOn procedure, 379-381
AssistantNotVisible procedure, 380
Assistant object, 370, 379-394
 animation types for, 383-384
 Assistant characters for, 384
 automatically referencing the Office Object Library, 381-382
 displaying a searching animation, 382-383
 previewing animations, 385-387
 showing and animating, 380-381
AssistantSearchOn procedure, 381
AssistantSearchOn2 procedure, 381-382
Attributes property
 Column object's, 138
 Field object's, 116
AutoForm wizard, 276-277, 281

autoincrement fields

ALTER TABLE and ALTER COLUMN used to reset, 251-255

start and step values for, 45

autoincrement field values, ADOX library and, 145-146

automation, 412-414

Word and, 430

AutoNumber fields (AutoNumber data type), 166, 180

AutoReport wizard, 302

AVG aggregate function, 224

B

balloonHeadTextCheck procedure, 391-392

balloonHeadTextLabel procedure, 389-390

Balloon object, 380, 387-394

collecting user input with, 390-392

graphics, 387-388

labels, 389-390

modeless, 392-394

balloonTextImageIcon procedure, 388

batch updating, in ODBC Direct workspaces, 97-98

BeginTrans method, 89, 447, 465

bidirectional replication, 476

breakpoint, 38

Briefcase, replication of databases and, 469

builtinCommandBarCount procedure, 395

built-in functions, 63-73

calculated fields using, 220-221

conversion, 68

date/time, 64-65, 68

domain aggregate, 68, 71-72

error handling, 68

Immediate window used to run, 41

inspection, 68

math, 68

messages, 68

summary of selected, 67-73

text, 68

Button property, Balloon object's 390

C

CacheSize property, 107
calculated fields, query design using, 220-225
 aggregate SQL functions, 223-224
 arithmetic operators, 220-221
 built-in functions, 220-221
 computing a value from more than one query, 224-225
 custom functions, 222-223
callAddGroupToUser procedure, 457
Callback procedure, 392-393
Callback property, 392-393
callCompactADB procedure, 496
callDeleteGroup procedure, 460
callDeleteUser procedure, 456
Call keyword, 29
callMakeGroup procedure, 459
callMakePartialFilter procedure, 485, 487
callMakeUser procedure, 456
callPrintTypePriority procedure, 492-495
callRemoveUserFromGroup procedure, 458
callSetAllTablePermissionsForGroup procedure, 463
callSetTheme procedure, 564-565
Cancel method, Connection object's, 97
cascading delete feature, 162
cascading updates feature, 162
Case Else clause, 54-55
 in error handler, 77
Catalog object, 134-137
 creating, deleting, and tracking groups in, 459-462
 creating a table and, 178
CatCon procedure, 134-137
.cboLookup_AfterUpdate procedure, 286-287
.cboPickAYear_AfterUpdate procedure, 291
Cells collection, Word, 432
ChangeAView procedure, 147, 148-149
ChangeInTrans procedure, 465-466
ChangePassword method, 456
Chart Component, on data access pages, 561-562
Chart objects, 289-291

Chart wizard, 289
 creating reports with, 302-303

CheckMailItTag procedure, 334

Circle method, 326

classes
 custom, 337
 programming events into, 353-357

form, 295-299
 manipulating, 296-298
 references to instances of, 298-299
 viewing properties and methods of, 295
 instantiating, 337

class modules, 27, 35-36, 335. See also custom class modules; form class modules; report class modules
 built-in events, 354
 custom classes and events and, 337-338
 custom properties and methods and, 337-338
 exposing a property with a property function, 339-340
 exposing a property with a public variable, 338-339
 public variables vs. Property functions, 340
 updating data sources, 340-343
 custom property functions in, 337
 instantiating classes and, 337
 standalone, 336-338
 login interface (case study), 343-347
 programmatically editing, 362-365

Click event procedure, 59

Close Button property, 24

Close event, 323, 336

Close method, arguments of, 24

cmdNewPassword_Click event procedure, 345-346

cmdPrintThem_Click event procedure, 329

cmdShowCustomer_Click event procedure, 288-289

cmdSubmit_Click event procedure, 347

Code button, 27-28, 37

Code Commentator, 569

CodeData object, 358

Code Librarian, 569, 571-574

CodeProject object, 357-358

Code window, 37-38, 38

collections, 22-23. See also specific collections
colors, of hyperlinks, 273
color viewer, 72-73
column-level updates, 477
Column object, 138
Columns collection, 136-137
ColumnType function, 142
ColumnType procedure, 140-142
COM (Component Object Model) Add-In Designer, 568
COMAddin object, 370
combo box lookup forms, 283-284
CommandBarButton object, 394, 399
CommandBarComboBox object, 394
CommandBarControl object, 396
CommandBar object, 370, 392-408. See also command bars
 enumerating command bar elements, 394-396
 listing menu commands, 396-397
 listing visible command bars, 395-396
CommandBarPopup object, 394
command bars. See also CommandBar object
 built-in, manipulating, 397-401
 adding commands, 399
 disabling and reenabling command bars and their controls, 397
 making invisible command bars visible, 399
 custom
 creating, 401-403
 deleting, 407-408
 modifying, 403-404
 popup, 404-407
 defined, 394
 listing visible, 395-396
 restoring, 401
Command Button wizard, 27
Command object, 121-129
 ADOX library procedures and views and, 150-154
 benefits of, 121
 CommandText property, 121
 CommandType property, 121
 CreateParameter method, 122
 creating a recordset with a select query and, 123

creating a recordset with a parameter query and, 124-126
deleting records with, 125-126
Execute method, 122
executing SQL commands with, 245-246
inserting records with, 126-128
updating record values with, 128-129

commands

- adding, to built-in command bars, 399-401
- menu, listing, 396-397

CommandText property

- Command object's, 121, 125, 127, 129, 205
- Procedure object's, 150, 152

CommandTimeout property, 121

CommandType property, 121-122

- Procedure object's, 152

CommitTrans method, 89, 447, 465

CompactDatabase method, 478

compacting replicas, 495-496

Compact method, 89

comparison operators, 201, 279

Component Object Model (COM)

- Add-In Designer, 568
- computeOnGas procedure, 421-422
- concurrency, single-row locking and, 43-45
- conditional code execution, 50-63
 - with Do...Loop statements, 61-63
 - with For Each...Next and With...End
 - With statements, 57-61
 - with For...Next statements, 55-57
 - with If...Then statements, 50-54
 - with Select Case statements, 54
- conditional formatting, 277-279
- Conditional Formatting dialog box, 279
- Conflict Resolution wizard, 468, 479
- Connection Control property, 440-441
- Connection object, 409
 - ADODB library, 100-104, 121
 - Mode property, 102-103
 - Open method, 101-102
 - OpenSchema method, 103-104

Catalog object and, 134
DAO, 96
Connections collection, 96-98
Connector property, multiple search criteria and, 377-379
connect2XLPrintFromFirst procedure, 411
constants, 48
ContactItem object, 426, 428
Contacts folder
 adding an item to, 425-426
 adding multiple items to, 427-428
 deleting an item from, 428
 enumerating items in, 424-426
container objects, 95
containers, 22-23
Containers collection, 95
Continue command, 39
controls, 23
 enumerating forms and, 291-292
 report, 304
ControlsInReports procedure, 332
ControlType property, 292, 333
conversion of data structure, in FrontPage guestbook case study, 307-310
Copyfile method, 482
Count property, 23
cpw procedure, 346
createAndShowPopUpMenu procedure, 406-407
CreateCustomView procedure, 257
CreateDatabase method, 89
CreateField method, 93
CreateIndex method, 93
CreateItem method, 426
Create method, Catalog object's, 134-135
CreateObject function, 413-414, 424-426, 432
CreateParameter method, 125, 342
 Command object's, 122-123
CREATE PROC statement, 257-261
CreateProcToDelete procedure, 258
CreateQueryDef method, 94, 97
CreateRelation method, 95
CreateReplaceView procedure, 257

CreateReplica method, 483-484, 487, 489
CreateTableDef method, 94
createTableFromXL procedure, 418-421
CreateView procedure, 255-256
CREATE VIEW statement, 255-257
CreateWorkspace method, 89
criteria
 manual query design and, 225-228
 for select queries, 200-201
crosstab queries, 205
Crosstab Query wizard, 215-216
cubes, form for calculating, 51-52
CurrentData object, 358
CurrentDb function, 90-91
Current event, 25
CurrentProject object, 357-358
CursorLocation property, 527, 533-535
cursors, ODBC Direct workspace, 87
cursor types, ADO, 105-106
custom class modules, 35-36
Customers table, in Northwind database, 101-103, 113, 156, 160
custom functions, query design and, 222-223
CVErr function, 79

D

DAO (Data Access Objects), 42, 85
 Database Container object, 371-372
 overview of, 86-98
 Jet workspace objects, 93-95
 Jet workspaces, 86
 objects common to Jet and ODBC Direct workspaces, 87-93
 ODBC Direct workspace object, 96-98
 ODBC Direct workspace, 86-93
data access. See also remote data sources in non-Access applications, 576-578
 ODE tools for, 569-570
data access models. See ADO (ActiveX Data Objects); DAO (Data Access Objects)
Data Access Objects. See DAO (Data Access Objects)
DataAccessPage object, 563-565
data access pages, 553-565
 creating, 553-555
 simple columnar page, 555-557

grouping records in, 557-559
Office 2000 Web components on, 559-562
 chart sample, 561-562
 PivotTable list sample, 561-562
 spreadsheet sample, 559-561
programmatic issues for, 563-565
database diagrams, 508-512
Database object
 in Jet workspace, 93-94
 in ODBCDirect workspace, 96
database replication, 99
Databases collection, 90-91
Database Splitter wizard, 440
Database window, opening VBE from, 36
Database wizards, 162
Data Definition Language (DDL), 205
data definition queries (data definition functions), 205, 249-255
Data Environment Designer, 570
Data Link Properties dialog box, 264, 515, 555
Data Report Designer, 570
datasheets, publishing, 538-544
Datasheet view, union queries in, 231-232
Data Source Name. See DSN (Data Source Name)
data sources
 class modules and, 340-343
 remote (see remote data sources)
data types
 for fields, 165-172
 for variables, 45-48
DateDiff function, 64, 71
Date function, 64-65
DateSerial function, 68
Date/Time data types, 172
dBase, 45
DBEngine function, 90
DBEngine object, 89
dbFailOnError option, 94
dbRunAsync constant, 97
DCount function, 72
DDL (Data Definition Language), 205

Deactivate event, 323
debugging, 38-42. See also error trapping
declarations, 48-50
DefinedSize property
 Column object's, 138
 Field object's, 116
deleteAContact procedure, 426-427
deleteAContact2 procedure, 429
DeleteAllRecords procedure, 125-126
DeleteARecord procedure, 125
DeleteAView procedure, 149
deleteCustomCbr procedure, 407-408
DeletefromModules procedure, 365
deleteGroup procedure, 460
DeleteLines method, 362-365
Delete method, 92
DeleteOrUpdateARecord procedure, 111-112
delete queries, 203, 239-242
deleting
 Contacts folder items, 426-430
 custom command bars, 407-408
 records, 111-112
 with Command object, 125-126
 duplicate records, 212
text
 from form class modules, 366-367
 from modules, 364-365
users, 455-457
 a view, 149
Description property, Error object's, 129
design master, 468
Design view, 24, 36, 37
 query design in, 216-228
 action queries, 234-243
 adding tables or other queries, 216-219
 calculated fields, 220-225
 criteria, 225-228
 parameter queries, 228-231
 referencing subdatasheets, 219-220
 update queries, 234-237

DFirst function, 71
diagrams, database, 508-512
digitally signing VBA projects, 578-580
Dim statement, 33, 49-50
 custom class modules and, 35-36
disableMenuBar procedure, 397
disableViewMenuAndControl procedure, 398
display duration, controlling, 271
DISTINCT keyword, 245
DLast function, 71
DLookup function, 71
DMax function, 71
DMin function, 71
DoCmd object, 24-25, 28, 178
DoCmd OpenReport method, 305
DoCmd OutputTo method, snapshot files and, 323
DocumentProperty object, 370, 375
Do...Loop statement, 61-63
domain aggregate functions, 69, 71-72
domainname argument, in domain aggregate functions, 71
DSN (Data Source Name), 194
 publishing datasheets in IDC/HTX format and, 540
duplicate records, Find Duplicates Query wizard, 212-213
dynamic cursor, 105
Dynamic recordset type, 91
Dynaset recordset type, 91
dynasets, 202

E

EasyLoop procedure, 108-109
EasyLoop2 procedure, 109
editing records, 111-112
Edit method, 92
Edit Relationships dialog boxes, 176-177
Elseif clauses, 51, 53
e-mail, snapshot files sent as attachments to, 323, 333-334
Empty keyword, 48
encrypting replicas, 495-496
End Function statement, 30
End Property statement, 35
EnumerateAllForms procedure, 359

EnumerateAllFormsToInsert procedure, 365
EnumerateAllModules procedure, 359
EnumerateAllModulestoInsert procedure, 363-364
EnumerateAllTables procedure, 359
EnumerateAllViews2 procedure, 360
enumerateControlCaptions procedure, 396, 399
enumerateVisibleCommandBars
procedure, 395-396
enumerating
 All collection members, 359-360
 command bar elements, 394-396
 Contacts folder items, 424-426
 fields, 140-142
 forms in another project, 294-295
 forms and controls, 291-292
 reports, 330-332
 tables, 139-140, 140
 views and procedures, 150-152
EP method, 339
ep2 method, 339
Err object, 74, 89-90, 129-130
 424 error, 132
 3251 error, 132
Error object, properties of, 129-130
Errors collection, 89-90, 129-133
error trapping, 73-80
 avoiding replacing an existing table, 180-181
 raising errors and, 79-80
 samples of, 75-80
 syntax for, 74-75
EvenToOdd procedure, 128-129
event procedures, 25-26
events, 25-26
 class module built-in, 354
 custom, 337-338
 programming into custom classes, 353-357
 standard module syntax for, 355-357
Excel
 accessing data in, 192-193
 integrating Access 2000 with, 409-410, 415-423

dynamically creating Access tables
based on Excel worksheets, 418-421
running Excel procedures from an Access procedure, 421-423
values from Excel worksheets, 415-418

Exchange ISAM, 45

Execute method, 94, 97

- Command object's, 122
- FileSearch object's, 373-379

Exit Do statements, 62

Exit For statements, 56, 58

Exit Function statements, 30, 75

Exit Sub statements, 74

Export dialog box, 538-539

Export Report dialog box, 320

Expression Builder, 173-174

F

Field Builder, 164-165

Field Builder dialog box, 164-165

fieldexpression argument, in domain aggregate functions, 71

Field List dialog box, 554-555

field lists, in SELECT statements, 244

FieldNameType procedure, 117

Field object, 115-121

- finding the longest field entry in, 120-121
- methods of, 116
- Name and Value properties of, 116-117
- printing field data types, 117
- properties of, 116
- Type property of, 117

fields. See also Field object

- adding
 - with Field Builder, 164
 - manually, 165
- auto-incrementing
 - setting autoincrement field values, 145-146
 - start and step values for, 45

AutoNumber, 166, 180

data types for, 165-172

Date/Time, 172

enumerating, 140-142

Lookup, 166-169
Memo, 171-172
Number, 169-171
OLE Object, 172
refreshing, 443-445
replication system, 472-473
Table wizard and, 163
Text, 166
Yes/No, 172

Fields collection, 93-95
FieldSizes procedure, 120-121
FieldType function, 117
FileLen function, 375
FileName property, FileSearch object's, 373, 376-377
FileSearchAct procedure, 383
FileSearch object, 370, 372-379
 basic file search with, 373-374
 displaying a searching animation, 382-383
 multiple search criteria and, 377-379
 sorting the return set from a search, 374-375
 text searches, 375-377
FileSearch1 procedure, 373
FileSearch2 procedure, 374-375
FileSearch3 procedure, 376
FileSearch property, Application object's, 373, 376-377
file size, sorting the return set from a file search by, 374-375
FileSystemObject object, 482-483
FileType property, 376-377
Filtered property, 108
filtering records, 114-115
FilterLikeField function, 114-115
Filter object, 471
Filter property, 114-115
FilterRecordset procedure, 114-115, 115
FindAMatch procedure, 112-113
findByID procedure, 528-529
Find Duplicates Query wizard, 212-213
FindFirst method, 93
finding records, 112-114
FindLast method, 93

Find methods, 92-93, 107-108, 112-114, 362-365
FindNext method, 92-93
FindPrevious method, 92-93
Find Unmatched Query wizard, 214
First Normal Form, 157-158
For Each...Next and With...End With statements, 57-61
For Each...Next statements, 67
 error trapping and, 78-79
foreign keys, 172, 175-176
For loops, error trapping and, 78-79
Format event, 323
Format function, 71
Format property of fields, 172
form classes, 295-299
 manipulating, 296-298
 references to instances of, 298-299
 viewing properties and methods of, 295
form class modules, 35-36
 inserting text into, 365-367
 login interface (case study), 343, 347-352
Form_Close event procedure, 443, 445
form letters, 311-313, 435-437
Form_Open event procedure, 271, 443
forms, 23, 269-299
 Access Projects and, 522-527
 enumerating, 291-292
 in another project, 294-295
 hiding and showing, 293
 HTML, 544-547
 linking to data, 276-281
 with AutoForm wizard, 276-277
 conditional formatting, 277-279
 looking up and displaying data in, 281-291
 charting a subset of data, 289-291
 creating a lookup form, 281-287
 dynamically displaying information, 287-289
 resynchronizing with one-to-many data, 524-526
 splash screen, 269-270
 switchboard, 272-276
 VBA used to manipulate, 291-295

viewing and editing data using, 522-524
Form view, property sheet in, 23-25
For...Next statements, 55-57
forward-only cursor (fire hydrant cursor), 105
Forward-only recordset type, 91
FoundFiles object, 372-379
FoxPro databases, 410
fromAccessToWordTable procedure, 430-432
FROM clause, 244
FrontPage 2000
 guestbook created with (case study), 306-313
 converting the data structure, 307-310
 creating a form letter, 311-313
 creating mailing labels, 310-311
 importing data from, 306-307
 publishing datasheets with, 538-540
FTP snapshots with Netscape browsers, 548-549
FullName property, 563
functions (function procedures), 30-34
 aggregate (see aggregate functions)
 built-in (see built-in functions)
 custom, query design and, 222-223
 definition of, 30
 invoking (running), 30-31
 procedures compared to, 30
 property
 custom, 337
 exposing properties with, 339-340
 public variables vs., 340

G

GetChunk method, 116
GetDefaultFolder method, 424-426
getfp procedure, 308-310
GetNameSpace method, 424-426
GetObject function, 413-414
GetODBCThroughOLEDB procedure, 194-195
GetOption method, 440, 445
GetString method, 110
GoToControl method, 25, 34
GROUP BY clause, 200, 212, 245, 248-249

grouping records for display in a report, 314-315
Grouping And Sorting dialog box, 314-315
groups, assigning to users, 457-459
Groups collection
 ADOX library, 136
 in Jet workspaces, 86, 95

H

HasModule property, 272, 336, 357, 361, 453
HAVING clause, 200, 245
Hex function, 67, 69-70
hextest function, 69
HideAForm procedure, 293
hiding and showing
 action queries, 239
 forms, 293
HTML (Hypertext Markup Language)
 Data Report Designer and, 570
 forms, 544-547
 publishing datasheets in, 538-540
HTML Help Image Editor, 571
HTML Help Workshop, 571
HTMLProject object, 370
Hyperlink data type, 171, 179-180, 272, 520-522, 549
 indexing and, 45
Hyperlink object, 537
hyperlinks, 549-553
 adding, 521-522
 colors of, 273
 inserting and editing, 550-552
 navigating with, 272-274
 samples of, 552-553
 types of, 549
Hypertext Markup Language. See HTML (Hypertext Markup Language)

I

Icon property, Balloon object's 387-388
IDC/HTX format, publishing datasheets in, 540-541
IDENTITY data type, 145-146, 166, 180
IDENTITY keyword, 252
If...Then...ElseIf...Else statement, in error trapping procedure, 76-77

If...Then...Else statement, in error trapping procedure, 76-77
If...Then statement, 50-54
Ignore Nulls property, 138-139
IIF (Immediate If) function, 65
Immediate If function, 292
Immediate window, 29, 40-41
 invoking a function from, 30-31
 uses of, 41
importing data, from FrontPage guestbook, 306-307
Indexed Sequential Access Method. See ISAM (Indexed Sequential Access Method)
indexes
 adding, with ADOX library, 143
 creating, 174-178
 programmatically adding, 187-188
Indexes collection, 93, 143
indexing Memo data types, 45
IndexNulls property, 138-139
Index object, 138-139
Initialize event, 336, 354
inner joins, SELECT statement syntax for, 246-247
Input Mask property, 172-173
Input Mask wizard, 172-173
Insert Hyperlink dialog box, 273-274, 550-552
inserting
 records, with a Command object, 126-128
 text into modules, 363-364
InsertIntoForms procedure, 366
InsertIntoModules procedure, 364
InsertLines method, 362, 364
Insert Picture dialog box, 269-270
InsertRecords procedure, 126-128
inspection functions, 68
instantiating classes, 337
IntelliSense feature, 28
Internet synchronization, replication of databases and, 471-472
InvoiceDates table, 63-67, 72
invoices, built-in functions used to report on past-due, 63-67
InvokeAddReference procedure, 355-357
InvokeRemoveReference procedure, 356

ISAM (Indexed Sequential Access Method)
 data sources, Jet workspaces and, 86
 installable ISAM drivers, 410-411
 Jet 4 and, 45
isAppThere procedure, 413-414
IsEmpty function, 48
IsNull function, 46
 error trapping and, 78
IsNull operator, 46
ISO 10646 standard, 44
ItemAdded event, 354-355
Item property, 23
ItemRemoved event, 354-355

J

Jet database engine (Jet 4; Jet 4 Server), 43-45
 bidirectional replication between SQL Server replicas and, 476
 MSDE vs., 498-500
Jet Engine object, 470-471
Jet workspaces, 86
 objects common to ODBC Direct workspaces and, 87-93
joins, 201-202
 inner, SELECT statement syntax for, 246-247
 in SELECT statements, 244
 Simple Query wizard and, 207
JRO library, 99
JRO programming, replication of databases and, 470-471, 480-496
 compacting and encrypting replicas, 495-496
 creating additional full replicas, 483-484
 creating partial replicas and filters, 484-487
 making a database replicable, 480-483
 Prevent Deletes replicas, 489-491
 replica properties, 492-495
 synchronizing replicas, 488-489
jrRepTypeNotReplicable procedure, 495

K

Key object, 139
keyset cursor, 105

L

labels, balloon, 389-390
Label wizard, creating reports with, 302
LanguageSettings object, 370
leap years, 32
Left\$ function, 221
LIKE operators, 245
linelabel argument, On Error statement, 74
Line method, 325
linkODBCAuthors procedure, 196
ListAllForms procedure, 292
ListAllReportsElsewhere procedure, 331
ListAllReports procedure, 330
listCommands procedure, 397
listContacts procedure, 424-426
listGroupsInCat procedure, 460-461
ListMyProcs procedure, 150-151
ListTables procedure, 139-140
ListTableTypeColumns procedure, 140-142
ListViews procedure, 137
Load event, 25, 336
Locals window, 40
locateEmployee procedure, 528
LockEdits property of a recordset, 92
locking
 page-level, 43-45, 449-450
 records
 manually, 443
 programmatically, 445-447
 row-level, 447-449
 single-row, 43-44
LockType property, 106
login interface (case study), 343-352
LookIn property, 373
Lookup fields, 166-169
lookup forms, creating, 281-287
loops, Do, 61-63
LoopToUsingErrors procedure, 132

M

macros, 81-83
 translating to VBA code, 82
 VBA compared to, 83
 working with, 81-82

Macro window, 82

mailing labels
 creating, 310
 with Word mail merge feature, 432-435

Mailing Label wizard, 310-311

MailMerge object, 411, 430, 432-435

MakeAView procedure, 147

makeDesignMaster procedure, 481-482

makeFullReplica procedure, 487

makeGroup procedure, 464

MakeLocalTableErrCatcher procedure, 181

MakeLocalTableErrCatcher2 procedure, 182-183

MakeLocalTable procedure, 178

makePartialFilter procedure, 487

MakeReplicable method, 480-483

Make Table dialog box, 243

make-table queries, 242-243
 with SQL statements, 249-255

makeUser procedure, 456

many-to-many relationships, 160-161

MarkFieldsToEdit procedure, 59, 61

.mdb files, 358, 360

.mde files, 453-454

Memo data types, 171-172
 indexing of, 45

menu bar, popup, 404-407

menu commands, listing, 396-397

Me prefix, 33

message box, displaying results in, 283-287

methods, 22-25
 custom, 337

Microsoft Access Run Time, 570

Microsoft Data Engine (MSDE), 497-501. See also Access Projects
 Office 2000 Developer Edition (ODE) and, 569-570

Microsoft Outlook

- integrating Access with, 424-430
- adding an item to the Contacts folder, 425-426
- adding multiple items to the Contacts folder, 427-428
- deleting an item from the Contacts folder, 426
- deleting multiple Contacts folder items, 428
- enumerating Contacts folder items, 424-426
- Jet 4 and, 45

Microsoft Replication Manager, 571

Microsoft Visual SourceSafe, 569

modeless balloons, 392-394

Mode property

- Balloon object's 387-388
- Connection object's 102-103

Modify Design permission, 453

Module object, programmatically editing modules and, 362

modules, 35-36

- class, 27
- programmatically editing, 362-367
 - deleting text, 364-365
 - inserting text into class modules, 363-364
 - inserting text into form class modules, 365-367
- standalone, 336-338, 530-535
- standard, 27, 336
 - cause events, 355-357
- types of, 336

MoveFirst method, 92, 106

MoveLast method, 92, 107

Move methods, 92, 93, 106-107

MoveNext method, 92, 107

MovePrevious method, 92, 107

MSDASQL driver, 265-266

MSDASQL provider, 194-195

MSDE (Microsoft Data Engine), 497-501. See also **Access Projects**

- Office 2000 Developer Edition (ODE) and, 569-570

MSysDB object, 371

multi-user databases, 439-466

- security features and, 450-464
- alternatives to user-level security, 450-454
- custom interface, 451

database password, 451-452
.mde files, 453-454
module password, 451-452
programmatically controlling user-level security, 452-464
sharing files in, 440-442
sharing forms in, 442-447
sharing recordsets in, 447-450
transactions and, 465
MyMakeTable procedure, 249-251
MySelect procedure, 246

N

Name property
 Catalog object's, 134, 137
 Column object's, 138
 Field object's, 116-117
 Index object's, 138-139
 Procedure object's, 150
NameSpace object, 424-426
NativeError property, 129-130
navigation
 with code, 272-274
 recordset, 106-107
Netscape browsers, FTP snapshots with, 548-549
newCommandBarAndButton procedure, 401-403
New keyword, 336-338, 420
newMenuItem procedure, 399-400
NewSearch method, 373
NoData event, 323
NoEasyLoop procedure, 110
normalization, 156-159
Now function, 64-65, 319
Null values, 46, 48
Number fields, 169-171
Number property
 Err object's, 74
 Error object's, 129
NumericScale property, Column object's, 138

O

- Object Browser, 42
 - form class properties and methods in, 295
- objects
 - definition of, 22-23
 - methods of, 22-23
 - properties of, 22-23
- Oct function, 67
- ODBC data sources, 86-87, 90-91, 137-138, 194-195
- ODBCDirect, 206
- ODBCDirect workspaces, 86-93
 - batch updating in, 97-98
 - benefits of, 87
 - objects common to Jet workspaces and, 87-93
 - objects in, 96-98
- ODBC sources, querying, 261-262
- OddToEven procedure, 128, 129
- ODE. See Office 2000 Developer
- Edition (ODE)
- Office applications. See also specific Office applications
 - integrating Access 2000 with, 409-437
 - automation, 412-414
 - Excel (see Excel)
 - installable ISAM drivers, 410-411
 - OpenDataSource method, 411
 - Outlook, 424-430
 - Word, 430-437
- Office 2000 Developer Edition (ODE), 567-580
 - Code Librarian, 571-574
 - data access tools, 569-570
 - in non-Access applications, 576-578
 - deployment and management tools, 570-571
 - digitally signing VBA projects, 578-580
 - overview of, 568-571
 - Package And Deployment wizard, 574-576
 - VBA productivity tools, 568-569
- Office Object Library, 382-383
 - animation types in, 383-384
 - automatically referencing, by Assistant object, 381-382

Office objects, 369-408
OLAP (online analytical processing database) technology, 215-216
OLE DB (OLE database) providers, 85, 97, 178, 192, 194-196, 206
OLE Object data type, 172
OnAction property, 401
On Error statement, 74, 76
one-to-many relationships, 160-161
one-to-one relationships, 159
OpenConnection method, 96-97
OpenDatabase method, 89
OpenDataSource method, 411, 434, 437
Open event, 25, 323
OpenFast procedure, 102
openForm procedure, 530
OpenLookOnlyErrors procedure, 130-131
OpenLookOnly procedure, 103
Open method
 Connection object's, 100-102
 Recordset object's, 104-106
OpenPrintXLDataSource procedure, 193
OpenQuery method, 237
OpenRecordset method, 45, 91-92, 94, 97-98
openRemoteWithCursorLocation procedure, 533-535
OpenSavedRST procedure, 190-191
OpenSchema method, 441
 Connection object's, 103-104
openTableOnRemoteServer procedure, 530-531
OpenUserLevel procedure, 454-455
openXLComputePrint procedure, 416, 418
Option Base 1 statement, 50
Option Compare Database statement, 34
Option Compare statement, 68
Option Explicit statement, 33-34, 50, 132
ORDER BY clauses, 245, 248-249, 513
Order Details table, in Northwind database, 157-158
orphan records, 161-162
Outlook. See Microsoft Outlook.

P

Package And Deployment wizard, 570-571, 574-576
PageFooterSection_Format event procedure, 327
page-level locking, 449-450
pages
 data access (see data access pages)
 locking, 43-45
Pages collection, 554
Paradox, 45
ParameterQCommand procedure, 124, 152
parameter queries, 204, 228-231
 creating a recordset with, 124-125
 updating data sources with, 342-343
Parameters collection, 94, 125
Parameters declaration, 125
pass-through queries, 206
password
 database, 451-452
 module, 452-453
password mask, 344
permissions, 453-454
 for groups, 457
 setting, 462-463
Picture property, of splash screens, 269-270
PivotTable list control, on data access pages, 561-562
PKErrorCatcher procedure, 144-145
PO1 procedure, 341
PO2 procedure, 342
PopulatePartial method, 485, 487
popup command bars, 404-407
Precision property, Column object's, 138
Prepared property, 122
Prevent Deletes replica, 469, 478, 489-491
primary keys, 156, 165, 175
 dynamically creating, 183-187
Print event, 323-324
printing
 field values of a recordset, 108-110
 a view, 147-148
printPreviewLabels procedure, 434

printPreviewLetters procedure, 437
printPriority procedure, 492-495
printReplicaType procedure, 492-495
priority-based conflict resolution, 478
Priority property, 478, 492
Private keyword, 35
Private statement, 50
Procedure object, 150-154
procedures
 creating and running stored, 152-154
 enumerating, 150-152
 Excel, running from an Access procedure, 421-423
 functions compared to, 30
 invoking (calling), 29-30
 naming conventions for, 27
 property, 34-35
 stored, 516-519
 creating, with SQL CREATE PROC statement, 257-261
 opening, 532
 subprocedures, 27-30
 types of, 27
processComboBoxChoice procedure, 406-407
procLookupNumber procedure, 153-154
ProjectType property, 360
Project window, 37-38
properties, 22-23, 23-25
 exposing, with a property function, 339
Properties collections, 134
Properties window, 37-38
property functions
 custom, 337
 exposing properties with, 339-340
 public variables vs., 340
Property Get function, 339-340, 350
Property Get statement, 34-35
Property Let function, 339-340, 350
Property Let statement, 34-35
property procedures, 34-35
Property Set statement, 34-35
property sheet, 23-25

PropertyTests collection, 372, 377
Provider property, 454
Public keyword, 35
Public statement, 50
public variables
 exposing properties with, 338-339
 property functions vs., 340
publishing datasheets, 538-544

Q

queries, 199-267
 action, 202-203, 234-243
 append queries, 237-239
 delete queries, 239-242
 hiding, 239
 make-table queries, 242-243
 update queries, 234-237
 append, 203, 237-239
 crosstab, 205
 data definition operations and, 205
 delete, 203, 239-242
 make-table, 205, 242-243
 manual design of, 207-243
 action queries, 234-243
 Design view, 216-228 (see also Design view, query design in)
 parameter queries, 228-231
 wizards, 207-216
 overview of, 199-207
 parameter, 204, 228-231
 creating a recordset with, 124-125
 updating data sources with, 342-343
 pass-through, 206
 programming, with SQL and ADO, 243-261
 data definition functions, 249-255
 SELECT statements, 244-249
 stored procedures, 257-261
 views, 255-257
 to remote data sources, 261-267
 Access Projects, 263-265
 linked ODBC sources, 261-262
 programmatic, 265-267

remote data sources and, 206-207
saving, 202
select (see select queries)
subqueries, 205
types of, 199-207
union, 204, 231-232
update, 203

QueryDef object, 94, 97-98
QueryDefs collection, 94
Query Designer, 514-519
Query Properties dialog box, 219
Quit method, Excel, 421

R

Raise method, 79
raising errors, 79-80
Range object, in Word, 430
RecordLocks property, 445-447
records
 adding, 110-111
 deleting, 111-112
 with Command object, 125-126
 duplicate records, 212
 duplicate, Find Duplicates Query
 wizard, 212-213
 locking
 manually, 443
 programmatically, 445-447
 unmatched, Find Unmatched Query wizard, 214

RecordsetClone property, 527

Recordset object (recordset)
 ADODB library, 101-102, 104-115
 ActiveConnection property, 105
 adding records, 110-111
 AddNew method, 108
 cursor type, 105
 editing or deleting a record, 111-112
 Filtered property, 108
 filtering records, 114-115
 finding records, 112-114
 Find method, 107

- LockType property, 106
- navigation methods, 106-107
- Open method, 105-106
- printing field values, 108-110
- Sort property, 108
- SQL statements, recordsets based on, 115
- creating, with a select query, 123
- recordsets
 - dynamically populating a table and, 188-189
 - saving, 190-191
 - sharing, 447-450
- Recordsets collection, ADO, 91-93
- Recordset Type property, 522
- Record Source property, 527
- ReDim keyword, 50
- ReferenceOffice procedure, 381-383
- References collection, 337, 353, 356
 - WithEvents keyword used to trap events propagated by, 354-355
- References dialog box, 99, 413
- referential integrity, 161-162, 174-178
- refreshing field values, 443-445
- Refresh method, 444
- Relations collection, 95
- Relations object, 95
- remote data sources, 206-207
 - querying, 261-267
 - Access Projects, 263-265
 - linked ODBC sources, 261-262
 - programmatic, 265-267
- removeEmails procedure, 428
- removeOneEmail procedure, 426
- removeUserFromGroup procedure, 458
- Repair method, 89
- Replicaid property, 477-478
- Replica object, 470-471
- replica sets, 468
- Replication command, 469
- Replication Manager, 471, 571
- replication of databases, 467-496
 - Briefcase icon and, 469

design changes and, 472-476
Internet synchronization and, 471-472
JRO development techniques and, 480-496
 compacting and encrypting replicas, 495-496
 creating additional full replicas, 483-484
 creating partial replicas and filters, 484-487
 making a database replicable, 480-483
 Prevent Deletes replicas, 489-491
 replica properties, 492-495
 synchronizing replicas, 488-489
JRO programming and, 470-471
new features for, 476-480
 column-level updates, 477
 Jet-SQL Server bidirectional replication, 476
 priority-based conflict resolution, 478
 replica visibility levels, 477-478
Replication command and, 469
Replication Manager and, 471
ReplicaType property, 483, 487, 495
report class modules, 35
ReportHeader_Print procedure, 325-326
reports
 Access Projects and, 519-521
 code behind, 305-306
 controls for, 304
 creating, 301-334
 FrontPage guestbook (case study), 306-313
 manually, in Design view, 303-306
 with wizards, 302-303
 distributing, with snapshots, 320
 with dynamic formatting and content, 323-327
 formatting and adding content, 323-326
 summing page values, 326-327
 manipulating report controls and, programmatically, 330-334
 with multiple columns, 315-319
 custom reports, 317-319
 Report wizard used to create, 315-316
 programmatically manipulating report controls and enumerating reports,
 330-332
 mailing snapshots, 333-334

- modifying report control properties, 332-333
- sorting, grouping, and calculating functionality for, 314-315
- updating, programmatically, 328-329

Report wizard, 302

- multicolumn reports created with, 315-316

Requery method, 105, 444

Required property of fields, 172

ResetCounter procedure, 251-252, 254-255

Reset method, 401

Resize event, 25

restoring command bars, 401

Resume linelabel statement, 75

Resume Next statement, 75

Resume statement, 74-75, 77

Resync Command property, 524-526

resynchronizing a form with its one-to-many data, 524-526

RGB function, 72-73

Rollback method, 89, 447, 465-466

round function, 375

row-level locking, 447-449

RowLockingPessimistic procedure, 448-449

RunCommand method, 24

runFormletters procedure, 437

runLabels procedure, 437

RunLookUpProc procedure, 153-154

Run method, Application object's, 421

runMLLabels procedure, 435

S

- Save method, for recordsets, 190-191
- SaveRST procedure, 190-191
- saving queries, 202
- s_Collineage field, 473
- scope
 - of arrays, 50
 - of variables, 49
- Script object, 371
- SearchSubFolders property, 373
- Second Normal Form, 158
- security features, multi-user Access applications and, 450-464
 - alternatives to user-level security, 451-454

- custom interface, 451
- database password, 451-452
- .mde files, 453-454
- module password, 452-453
- programmatically controlling user-level security, 454-464
- Seek method, 93
- Select Case statement, 54-55
 - in error handler, 77
- SelectCommand procedure, 123
- SELECT @@IDENTITY statement, 45
- SELECT...INTO statement, 249-251
- select queries, 199-202
 - aggregate functions and, 201
 - creating a recordset with, 123
 - criteria for, 200-201
 - joins and, 201-202
 - saving, 202
 - subqueries, 233-234
 - union queries and, 231-232
 - updating source data and, 202
- SELECT statement, SQL, 115
 - Command objects used to execute SQL commands in, 245
 - DISTINCT keywords, 245
 - field lists, 244
 - FROM clause, 244
 - GROUP BY clause, 200, 212, 245, 248-249
 - HAVING clause, 245
 - inner join syntax, 246-247
 - joins in, 244
 - LIKE operators, 245
 - nested, 205
 - ORDER BY clause, 245
 - subqueries, 233-234
 - SUM aggregate function and ORDER BY clause, example using, 248-249
 - UNION SELECT statement, 231-232
 - WHERE clause, 244
- SendObject method, snapshot files and, 323
- SendSnapShots procedure, 333
- setCheckCount procedure, 390-391
- SetFocus method, 25

SetHiddenAttribute method, 294
setLabelCount procedure, 389
SetOption method, 440, 443, 445
SetPermissions method, 462-463
SetResetCounter procedure, 252-255
SetStartAndStep procedure, 145-146
SetWarnings method, 236-237, 254
s_Generation field, 473
s_GUID field, 473
sharing
 files, 440-442
 forms, 442-447
 recordsets, 447-450
showAndProcessComboBox procedure, 406-407
showClippit procedure, 400
showF1 procedure, 400
Show method, Balloon object's 389
ShowPopup method, 406
showRocky procedure, 400
Show Table dialog box, 217, 234-235
showWebBar procedure, 399
Simple Query wizard, 207-211
Simple Query Wizard dialog box, 208-209
single-row locking, 43-45
Size Mode property, 271
s_Lineage field, 473
Snapshot recordset type, 91
snapshots
 distributing reports using, 320-323, 332-333
 FTP, with Netscape browsers, 548-549
Snapshot viewer, 320-323
sorting
 the return set from a file search, 374-375
 Windows NT-compatible, 44-45
Sort property, recordset, 108
Source property, Error object's, 130
splash screens, 269-270
spreadsheets, on data access pages, 559-561
SQL statements
 aggregate functions, 68

query design and, 223-224
CREATE PROC statement, 257-261
CREATE VIEW statement, 255-257
Data Definition Language (DDL), 205
pass-through queries, 206
recordsets based on, 115
SELECT statement (see SELECT statement, SQL)
union queries, 231-232
 updating data sources with a SQL string, 340-341
SQLOLEDB driver, 265-266
SQLOleDBQuery procedure, 266-267
SQLRecordset procedure, 115
SQL Server, 43-45
 bidirectional replication between Jet and Microsoft SQL Server replicas, 476
 recovering SQL Server databases, 507
 stored procedures and, 516-519
SQLState property, Error object's, 130
squares, form for calculating, 51
standard modules. See modules
Startup dialog box, 81
start value, for auto-incrementing fields, 45
static cursor, 105
Static keyword, 50
Static statement, 50
static variables, 50
step value, for auto-incrementing fields, 45
stored procedures, 516-519
 creating, with SQL CREATE PROC statement, 257-261
 opening, 532
StrComp function, 67
String Editor, VBA, 569
string functions, 68
subdatasheets, referencing, 219-220
subforms, 279-281
subprocedures, 27-30. See also procedures
subqueries, 205, 233-234
subreports, 304-305
Sub statement, 27
subtotals, calculating, 315

SUM aggregate function, query procedure example using ORDER BY clause and, 248-249
SummaryInfo object, 371
Supports method, 106
switchboard forms, 272-276
SynchAddToFoo2 procedure, 489-491
synchFooFoo2ToDelete procedure, 489-491
synchFooNorthwindToDelete procedure, 488-489
synchNorthwindFooToAdd procedure, 488
synchronizing replicas, 488

T

TableDef object, 93-94
TableDefs collection, 93-94
TableErrCatcher error handling routine, 181
Table object, 137
 automating Word from Access and, 430-432
Table Properties dialog box, 221
Table recordset type, 91
tables, 155-197. See also Table object
 ADO techniques for programmatically creating and managing, 178-197
 avoiding replacing an existing table, 180-181
 creating a table, 178-183
 data in other formats, 192-197
 dynamically populating a table, 188-192
 indexes, 183-187
 MSDASQL provider, 194-195
 primary keys, 183-187
 recordsets, using, 188-189
 replacing a table, 182-183
 saving recordsets, 190-191
 SQL, using, 191-192
 TransferDatabase and TransferSpreadsheet methods, 195-197
based on Excel worksheets, dynamically creating, 418-421
creating, with ADOX library, 143
editing and creating using a worksheet, 512-513
editing in database diagram windows, 512
enumerating, 139-140
joining (see joins)
manually creating, 165-178
normalization of, 156-159

- opening, 530-531
- relationships between, 159-162
- replication system, 473-476
- wizards for creating, 162-165

Tables collection, 178

Tables container object, 95

Table wizard, 163-165

Table Wizard dialog box, 163-165

Terminate event, 336, 354

testformclass2 procedure, 298-299

Text fields, 166

text functions, built-in, 68

Text/HTML ISAM, 45

Text Import wizard, 306

TextOrProperty property, 375

text searches, 375-377

Third Normal Form, 158

TimedLoop procedure, 71

TimerInterval property, 271

timing the performance of code, procedure for, 69-70

toggleNewUserInAdminsGroup procedure, 461-462

toolbars. See CommandBar object transactions, 465-466

Transact-SQL, 499, 504, 516

TransferDatabase method, 195-197

TransferSpreadsheet method, 195-197

TryToDeleteFromFoo2 procedure, 489-491

Type...End Type statement, 48

TypeOf keyword, 59, 61, 292

Type property

- AccessObject object's, 357
- Column object's, 138
- Field object's, 117
- Key object's, 139
- Table object's, 137

U

- Undo Delete function, 364
- Unicode, 44, 44-45
- union queries, 204, 231-232, 516-521
- UNION SELECT statement, 231-232
- Unique Table property, 524-526

Unique Values property, 234
unmatched records, Find Unmatched Query wizard, 214
Until keyword, 62
Updatable Snapshot property, 522, 524-526
Update method, 92, 108, 111
update queries, 203, 234-237
updating
 data sources
 select queries and, 202
 with a parameter query, 342-343
 with a SQL string, 340-341
 record values, with Command object, 128-129
UserDefined object, 371
UserForms, 269
user-level security
 alternatives to, 451-454
 programmatically controlling, 452-464
 adding and deleting users, 455-457
 assigning groups to users, 457-459
 connecting to a secure database, 454-455
 creating, deleting, and tracking groups in a catalog, 459-462
 setting permissions, 462-463
User List, sharing files and, 441-442
Users collection
 ADOX library, 136
 in Jet workspaces, 86, 95

V

validating data, 172-174
Validation Rule property, 173
Validation Text property, 173
Value property, Field object's, 116-117
variables
 data types for, 45-48
 declaration of, 49-50
 public
 exposing properties with, 338-339
 property functions vs., 340
 scope of, 49
 static, 50
 user-defined, 48

Variant data type, 46, 48
VarType function, 46
VBA (Visual Basic for Applications), 21
 digitally signing VBA projects, 578-580
 productivity tools in ODE, 568-569
VBA Code Commentator, 569
VBADAOUnUsedErrorList function, 80
VBADAOUsedErrorList function, 80
VBA Multi-Code Import/Export add-in, 569
VBA String Editor, 569
VBE (Visual Basic Editor), 21, 28
 debugging in, 38-42
 opening, 36
 shortcut for toggling between Database window and, 36
 windows in, 36-38
ViewAView procedure, 147-148
View object, 146-149
views
 creating, 146-147
 with SQL CREATE VIEW statements, 255-257
 deleting, 149
 enumerating, 150-152
 modifying, 148-149
 printing, 147-148
 Query Designer and, 513-516
Views collection, 137, 146-147
Visibility property, 471, 476, 480
Visible property, CommandBar object's 399
Visual Basic 6, 45
Visual Basic Editor. See VBE (Visual Basic Editor)
Visual Basic for Applications. See VBA (Visual Basic for Applications)
Visual SourceSafe, Microsoft, 569

W

Watch window, 40
WebPageFont object, 371
Web technologies, 537-565
 data access pages (see data access pages)
 FTP snapshots with Netscape browsers, 548-549
 HTML forms, 544-547
 hyperlinks, 549-553

publishing datasheets, 538-544
WHERE clause, 200, 244
While keyword, 62
While...Wend statement, 61-62
wildcard characters, in queries, 206, 245
Windows Address Book, 45
With...End With statements, 57-61
WithEvents keyword, 338, 353-355
 trapping events propagated by the References collection with, 354-355
wizards
 AutoForm wizard, 276-277, 281
 AutoReport wizard, 302
 Chart wizard, 289
 Command Button wizard, 27
 Conflict Resolution wizard, 468, 478-480
 Database Splitter wizard, 440
 Database wizards, 162-163
 Input Mask wizard, 172-173
 Mailing Label wizard, 310-311
 Package And Deployment wizard, 570-571, 574-576
 query, 207-216
 Crosstab Query wizard, 215-216
 Find Duplicates Query wizard, 212-213
 Find Unmatched Query wizard, 214
 Simple Query wizard, 207-211
 Report wizard, 302
 Table wizard, 163-165
 Text Import wizard, 306
Word, integrating Access with, 430-437
 automation, 430-432
 form letters, 435-437
 mailing labels, 432-435
wordThere procedure, 413
Workspaces collection, 89, 90

X

xlThere procedure, 413-414

Y

Year2KTest function, 30-31

Yes/No data type, 172

Y2K (Year 2000) compliance, 32

Z

Zoom dialog box, 218, 221