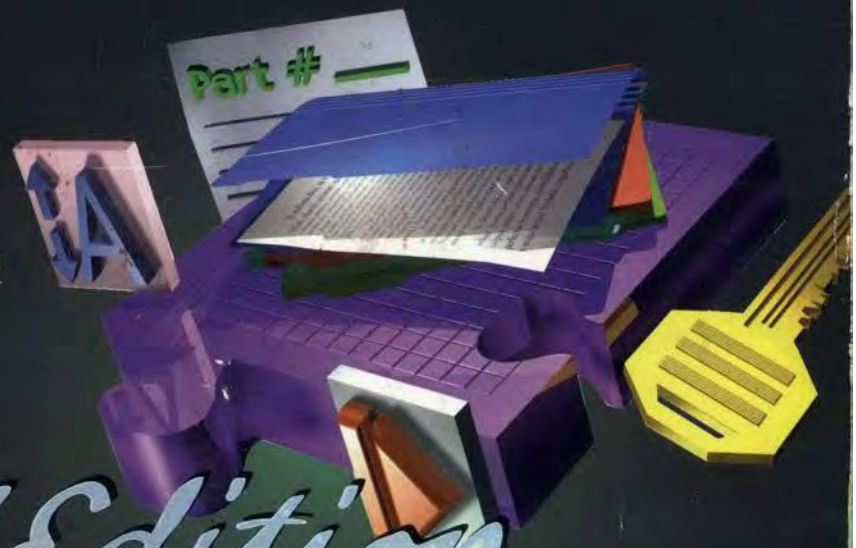
The Most Complete Reference

"Indispensable! The first Access book I get for my staff and clients."

Stan Leszynski
Leading Access Developer and Writer

- Create forms, reports, tables, and mailing labels with new Access wizards
- Export data to HTML tables and generate data-driven Web pages



Special Edition

USING MICROSOFT® ACCESS 97

Roger Jennings

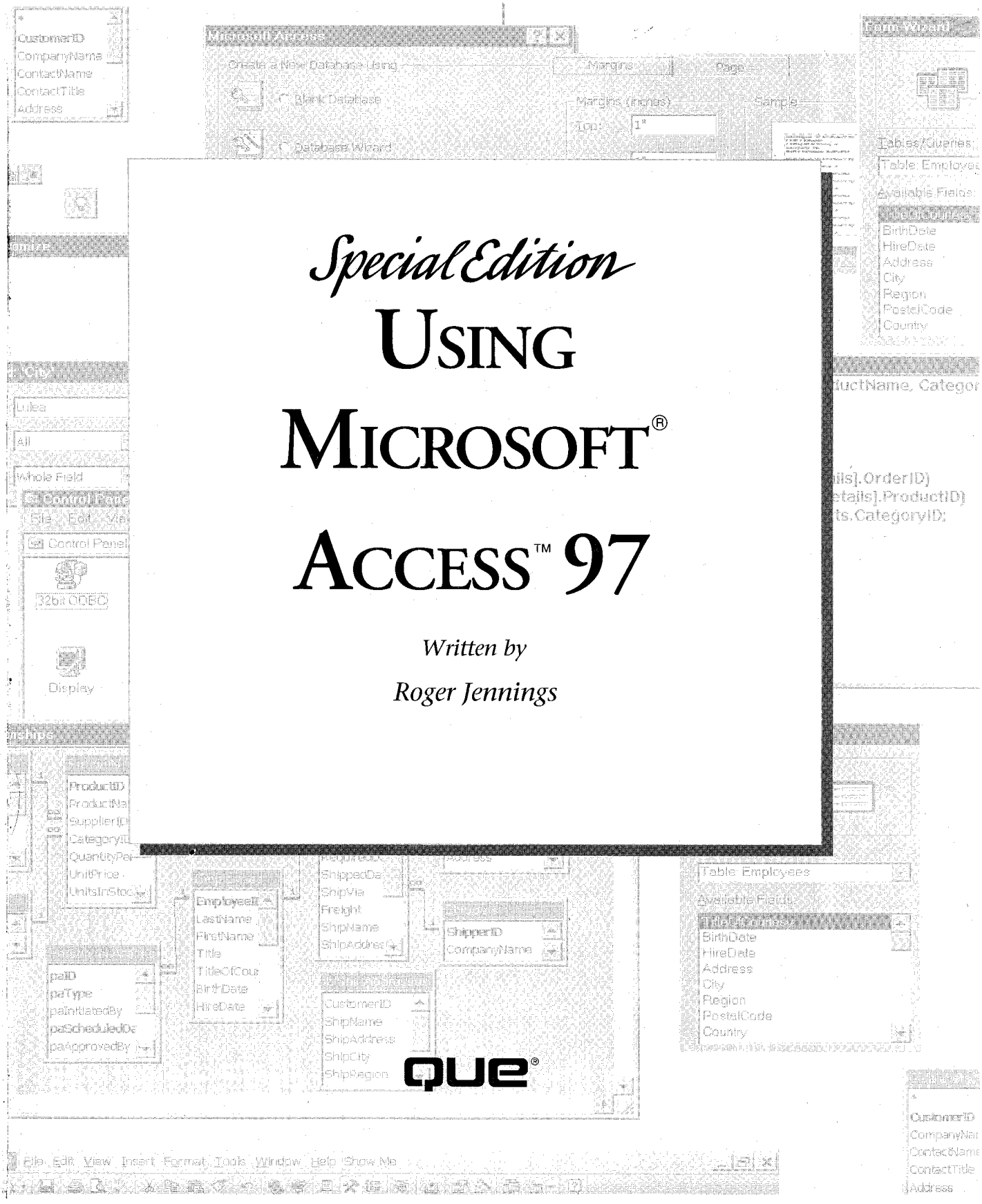
mcp.com/que

For the hottest tips, information, and resources on Office 97, visit Que's Office 97 Resource Center at: http://www.mcp.com/que/msoffice

Designed for Microsoft® Windows®95

que®

Oracle Ex. 1009, p. 1
Oracle v. DeBellis
IPR2024-00221



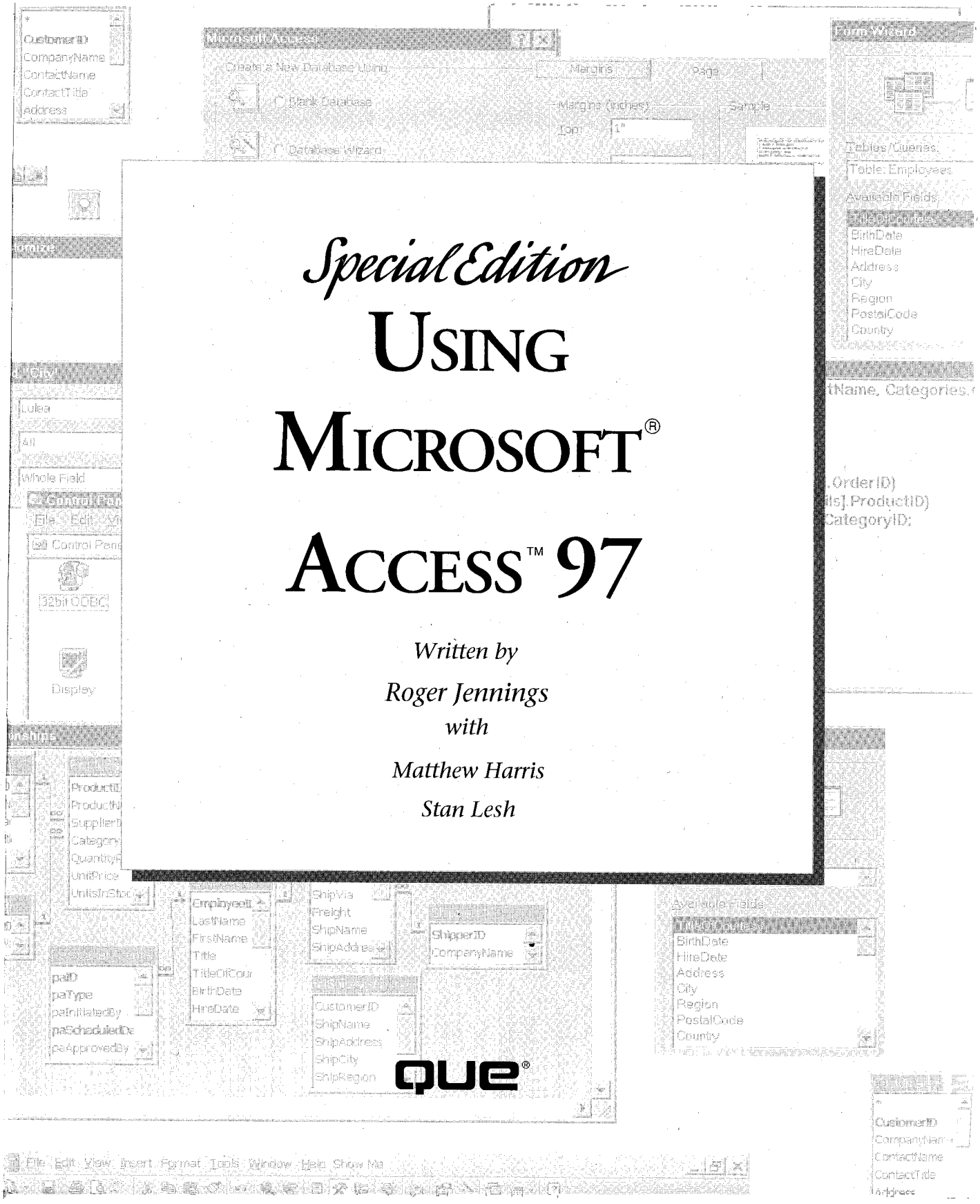
Special Edition Using Microsoft® Access™ 97

Written by

Roger Jennings

Que®

Oracle Ex. 1009, p. 2
Oracle v. DeBellis
IPR2024-00221



Special Edition Using Microsoft® Access™ 97

Written by

Roger Jennings

with

Matthew Harris

Stan Lesh

que®

Oracle Ex. 1009, p. 3
Oracle v. DeBellis
IPR2024-00221

Special Edition Using Microsoft® Access™ 97

Copyright© 1997 by Que® Corporation

All rights reserved. Printed in the United States of America. No part of this book may be used or reproduced in any form or by any means, or stored in a database or retrieval system, without prior written permission of the publisher except in the case of brief quotations embodied in critical articles and reviews. Making copies of any part of this book for any purpose other than your own personal use is a violation of United States copyright laws. For information, address Que Corporation, 201 W. 103rd Street, Indianapolis, IN, 46290. You may reach Que's direct sales line by calling 1-800-428-5331.

Library of Congress Catalog No.: 96-71443

ISBN: 0-7897-0916-3

This book is sold *as is*, without warranty of any kind, either express or implied, respecting the contents of this book, including but not limited to implied warranties for the book's quality, performance, merchantability, or fitness for any particular purpose. Neither Que Corporation nor its dealers or distributors shall be liable to the purchaser or any other person or entity with respect to any liability, loss, or damage caused or alleged to have been caused directly or indirectly by this book.

99 98 97 6 5 4 3 2 1

Interpretation of the printing code: the rightmost double-digit number is the year of the book's printing; the rightmost single-digit number, the number of the book's printing. For example, a printing code of 96-1 shows that the first printing of the book occurred in 1996.

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Que cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Screen reproductions in this book were created using Collage Plus from Inner Media, Inc., Hollis, NH.

Composed in *Stone Serif* and *MCPdigital* by Que Corporation

Oracle Ex. 1009, p. 4
Oracle v. DeBellis
IPR2024-00221

Contents at a Glance

Introduction 1

Learning Access Fundamentals 19

1 Access 97 for Access 95 and 2.0 Users: What's New 21
2 Up and Running with Access 97 47
3 Navigating Within Access 63
4 Working with Access Databases and Tables 107
5 Entering, Editing, and Validating Data in Tables 161
6 Sorting, Finding, and Filtering Data in Tables 179
7 Linking, Importing, and Exporting Tables 203

Learning Fundamentals

Querying for Specific Information 259

8 Using Querying by Example 261
9 Understanding Operators and Expressions in Access 287
10 Creating Multitable and Crosstab Queries 321
11 Using Action Queries 371

for Information

Creating Forms and Reports 393

12 Creating and Using Forms 395
13 Designing Custom Multitable Forms 443
14 Printing Basic Reports and Mailing Labels 513
15 Preparing Advanced Reports 551

Creating Forms and Reports

Publishing Data on Intranets and the Internet 573

16 Working with Hyperlinks and HTML 575
17 Exporting Data to World Wide Web Pages 597
18 Creating Dynamic Web Pages 625

Publishing Data

Integrating Access with Other Office 97 Applications 663

19 Using 32-Bit OLE Components 665
20 Adding Graphics to Forms and Reports 691
21 Using Access with Microsoft Excel 729
22 Using Access with Microsoft Word and Mail Merge 763

Integrating Access

Using Advanced Access Techniques 785

23 Exploring Relational Database Design and Implementation 787
24 Working with Structured Query Language 835
25 Securing Multiuser Network Applications 865
26 Connecting to Client/Server Databases 913
27 Replicating Access Databases 955

Using Advanced Techniques

Oracle Ex. 1009, p. 5
Oracle v. DeBellis
IPR2024-00221

Programming, Distributing, and Converting Access Applications **975**

28 Writing Visual Basic for Applications Code 977

29 Understanding the Data Access Object Class 1023

30 Responding to Events with VBA 5.0 1049

31 Exchanging Data with Automation and ActiveX Controls 1095

32 Using the Office 97 Developer Edition 1135

33 Migrating Access 2.0 and Access 95 Applications to Access 97 1151

Developing Access

Appendixes **1161**

A Glossary 1163

B Naming Conventions for Access Objects and Variables 1207

C Data Dictionary for the Personnel Actions Table 1231

Appendixes

Oracle Ex. 1009, p. 6
Oracle v. DeBellis
IPR2024-00221

Contents

Introduction 1

Who Should Read This Book? 4
How This Book Is Organized 5
Part I, "Learning Access Fundamentals" 5
Part II, "Querying for Information" 6
Part III, "Creating Forms and Reports" 6
Part IV, "Publishing Data on Intranets and the Internet" 7
Part V, "Integrating Access with Other Office 97 Applications" 7
Part VI, "Using Advanced Access Techniques" 8
Part VII, "Programming, Distributing, and Converting Access Applications" 8
Part VIII: "Appendixes" 9
How This Book Is Designed 10
Typographic Conventions Used in This Book 11
Key Combinations and Menu Choices 11
SQL Statements and Keywords in Other Languages 11
Typographic Conventions Used for VBA 12
System Requirements for Access 97 13
Other Sources of Information for Access 14
Books 14
Periodicals 15
Online Sources 16

I Learning Access Fundamentals 19

1 Access 97 for Access 95 and 2.0 Users: What's New 21

Analyzing Access 97's Enhancements and Changes 22
Major Access 97 Enhancements 22
Important Access 97 Changes 27
Summarizing Access 95's Improvements to Access 2.0 29
Installing Access 97 32
Making an Initial Installation of Access 97 32
Customizing a Typical Installation 40
Installing ValuPack Components 42
Installing the Office Developer Edition Components 44

Oracle Ex. 1009, p. 7
Oracle v. DeBellis
IPR2024-00221

2 Up and Running with Access 97 47

Starting Access and Opening Databases 47
Opening a Database During Access Startup 48
Opening a Database When Access is Already Running 50
Understanding the Database Window 51
Using Database Tables 52
Viewing Data in Tables 52
Selecting and Editing Data Records 53
Viewing and Editing Graphics 56
Using Design Mode 57
Printing the Contents of a Table 59

3 Navigating Within Access 63

Understanding Access's Functions and Modes 63
Defining Access Functions 63
Defining Access Operating Modes 66
Understanding Access's Table Display 67
Maximized Document Windows 69
Document Windows Minimized to Icons 70
The Toolbars in Table View 71
Manipulating Access's Toolbars 75
Right Mouse Button Shortcut Menus 78
Using the Function Keys 79
Global Function Keys 79
Function-Key Assignments for Fields, Grids, and Text Boxes 80
Function Keys in the Module Window 80
Setting Default Options 81
System Defaults 82
Defaults for Datasheet View 88
Using Access Help 89
Context-Sensitive Help 90
The Help Menu 91
Using the Help Topics Dialog 91
Using the Office Assistant 97
Using the Database Utilities 102
Compacting Databases 102
Converting Databases to Access 97 Format 104
Repairing Databases 104
Creating MDE Files 105

4 Working with Access Databases and Tables 107

Defining the Elements of Access Databases 107
Understanding Relational Databases 110
Using Access Database Files and Tables 111
The Access System Database 112
Access Library Databases 112

Oracle Ex. 1009, p. 8
Oracle v. DeBellis
IPR2024-00221

Creating a New Database 112
Understanding the Properties of Tables and Fields 115
Choosing Field Data Types, Sizes, and Formats 119
Choosing Field Sizes for Numeric and Text Data 120
Selecting a Display Format 123
Using Input Masks 128
Using the Northwind Traders Sample Database 130
Using the Table Wizard to Create New Tables 130
Creating a Table Directly in Datasheet View 134
Adding a New Table to an Existing Database 137
Designing the Personnel Actions Table 138
Creating the Personnel Actions Table 141
Setting Default Values of Fields 145
Working with Relations, Key Fields, and Indexes 146
Establishing Relationships Between Tables 146
Selecting a Primary Key 152
Adding Indexes to Tables 154
Altering Fields and Relationships 155
Rearranging the Sequence of Fields in a Table 155
Changing Field Data Types and Sizes 156
Changing Relationships Between Tables 158
Copying and Pasting Tables 159

5 Entering, Editing, and Validating Data in Tables 161

Using Keyboard Operations for Entering and Editing Data 162
Creating an Experimental Copy of Northwind.mdb 162
Using Data Entry and Editing Keys 163
Using Key Combinations for Windows Clipboard Operations 165
Using Shortcut Keys for Fields and Text Boxes 166
Setting Data Entry Options 167
Adding Records to a Table 168
Selecting, Appending, Replacing, and Deleting Table Records 170
Validating Data Entry 170
Adding Field-Level Validation Rules 171
Adding Table-Level Validation Rules and Using the Expression Builder 173
Adding Records to the Personnel Actions Table 175
Entering Personnel Actions Table Data and Testing Validation Rules 177

6 Sorting, Finding, and Filtering Data in Tables 179

Sorting Table Data 179
Freezing Display of a Table Field 180
Sorting Data on a Single Field 181
Sorting Data on Multiple Fields 182
Removing a Table Sort Order and Thawing Columns 182

Oracle Ex. 1009, p. 9
Oracle v. DeBellis
IPR2024-00221

Finding Matching Records in a Table 183
Replacing Matched Field Values Automatically 186
Filtering Table Data 186
Filtering by Selection 187
Filtering by Form 189
Advanced Filters and Sorts 193
Adding a Multi-Field Sort and Compound Filter Criteria 193
Using Composite Criteria 195
Saving Your Filter as a Query and Loading a Filter 197
Customizing Datasheet View 199
Copying, Exporting, and Mailing Sorted and Filtered Data 201

7 Linking, Importing, and Exporting Tables 203

Learning How Access Handles Tables in Other Database File Formats 204
Identifying PC Database File Formats 205
Linking and Importing External Tables 206
Solving Problems with Importing or Linking Files 209
Dealing with Graphics in External Files 210
Converting Field Data Types to Access Data Types 212
Setting Primary Keys and Using Index Files 213
Linking Tables from Other Access Databases 217
Using the Linked Table Manager Add-In to Relink Tables 218
Importing versus Linking Database Files as Tables 219
Importing and Linking Spreadsheet Files 220
Creating a Table by Importing an Excel Worksheet 221
Linking an Excel Worksheet by Using the ODBC API 228
Importing Text Files 233
Using the Text Import Wizard 234
The Text Import Wizard Advanced Options 239
Using Delimited Text Files 241
Handling Fixed-Width Text Files 244
Appending Text Data to an Existing Table 245
Using the Clipboard to Import Data 245
Pasting New Records to a Table 246
Replacing Records by Pasting from the Clipboard 249
Modifying Linked and Imported Tables 251
Restructuring the Presentation of Tables 251
Adding Fields to Tables with Imported Data 251
Changing Field Data Types of Imported Files 251
Adding or Modifying Other Field Properties 252
Deleting Redundant Fields in Imported Tables 252
Exporting Data from Access Tables 253
Exporting Data Through the Windows Clipboard 253
Exporting Data as a Text File 255
Exporting Data in Other File Formats 257

Oracle Ex. 1009, p. 10
Oracle v. DeBellis
IPR2024-00221

II Querying for Specific Information 259

8 Using Query by Example 261

Introducing Queries 261
Creating Your First Real Query 262
Choosing Fields for Your Query 264
Selecting Records by Criteria and Sorting the Display 267
Creating More Complex Queries 269
Editing Table Data in Query View 271
Changing the Names of Query Column Headers 271
Printing Your Query as a Report 273
Using the Data from Your Query 276
Saving and Naming Your Query 278
Creating Other Types of Queries 279
Creating and Using a Simple Make-Table Action Query . 280
Adding a Parameter to Your Make-Table Query 282
Translating Graphical QBE to Structured Query Language 284

9 Understanding Operators and Expressions in Access 287

Understanding the Elements in Expressions 288
Operators 289
Literals 293
Identifiers 294
Functions 294
Intrinsic and Named Constants 309
Creating Access Expressions 309
Expressions for Creating Default Values 309
Expressions for Validating Data 310
Expressions for Query Criteria 310
Expressions for Calculating Query Field Values 316
Other Uses for Expressions 319

10 Creating Multitable and Crosstab Queries 321

Joining Tables to Create Multitable Queries 322
Creating Conventional Single-Column Equi-Joins 323
Specifying a Sort Order for the Query Result Set 326
Creating Queries from Tables with Indirect Relationships 326
Creating Multicolumn Equi-Joins and Selecting Unique Values 329
Using Lookup Fields in Tables 332
Adding a Foreign Key Drop-Down List with the Lookup Wizard 333
Adding a Fixed Value Lookup List to a Table 337

Oracle Ex. 1009, p. 11
Oracle v. DeBellis
IPR2024-00221

Outer, Self, and Theta Joins 340
Creating Outer Joins 340
Creating Self-Joins 342
Creating Not-Equal Theta Joins with Criteria 344
Updating Table Data with Queries 345
Characteristics That Determine Whether You Can Update a Query 346
Formatting Data with the Output Field Properties Window 347
Making All Fields of Tables Accessible 349
Using the Field List Title Bar to Add All Fields of a Table 349
Using the Asterisk to Add All Fields Without Field Names 350
Selecting All Fields with the Output All Fields Property .. 350
Making Calculations on Multiple Records 351
Using the SQL Aggregate Functions 351
Making Calculations Based on All Records of a Table 352
Making Calculations Based on Selected Records of a Table 356
Designing Parameter Queries 358
Adding a Parameter to the Monthly Sales Query 359
Specifying the Parameter's Data Type 360
Creating Crosstab Queries 361
Creating a Monthly Product Sales Crosstab Query 362
Using Fixed Column Headings with Crosstab Queries 363
Decreasing the Level of Detail in Crosstab Queries 365
Creating Queries from Tables in Other Databases 367

11 Using Action Queries 371

Creating New Tables with Make-Table Queries 371
Designing and Testing the Select Query 372
Converting the Select Query to a Make-Table Query 374
Establishing Relationships for the New Table 376
Using the New tblShipAddresses Table 377
Creating Action Queries to Append Records to a Table 378
Deleting Records from a Table with an Action Query 380
Updating Values of Multiple Records in a Table 383
Testing Cascading Deletion and Updating 388
Creating the Test Tables and Establishing Relationships 388
Testing Cascading Deletion 390
Testing Cascading Updates 390

Oracle Ex. 1009, p. 12
Oracle v. DeBellis
IPR2024-00221

III Creating Forms and Reports 393

12 Creating and Using Forms 395

Identifying Types of Forms 396
Forms for Transaction Processing 396
Forms for Decision Support 397
Creating a Transaction-Processing Form with the Form Wizard 397
Choosing the Form Type 398
Creating the Basic Form with the Form Wizard 399
Using the Form Design Window 404
Elements of the Form Design Window 405
Form Design Toolbar Buttons and Menu Choices 406
The Formatting Toolbar 409
Default Values for Forms 410
Using AutoFormat 411
Changing an Object's Colors 413
Selecting, Editing, and Moving Form Elements and Controls 419
Changing the Size of the Form Header and Form Footer 421
Selecting, Moving, and Sizing a Single Control 421
Aligning Controls to the Grid 422
Selecting and Moving a Group of Controls 423
Aligning a Group of Controls 424
Using the Windows Clipboard and Deleting Controls 424
Changing the Color and Border Style of a Control 424
Changing the Content of Text Controls 425
Using the Format Painter 426
Rearranging the Personnel Actions Form 426
Setting Properties of the Main Form 427
Setting the Properties of a Subform 429
Using Transaction-Processing Forms 431
Toolbar Buttons in Form View 431
Using the Personnel Actions Form 433
Appending New Records to the Personnel Actions Table 433
Editing Existing Data 436
Committing and Rolling Back Changes to Tables 436
Deleting a Record in Form View 437
Modifying the Properties of a Form or Control After Testing 437
Changing the Order of Fields for Data Entry 438
Removing Fields from the Tab Order 439
Changing the Source of Data for the Subform and Sorting the Subform Data 439

Oracle Ex. 1009, p. 13
Oracle v. DeBellis
IPR2024-00221

13 Designing Custom Multitable Forms 443

Understanding the Access Toolbox 443
Control Categories 444
The Toolbox 444
Access's Control Wizards, Builders, and Toolbars 446
The Appearance of Controls in Design and Run Modes .. 450
Using the Toolbox to Add Controls 451
Creating the Query on Which to Base the Main Form 452
Creating a Blank Form with a Header and Footer 454
Adding a Label to the Form Header 455
Formatting Text and Adjusting Text Control Sizes 457
Creating Bound, Multiline, and Calculated Text Boxes ... 458
Changing the Default View and Obtaining Help for Properties 464
Adding Option Groups, Binding Controls, and Locking Tools 464
Using the Clipboard to Copy Controls to Another Form 472
Using List Boxes and Combo Boxes 474
Creating a Combo Box to Find Specific Records 486
Creating a Tab Control 489
Changing One Control Type to Another 495
Completing the Main Personnel Actions Entry Form 495
Creating a Subform Using the Subform/Subreport Wizard 499
Creating and Using Continuous Forms 503
Overriding the Field Properties of Tables 509
Adding Page Headers and Footers for Printing Forms 510

14 Printing Basic Reports and Mailing Labels 513

Differences and Similarities Between Forms and Reports 513
Types of Access Reports 514
Creating a Grouping Report with the Report Wizard 516
Using Access's Report Windows 523
Using AutoFormat and Customizing Report Styles 526
Modifying a Basic Report Wizard Report 526
Deleting, Relocating, and Editing Existing Controls 526
Using the DLookUp() Domain Aggregate Function to Print Product Category Names 532
Adding Other Calculated Controls to the Report 534
Aligning and Formatting Controls, and Adjusting Line Spacing 539
Adjusting Margins and Printing Conventional Reports 543
Preventing Widowed Records with the Group Keep Together Property 546
Printing Multicolumn Reports as Mailing Labels 547

Oracle Ex. 1009, p. 14
Oracle v. DeBellis
IPR2024-00221

15 Preparing Advanced Reports 551

Grouping and Sorting Report Data 551
Grouping Data 552
Sorting Data Groups 555
Working from a Blank Report 555
The Basis for a Subreport 555
Adding and Deleting Sections of Your Report 557
Controlling Page Breaks and the Printing of Page Headers and Footers 559
Creating the Monthly Sales by Category Report 559
Incorporating Subreports 564
Adding a Linked Subreport to a Bound Report 564
Using Unlinked Subreports and Unbound Reports 568
Adding Other Controls to Reports 569
Sending Reports by Microsoft Exchange and Outlook 570

IV Publishing Data on Intranets and the Internet 573

16 Working with Hyperlinks and HTML 575

Putting Microsoft's Internet Program in Perspective 575
An Overview of Microsoft's Internet Products 576
Microsoft Technologies Supporting Internet-Related Products 580
Navigating with Hyperlinks 583
Hyperlinks and File Paths 584
Hyperlinks and Intranets 584
Understanding Access 97's Hyperlink Field Data Type 585
Testing Hyperlinks in the Northwind Orders Table 586
Editing and Inserting Conventional Hyperlinks 586
Linking to Bookmarks in a Word Document 588
Specifying Hyperlinks to Pages on an Intranet Server 590
Using Hyperlinks to Open an Access Object 592
Specifying Internet Uniform Resource Locators 594
Using Hyperlinks with Access Controls 595

17 Exporting Data to World Wide Web Pages 597

Exporting Table and Query Datasheets to HTML 597
Creating an Unformatted Web Page 601
Creating a Web Page from a Query 603
Using HTML Templates 606
Using the Access HTML Templates Included with Office 97 606
Exporting a Query Datasheet with a Template 608

Oracle Ex. 1009, p. 15
Oracle v. DeBellis
IPR2024-00221

Exporting Reports to HTML 610
Using the Web Publishing Wizard
to Create a Static Web Site 614
Importing Data from HTML Tables 618

18 Creating Dynamic Web Pages 625

Understanding Microsoft's View of Databases and the Web ... 626
Rolling Up Data to Marts and Malls 626
Managing Distributed Transactions 627
Using the IDC for Delivering Information 627
The IDC Demonstration Applications Included
with IIS 2.0 628
A Dynamic Web-Based Access Application 635
Exporting Datasheets to Dynamic Web Pages 637
Creating a System ODBC Data Source by
Using Access 97 637
Specifying a Default ODBC Data Source 642
Using HTML Export to Create the Web Pages 643
Exporting Tables, Queries, and Forms
to Active Server Pages 652
Creating \InetPub Subfolders for ASP Files 652
Exporting a Query to an ASP File 655
Exporting Access Forms to ASP Files 660

V Integrating Access with Other Office 97 Applications 663

19 Using 32-Bit OLE Components 665

Understanding the Importance of OLE 665
Defining OLE 666
Classifying OLE 2.1-Compliant Applications 668
Introducing Object Building Blocks 670
Properties and Methods Encapsulated in Objects 670
Object-Enabled Applications 671
The Advantages of Objects 671
Object Classes, Types, Hierarchies, and Inheritance 672
Containing Objects in Other Objects 673
Understanding Differences Between OLE and DDE 675
Embedding versus Linking Source Documents 676
Activating OLE 2+ Objects in Place 680
Taking Advantage of Automation with OLE Objects 681
The Windows Registry 682
The Registry Editor 683
How Access and Other OLE Applications Use
the Registry 686
Replacing the Registry Files with Backups 687

Oracle Ex. 1009, p. 16
Oracle v. DeBellis
IPR2024-00221

20 Adding Graphics to Forms and Reports 691

Adding a Bound Object Control to a Form or Report 691
Including Photos in the Personnel Actions Query 692
Displaying the Employee's Picture in the Personnel Actions Form 693
Scaling Graphic Objects 696
Examining Bitmap Image File Formats 699
Adding an Unbound Object Frame to a Form or Report 701
Creating a New Logo for Your Personnel Actions Form 702
Sizing and Setting the Properties of the Unbound Object Frame 705
Using the Image Control 707
Creating Graphs and Charts with Microsoft Graph 97 709
Creating the Query on which to Base the Graph 709
Using the Chart Wizard to Create an Unlinked Graph 711
Modifying the Design Features of Your Graph 715
Creating a Graph from a Crosstab Query 721
Linking the Graph to a Single Record of a Table or Query 723

21 Using Access with Microsoft Excel 729

Importing and Reorganizing Worksheet Data 730
Obtaining the Example Worksheet 731
Importing the Example Worksheet 732
Developing a Conversion Strategy 735
Extracting Data 737
Modifying the Table Structure 739
Adding the Type Value to the Close Table 742
Creating the Final Tables 743
Verifying the Tables by Re-Creating the Worksheet 746
Using the Tables You Created 750
Exporting Stock Data as a Worksheet 754
Using Excel as an OLE Server 754
Embedding an Excel Worksheet in a Form 755
Extracting Values from an OLE Object 759
Linking to a Range of Cells in an Excel Worksheet 760

22 Using Access with Microsoft Word and Mail Merge 763

Using the Access Mail Merge Wizard 763
Creating and Previewing a New Form Letter 764
Using an Existing Main Merge Document with a New Data Source 767

Oracle Ex. 1009, p. 17
Oracle v. DeBellis
IPR2024-00221

Using Word 97's Mail Merge Feature with Access Databases 770
Creating a New Mail Merge Data Source with Microsoft Query and an ODBC Data Source 770
Creating Form Letters from an Existing Query 777
Embedding or Linking Word Documents in Access Tables 778
Embedding or Linking a Word 97 Document in a Table 779
Creating a Form to Display the Document 781

VI Using Advanced Access Techniques 785

23 Exploring Relational Database Design and Implementation 787

Integrating Objects and Relational Databases 787
Understanding Database Systems 788
The Objectives of Database Design 788
The Process of Database Design 790
The Object-Oriented Approach to Database Design 790
Database Terminology 793
Types of Tables and Keys in Relational Databases 796
Data Modeling 797
Using Data Modeling Tools to Create Access Databases 800
Normalizing Data to the Relational Model 802
Normalization Rules 802
Types of Relationships 809
Using Access 97's Table Analyzer Wizard 813
Working with Data Dictionaries 818
Conventional Data Dictionaries 818
Access's Integrated Data Dictionary System 819
Using Access Indexes 823
Indexed Sequential Access Method 824
Access Data Pages and Variable Length Records 825
Balanced Binary Trees, Roots, and Leaves 826
Nonclustered and Clustered Indexes 827
Query Speed versus Update Performance 828
Enforcing Database Integrity 829
Ensuring Entity Integrity and Auditability 830
Maintaining Referential Integrity 834

24 Working with Structured Query Language 835

Using Access to Learn SQL 835
What Is Structured Query Language? 836
Looking at the Development of SQL 837
Comparing ANSI and Jet SQL 838
SQL Reserved Words in Access 839

Oracle Ex. 1009, p. 18
Oracle v. DeBellis
IPR2024-00221

Jet SQL Reserved Words Corresponding to ANSI SQL Keywords 840
Access Functions and Operators Used in Place of ANSI SQL Keywords 840
Jet SQL Reserved Words, Operators, and Functions Not in ANSI SQL 841
Jet's *DISTINCTROW* and SQL's *DISTINCT* Keywords 842
Creating Tables with Jet DDL 846
Common ANSI SQL Keywords and Features Not Supported by Jet SQL Reserved Words 846
Writing Select Queries in SQL 847
Using SQL Punctuation and Symbols 849
Using SQL Statements to Create Access Queries 850
Using the SQL Aggregate Functions 852
Creating Joins with SQL 855
Using *UNION* Queries 857
Implementing Subqueries 858
Specifying Action Query Syntax 860
Adding *IN* to Use Tables in Another Database 861
Working with Another Access Database 861
Using the *IN* Clause with Other Types of Databases 863
Using SQL Statements in Forms, Reports, and Macros 863

25 Securing Multiuser Network Applications 865

Networking Access Applications 865
Installing Access in a Networked Environment 866
Sharing Your Access Database Files with Other Users 868
Creating a Folder and System File for File Sharing 868
Preparing to Share Your Database Files 872
Splitting Databases for File Sharing 873
Choosing Workgroups with the Workgroup Administrator 876
Using Command-Line Options to Open a Shared Database 877
Maintaining Database Security 881
Specifying the Principles of Database Security on a LAN 881
Password-Protecting a Single Database 883
Managing Groups and Users 884
Understanding Database Object Ownership 892
Owner Permissions for Objects 892
Changing the Ownership of Database Objects 892
Granting and Revoking Permissions for Database Objects 895
Altering Group Permissions 897
Granting Additional Permissions to Specific Users 900
Granting Permissions for a Database in Another Workgroup 901
Sharing Databases on the Network 902

Oracle Ex. 1009, p. 19
Oracle v. DeBellis
IPR2024-00221

Sharing Database Files on a Windows 95 Network 902
Sharing Files with User-Level Security 904
Sharing Database Files from a Network Server 907
Accessing the Shared Files 907
Administering Databases and Applications 910
Backing Up and Restoring Databases 910
Compacting and Repairing Database Files 911
Encrypting and Decrypting Database Files 911

26 Connecting to Client/Server Databases 913

Creating Scalable Access Applications 913
Defining the Client/Server Environment 915
Defining Open Database Connectivity 917
Understanding ODBC Drivers 918
Installing the Microsoft ODBC Driver for SQL Server 920
Adding and Removing SQL Server Data Sources 924
Using the ODBC Data Source Administrator to Create a SQL Server Data Source 924
Creating a User or System SQL Server Data Source Within Access 97 929
Creating a File Data Source Within Access 97 931
Using Databases Connected by ODBC 935
Using Access 97's SQL Pass-Through Queries 940
Executing a Simple Pass-Through Query 940
Setting the Connect String Property for SQL Pass-Through Queries 942
Logging ODBC Queries 944
Exporting Access Tables to a Client/Server Database 945
Using Access's Built-In Export Feature 945
Using the Access 97 Upsizing Wizard 947
Using the SQL Server Browser Add-In 951

27 Replicating Access Databases 955

Understanding the Principles of Briefcase Replication 956
Creating and Updating a Briefcase Replica 957
Creating the Contact Management Application for Replication 958
Creating a Replica Set 962
Synchronizing the Design-Master and Briefcase Replicas 965
Emulating Replica Set Reconciliation with Disks 967
Replicating Part of a Database 971
Obtaining and Installing the Partial Replica Wizard 972
Using the Partial Replica Wizard 972

Oracle Ex. 1009, p. 20
Oracle v. DeBellis
IPR2024-00221

VII Programming, Distributing, and Converting Access Applications 975

28 Writing Visual Basic for Applications Code 977

Introducing Access VBA 978
Access VBA Compared with xBase, PAL, and Visual VBA 979
Where You Use Access VBA Code 980
Typographic and Naming Conventions Used for Access VBA 980
Modules, Functions, and Subprocedures 981
References to VBA and Access Modules 985
Data Types and Database Objects in Access VBA 986
Variables and Naming Conventions 988
Symbolic Constants 995
Access VBA Named and Optional Arguments 997
Controlling Program Flow 998
Branching and Labels 998
Conditional Statements 999
Repetitive Operations: Looping 1002
Handling Run-Time Errors 1005
Detecting the Type of Error with the Err Function 1006
Using the Error Event in Form and Report Modules 1007
Exploring the Module Window 1007
The Toolbar of the Module Window 1008
Module Shortcut Keys 1010
Menu Commands in the Module Window 1010
The Access VBA Help System 1011
The Access VBA Compiler 1012
Examining the Utility Functions Module 1014
Adding a Breakpoint to the IsLoaded() Function 1015
Using the Debug Window 1016
Printing to the Debug Window with the Debug Object 1018
Using Text Comparison Options 1019
Writing Your Own Functions and Procedures 1020

29 Understanding the Data Access Object Class 1023

Understanding Objects and Object Collections 1023
Naming Standards for Object Collections 1024
Creating a Reference to the Data Access Object 1025
Using the Object Browser 1027
Referring to Data Access Objects in VBA 1029
Properties of and Methods Applicable to DAOs 1030

Oracle Ex. 1009, p. 21
Oracle v. DeBellis
IPR2024-00221

Creating Object Variables 1032
Using Object Variables in the Debug Window 1033
Creating New Data Access Objects 1033
Opening a New Recordset Object 1034
Defining a New QueryDef Object 1040
Writing a Function that Uses Database Objects 1041
Using ODBCDirect and the Remote Data Object 1044
Experimenting with ODBCDirect 1046

30 Responding to Events with VBA 5.0 1049

Understanding the Role of Class Modules 1049
Event-Handling Code in the Main Switchboard Form 1050
Understanding Access 97's Event Repertoire 1062
Working with Access 97's *DoCmd* Methods 1068
Arguments of *DoCmd* Methods 1070
Replacing *AddMenu* Macro Actions with *CommandBar* Objects 1075
Replacing the AutoExec Macro and Specifying a Custom *CommandBar* with Startup Properties 1079
Converting Access Macros to VBA Code 1080
Referring to Access Objects with VBA 1087
Referring to Open Forms or Reports and Their Properties 1087
Referring to Controls and Their Properties 1088
Referring to Controls on a Subform or the Main Form 1089
Responding to Data Events 1090

31 Exchanging Data with Automation and ActiveX Controls 1095

Introducing Microsoft's ActiveX Technology 1096
Understanding Automation 1097
Categorizing Automation Components 1097
Interacting with 16-Bit OLE Automation Servers 1099
Using Automation Servers with Access 97 1100
Manipulating an Excel 97 Workbook Object 1102
Creating a New Excel Worksheet with Access VBA Code 1112
Sending Data to Microsoft Word 97 with Automation 1116
Adding ActiveX Controls to Your Application 1122
Registering OLE Controls 1123
Using the Calendar Control 1124
Using Access 97 as an Automation Server 1131

Oracle Ex. 1009, p. 22
Oracle v. DeBellis
IPR2024-00221

32 Using the Office 97 Developer Edition 1135

Examining the Content of the Office 97 Developer Edition 1135
Differences Between Run-Time and Retail Access 1136
Creating Distribution Disks with the Setup Wizard 1138
Preparing Sample Files to Use the Setup Wizard 1138
Creating Distribution Disk Images 1138
Using the 32-Bit Windows API Viewer 1147

33 Migrating Access 2.0 and Access 95 Applications to Access 97 1151

Using Access 2.0 Application .MDB Files with Access 97 1151
Using Access 1.x and 2.0 Data .MDB Files with Access 97 1154
Converting Access 1.x and 2.0 Files to Access 95 1155
Converting versus Importing .MDB Files 1155
Handling Conversion and Import Errors 1156
Converting Access 95 Databases to Access 97 Format 1159

VIII Appendixes 1161

A Glossary 1163

B Naming Conventions for Access Objects and Variables 1207

The Leszynski Naming Conventions for Microsoft Access: A Primer 1208
Access Object Types 1209
Structuring Object Names 1210
What Is a Prefix? 1210
What Is a Tag? 1211
What Is a Base Name? 1211
What Is a Qualifier? 1211
What Is a Suffix? 1212
Creating Database Object Base Names 1212
Rules for Base Names 1212
Base Name Length Limits 1213
Compound Base Names 1214
Field Base Names 1214
Ordering Base Name Elements 1214
Naming Conventions for Database Objects 1215

Oracle Ex. 1009, p. 23
Oracle v. DeBellis
IPR2024-00221

Tags for Database Window Objects 1215
Tags for Form and Report Control Objects 1217
Using Menu Macros 1218
Creating VBA Object Base Names 1219
Rules for Base Names 1219
Base Name Lengths 1219
Compound Base Names 1220
Naming Conventions for VBA Objects 1220
Tags for Variables 1221
Prefixes for Variables 1224
Naming Constants 1225
Tags and Prefixes for Procedures 1225
Using Macros Instead of VBA 1226
VBA Object Name Examples 1227
Creating Your Own Tags 1227
How to Get the Complete Text of the LNCs 1229

C Data Dictionary for the Personnel Actions Table 1231

Oracle Ex. 1009, p. 24
Oracle v. DeBellis
IPR2024-00221

Credits

President
Roland Elgey

Publishing Director
David W. Solomon

Title Manager
Kathie-Jo Arnoff

Editorial Services Director
Elizabeth Keaffaber

Managing Editor
Michael Cunningham

Director of Marketing
Lynn E. Zingraf

Acquisitions Manager
Elizabeth South

Acquisitions Editor
Angela Wethington

Senior Product Director
Lisa D. Wagner

Product Director
Kevin Kloss

Production Editor
Lisa M. Gebken

Editors
Sarah R. Backer
Kate Givens
Linda Seifert
Nick Zafran

Strategic Marketing Manager
Barry Pruett

Product Marketing Manager
Kris Ankney

Assistant Product Marketing Managers
Karen Hagen
Christy M. Miller

Technical Editors
Jim O'Connor
Tim Schubach
Sara Strock

Media Development Specialist
David Garratt

Technical Support Specialist
Nadeem Muhammed

Acquisitions Coordinator
Tracy C. Williams

Software Relations Coordinators
Patty Brooks
Susan Gallaher

Editorial Assistant
Virginia Stoller

Book Designer
Ruth Harvey

Cover Designer
Dan Armstrong

Production Team
Marcia Brizendine
Jason Carr
Jenny Earhart
Brian Flores
Jessica Ford
Staci Somers

Indexers
Chris Barrick
Brad Herriman

Oracle Ex. 1009, p. 25
Oracle v. DeBellis
IPR2024-00221

This book is dedicated to my wife, Alexandra.

Oracle Ex. 1009, p. 26
Oracle v. DeBellis
IPR2024-00221

About the Author

Roger Jennings is a consultant and author specializing in Windows database and digital video applications. He was a member of the Microsoft beta-test team for Microsoft Access 1.0, 1.1, 2.0, 95, and 97; Windows 3.1 and 95; Windows NT 3.5, 3.51, and 4.0; Visual Basic 2.0, 3.0, 4.0, and 5.0; SQL Server 6.0 and 6.5; Internet Information Server 3.0 (code-named Denali); and ActiveMovie. In addition to writing the four preceding editions of this book, he is the author of Que's *Special Edition Using Windows NT Server 4*, *Unveiling Windows 95*, *Discover Windows 3.1 Multimedia*, and *Access for Windows Hot Tips*. Roger was a contributing author to Que's *Special Edition Using Windows 95*, *The Official VBPJ Guide to Visual Basic*, *Excel Professional Techniques*, *Killer Windows Utilities*, and *Using Visual Basic*. He has written two other books about creating database applications with Access and Visual Basic, co-authored a book on using Microsoft Visual C++ for database development, is a contributing editor of *Visual Basic Programmer's Journal*, and has written articles for the *Microsoft Developer Network News* and the MSDN CD-ROMs.

Roger has more than 25 years of computer-related experience and has presented technical papers on computer hardware and software to the Academy of Sciences of the former USSR, the Society of Automotive Engineers, the American Chemical Society, and a variety of other scientific and technical organizations. He is a principal of OakLeaf Systems, a Northern California software development and consulting firm. You may contact him via CompuServe (**70233,2161**) or on the Internet (**70233.2161@compuserve.com**).

Oracle Ex. 1009, p. 27
Oracle v. DeBellis
IPR2024-00221

Acknowledgments

Matthew Harris, a consultant and author living in Oakland, California, updated Chapters 2 through 15, 19 through 22, 27, and 28 from the fourth edition of this book, *Special Edition Using Access 95*. Matthew has been involved with the microcomputer industry since 1980. He has provided programming technical support, training, and consulting services to the 6th international Conference on AIDS, the University of California at San Francisco, and many private firms. Matthew began programming applications for IBM PCs and compatibles in 1983 and has written commercially-distributed applications and in-house programs for a variety of clients. He also has taught classes on using MS-DOS and on programming in BASIC and Pascal. Matthew is the author of *The Disk Compression Book*, and the co-author of *Using FileMaker Pro 2.0 for Windows* (both published by Que). He was a contributing author for Que's *Special Edition Using Word for Windows 6*, *Special Edition Using Excel 5*, *Excel Professional Techniques*, *Using Paradox 4.5 for DOS*, *Special Edition Using Paradox for Windows 5.0*, *The Paradox Developer's Guide*, *Using MS-DOS 6*, and *Unveiling Windows 95*. Reach Matthew on CompuServe at **74017,766**.

Stan Leszynski is the author of the *Leszynski Naming Conventions for MicroSoft Access*, an abridged version of which appears in Appendix B, "Naming Conventions for Access Objects and Variables." He also is the author of *Access Expert Solutions* published by Que. Stan founded Leszynski Company, Inc., in 1982 to create custom PC database applications, and since that time the firm has created solutions for hundreds of clients, including Microsoft Corporation. The company also has written retail products sold by Microsoft, Microrim, Qualitas, and Kwery, which have a user base of more than two million people. Successful products include the OLE Calender control shipped with Access 2.0 and Access 95, the 386MAX memory manager, and four R:BASE developer tools. Stan's company currently specializes in Access, Visual Basic, SQL Server, and Visual C++ applications. Stan's second company, Kwery Corporation, shipped the first Access add-in, Access to Word, and the first OLE Controls for Access 2.0, Kwery Control Pak 1. Stan is a col-

Oracle Ex. 1009, p. 28
Oracle v. DeBellis
IPR2024-00221

umnist on developer issues for *Access/Visual Basic Advisor* magazine and speaks regularly at Access conferences in the U.S., Canada, and Europe. Contact Stan on CompuServe (**71151,1114**).

Thanks to Jim Sturms, Monte Slichter, Richard Dickinson, and all of the other members of Microsoft Corporation's product support and product management staff during the beta-testing period for Access 97. Their prompt replies to questions concerning the pre-release versions of Access are especially appreciated. Thanks also to Access Insiders and other members of Access 97 beta program for aiding in the resolution of many prerelease issues.

Kevin Kloss, development editor, provided valuable insight and suggestions for the development of this book's content and organization. Angela Wethington, acquisitions editor, made sure that I did not fall too far behind the manuscript submission schedule. Lisa Gebken, production editor, put in long hours to add last-minute changes incorporated in the final release candidate of Access 97 and still met a very tight publication schedule. Technical editing was done by Jim O'Connor, Tim Schubach, and Sara Strock. Their contributions to this book are gratefully acknowledged. The responsibility for any errors or omissions, however, rests solely on my shoulders.

Oracle Ex. 1009, p. 29
Oracle v. DeBellis
IPR2024-00221

We'd Like to Hear from You!

As part of our continuing effort to produce books of the highest possible quality, Que would like to hear your comments. To stay competitive, we *really* want you, as a computer book reader and user, to let us know what you like or dislike most about this book or other Que products.

You can mail comments, ideas, or suggestions for improving future editions to the address below, or send us a fax at (317) 581-4663. For the online inclined, Macmillan Computer Publishing has a forum on CompuServe (type **GO QUEBOOKS** at any prompt) through which our staff and authors are available for questions and comments. The address of our Internet site is **http://www.mcp.com** (World Wide Web).

In addition to exploring our forum, please feel free to contact me personally to discuss your opinions of this book: I'm **74201,1064** on CompuServe, and I'm **kkloss@que.mcp.com** on the Internet.

Thanks in advance—your comments will help us to continue publishing the best books available on computer topics in today's market.

Kevin Kloss
Product Development Specialist
Que Corporation
201 W. 103rd Street
Indianapolis, Indiana 46290
USA

Oracle Ex. 1009, p. 30
Oracle v. DeBellis
IPR2024-00221

Introduction

Microsoft Access 97, Version 8.0 (called Access 97 in this book) is a powerful and robust 32-bit relational database management system (RDBMS) for creating desktop and client/server database applications that run under Windows 95 and Windows NT 3.51 and 4.0. As a component of the Professional and Developer Editions of the Microsoft Office 97 suite, Access 97 has an upgraded user interface that is consistent with Microsoft Excel 97 and Word 97, as well as Windows 95 common controls, such as an Explorer-style Database window and common file open and save dialogs.

Like all members of Office 97, Access 97 offers a variety of new Internet-related features for creating HTML documents used by the World Wide Web. The upgraded Jet 3.5 database engine speeds queries against client/server databases with ODBCDirect. Aside from Internet enhancements and ODBCDirect, Access 97 does not differ dramatically from Access 95. Improved performance was Microsoft's first priority for the developers of Access 97, because Access 95 had gained a well-deserved reputation for slow operation. Ease of use was the second priority; Access 97 includes many new or improved wizards and other aids designed for first-time database users.

If you're a new Access user, a brief history of Microsoft Access is useful to put Access 97 in perspective. Version 1.0 of Access revolutionized the Windows database market and achieved a new record for sales of a Windows application—Microsoft Corporation received orders for more than 750,000 copies of Access 1.0 between its release data in mid-November 1992, and January 31, 1993. Access 1.1, introduced in May 1993, solved some of the shortcomings in Access 1.0, and the Access 1.1 Distribution Kit (ADK) gave developers the ability to release royalty-free, runtime versions of their Access applications.

Access 2.0, released about a year after Access 1.1, added OLE 2.0 client capability and was the first Microsoft application to use 16-bit OLE Controls, prepackaged objects that extended the already rich set of control objects of Access. Access 2.0 was a member of the Professional Edition of the phenomenally successful Microsoft Office 4.x software suite. Office 4.x gained almost 90 percent of the Windows productivity software suite market. By far, the majority of Access 2.0's multimillion-copy sales resulted from the product's inclusion in Office 4.x.

Oracle Ex. 1009, p. 31
Oracle v. DeBellis
IPR2024-00221

Microsoft introduced Access 95, the first 32-bit version of Access, in late 1995 as a component of Microsoft Office 95. Access 95 adopted VBA as its programming language and was the first multithreaded member of the Office suite. The success of Office 95, and thus Access 95, was dependent on adoption of Windows 95 and, to a lesser extent, Windows NT Workstation 3.5+, by major corporate accounts. Windows 95 was installed on more than 80 percent of all new PCs shipped, but the rate of migration from 16-bit to 32-bit Windows on existing PCs turned out to be much slower than anticipated by Microsoft and other software publishers. In many cases, moving to Access 95 required additional RAM. Even with 16M of RAM, however, many 32-bit Access applications ran slower than their 16-bit predecessors in 8M of RAM. Fortunately, the trend toward upgrading 80486DX PCs to 133-MHz and faster Pentium models accelerated during 1996, and 16M became the standard for all but the most parsimonious organizations. Use of Access 95 began to really take off in mid-1996. In the fall of 1996, Microsoft announced that the firm had sold a cumulative total of 10 million copies of Access.

One of the primary reasons for Access's success is that Access duplicates on the PC desktop many of the features of client/server relational database systems, also called *SQL databases*. Client/server RDBMSs are leading the way in transferring database applications from minicomputers and mainframes to networked PCs—a process called *downsizing*. Despite Access's power, this desktop RDBMS is easy for nonprogrammers to use. Buttons on upgraded multiple toolbars, which are almost identical across the Office 97 members, offer shortcuts for menu commands. An extensive collection of wizards and add-ins handle most of the mundane chores involved in creating and modifying tables, queries, forms, graphs, and reports. Builders aid you in creating complex controls on forms and reports, as well as in writing expressions. Last, and in this case least, an animated Office Assistant attempts to anticipate users' questions about Access 97.

Microsoft Access 1.0 introduced a new approach to writing macros that automate repetitive database operations. Access 95's 40+ macro instructions were remarkably powerful; you could create quite sophisticated database applications using only Access macros. Access 97 relegates macros to the "for backward compatibility only" category. Office 97 brings a common version 5.0 of 32-bit Visual Basic for Applications (VBA) to Access, Excel, Word, and even PowerPoint. Visual Basic 5.0 shares the same VBA engine with members of Office 97. VBA's syntax is easy to learn, yet VBA provides a vocabulary rich enough to satisfy veteran xBase and Paradox application developers. Making the transition from Access macros to VBA is important, because there's no guarantee that the next version of Access will continue to support macro programming.

Access 97 supports 32-bit Object Linking and Embedding (OLE) 2.1 as both a container (client) and as an ActiveX component (formerly OLE Automation server) application, giving you the benefits of in-place activation of objects, such as Excel 97 worksheets and Word 97 documents stored in Access databases. (Access 2.0 was an OLE 2.0 container application only.) Conversely, you can manipulate Access objects within an Excel 97, Word 97, Project 97, or 32-bit Visual Basic 4+ application. Access 97 also lets you manipulate ActiveX components created with Visual Basic 4+. *LOBjects* (Line of Business Objects) built with Visual Basic 4+ and shared with other VBA-enabled applications

Oracle Ex. 1009, p. 32
Oracle v. DeBellis
IPR2024-00221

promise to make a major change in the methodology of creating enterprise-scale database applications.

Access 97 and Visual Basic 4+ share the ability to take advantage of the new ActiveX Controls (formerly OLE Controls) that Microsoft, third-party add-in software publishers, and you create. (The Control Creation Edition of Visual Basic 5.0 lets you design your own ActiveX controls that are compatible with Access 97.) ActiveX Controls provide Access 97 with the extensibility that VBX custom controls brought to Visual Basic. Access 97 can accommodate almost every 32-bit OLE Control included with the Professional and Enterprise editions of Visual Basic 4.0. The new, lightweight ActiveX controls for Internet applications, many of which are available for downloading from **http://www.microsoft.com**, carry much less overhead. Other members of Office 97 also can take advantage of ActiveX controls in dialogs created with Microsoft's new Forms 2.0 technology.

Access is specifically designed for creating multiuser applications where database files are shared on networks, and Access incorporates a sophisticated security system to prevent unauthorized persons from viewing or modifying the databases you create. Access's security system is modeled on that of Microsoft SQL Server 4+. Access 2.0 simplified the labyrinthine security employed by versions 1.x and made creating secure Access applications much easier. No substantial changes have been made to the Access security system in Access 97, but the User-Level Security Wizard introduced with Access 95 makes secure applications easier to implement.

Access has a unique database structure that is capable of combining all related data tables and their indexes, forms, reports, macros, and VBA code within a single .mdb database file. It is now a generally-accepted database design practice (GADBDP) to use separate .mdb files to contain data and application objects; your application .mdb links tables contained in the data .mdb. (The term *link* replaced *attach* beginning with Access 95.) Access 97 introduces a new file format, .mde, so you can distribute Access applications without including your original VBA source code.

Access has the capability to import data from and export data to the more popular PC database and spreadsheet files, as well as text files. Access also can attach dBASE, FoxPro, and Paradox table files to databases and manipulate these files in their native formats. You also can use Access on workstations that act as clients of networked file and database servers in client/server database systems. Access, therefore, fulfills all the requirements of a professional relational database management system, as well as a front-end development tool for use with client/server databases. Microsoft has made several improvements to these features in Access 97. The most important new features of Access 97 are discussed in Chapter 1, "Access 97 for Access 95 and 2.0 Users: What's New."

Several chapters of this book are devoted to using Access 97 with other members of Office 97, such as Microsoft Excel 97 and Microsoft Word 97, plus Microsoft Graph 8.0 and the Paint applet supplied with Windows 95. Applets are small but useful applications supplied as components of major applications; Graph 8.0, for example, is an OLE 2.1 applet.

Oracle Ex. 1009, p. 33
Oracle v. DeBellis
IPR2024-00221

Who Should Read This Book?

Special Edition Using Microsoft Access 97 takes an approach that is different from most books about database management applications. This book doesn't begin with the creation of a database for Widgets, Inc., nor does it require you to type a list of fictional customers for their new Widget II product line to learn the basics of Access. Instead, this book makes the following basic assumptions about your interest in Microsoft's relational database management system:

- You have one or more PCs operating in a business, professional, institutional, or government agency setting.
- You are using or have decided to use Microsoft Windows 95 or Windows NT Workstation 4.0 as the operating environment for at least some, if not all, of your PCs. (Applications you create with Access 97 run only under Windows 95 or Windows NT 3.51+.)
- You are able to navigate Microsoft Windows 95 using the mouse and keyboard. Books about DOS database managers no longer attempt to teach you DOS, nor does *Special Edition Using Microsoft Access 97* try to teach you Windows fundamentals.
- You aren't starting from "ground zero." You now have or will have access via your PC to data that you want to process with a Windows database manager. You already may have acquired Access and want to learn to use it more quickly and effectively. Or, you may be considering using Access as the database manager for yourself, your department or division, or your entire organization.
- Your existing data is in the form of one or more database, spreadsheet, or even plain text files that you want to manipulate with a relational database management system. Access can process the most common varieties of all three types of files.
- If you're planning to use Access 97 as a front-end to a client/server RDBMS, you have SQL Server 6.5 installed or have a 32-bit ODBC driver and the required client license for your SQL database.
- If your data is on a mini- or mainframe computer, you are connected to that computer by a local area network and a database gateway or through terminal emulation software and an adapter card. Otherwise, you are able to obtain the data on PC-compatible disks; some people call this method *SneakerNet* or *FootWare*.

If some or all of your data is in the form of ASCII or ANSI text files, or files from a spreadsheet application, you need to know how to create an Access database from the beginning and import the data into Access's own .mdb file structure. If your data is in the form of dBASE, FoxPro, or Paradox files, you can link the files as tables and continue to use them in the format native to your prior database manager. Access 97 lets you link Excel and text files to Access databases. (Access 2.0 required the Microsoft ODBC Desktop Database Driver kit to attach Excel and text files. Office 97 includes the required ODBC drivers.) The capability to link files in their native format is an important advantage to have during conversion from one database management system to another. Each of these subjects receives thorough coverage in this book.

Oracle Ex. 1009, p. 34
Oracle v. DeBellis
IPR2024-00221

Special Edition Using Microsoft Access 97 is designed to accommodate readers who are new to database management; who are occasional or frequent users of dBASE, FoxPro, or Paradox for DOS or Windows; or who are seasoned Windows database application developers who are migrating to Microsoft Access.

How This Book Is Organized

Special Edition Using Microsoft Access 97 is divided into eight parts that are arranged in increasing levels of detail and complexity. Each division after Part I draws on the knowledge and experience you have gained in the prior parts, so use of the book in a linear, front-to-back manner through Part IV, "Publishing Data on Intranets and the Internet," is recommended during the initial learning process. After you've absorbed the basics, *Special Edition Using Microsoft Access 97* becomes a valuable reference tool for the advanced topics.

As you progress through the chapters in this book, you create a model of an Access application called *Personnel Actions*. In Chapter 4, "Working with Access Databases and Tables," you create the Personnel Actions table. In the following chapters, you add new features to the Personnel Actions application. Make sure to perform the example exercises for the Personnel Actions application each time you encounter them, because succeeding examples build on your prior work.

The eight parts of *Special Edition Using Microsoft Access 97*, and the topics they cover are described in the following sections.

Part I, "Learning Access Fundamentals"

Part I introduces you to Access and many of the unique features that make Access the easiest to use of all database managers. The chapters in Part I deal almost exclusively with tables, the basic elements of Access databases.

Chapter 1, "Access 97 for Access 95 and 2.0 Users: What's New" provides a summary of the most important new features of Access 97 and a detailed description of each of these additions and improvements. Much of the content of this chapter is of interest primarily to readers who now use Access 2.0, because most of the changes from Access 95 to Access 97 are incremental in nature. Readers new to Access, however, benefit from the explanations of why many of these features are significant in everyday use of Access 97.

In Chapter 2, "Up and Running with Access 97," you learn how to open an Access database, view a table, use a typical query, add a few new data items, view the results of your work, and finally print a formatted report. Chapter 2 shows you how to use the Database Wizard, introduced by Access 95, to create a database from the standard database templates included with Access 97.

Chapter 3, "Navigating Within Access," shows you how to navigate Access by explaining its toolbar and menu choices and how they relate to the structure of Access.

Chapter 4, "Working with Access Databases and Tables," delves into the details of Access tables and shows you how to create tables, and choose the optimum data types from the many new types Access offers.

Oracle Ex. 1009, p. 35
Oracle v. DeBellis
IPR2024-00221

Chapter 5, "Entering, Editing, and Validating Data in Tables," shows you how to arrange the data in tables to suit your needs and limit the data displayed to only that information you want. Finding and replacing data in the fields of tables also is covered here.

Chapter 6, "Sorting, Finding, and Filtering Data in Tables," describes how to add new records to tables, enter data in the new records, and edit data in existing records. Chapter 6 describes how to make best use of the Filter by Form and Filter by Selection features of Access 97.

Chapter 7, "Linking, Importing, and Exporting Tables," explains how you import and export files of other database managers, spreadsheet applications, and even ASCII files you may download from the Internet or information utilities such as The Microsoft Network, CompuServe, America Online, Dow Jones News Service, or government-sponsored databases. Chapter 7 explains the Table Analyzer Wizard that aids in creating a relational database structure from "flat files" in ASCII and spreadsheet formats.

Part II, "Querying for Information"

Part II explains how to create Access queries to select the way you view data contained in tables and how you take advantage of Access's relational database structure to link multiple tables with joins.

Chapter 8, "Using Query by Example," starts you off with simple queries created with Access's graphic query-by-example (QBE) design window. You learn how to choose the fields of the tables that are included in your query and return query result sets from these tables. Chapter 8 shows you how to use the Select Query Wizard to simplify the QBE process.

Chapter 9, "Understanding Operators and Expressions in Access," introduces you to the operators and expressions that you need to create queries that provide a meaningful result. You use the improved Debug window of the VBA 5.0 code editor to evaluate the expressions you write.

In Chapter 10, "Creating Multitable and Crosstab Queries," you create relations between tables, called *joins*, and learn how to add criteria to queries so that the query result set includes only those records you want. Chapter 10 also takes you through the process of designing powerful crosstab queries to summarize data and to present information in a format similar to that of worksheets.

Chapter 11, "Using Action Queries," shows you how to develop action queries that update the tables underlying append, delete, update, and make-table queries. Chapter 11 also covers Access 95's referential integrity features, including cascading updates and cascading deletions.

Part III, "Creating Forms and Reports"

Part III is your introduction to the primary application objects of Access. (Tables and queries are considered database objects.) Forms make your Access applications come alive with the control objects you add using Access 97's toolbox. Access's full-featured report generator lets you print fully formatted reports or save reports to files that you can process in Excel 97 or Word 97.

Oracle Ex. 1009, p. 36
Oracle v. DeBellis
IPR2024-00221

Chapter 12, "Creating and Using Forms," shows you how to use Access's Form Wizards to create simple forms and subforms that you can modify to suit your particular needs. Chapter 12 introduces you to the Subform Builder Wizard that uses drag-and-drop techniques to automatically create subforms for you.

Chapter 13, "Designing Custom Multitable Forms," shows you how to design custom forms for viewing and entering your own data with Access's advanced form design tools.

Chapter 14, "Printing Basic Reports and Mailing Labels," describes how to design and print simple reports with Access's Report Wizard and how to print preformatted mailing labels using the Mailing Label Wizard.

Chapter 15, "Preparing Advanced Reports," describes how to use more sophisticated sorting and grouping techniques, as well as subreports, to obtain a result that exactly meets your detail and summary data reporting requirements.

Part IV, "Publishing Data on Intranets and the Internet"

Part IV describes the new Internet enhancements of Access 97.

Chapter 16, "Working with Hyperlinks and HTML," describes Microsoft's Internet strategy and introduces you to Access 97's new Hyperlink field data type. The chapter shows you how to use hyperlinks to connect to Word documents and Excel worksheets, as well as open Access form and report objects.

Chapter 17, "Exporting Data to World Wide Web Pages," describes how to export formatted static Web pages from table and query datasheets. You also learn how to use HTML templates and the Publish to the Web Wizard to create multipage Web documents from Access reports.

Chapter 18, "Creating Dynamic Web Pages," describes Internet Information Server 2+'s Internet Database Connector (IDC) for creating dynamic Web pages from Open Database Connectivity (ODBC) data sources. The chapter explains how to set up an Access ODBC data source and leads you through exporting Access datasheets to the .idc/.htx file combination required by IDC.

Part V, "Integrating Access with Other Office 97 Applications"

Part V shows you how to use the new 32-bit Object Linking and Embedding (OLE) 2.1 features of Access 97 with Microsoft Graph 8.0, plus OfficeLinks to Excel 97 and Word 97.

Chapter 19, "Using 32-Bit OLE Components," explains Object Linking and Embedding, the principles that make OLE 2.1 useful in Windows application development, how these principles apply to your Access database applications, and how you use 32-bit ActiveX server applications with Access 95. Chapter 19 also explains Windows 95's Registry that replaces the registration database (REG.DAT) and .INI files of Windows 3.x.

Chapter 20, "Adding Graphics to Forms and Reports," describes how to best take advantage of Access OLE Object field data type and bound object frames to display graphics

Oracle Ex. 1009, p. 37
Oracle v. DeBellis
IPR2024-00221

and play multimedia objects stored in your Access tables. Adding static graphics to forms and reports with unbound object frames is also covered.

Chapter 21, "Using Access with Microsoft Excel," gives you detailed examples of exchanging data between Access and Excel 97 workbooks by using Access as an OLE 2.1 client and server, without the need to write VBA code. The "Analyze It with MS Excel" OfficeLink feature of Access 97 also is covered.

Chapter 22, "Using Access with Microsoft Word and Mail Merge," shows you how to store documents in OLE Object fields, explains the OfficeLink "Publish It with MS Word" option for database publishing, and how to use Access 97's "Merge It" OfficeLink feature to interactively create form letters and envelopes addressed with data from your Access applications.

Part VI, "Using Advanced Access Techniques"

Part VI covers the theoretical and practical aspects of relational database design and Structured Query Language (SQL), and then goes on to describe how to set up and use secure Access applications on a local area network. Part VI also describes how you use the ODBC Application Programming Interface (API) to create Access front-ends for client/server databases.

Chapter 23, "Exploring Relational Database Design and Implementation," describes the process you use to create relational database tables from real-world data—a technique called *normalizing the database structure*. This chapter explains how to use the Database Documentor tool included with Access 97 to create a data dictionary that fully identifies each object in your database.

Chapter 24, "Working with Structured Query Language," explains how Access uses the Jet dialect of SQL to create queries and how to write your own SQL statements. Special emphasis is given to the newer Jet SQL extensions, such as `UNION` queries and subqueries, as well as Jet 3.5's implementation of SQL's Data Definition Language (DDL).

Chapter 25, "Securing Multiuser Network Applications," explains how to set up Access to share database files on a network and how to use the security features of Access to prevent unauthorized viewing of or tampering with your database files.

Chapter 26, "Connecting to Client/Server Databases," introduces you to the 32-bit ODBC 2.5 API and shows you how to create ODBC data sources from client/server databases, as well as the basics of designing Access front-ends for client/server databases.

Chapter 27, "Replicating Access Databases," shows you how to create independent replicas of Access databases on network servers or disks. Users update the replica databases, and the updated replicas later are merged with the design master database. The chapter briefly describes Access 97's new partial replication features.

Part VII, "Programming, Distributing, and Converting Access Applications"

Part VII assumes that you have no programming experience in any language. Part VII explains the principles of writing programming code in object-enabled VBA and applies

Oracle Ex. 1009, p. 38
Oracle v. DeBellis
IPR2024-00221

these principles to using Automation to exchange data with Excel 97 worksheets and Word 97 documents. Part VII also describes the Office 97 Developer Edition's Setup Wizard and supplies tips for converting Access 2.0 and 95 applications to Access 97.

Chapter 28, "Writing Visual Basic for Applications Code," describes how to use VBA to create user-defined functions stored in modules and to write simple procedures that you activate with macros or directly from events. Access's class modules (called *Code Behind Forms* in Access 2.0), which let you store event-handling code in Form and Report objects, also are described.

Chapter 29, "Understanding the Data Access Object Class," shows you how to declare and use members of the Jet 3.5 database engine's object collections, such as `TableDefs`, to create new tables and modify the properties of tables with VBA code.

Chapter 30, "Responding to Events with VBA 5.0," describes how to use VBA event-handling subprocedures in class modules to replace the macros used by earlier versions of Access. Chapter 30 explains the events triggered by Access form, report, and control object, and how to use methods of the `DoCmd` object to respond to events, such as clicking a command button.

Chapter 31, "Exchanging Data with Automation and ActiveX Controls," gives you a complete, working application that uses VBA and Automation (formerly OLE Automation) to transfer data to and from an Excel 97 worksheet. Using ActiveX controls, which replace the OLE Controls of prior Access versions, also is covered in this chapter.

Chapter 32, "Using the Office 97 Developer Edition," describes the content of the ODE, the distributable 32-bit ActiveX Controls included in the ODE, and how to use the Setup Wizard to create distribution disks for your runtime Access applications.

Chapter 33, "Migrating Access 2.0 and Access 95 Applications to Access 97," tells you what changes you need to make when you convert your current Access 2.0 and Access 95 database applications to Access 97. This chapter also describes how to use 32-bit Access 97 front-end applications with 16-bit Access 2.0 databases.

Part VIII: "Appendixes"

Appendix A, "Glossary," presents a glossary of the terms, abbreviations, and acronyms used in this book that may not be familiar to you and cannot be found in commonly used dictionaries.

Appendix B, "Naming Conventions for Access Objects and Variables," incorporates the most commonly used set of standardized rules for naming Access objects and Access VBA variables, the *Leszynski Naming Conventions for Microsoft Access*.

Appendix C, "Data Dictionary for the Personnel Actions Table," shows you how to implement the Personnel Actions table that is used for many of the examples in this book.

Oracle Ex. 1009, p. 39
Oracle v. DeBellis
IPR2024-00221

How This Book Is Designed

The following special features are included in this book to assist readers.



Readers who have never used a database management application are provided with quick-start examples to gain confidence and experience while using Access with the Northwind Traders sample database. Like Access, this book uses the *tabula rasa* approach: each major topic begins with the assumption that the reader has no prior experience with the subject. Therefore, when a button from the toolbar or control object toolbox is used, its icon is displayed in the margin.

Note

Notes offer advice to aid you in using Access, describe differences between Access 2 and 1.x, and explain the few remaining anomalies you find in version 2.0 of Access. In a few instances, notes explain similarities or differences between Access and other database management applications.

Tip

Tips describe shortcuts and alternative approaches to gaining an objective. These tips are based on the experience the authors gained during more than two years of testing successive alpha and beta versions of Access and the Access Distribution Kit.

Caution

Cautions are provided when an action can lead to an unexpected or unpredictable result, including loss of data; the text provides an explanation of how you can avoid such a result.



Features that are new or have been modified in Access 97 are indicated by the Best of 97 icon in the margin, unless the change is only cosmetic. Where the changes are extensive and apply to an entire section of a chapter, the icon appears to the left or right of the section head.

All sample databases and associated queries, forms, reports, and the like, used in this book are posted to a page accessible from the Que Microsoft Office page of the Macmillan Superlibrary on the Internet (**http://www.mcp.com/que/MSoffice**).

Most software manuals require you to wade through all the details relating to a particular function of an application in a single chapter or part, before you progress to the next topic. In contrast, *Special Edition Using Microsoft Access 97* first takes you through the most frequently used steps to manipulate database tables and then concentrates on using your existing files with Access. Advanced features and nuances of Access are covered in later chapters. This type of structure requires cross-referencing so that you can easily locate more detailed or advanced coverage of the topic. Cross-references to specific sections in other chapters occur in the margins next to the material they pertain to, such as in the sample reference next to this paragraph:

Oracle Ex. 1009, p. 40
Oracle v. DeBellis
IPR2024-00221

▶▶ See "Opening a Database During Access Startup," p. 48



Troubleshooting

Troubleshooting boxes provide commonly asked questions about problems or challenges typically encountered by users.

These troubleshooting tips are presented in the form of a commonly asked question and its answer.

Typographic Conventions Used in This Book

This book uses various typesetting styles to distinguish between explanatory and instructional text, text you enter in dialogs, and text you enter in code-editing windows.

Key Combinations and Menu Choices

Key combinations that you use to perform Windows operations are indicated by joining the keys with a plus sign: Alt+F4, for example. This indicates that you press and hold the Alt key while pressing the function key F4. In the rare cases when you must press and release a control key, and then enter another key, the keys are separated by a comma without an intervening space: Alt,F4, for example. Key combinations that perform menu operations that require more than one keystroke are called *shortcut keys*. An example of such a shortcut is the Windows 95 key combination, Ctrl+C, which substitutes for the Copy choice of the Edit menu in almost all Windows applications.

To select a menu option with the keyboard instead of the mouse, you press the letter that appears with an underscore (_) in the menu option. Sequences of individual menu items are separated by a comma: Edit, Cut, for example. The Alt key required to activate a choice from the main menu with an accelerator key is assumed and not shown.

Successive entries in dialogs follow the tab order of the dialog. Tab order is the sequence in which the caret moves when you press the Tab key to move from one entry or control option to another, a process known as *changing the focus*. The entry or control option that has the focus is the one that receives keystrokes or mouse clicks. Command buttons, option buttons, and check box choices are treated similarly to menu choices, but their accelerator key letters in the text don't include the underscore attribute.

File and folder names are initial-letter-capitalized in the text and headings of this book, in conformity to Windows 95 and Windows NT 4.0 filenaming conventions and the appearance of file names in Windows Explorer.

SQL Statements and Keywords in Other Languages

SQL statements and code examples, including macro commands, are set in a monospace font. Keywords of SQL statements, such as SELECT, are set in all uppercase, as are the

Oracle Ex. 1009, p. 41
Oracle v. DeBellis
IPR2024-00221

keywords of foreign database programming languages when they are used in comparative examples, such as xBase's `DO WHILE ... ENDDO` structure. Ellipses indicate intervening programming code that isn't shown in the text or examples.

Square brackets in **`monospace boldface`** type (**`[]`**) that appear within Jet SQL statements don't indicate optional items as they do in syntax descriptions. In this case, the square brackets are used in lieu of quotation marks to frame a literal string or to allow use of a table and field names, such as `[Personnel Actions]`, that include embedded spaces or special punctuation, or field names that are identical to reserved words in VBA.

Typographic Conventions Used for VBA

This book uses a special set of typographic conventions for references to Visual Basic for Applications keywords in the presentation of VBA examples:

- `Monospace` (MCPdigital) type is used for all examples of VBA code, as in the following statement:

```
Dim NewArray ( ) As Long
ReDim NewArray (9, 9, 9)
```

- `Monospace` type also is used when referring to names of properties of Access database objects, such as `FormName.Width`. The captions for text boxes and drop-down lists in which you enter values of properties, such as Source Connect String, are set in the proportionally spaced font (Stone Serif) of this book.

- **`Bold monospace`** type is used for all VBA reserved words and type-declaration symbols (which are not used in the code examples in this book), as shown in the preceding example. Standard function names in VBA also are set in bold type so that reserved words, standard function names, and reserved symbols stand out from variable and function names and values you assign to variables.

- *`Italic monospace`* type indicates a replaceable item. For example,

```
Dim DataItem As String
```

- ***`Bold italic monospace`*** type indicates a replaceable reserved word, such as a data type, as in

```
Dim DataItem As DataType
```

  `DataType` is replaced by a keyword corresponding to the desired VBA data type, such as **`String`** or **`Variant`**.

- An ellipsis (`...`) substitutes for code not shown in syntax and code examples, as in

```
If...Then...Else...End If
```

- French braces (`{}`) surrounding two or more identifiers separated by the pipe symbol (|) indicate that you must choose one of these identifiers, as in

```
Do {While|Until}...Loop
```

  In this case, you must use either the **`While`** or the **`Until`** reserved word in your statement.

Oracle Ex. 1009, p. 42
Oracle v. DeBellis
IPR2024-00221

- Square brackets ([], not in bold type) surrounding an identifier indicate that the identifier is optional, as in

```
Set tblName = dbName.OpenTable(strTableName[, fExclusive])
```

  Here, the `fExclusive` flag, if set **`True`**, opens the table specified by `strTableName` for exclusive use. `fExclusive` is an optional argument.

System Requirements for Access 97

Access 97 is a very resource-intensive application. Access must be installed on an 80486DX33 or better PC running Windows 95, Windows NT 3.51, or Windows NT 4.0. Although it's possible to install Windows 95 and Access 97 on an 80386 PC, you will find execution of Access on computers using the 80386-series CPUs to be very slow, and operation may be glacial with large tables. Access 97 requires 12M of RAM for adequate performance under Windows 95. If you plan to use Access 97 for extensive handling of graphic images or run it often with other applications using Object Linking and Embedding (OLE), 16M to 20M of RAM should be installed for use under Windows 95 and 24M of RAM or more for Windows NT 3.51+. Using OLE 2.1 to manipulate complex objects, such as large bitmaps, requires substantial amounts of memory. As a rule, adding more RAM is more cost-effective than increasing processor speed or power if you're using an 80486DX-based PC.

Most users of Access 97 install the product in conjunction with Office 97. A complete installation of Office 97 requires approximately 170M of disk space. A complete installation of the stand-alone version of Access 97 requires a total of about 70M of free disk space, and you should reserve at least 10M to store the new databases you create. (The incremental disk space requirement of Access 97 is somewhat less if you've already installed Office 97; Access 97 and the other members of Office 97 share a variety of files.) Installing the Office 97 Developer Edition requires another 20M of disk space. You also should have at least 25M of space available for a Windows 95 swap file or a dedicated Windows NT paging file of at least 50M. Access also makes use of numerous temporary files when processing large client/server database tables or storing large amounts of data returned from queries against client/server databases.

Access 97 and Windows 95 have been tested under Microsoft's DriveSpace disk data compression system included with Microsoft Plus! for Windows 95. DriveSpace reduces the physical fixed disk space required to install Access 95 by a factor of about 1.5 to 1.7. Bear in mind that DriveSpace compression is not compatible with the file compression system included with Windows NT 3.51 and Windows NT 4.0, so you cannot share compressed disk volumes between Windows 95 and Windows NT in dual-boot mode.

Note

Fixed disk compression utilities such as DriveSpace do not compress encrypted Access .mdb files significantly. Compression utilities rely on creating tokens that represent repeating groups of bytes in files. The utility stores the tokens and a single copy of the translation of each token. Encryption removes most, if not all, of the repeating groups of bytes in the file.

Oracle Ex. 1009, p. 43
Oracle v. DeBellis
IPR2024-00221

A mouse or trackball isn't a requirement for using the Access applications you create, but you need one of these two pointing devices to select and size the objects that you add to forms and reports using Access's toolbox. Because a pointing device is required to create Access applications, this book dispenses with the traditional "Here's how you do it with the mouse..." and "If you want to use the keyboard..." duplicate methodology in step-by-step examples. Designing your Access applications with accelerator keys to eliminate mouse operations speeds keyboard-oriented data entry by enabling the operator to keep his or her fingers on the keyboard during the entire process. Users accustomed to mouseless DOS database applications will appreciate your thoughtfulness.

Other Sources of Information for Access

SQL and relational database design, which are discussed in Chapters 19 and 20, are the subject of myriad guides and texts covering one or both of these topics. Articles in database-related periodicals and files you read on the Internet or download from online information utilities, such as CompuServe, provide up-to-date assistance in using Access 97. The following sections provide a bibliography of database-related books and periodicals, as well as a brief description of Internet Web sites and CompuServe forums of interest to Access users.

Books

The following books published by Que complement the content of this book by providing detailed coverage of Access programming techniques, application design, Structured Query Language, client/server databases, and the Windows 95 and Windows NT operating systems:

- *Access 97 Power Programming,* by F. Scott Barker, shows you how to get the most out of the Access flavor of VBA 5.0 and complements the VBA programming chapters of Part VII of this book. (Que, ISBN 0-7897-0915-5.)
- *Access 97 Expert Solutions,* by Stan Leszynzki, describes the Access 97 application design and implementation techniques employed by one of North America's top database developers. Stan is the author of the *Leszynski Naming Conventions for Microsoft Access,* Appendix B of this book. (Que, ISBN 0-7897-0367-X.)
- *Introduction to Databases,* by James J. Townsend, gives a thorough explanation of personal computer databases and their design. This book is especially recommended if the subject of PC databases is new to you. (Que, ISBN 0-88022-840-7.)
- *Understanding the New SQL: A Complete Guide,* by Jim Melton and Alan R. Simpson, describes the history and implementation of the American National Standards Institute's X3.135.1-1992 standard for the latest official version of Structured Query Language, SQL-92, on which Jet SQL is based. Jim Melton of Digital Equipment Corp. was the editor of the ANSI SQL-92 standard, which consists of more than 500 pages of fine print. (Morgan Kaufmann Publishers, ISBN 1-55860-245-3.)
- *Special Edition Using Microsoft SQL Server 6.5,* by Bob Branchek, Peter Hazelhurst, Stephen Wynkoop, and Scott Warner is designed to bring system and database

Oracle Ex. 1009, p. 44
Oracle v. DeBellis
IPR2024-00221

administrators, as well as developers, up-to-date on the latest version of Microsoft SQL Server. (Que, ISBN 07897-0097-2.)

- *Special Edition Using Microsoft Internet Information Server* supplies detailed instructions for setting up an Internet or intranet Web site using IIS 2.0. This book extends the coverage of Part IV, "Publishing Data on Intranets and the Internet." (Que, ISBN 07897-0850-7.)
- *Special Edition Using Windows NT Server 4*, by Roger Jennings, provides all of the information you need to set up Windows NT Server 4.0 for sharing Access databases, install and run SQL Server 6.5, and create your own intranet Web site with Internet Information Server. (Que, ISBN 07897-0251-7.)
- *Special Edition Using Windows NT Workstation 4.0* by Paul Sanna, et al., complements this book by providing coverage of the client-side features of Windows NT 4.0 that are beyond the scope of this book. (Que, ISBN 07897-0673-3.)
- *Platinum Edition Using Windows 95* by Ron Person, et al., is a 1,300+ page book that covers all aspects of Windows 95 in detail and is especially useful as a reference for Windows 95 client networking and user/policy management. (Que, ISBN 07897-0797-7.)

Periodicals

The following are a few of the magazines and newsletters that cover Access exclusively or in which articles on Microsoft Access appear on a regular basis:

- *Access/Visual Basic Advisor,* published by Advisor Communications International, Inc., is a full-color, bimonthly magazine intended to serve Access users and developers. You can supplement your subscription with an accompanying disk that includes sample databases, utilities, and other software tools for Access.
- *Data Based Advisor* is published by Data Based Solutions, Inc., a firm related to the publishers of Access Advisor. Data Based Advisor covers the gamut of desktop databases, with emphasis on xBase products, but Access receives its share of coverage, too.
- *DBMS* magazine, published by M&T, a Miller-Freeman company, is devoted to database technology as a whole, but DBMS concentrates on the growing field of client/server RDBMS. DBMS covers subjects, such as SQL and relational database design, that are of interest to all developers, not just those who use Access.
- *Inside Microsoft Access* is a monthly newsletter of Access tips and techniques of the Cobb Group, which publishes a variety of newsletters on products such as Visual Basic and Paradox.
- *Microsoft Interactive Developer,* published by Fawcette Technical Publications, Inc., covers Internet- and intranet-related topics, with emphasis on Internet Information Server and Visual Basic, Scripting Edition (VBScript).
- *Smart Access* is a monthly newsletter of Pinnacle Publishing, Inc., which publishes several other database-related newsletters. Smart Access is directed primarily to

Oracle Ex. 1009, p. 45
Oracle v. DeBellis
IPR2024-00221

developers and Access power users. This newsletter tends toward advanced topics, such as creating libraries and using the Windows API with VBA. A disk is included with each issue. Like other publications directed to Access users, much of the content of Smart Access is of equal interest to Visual Basic database developers.

- *Visual Basic Programmer's Journal* is a monthly magazine from Fawcette Technical Publications that covers all of the dialects of VBA. *Visual Basic Programmer's Journal* has a monthly column devoted to database topics of interest to Access and Visual Basic developers.

Online Sources

The following sections describe the primary online sources of information of interest to users of Access 97. Prior to 1996, Microsoft provided end-user technical support for its products through a large number of CompuServe Information Services forums. Microsoft's Web site now is the primary source of new and updated information for Access users and developers. The former Microsoft CompuServe forums now are run by independent organizations, such as the Windows Users Group Network (WUGNET).

The Internet. Internet Web sites and newsgroups rapidly are replacing commercial online services as the principal source of information of interest to all users of Windows and its applications. Following are the primary Web sites and news groups for Access 97 users and developers:

- Microsoft's Access home page, **http://www.microsoft.com/msaccess/**, is the jumping-off point for Access users and includes links to all related home pages on the Microsoft Web site.
- Microsoft's Access Developer home page, **http://www.microsoft.com/accessdev/**, provides a variety of links to information of particular interest to the Access developer community. This home page provides a link for downloading the Access 97 Upsizing Wizard for automating the migration of Access multiuser applications to SQL Server 6.5 databases.
- Microsoft's AnswerPoint home page, **http://www.microsoft.com/supportnet/answerpoint/**, provides information about Microsoft's no-fee and low-fee support services for all its products. For other support options, go to **http://www.microsoft.com/Support/**.
- *DevX*, Fawcette Technical Publications' new Web site for Windows developers at **http://www.windx.com/**, offers a wide range of news, features, and product reviews of ActiveX controls.
- *Avatar Magazine*, an online service of Fawcette Technical Publications, Inc., complements *Microsoft Interactive Developer* as "an interactive publication for creators of interactive media...from Web sites to digital video." Check out the Avatar site at **http://www.avatarmag.com/**.
- Microsoft's **msnews.microsoft.com** news server offers a variety of Access-related newsgroups at **microsoft.public.access.*subject***. When this book was written,

Oracle Ex. 1009, p. 46
Oracle v. DeBellis
IPR2024-00221

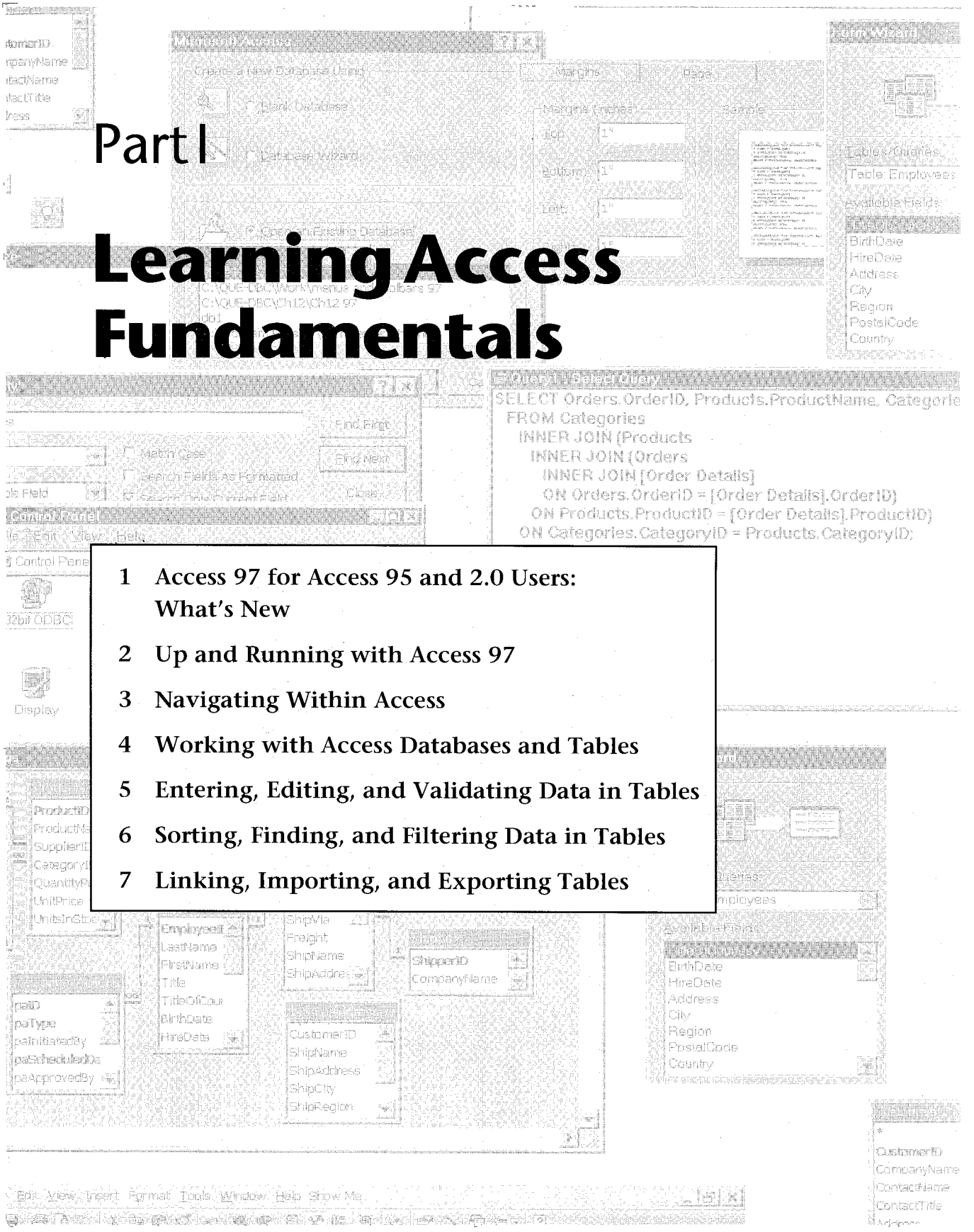
there were more than 20 Access subject areas. A list of Microsoft Access newsgroups is available at **http://www.microsoft.com/support/news/Access.htm**.

- The Access Developer Network site at **http://www.wji.com/access/homepage.html** provides a forum for Access developers with tips and source code, Access User Group meeting announcements, and links to other sites with Access content.
- The UseNet **comp.databases.ms-access** newsgroup is an active community of Access users and developers.

CompuServe. CompuServe offers the following forums of interest to Access 97 users and developers:

- The Microsoft Access forum (**GO MSACCESS**) provides peer-to-peer technical support for all versions of Access. The MSACCESS forum is one of the most active application forums on CompuServe. Many authors of books about Access and contributors to publications that feature Access, as well as professional Access developers, participate regularly in this forum. A variety of useful utility and sample applications written by third-party developers also are available for no charge other than the cost of connection time to CompuServe.
- The Microsoft Knowledge Base (**GO MSKB**) contains text files of technical tips, workarounds for bugs, and other useful information about Microsoft applications. Use **Access** as the search term, but remember to turn on your communication software's capture-to-file feature because the information is in the form of messages that scroll down your screen, not in the form files. You also can search the Knowledge Base on America Online; use the keyword **Microsoft**.
- Fawcette Technical Publications operates the *Visual Basic Programmers Journal and Discussion* forum (**GO VBPJFO**) and two Windows Components forums (**GO COMPA** and **GO COMPB**). VBPJFO offers message and library sections devoted to OLE, VBA/Office, Database, and Client/Server topics. Now that Access 97 has adopted the VBA programming language and shares 32-bit ActiveX Controls with Visual Basic 4+, Access and Visual Basic developers have many areas of common interest. As publishers of Visual Basic .vbxcustom controls migrate their products to ActiveX Controls, you'll find the Windows Components forums to be an excellent source of information and product support for extensions to Access 97.
- *Data Based Advisor* magazine operates a forum on CompuServe (**GO DBA**) that covers a wide range of database topics, including client/server systems. One section of the DBA forum is devoted to Microsoft Access and Visual Basic.
- *DBMS* magazine runs a forum (**GO DBMS**) on CompuServe primarily devoted to client/server database issues.

Oracle Ex. 1009, p. 47
Oracle v. DeBellis
IPR2024-00221



Part I

Learning Access Fundamentals

1 Access 97 for Access 95 and 2.0 Users: What's New

2 Up and Running with Access 97

3 Navigating Within Access

4 Working with Access Databases and Tables

5 Entering, Editing, and Validating Data in Tables

6 Sorting, Finding, and Filtering Data in Tables

7 Linking, Importing, and Exporting Tables

Oracle Ex. 1009, p. 48
Oracle v. DeBellis
IPR2024-00221

Chapter 1

Access 97 for Access 95 and 2.0 Users: What's New

Access 97 is a member of the Microsoft Office 97 Professional software suite that consists of second-generation, 32-bit versions of Microsoft Word, Excel, PowerPoint, and Binder, plus the first iteration of Microsoft Outlook. Most users acquire Access 97 by upgrading from Office 4.x or Office 95, or through purchase of Office 97 Professional licenses. The Office 97 Professional CD-ROM includes the Office 97 ValuPack, a collection of supplemental files for each of the Office members. The Office 97 Developer Edition (ODE), which replaces the Access Developer's Toolkit (ADT) of Access 95 and 2.0, lets you distribute run-time versions of your Access 97 applications. The ODE includes all of the components of the Professional Edition; if you already have Office 97 Professional Edition, you can buy an ODE upgrade. You also can purchase a stand-alone version of Access 97.



This chapter begins with a capsule summary of what's new in Access 97 and the features that Microsoft introduced in Access 95. The reason for discussion of Access 95 in a book about Access 97 is that many Access 2.0 (and even 1.x) users skipped the Access 95 release. The chapter concludes with sections devoted to installing Access 97 with the newly revised Setup application of the Office 97 Professional CD-ROM, installing ValuPack components, and installing the Office Developer Edition files from the Office 97 Resource Kit and Office 97 Developer Tools CD-ROM. In all of the following chapters of this book, the "Best of 97" icon, shown here in the margin, indicates a new or altered feature. Because this chapter is devoted primarily to the new features of and improvements to Access 97, the icon appears only once in this chapter.

Note

This chapter assumes familiarity with Access 2.0 and, to a lesser extent, with Access 95. If you're a new Access user, consider skipping to the section "Installing Access 97" near the end of this chapter. After you've worked your way through the first four parts of this book (Chapters 2 through 18), you're likely to find most of the "What's New" information presented here to be much more meaningful.

Oracle Ex. 1009, p. 49
Oracle v. DeBellis
IPR2024-00221

Analyzing Access 97's Enhancements and Changes

Microsoft is determined to move all users of 16-bit Windows 3.1+ to 32-bit Windows 95 or, better yet, Windows NT Workstation 4.0. Press reports from the July 31, 1996, Microsoft briefing for security analysts quote Bill Gates as saying that the software market is an "annuity business," which explains (at least in part) the almost-annual updates to Microsoft Office and thus to Microsoft Access. Many large organizations, especially those with thousands of PCs, decided not to pay the 1995-96 annuity premium for the Windows 95 upgrade, opting to stay with "tried-and-true" Windows 3.1+ and Microsoft Office 4.x.

PC vendors installed Windows 95 on about 80 percent of all new PCs delivered after August 1995, but sales of Windows 95 and its 32-bit applications to corporations and other large organizations did not meet distributors' expectations. According to Computer Intelligence, a well-respected market research firm, corporate sales of Windows 3.1+ applications still were outpacing Windows 95 versions by better than 2:1 as of August 1996, and retail 16-bit and 32-bit application sales were about neck and neck.

1997 promises to be the year for corporate adoption of Windows 95 and Windows NT 4.0, changing the ratio of application sales in favor of 32-bit versions. Web-based corporate intranets provide much of the impetus for moving to 32-bit Windows operating systems; today's 32-bit Web browsers offer better performance and more features than the 16-bit variety. Rapidly declining prices of Pentium-based PCs and dynamic RAM during 1996 reduced the cost of upgrading to "Designed for Windows 95" PCs. Replacing 80486 and slower processors with Pentiums disguises the slower operating speed of many large 32-bit Windows applications. A continuing trend toward the use of laptop PCs in the corporate environment makes the Plug and Play, energy management, and PCMCIA card support features of Windows 95 compelling reasons for upgrading to 32-bits.

Major Access 97 Enhancements

Continuing enhancement of the members of the 32-bit Office software suite are intended to accelerate the trend toward 32-bit Windows operating systems. Access 97 is an incremental upgrade to Access 95, not the major upgrade represented by the transition from Access 2.0 to Access 95. Following is a list of the most significant new features of Access 97:

- *Internet features* are the primary addition to Access 97, as well as to other members of Office 97. You can export datasheets to static or active HTML pages for the World Wide Web, publish multipage reports in a series of static Web pages, and use the Publish to the Web Wizard to create the HTML 3.2 equivalent of Access forms (see Figure 1.1). Access now sports an HTML field data type to store Web URLs (Uniform Resource Locators), as well as pointers to content stored within Office documents. You also can synchronize Briefcase replicas over the Internet. Chapters 16, 17, and 18 show you how to take advantage of Access 97's new features for private intranets and the public Internet.

Oracle Ex. 1009, p. 50
Oracle v. DeBellis
IPR2024-00221

▶▶ See "Putting Microsoft's Internet Program in Perspective," p. 575

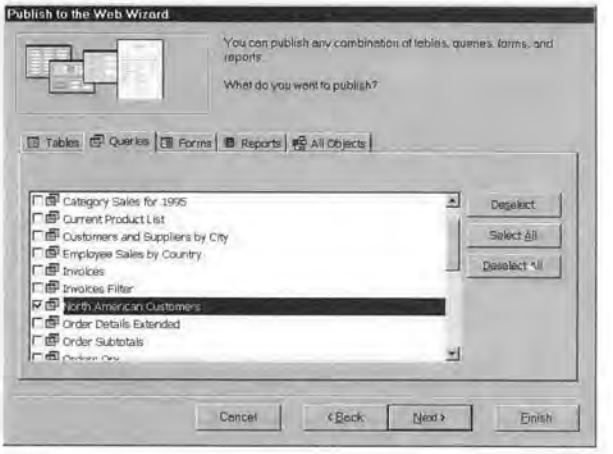


FIG. 1.1 The second dialog of the Publish to the Web Wizard.

- *Performance is enhanced* compared with Access 95, which many Access 95 users might consider damnation by faint praise. The major complaint among Access developers was that most Access 95 applications ran slower than their Access 2.0 counterparts. Forms that don't include event-handling code load faster. Microsoft has fine-tuned the Jet 3.5 database engine and Access 97 for increased operating speed. The new lightweight ActiveX controls also are faster than Access 95's OLE Controls.

▶▶ See "Adding ActiveX Controls to Your Application," p. 1122

- *`CommandBar` objects* let you customize menus and create toolbars with the ability to show or hide, customize, resize, dock, or move the toolbar. You also use `CommandBar` objects to create custom popup menus. You can program `CommandBar` objects with VBA, eliminating the need for the menu macros of Access 95 and prior versions.

▶▶ See "Replacing AddMenu Macro Actions with `CommandBar` Objects," p. 1075

- *The Tab control* is an Access native control, included in the toolbox, instead of the add-on 32-bit OLE Control for Access 95. Tab controls let you create forms that emulate the tabbed dialogs of Windows 95 properties sheets. Figure 1.2 shows the tabbed SolutionsHyperForm of the Solutions.mdb database included with Office 97.

▶▶ See "Creating a Tab Control," p. 489

Oracle Ex. 1009, p. 51
Oracle v. DeBellis
IPR2024-00221

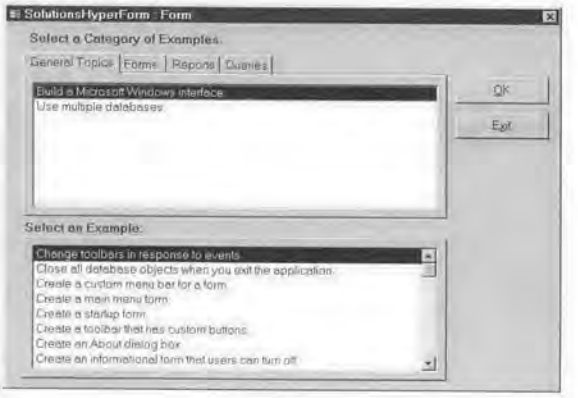


FIG. 1.2 The Solutions example database's SolutionsHyperForm, which uses Access 97's built-in Tab control.

- *Jet 3.5* lets you limit the number of records returned by queries. This feature prevents *runaway queries*, which can return thousands of rows when users specify inappropriate criteria. All members of Office 97 now can use the Jet 3.5 Data Access Object for database connectivity.

▶▶ See "Creating a Reference to the Data Access Object," p. 1025

- *Open Database Connectivity (ODBC) 3.0* brings you the new tabbed 32-bit ODBC Data Source Administrator control panel tool shown in Figure 1.3. You now can create system-wide data sources (System DSN page) that aren't specific to a single user (User DSN page), and you can share *Datasource*.dsn files (File DSN page) that specify ODBC connections to simplify deployment of Access client/server front ends. ODBC 3.0 provides enhanced tracing capability (Tracing page) and now displays of the version numbers of all ODBC-related files (ODBC Drivers and About pages). You also can create ODBC 3.0 data sources from within Access 97. ODBC 3.0 is required to use Microsoft's new OLE DB technology and ActiveX Data Objects with Internet Information Server 3.0's Active Server Pages (ASPs).

▶▶ See "Using the ODBC Data Source Administrator to Create a SQL Server Data Source," p. 924
▶▶ See "Creating a File Data Source Within Access 97," p. 931

- *ODBCDirect* speeds client/server queries by using the Remote Data Object (RDO) of Visual Basic's Enterprise Edition to bypass the Jet 3.5 database engine. ODBCDirect, which is of primary interest to Access developers, also lets you execute asynchronous queries and batch updates of changes to `Recordset` objects.

▶▶ See "Using ODBCDirect and the Remote Data Object," p. 1044

Oracle Ex. 1009, p. 52
Oracle v. DeBellis
IPR2024-00221

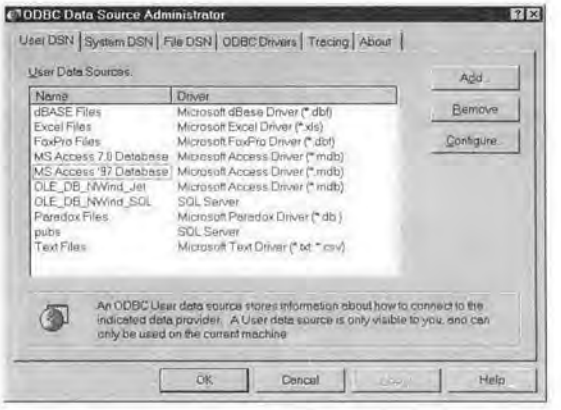


FIG. 1.3 ODBC 3.0's enhanced ODBC Data Source Administrator.

- *Microsoft Graph 8.0* represents a considerable improvement over version 5.0 included with Access 95. Graph 8.0 includes new chart types, such as Bubble, Cylinder, and Cone. Graph 8.0 is a 32-bit OLE applet used by all of the members of Office 97. Figure 1.4 shows most of the graph and chart types provided by the Graph 8.0.

▶▶ See "Using the Chart Wizard to Create an Unlinked Graph," p. 711

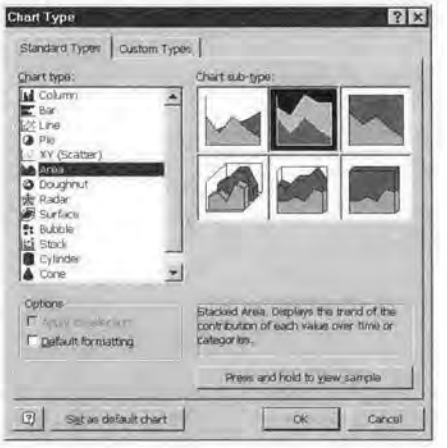


FIG. 1.4 Some of the graph and chart types you can create with Microsoft Graph 8.0.

- *Visual Basic for Applications 5.0* now is common to the Office 97 product line, including Word 97 but not Outlook. (Outlook uses Visual Basic, Scripting Edition, a subset of VBA 5.0, as its programming language). Visual Basic 5.0 also uses VBA 5.0. Access 97 makes extensive use of the Access `DoCmd` object, so this book uses the term *Access VBA* to identify code that relies on Access-specific objects.

Oracle Ex. 1009, p. 53
Oracle v. DeBellis
IPR2024-00221

▶▶ See "Introducing Access VBA," p. 978

- *The .mde file format* lets you save .mdb files without including readable VBA source code. This feature lets developers protect their intellectual property when creating Access 97 applications, libraries, and wizards.
- *Auto List Members* in the Module window displays a list of possible values for completion of statements. Typing the first letter of an object type, property, or method automatically positions the list (see Figure 1.5). Auto Quick Info provides syntax information for VBA. Auto Parameter provides parameter information for object methods. You can turn these features on and off with menu commands.

▶▶ See "Printing to the Debug Window with the **Debug** Object," p. 1018

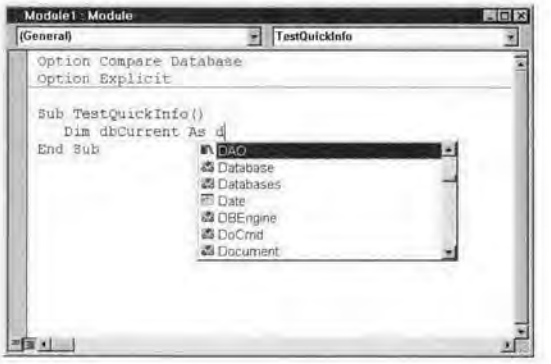


FIG. 1.5 The automatic Quick Info feature listing possible objects for completion of a variable declaration statement.

- *Programmatic references* to object libraries is an important new feature for Access developers. Access 97 provides the `References` collection, which you can manipulate with VBA 5.0.
- *Improvements to the Debug window* include a change from a modal dialog to a nonmodal window. You also can specify that the Debug window remain on top of other windows. The Expression pane of the Debug window shows you the current value of variables during execution of VBA code (see Figure 1.6).

▶▶ See "Using the Debug Window," p. 1016

- *A redesigned Object Browser* is shared by all Office 97 members (see Figure 1.7). You can display all of the objects to which you've created references, only the member objects of a specific reference, or search for an object by name. When you select a member object, such as a property or method, the syntax for the member object

Oracle Ex. 1009, p. 54
Oracle v. DeBellis
IPR2024-00221

automatically appears in the lower pane of the Object Browser. Clicking the Help button opens the help topic for the object or member object, if a help topic is available.

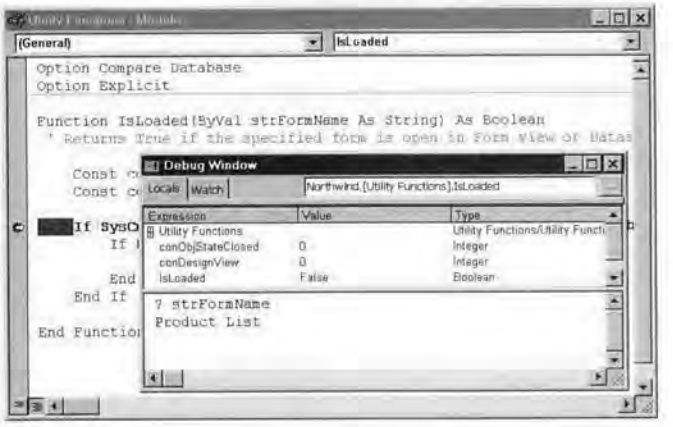


FIG. 1.6 Obtaining the value of a variable in the Debug window opened by a breakpoint placed in VBA code.

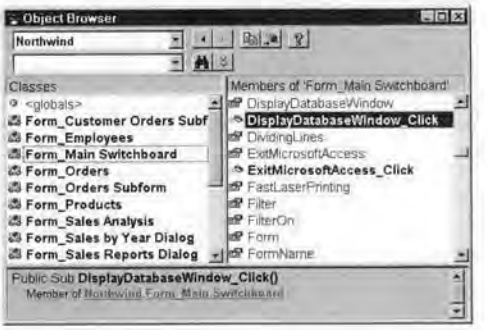


FIG. 1.7 Object Browser displaying objects in the Northwind project (Northwind.mdb).

To navigate through Microsoft's entire list of new Access 97 features, use Explorer to open the Acnew80.hlp file or choose Help, Contents and Index from Access 97's window, then click the Welcome to Microsoft Access 97—What's New txVic. Each of the individual help windows includes what's new in both Access 97 and Access 95.

▶▶ See "Examining Project Class Module Members in the Object Browser," p. 1054



Important Access 97 Changes

Changes primarily are of interest to users of prior versions of Access upgrading to Access 97. Following is a list of the principal differences between Access 95 and Access 97 that don't qualify as enhancements:

Oracle Ex. 1009, p. 55
Oracle v. DeBellis
IPR2024-00221

- *The Jet .mdb file structure has changed* for the fourth time. Access 97's Jet 3.5 .mdb file structure is not backwardly compatible with Access 95 or 2.0. You can open an Access 1.x, 2.0, or 95 database in Access 97, but you cannot open an Access 97 database in any prior version of Access. You cannot change in Access 97 the properties of database objects created with prior versions of Access; you must convert the file to Jet 3.5 format to make design-mode changes. Fortunately, you can link (the new term for *attach*) Access 1.x, 2.0, or 95 tables to Access 97 applications. Thus, you can accommodate simultaneous links to tables in Access 1.x, 2.0, and 95 .mdb files by database front-end applications created with Access 97 and its predecessors.



▶▶ See "Using Access 2.0 Application .mdb Files with Access 97," p. 1151

- *Access executable and support files are installed in a common default Office folder,* \Program Files\Microsoft Office\Office, together with the other members of Office 97. Example files, such as Northwind.mdb, are stored in \Program Files\Microsoft Office\Office\Samples.
- *System.mdw has moved.* Setup now installs the workgroup information file (previously called the *system file* or *system database*) in the *Windows*\System folder of Windows 95 or the *WinNT*\System32 folder of Windows NT 3.51+. Previously, System.mda (Access 2.0 and earlier) or System.mdw (Access 95+) was installed in the same folder as Msaccess.exe. According to Microsoft, the relocation of System.mdw is necessary to support the run-from-network and run-from-CD-ROM versions of Office 97.



▶▶ See "Creating a Folder and System File for File Sharing," p. 868

- *Wrkgadm.exe has moved,* apparently following System.mdw. Like System.mdw, the Workgroup Administrator application has moved from the ...\Access folder of prior versions to the *Windows*\System folder of Windows 95 or the *WinNT*\System32 folder of Windows NT 3.51+. Setup doesn't create a Start menu choice for Wrkgadm.exe; you should create a desktop shortcut to Wrkgadm.exe to expedite connection to multiuser Access applications.
- *Macro support is for backward compatibility only.* Now that all of the members of Office 97, except Outlook, support VBA 5.0, Microsoft wants you to substitute VBA event-handling code for macros. Access 95 required you to write custom menu macros; you program `CommandBar` objects to create custom menus and toolbars with Access 97. There's no assurance that future versions of Access will support macros, so Access macros are obsolescent, at best.



▶▶ See "Converting Access Macros to VBA Code," p. 1080

Oracle Ex. 1009, p. 56
Oracle v. DeBellis
IPR2024-00221

- *Some files you need are part of the Office 97 ValuPack.* As an example, the help files for programming Graph 8.0 are located in the \ValuPack\MoreHelp folder of the Office 97 CD-ROM. You must copy the .hlp (help) and .cnt (content) files you need to your ...\Office folder, then delete the existing .gid file to create a new search index for the help file.
- *Class modules replace code-behind-forms (CBF) modules* of Access 2.0 and Access 95. Class modules, which originated in Visual Basic 4.0, also let you write VBA code in the form of reusable components, each with its own set of custom properties and methods.

Many of the changes in the preceding list are the result of Microsoft's desire to fully integrate Access with the other members of the Office software suite. Microsoft's Office program management group is responsible for assuring "look and feel" consistency across all members of Office 97 and its successors.

Summarizing Access 95's Improvements to Access 2.0

Access 95 was a major upgrade from Access 2.0 and incorporated more than 100 significant changes to the last 16-bit version of Access. As mentioned at the beginning of this chapter, the "New in Access 95" features are included for the benefit of new Access users and those who skipped the Access 95 upgrade. Access 95's features and improvements, all of which are incorporated in Access 97, fell into the 10 categories of the following list:

- *User interface (UI) modifications* made Access 95 conform to the "look and feel" of the other members of Microsoft Office 95. The primary changes to the Access 95 UI included a tabbed Database window that features the Explorer look, new File Open common dialogs that also support import and link operations, and many relocated menu commands. Access 95's toolbars conformed to the Office 95 model. Tabbed dialogs used to set Access 95's operating environment options and set database file properties emulated Windows 95's property sheets. Windows 95 proportional scroll bars sized the scroll button to indicate the percentage of the data being displayed. When you clicked the scroll button, Access 95 displayed a "scroll tip" with the current record number and the total number of records. The primary change to Access 97's UI is the move to dynamic from 3-D toolbar buttons and addition of the optional Web toolbar (see Figure 1.8).

▶▶ See "Starting Access and Opening Databases," p. 47



- *Multithreaded preemptive multitasking* improved the performance of large Access 95 applications. Access and Visual Basic for Applications (VBA) each gained their own thread of execution and the Jet 3.0 Data Access Object became multithreaded. Very large Access 95 applications could take advantage of Windows NT's Symmetrical Multiprocessing (SMP) when running on a Windows NT workstation having two or

Oracle Ex. 1009, p. 57
Oracle v. DeBellis
IPR2024-00221

more Intel processors. There are no significant changes to the Access 95 and Jet 3.0 threading model in Access 95 and Jet 3.5.



FIG. 1.8 Access 97 displaying the new optional Web toolbar.

- *New and improved add-ins and wizards* aided Access users by automating a wider variety of tasks. The Database Splitter Add-In automated the process of separating data objects and application objects into individual .mdb files. The Table Analyzer Wizard found repeated data in tables imported from flat ASCII or spreadsheet files and automatically creates a related lookup table to eliminate the redundancy. The Performance Analyzer Wizard made suggestions for improving your application's operating speed. The User-Level Security Wizard automatically created a new secure database file from a conventional, unsecured database. Only minor changes have been made to Access 97's wizard collection; the new Publish to the World Wide Web Wizard is discussed in the section "Major Access 97 Enhancements" earlier in the chapter.
- *Data import and export operations* were enhanced. The Import/Export Wizard made setting up import and export specifications a snap. Exporting reports to other applications in .xls, .rtf, and .txt file formats included data in subreports, and data pasted into Excel worksheets included text formatting. Dragging selected data from an Access datasheet into an Excel worksheet also preserved formatting. Access 97 adds the ability to import and export HTML tables.
- *Lookup fields* let you make selections in table datasheets from a drop-down combo box or list box populated by a field of a related table or from a list of fixed values. Lookup fields generated controversy among relational database purists because the foreign key value stored in a column of the Lookup data type is replaced in Datasheet view by a value returned from another table. There is no change to the lookup field feature in Access 97.
- *Filter by Form and Filter by Selection* aided searches for the data. Filter by Form let you enter a value in a control of a form or a field of a datasheet, then let you limit by applying the filter the underlying data set to records that match the entered

Oracle Ex. 1009, p. 58
Oracle v. DeBellis
IPR2024-00221

value. Filter by Selection let you select a value in a form or datasheet and limit the underlying data set to records containing the selected value. Access 97 lets you type a filter criterion in the Filter For text box (see Figure 1.9).

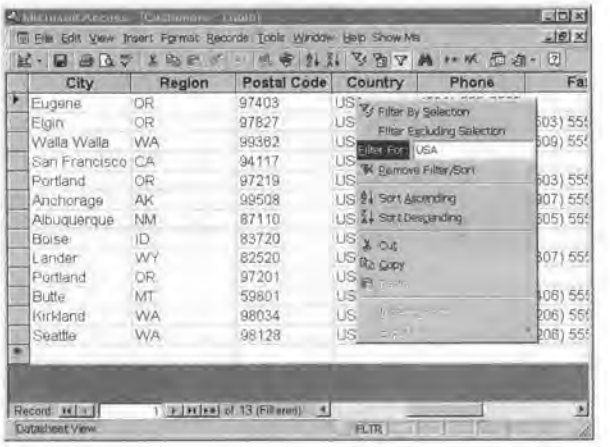


FIG. 1.9 Filtering by a criterion typed in the new Filter For text box on the popup menu for a table field.

- *Startup properties* eliminated the need for AutoExec macros and let you assign an opening form, customize the title bar caption, and control a variety of other application properties without writing macros or Access VBA code. There is no change to the Startup dialog in Access 97.
- *Briefcase replication* made easy the process of synchronizing changes to Access tables by users without network access to shared databases. You also can distribute updated versions of your application .mdbs with Windows 95 Briefcases. Access 97 adds the ability to create partial replicas, either with VBA code or by using the Partial Replica Wizard. The advantage of a partial replica is that you only include a subset of the records of a design-master replica table in the partial replica.

▶▶ See "Obtaining and Installing the Partial Replica Wizard," p. 972

The Partial Replica Wizard isn't included with Office 97 Professional Edition, Developer Edition, or the stand-alone version of Access 97. You can download the Partial Replica Wizard from

On the Web

http://www.microsoft.com/msaccess

- *Full support for 32-bit OLE 2.1* let you extend your repertoire of Access control objects with 32-bit OLE Controls (.ocxs) included with Visual Basic 4.0 and from third-party OLE Control publishers. Access 95 became an (OLE) Automation client and server. In addition to 32-bit OLE Controls, which are now obsolete, Access 97 is a container for ActiveX Controls. An ActiveX version of Access 95's Calendar control is included with Access 97 (see Figure 1.10).

Oracle Ex. 1009, p. 59
Oracle v. DeBellis
IPR2024-00221

▶▶ See "Using the Calendar Control," p. 1124



FIG. 1.10 A form containing the Calendar ActiveX Control, a date display text box, and command buttons to change the displayed date.

Installing Access 97

Office 97 Professional Edition includes a redesigned Setup program for installing the individual members of Office 97, as well as common files used by all of the member applications. If you choose the Typical Setup option, many of the files you need to take full advantage of Access 97's features aren't installed. The sections that follow describe the Setup process for a new installation and how to modify an existing Office 97 Typical installation to copy the added files required to gain maximum benefit from this book.

Making an Initial Installation of Access 97

The following steps describe how to install the Access 97 and Data Access components of Office 97 from the distribution CD-ROM, either as a new installation or as an upgrade to a prior version of Office, using the Custom Setup option:

1. Close all running applications and insert the distribution CD-ROM in your CD-ROM drive. Alternatively, you can use a CD-ROM drive shared by a server or another PC in your workgroup.
2. From the Windows 95 or Windows NT 4.0 Start menu, choose Run to open the Run dialog. Click the Browse button, select your CD-ROM drive, and double-click the Setup.exe item to close the Browse dialog.
3. Click OK to close the Run dialog and start the Setup program. After a few seconds, depending on the speed of your CD-ROM, Setup's opening dialog appears (see Figure 1.11).
4. Click continue to open the Name and Organization dialog. Setup obtains the default entries for the Name and Organization text boxes that came from the entries you made when installing Windows 95 or Windows NT (see Figure 1.12). Edit the entries, if necessary, and click OK to open the Confirm Name and Organization dialog. Click OK again to continue.

Oracle Ex. 1009, p. 60
Oracle v. DeBellis
IPR2024-00221



FIG. 1.11 Office 97's opening Setup dialog for a new installation.

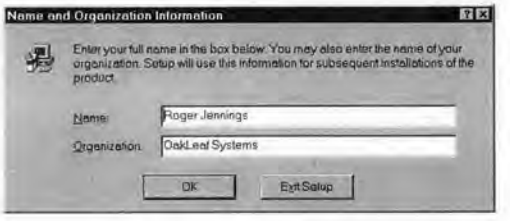


FIG. 1.12 Specifying your name and optional organization name for registering Office 97.

5. Type the CD Key number from the sticker on the Office 97 distribution CD-ROM cover in the two CD Key text boxes and click OK (see Figure 1.13).



FIG. 1.13 Providing the CD Key number to verify your Office 97 license.

6. Click OK in the Setup dialog that displays the Product ID for Office 97 to open the folder location dialog. Microsoft proposes to install most of the Office 97 files in subfolders of C:\Program Files\Microsoft Office (see Figure 1.14).

Oracle Ex. 1009, p. 61
Oracle v. DeBellis
IPR2024-00221



FIG. 1.14 The Setup dialog displaying the default folder for installation of most Office 97 files.

If you want to change to a different drive or folder, click Change Folder to display the Change Folder dialog (see Figure 1.15). Type the drive and path for the Office 97 files or select an existing folder, either on the C: drive or on another different logical drive, then click OK to close the Change Folder dialog. After selecting the location for the files, click OK to continue.



FIG. 1.15 The Change Folder dialog for specifying a non-default location for Office 97 files.

> **Note**
>
> The installation process also installs a substantial number of files in your \Windows\System or \WinNT\System32 folder, and into subfolders of the \Program Files\Common Files\Microsoft Shared folder. The location of these files is not determined by the folder you choose for installing Office 97.

7. Click the Custom button to specify user-selected options for all Office 97 applications (see Figure 1.16). By selecting optional components carefully, you can install Office 97 and consume substantially less disk space than selecting the Typical option.

Oracle Ex. 1009, p. 62
Oracle v. DeBellis
IPR2024-00221

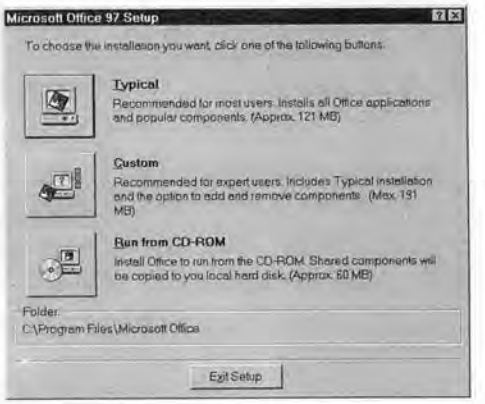


FIG. 1.16 The three installation options for Office 97.

8. In the Microsoft Office—Custom dialog, select Microsoft Access (see Figure 1.17). You can click the Select All button to install all Access files on the CD-ROM or the Change Option button to choose what files to install. To follow this procedure, click the Change Option button.

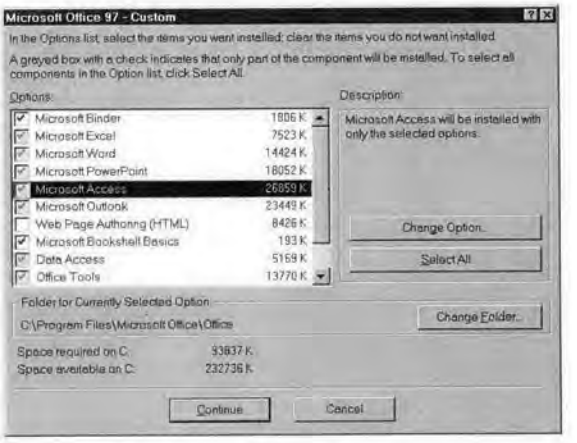


FIG. 1.17 The opening version of the Microsoft Office 97—Custom dialog.

Note

Boxes with a check mark and a white background indicate that all files in the category will be installed unless you change the option. Boxes with a gray check mark and light gray background indicate that only those files that correspond to the Typical Setup option will be installed.

Oracle Ex. 1009, p. 63
Oracle v. DeBellis
IPR2024-00221

9. Figure 1.18 shows the default settings for the Microsoft Office 97—Microsoft Access dialog. There are no options for installing Microsoft Access Program Files. All of the Help Topics files are not installed, so click to select the Help Topics item, then click the Change Option button.

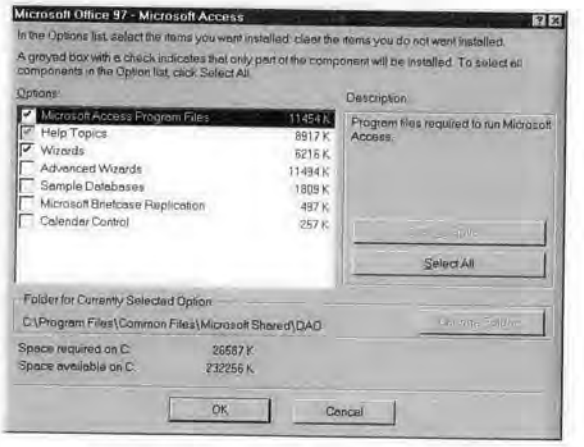


FIG. 1.18 The opening version of the Microsoft Office 97—Microsoft Access options dialog.

10. In the Microsoft Office 97—Help Topics dialog, click the Select All button or click the Programming Help item to specify installation of the help file (Acvba80.hlp) for Access VBA (see Figure 1.19). Click OK to return to the Microsoft Office 97—Microsoft Access dialog.

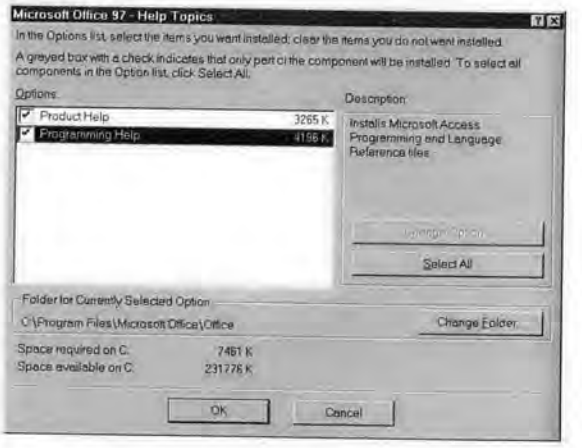


FIG. 1.19 Specifying installation of the Programming Help file for Access.

11. In the Microsoft Office 97—Microsoft Access dialog, click the Advanced Wizards item of the Options list to install all of the Access 97 wizards.

Oracle Ex. 1009, p. 64
Oracle v. DeBellis
IPR2024-00221

12. Select Sample databases to open the Microsoft Office 97—Sample Databases dialog. Click Select All to install all three of the sample databases: Northwind.mdb, Orders.mdb, and Solutions.mdb (see Figure 1.20). Click OK to return to the Microsoft Office 97—Microsoft Access dialog.

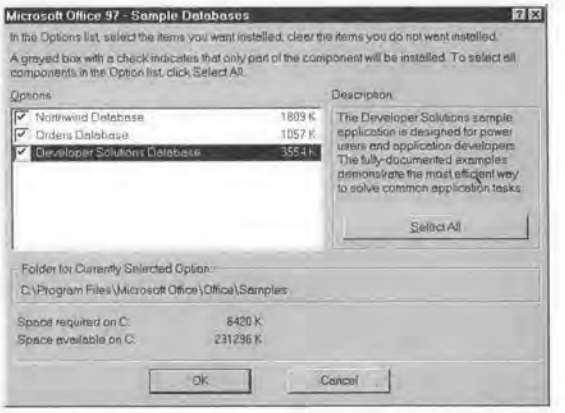


FIG. 1.20 Selecting all of the sample databases for installation.

> **Note**
>
> Northwind.mdb is used for most of the examples in this book. Orders.mdb demonstrates how to create a simple order entry application. Solutions.mdb offers a variety of interesting examples of the use of new features of Access 97, including VBA programming techniques.

13. In the Microsoft Office 97—Microsoft Access dialog, click to select the Microsoft Briefcase Replication and Calendar Control items. The dialog appears as shown in Figure 1.21.



FIG. 1.21 Adding Briefcase Replication and the Calendar Control to the installation options.

Oracle Ex. 1009, p. 65
Oracle v. DeBellis
IPR2024-00221

> **Note**
>
> Figure 1.21 shows that all of the components of Access 97 options are selected. It's faster to simply click the Select All button, as noted in preceding step 8, but going through the Access 97 option selection process prepares you for choosing installation options for other members of Office 97.

14. Click OK to return to the Microsoft Office 97—Custom dialog and select Data Access. Click the Change Option button to open the Microsoft Office 97—Data Access dialog (see Figure 1.22). The Data Access choices you make apply to all members of Office 97 that use the Jet 3.5 Data Access Object (DAO), including Access 97.

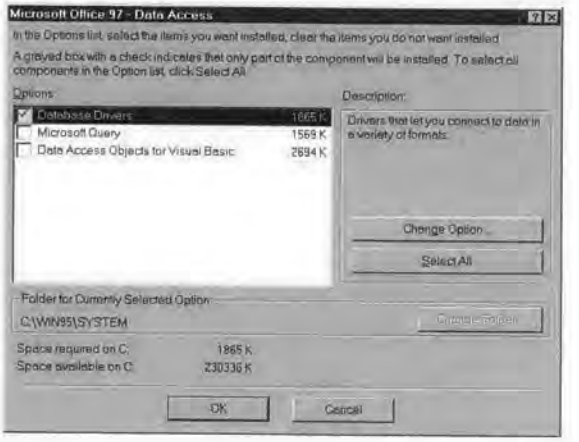


FIG. 1.22 The default option selections of the Microsoft Office 97—Data Access dialog.

15. Select the Database Drivers item in the Options list and click the Change Option button to open the Microsoft Office 97—Database Drivers dialog. All of the drivers except the Microsoft SQL Server Driver are selected by default.

> **Note**
>
> The Access, dBASE and FoxPro, Excel, and Text drivers formerly were called the *Desktop Database Drivers*. All drivers are 32-bit ODBC drivers. The HTML driver is a new Office 97 driver.

16. If you intend to use Access 97 as a front-end to SQL Server 6.x, click to select the SQL Server driver (see Figure 1.23). Click OK to return to the Microsoft Office 97—Data Access dialog.

You can save some disk space by deselecting the dBASE and Microsoft FoxPro Drivers, if you don't intend to import, export, or link xBase files. The Microsoft Access

Oracle Ex. 1009, p. 66
Oracle v. DeBellis
IPR2024-00221

Driver is required for other Office applications to import, export, or link Access databases.

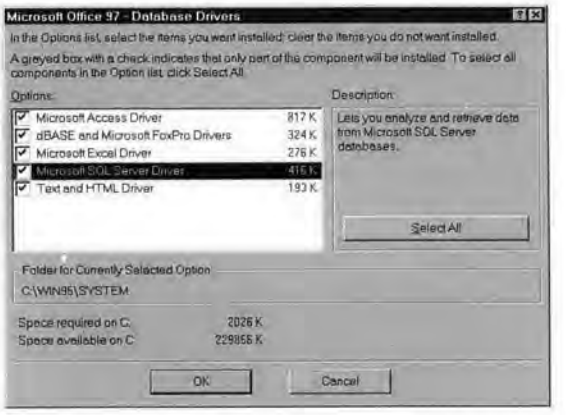


FIG. 1.23 Specifying installation of the 32-bit ODBC 3.0 driver for Microsoft SQL Server 6.x.

17. Click to select the Data Access Objects for Visual Basic item of the Options list. Each of the members of Office 97 can create a reference to the Jet 3.5 DAO. If you selected all of the database drivers in the preceding step, your Microsoft Office 97—Data Access dialog appears as shown in Figure 1.24. Click OK to return to the Office 97—Custom dialog.

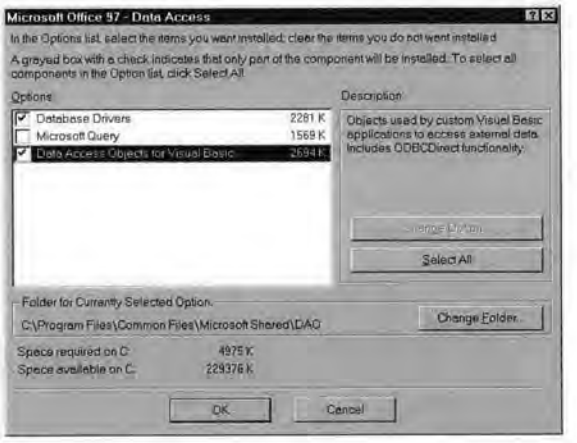


FIG. 1.24 Adding the Jet 3.5 DAO for installation.

18. Choose the options you want to install for the other members of Office 97, as well as for files that are common to Office 97. When you've completed the specification process, click Continue in the Microsoft Office 97—Custom dialog to begin copying files.

Oracle Ex. 1009, p. 67
Oracle v. DeBellis
IPR2024-00221

> **Note**
>
> Once you click Continue in the Microsoft Office 97—Custom dialog, there's no way to change the options you selected. If you forgot to specify options for other applications, you can click Cancel in the Copy Progress dialog to terminate the Setup program and start over. Alternatively, you can let Setup run its course, then modify the installation as described in the next section.

Setup checks for available disk space, then copies files from the CD-ROM to your fixed disk. During the copying process, the dialog shown in Figure 1.25 displays installation progress.



FIG. 1.25 The Progress Monitor dialog during the file copying process.

19. When installation is complete, the dialog shown in Figure 1.26 appears. You can elect to use the Registration Wizard to register your copy of Office 97 online; otherwise, click OK to complete the Setup process. Depending on the installation, you may be requested to reboot Windows at this point.



FIG. 1.26 The Setup dialog indicating completion of Office 97 installation.

> **Note**
>
> The installation of the stand-alone version of Access 97 is similar to that for the Office 97 Professional Edition, except that you complete only the Access 97 and Data Access option steps. The stand-alone version of Access 97 primarily is of interest to purchasers of the standard Office 97 or Small Business editions.

Customizing a Typical Installation

If you installed Office 97 with the Typical option or want to install or remove options from an existing Typical or Custom Office 97 installation, follow these steps:

Oracle Ex. 1009, p. 68
Oracle v. DeBellis
IPR2024-00221

1. Close all running applications and insert the distribution CD-ROM in your CD-ROM drive. Alternatively, you can use a CD-ROM drive shared by a server or another PC in your workgroup. The CD-ROM is required even if you intend only to delete previously selected options.
2. From the Windows 95 or Windows NT 4.0 Start menu, choose Run to open the Run dialog. Click the Browse button, select your CD-ROM drive, and double-click the Se‹up.exe item to close the Browse dialog.
3. Click OK to close the Run dialog and start the Setup program. After a few seconds, depending on the speed of your CD-ROM, Setup's maintenance dialog appears (see Figure 1.27).

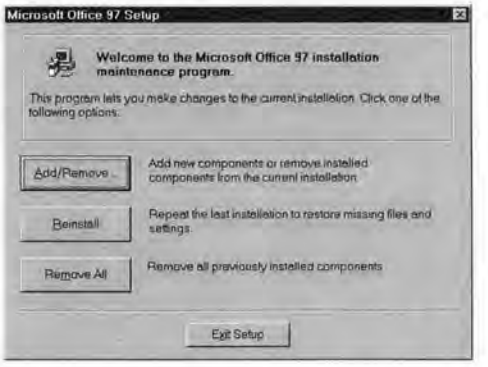


FIG. 1.27 The opening dialog of the maintenance version of the Setup program.

4. Click the Add/Remove button to open the Microsoft Office 97—Maintenance dialog. Select Microsoft Access (see Figure 1.28) or any other component whose setup options you want to change, and click the Change Options dialog.

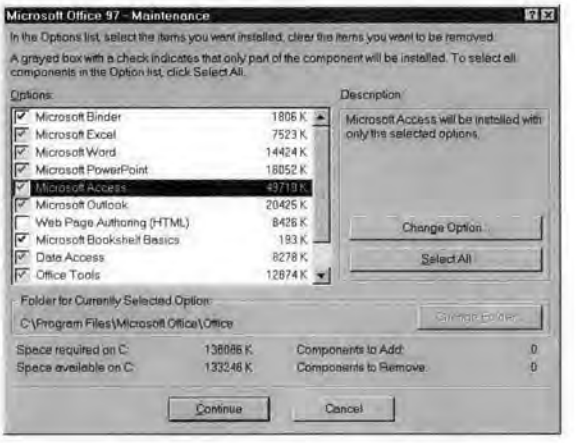


FIG. 1.28 The Office 97—Maintenance dialog that lets you add or remove options for all members of Office 97.

Oracle Ex. 1009, p. 69
Oracle v. DeBellis
IPR2024-00221

5. If you're changing the options for a Typical installation of Access 97, follow steps 9–18 of the preceding selection to add the features you need to take full advantage of Access 97 and the examples of this book.

Installing ValuPack Components

The Microsoft Office 97 ValuPack, which is included only on the CD-ROM version of Office 97, includes a variety of useful utilities and accessories for Office 97 users and developers. To check out the ValuPack's contents, follow these steps:

1. With the Office 97 Professional Edition CD-ROM in an accessible CD-ROM drive, open Explorer and navigate to the \ValuPack folder of the CD-ROM.
2. Double-click the Valupack.hlp file to open the main help topic, Welcome to the Microsoft Office 97 ValuPack (see Figure 1.29).



FIG. 1.29 The opening help topic of the Valupack.hlp file.

3. Click the button associated with the ValuPack topic you want to examine. Figure 1.30 shows the topic for the HTML version of the book *Building Applications with Microsoft Access 97*, the printed version of which is included with the Office 97 Developer Edition. Many of the ValuPack utilities have their own setup file to install the required files on your local fixed disk.
4. For a more graphic-intensive view of the content of the ValuPack folder of the CD-ROM, double-click the Overview.ppt file to launch PowerPoint 97. Figure 1.31 shows the first slide of the Overview.

Oracle Ex. 1009, p. 70
Oracle v. DeBellis
IPR2024-00221



FIG. 1.30 A description of the HTML version of *Building Applications with Microsoft Access 97*.

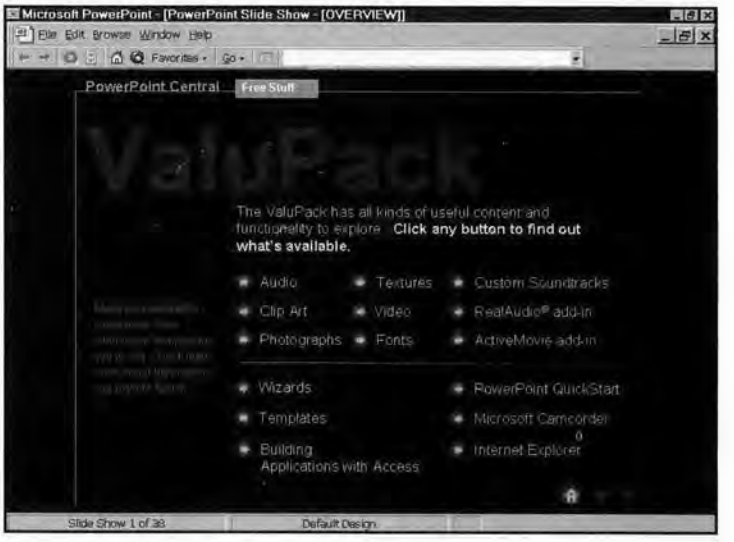


FIG. 1.31 The first PowerPoint 97 slide for the Overview of the ValuPack.

Oracle Ex. 1009, p. 71
Oracle v. DeBellis
IPR2024-00221

Installing the Office Developer Edition Components

The Office 97 Developer Edition CD-ROM consists of two components: the Office 97 Resource Kit (ORK) and Office 97 Developer Edition (ODE) Tools. Following are the steps you take to install the Access 97 Setup Wizard and other tools included with the Developer edition:

1. With the ODE CD-ROM in an accessible CD-ROM drive, from the Windows 95 or Windows NT 4.0 Start menu, choose Run to open the Run dialog. Click the Browse button, select your CD-ROM drive, and navigate to the \ODE folder. Double-click the Setup.exe item to close the Browse dialog and open the first Setup dialog (see Figure 1.32).

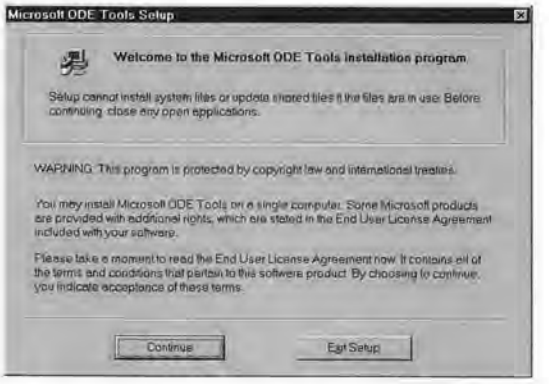


FIG. 1.32 The opening dialog of the Microsoft Office 97 Developer Edition ODE Tools Setup program.

2. Click Continue to enter and confirm your name and organization. When you click OK in the Product ID dialog, the dialog to select the installation folder appears (see Figure 1.33). Unless you have a good reason to do otherwise, select the default location for the ODETools folder.

3. Click OK to open the Installation Options dialog (see Figure 1.34). Most Access developers are likely to install all of the ODE tools, so click the Complete button to install the Setup Wizard, Replication Manager, and other ODE tools. The Setup application installs the files in the folder specified in the preceding step.

Oracle Ex. 1009, p. 72
Oracle v. DeBellis
IPR2024-00221

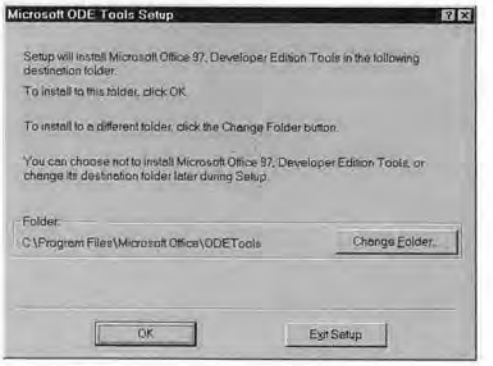


FIG. 1.33 Accepting the default location for the ODETools folder.

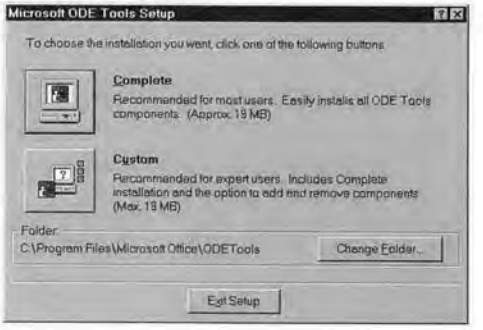


FIG. 1.34 Choosing between Complete and Custom installation of the ODE Tools.

Oracle Ex. 1009, p. 73
Oracle v. DeBellis
IPR2024-00221

Chapter 2

Up and Running with Access 97



This chapter is designed to give you an overview of Access's two most important database objects: databases and tables. Even if you're well acquainted with previous versions of Access, you might want to browse this chapter for the Access 97 icon in the margin that indicates a description of a new feature or an improvement over Access 95. You see the Access 97 icon at each location in this book where a new or improved feature is discussed except in Chapter 1, which is devoted to the novel elements of Access 97.

The examples in this chapter, like most other examples in this book, use the Northwind Traders sample database, Northwind.mdb, that is supplied with Access 97. Northwind Traders is a fictitious wholesaler of specialty food products whose accounting system is an Access database application created by Microsoft. The sample database includes tables, forms, and reports that a small firm might use to automate its invoicing, inventory control, ordering, and personnel operations.

The remainder of Part I (Chapters 3 – 7) expands on the subjects you learn in this chapter using Northwind.mdb and describes in detail how you create and use Access database tables.

Note

This chapter assumes that you have already installed Access. If you haven't installed Access, refer to your Access documentation for instructions on setting up Access on your computer.

Starting Access and Opening Databases

Before you can open any database in Access, you must start Access itself. You can open a database at the same time you start Access, or you can open a database any time after you've started Access. You also can run multiple

Oracle Ex. 1009, p. 74
Oracle v. DeBellis
IPR2024-00221

instances of Access, if you have enough system resources. Access allows you to have only one database per instance open at a time, however.

The next two sets of instructions first explain how to open a database at the same time you start Access 97, and then explain how to open a database after Access is already running.

Opening a Database During Access Startup

To simultaneously start Access and open the Northwind Traders database, Northwind.mdb, perform the following steps:

1. Choose Microsoft Access from Windows 95's or Windows NT 4.0's Start, Programs menu. (Access is installed on this menu by default.)

   As Access is loading, a copyright notice (called a *splash screen*) appears with the name and organization that was entered at the time Access was installed. After Access has loaded, it displays the database opening dialog shown in Figure 2.1.

   Notice the list of previously opened databases at the bottom of the dialog; if this is the first time you've ever run Access, this list does not appear.

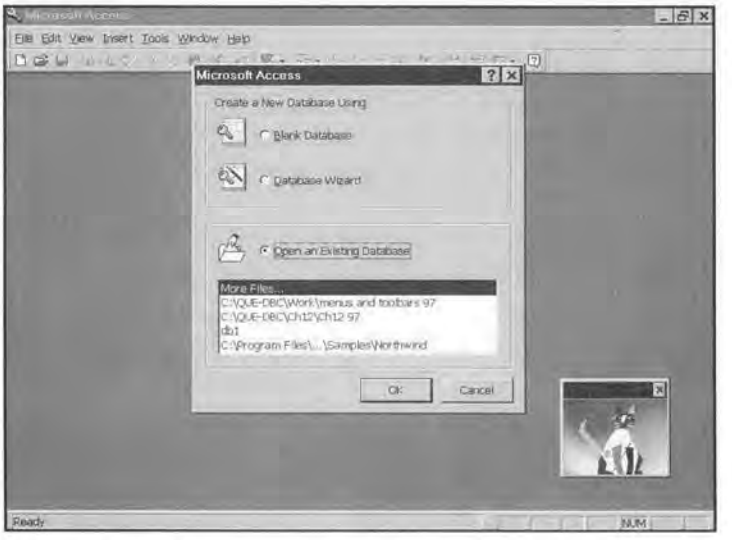


FIG. 2.1 The Access 97 startup dialog for opening a database.

2. Select the Open an Existing Database option.
3. Select the More Files choice in the list at the bottom of the dialog, and then click OK. (If this is the first time Access has been run, the list at the bottom of the dialog doesn't appear; in this case, just click OK.) Access displays a standard Windows 95 Open dialog, shown in Figure 2.2.

Oracle Ex. 1009, p. 75
Oracle v. DeBellis
IPR2024-00221

4. Select the \Program Files\Microsoft Office\Office\Samples folder in the Open dialog. (The Northwind.mdb files are located in the Samples subfolder of whatever folder in which you installed Access 97; the default location is \Program Files\Microsoft Office\Office.) Figure 2.2 shows the Open dialog with the contents of the Samples folder displayed.



FIG. 2.2 Displaying the sample database in the Open dialog.

5. Double-click Northwind.mdb to open the Northwind Traders sample database. Access opens the Northwind Traders database.

   If this is the first time you've opened the Northwind Traders sample database, the Access Visual Basic for Applications (VBA) program code in the sample database displays a special splash screen for the Northwind Traders database (see Figure 2.3). If you don't want to see this splash screen again, mark the Don't Show This Screen Again check box.

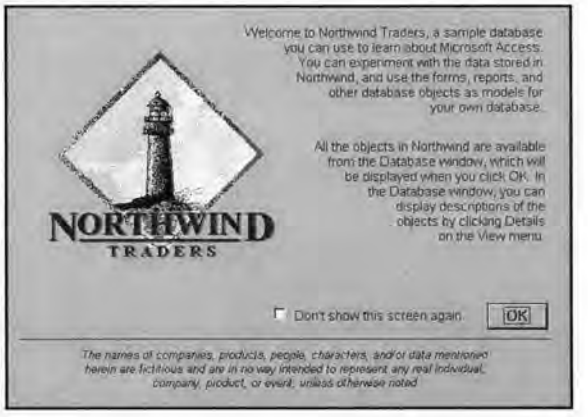


FIG. 2.3 The splash screen of the Northwind Traders sample database.

Oracle Ex. 1009, p. 76
Oracle v. DeBellis
IPR2024-00221

6. Click OK to close the Northwind Traders splash screen. After closing the Northwind Traders splash screen, Access displays the Database window shown in Figure 2.4.



FIG. 2.4 The Database window, Access's "home base."

Note

You can use several other methods to open a database in the common Open dialog. You can click Northwind.mdb to select it and then click Open. Or you can type **Northwind** in the File Name text box and click OK. Access adds the default .mdb extension for you. However, double-clicking is the fastest and most mistake-proof method of opening a database.

Opening a Database When Access Is Already Running

As you'd expect, it's also possible to open a database after you start Access. You can have only one database open at a time in an instance of Access, so opening a database causes Access to close any currently open database, and replaces the contents of the Database window with the newly opened database's information. (The Database window is explained in the next section of this chapter.)

To open the Northwind Traders database, Northwind.mdb, when Access is already running, follow these steps:



1. Click the Open Database button of the toolbar, or choose File from the main menubar. If you use the menu to open a database, the File menu appears (see Figure 2.5); if you click the Open Database toolbar button, the file Open dialog appears immediately.

   If you have run Access previously, the file names of up to four Access database files you have opened may appear above Exit.

2. Choose File, Open Database to display the Open dialog (refer to Figure 2.2).

3. Finish opening the Northwind Traders database by following the procedures in steps 4–6 of the preceding section ("Opening a Database During Access Startup").

Oracle Ex. 1009, p. 77
Oracle v. DeBellis
IPR2024-00221

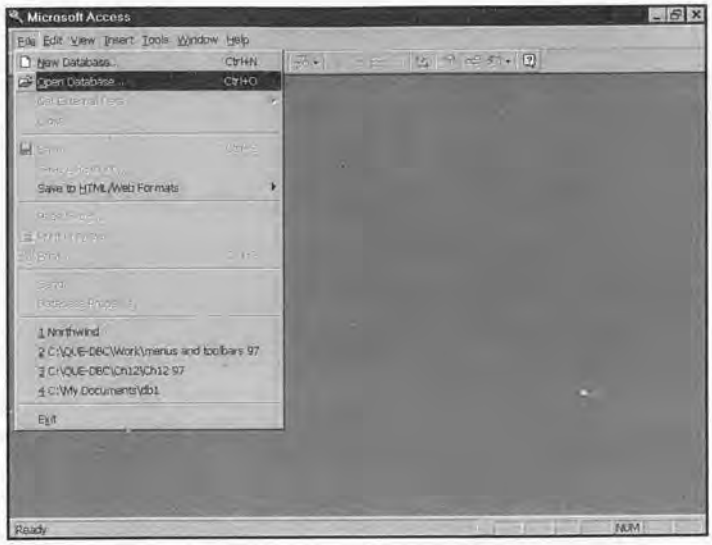


FIG. 2.5 Choosing File, Open Database to open an Access database.

Note

Access's toolbar buttons provide ScreenTips (called ToolTips in earlier versions of Access). *ScreenTips* are small captions that appear under a button after the mouse pointer has rested on a button for more than about 0.5 seconds. If you find ScreenTips to be distracting, you can turn them off by choosing View, Toolbars, Customize to display the Customize dialog for toolbars. Click the Options tab to display the Options sheet, and then click the Show ScreenTips on Toolbars check box to remove the check mark; click the Close button to close the Customize dialog. If you're using SVGA mode (800×600 pixels) or UVGA mode (1,024×768 pixels), you may want to try clicking the Large Buttons check box of the Toolbars dialog. The icons of larger buttons are easier to discern in SVGA mode and especially UVGA mode.

Note

You can close any open database by clicking the close button at the top right corner of the Database window, or by choosing File, Close. When there is no open database, Access displays an empty window; only the New Database, Open Database, and Office Assistant toolbar buttons are enabled; and the File, Close command is disabled.

Understanding the Database Window

The Database window is your "home base" for all operations with the sample database or databases you create. Almost every operation you perform with Access begins with a choice you make from the Database window.

Oracle Ex. 1009, p. 78
Oracle v. DeBellis
IPR2024-00221

After you open a database file, the 20 buttons to the right of the New Database and Open Database buttons of the Database toolbar (under the main menubar) are enabled or disabled, depending on Access 97's current status. *Toolbar buttons*, a common feature of contemporary Windows applications (and of Windows 95 itself), are shortcuts for menu choices.

Any operation you can perform by clicking a toolbar button can be performed by making two or more menu choices. Access 97 has 20 standard toolbars; Access displays the appropriate toolbar for the database object in the active window. You can customize the standard toolbars or create your own special-purpose toolbars. Using the toolbar buttons is much quicker and, once you learn what the symbols mean, more intuitive than using the standard menu structure. Chapter 3, "Navigating Within Access," describes the use of Access 97's new toolbars in detail.



▶▶ See "The Toolbars in Table View," p. 71

Note



Access 97 and the other members of the Microsoft Office 97 software suite use the term *command bars* to refer collectively to menubars and toolbars. Menubars and toolbars perform identical functions; they differ only in appearance. Toolbars, however, offer more flexibility than menubars. For instance, toolbars can include drop-down lists.

Using Database Tables

This chapter is devoted to exploring tables, which are the basic elements of all databases and the portion of the Access database file where data is stored.



▶▶ See "Understanding Access's Table Display," p. 67

Viewing Data in Tables

To display the contents of the Categories table, which describes the types of products in which Northwind trades, double-click Categories in the Database window's list of tables. The Categories table appears, as shown in Figure 2.6.

Tip



You also can display the Categories table by clicking Categories and then clicking the Open button or pressing Enter. The double-clicking method is quicker, however.

Oracle Ex. 1009, p. 79
Oracle v. DeBellis
IPR2024-00221

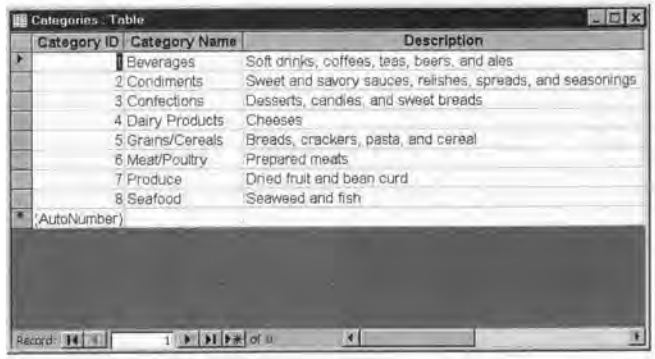


FIG. 2.6 The Datasheet display of the records in the Categories table.

The normal method of displaying the information contained in relational database tables is the familiar row-column technique used by spreadsheet applications. Every widely used PC RDBMS (relational database management system) uses row-column presentation for data contained in tables. The rows of a table are known as *records*, and the columns are referred to as *fields*. A record contains information on a single item, such as one class of product or a particular invoice. A field contains the same type of information, such as a product code, for all records in a table. In this book, the intersection between a row and a column—a single field of a single record—is called a *data cell*. A data cell contains a single piece of information. The terms *data item* and *data entity* often are used as synonyms for data cell.



Figure 2.6 shows the data in the Categories table in what Access calls *Datasheet view*—the default method of displaying a table. You select Datasheet view, if another type of view is active, by clicking the Datasheet button, which resembles a spreadsheet, on the toolbar (the Datasheet button only appears when the table is *not* in Datasheet view). You also can choose View, Datasheet. Datasheet view also is the default for viewing the results of queries; you can see a Datasheet view of the table or query when you are creating an Access form.

Selecting and Editing Data Records

To change the data contained in any cell of the table (other than in OLE Object fields used to display pictures or hyperlink fields), select the cell by clicking it with the mouse or by using the arrow keys. Selected content is indicated by the white on black (reverse) appearance of the cell. The default selected cell is the first field of the first record of the table. All contents of that cell are selected when you first display the table.

Caution

If you type a character into a cell when its entire content is selected, the character you type replaces what was in that cell. You can recover from an accidental replacement by pressing Esc; this action restores the original content, but only if you have not yet selected a different cell. When you

(continues)

Oracle Ex. 1009, p. 80
Oracle v. DeBellis
IPR2024-00221

(continued)

make a change in a data cell and then move to a new data cell, the change is made to the content of the table. Click the selected data cell to deselect the entire contents of the cell. You can choose Edit, Undo Saved Record or press Ctrl+Z to reverse the changes you made after saving a record.

Note

See Chapter 5, "Entering, Editing, and Validating Data in Tables," for detailed information on editing data records.

You can select any cell to edit by positioning the mouse pointer at the point within a cell where you want to make the change and then clicking your left mouse button. The mouse pointer resembles an I-beam when it is located within the contents of data cells. The editing cursor, a thin vertical line called the *caret* by Windows and this book (often referred to as the *insertion point* in other texts), appears.



The conventional text-editing functions of Windows applications apply to Access. At this point, you shouldn't change the data, called *values*, of the data cells. Changes to values may affect the appearance of examples in later chapters. If you do make a change, a pencil symbol appears in the gray box at the left of the window corresponding to the record whose value you have changed. You can return to the original value by pressing Esc.



If you click an OLE Object field, the cell has a gray border and the caret does not appear. (The Categories table contains an OLE Object field, although it isn't visible in Figure 2.6; the field stores a bitmap picture, and contains the words `Bitmap Image` to let you know that this is an OLE Object field containing picture data.) You cannot edit the text of an OLE Object field. If you click a hyperlink field, Access invokes your default Internet World Wide Web browsing application, and attempts to connect to the URL or UNC address specified by the hyperlink. (The Categories table doesn't contain a hyperlink field; the Suppliers table of Northwind.mdb does contain a hyperlink field.) To select a hyperlink field without starting your Web browser, move the caret into the field using the keyboard instead of the mouse. (A *URL* is a Universal Resource Link address and is used primarily to specify Web addresses; UNC stands for Universal Naming Convention, and is used primarily for addresses in a local or wide-area network.)



▶▶ See "Understanding Access 97's Hyperlink Field Data Type," p. 585



You use the empty record at the bottom of the table, indicated by the asterisk (*) in the selection button column for the record, to add a new record to the table. If you enter any data into a cell of this record, a new record containing the data you entered is added to

Oracle Ex. 1009, p. 81
Oracle v. DeBellis
IPR2024-00221

the table. If you accidentally add data in this *tentative append* record, click the Record Selection button to select the record, and then press Delete. A dialog appears, requesting confirmation of your deletion. Click Yes. (A record's *selection button* is the gray square that appears at the left of each row in Datasheet view; refer to Figure 2.6.)

If you use the arrow keys to select a data cell, all contents of the cell are selected. You cannot click the corner of a cell and drag the mouse pointer to a cell in the opposite corner of an imaginary rectangle to select a group of cells, as you do with Excel. You can, however, select an entire record by clicking the selection button for that record, or you can select a group of records by dragging the mouse down the selection button column. Records and groups of records most often are selected to copy the records to the Windows Clipboard. Notice that when you select a cell with the keyboard or the mouse, the triangular arrow moves to the selected record. The button with the triangular arrow is called the *current record pointer*, or sometimes the *active record pointer*. (The first record in Figure 2.6 is the current record; notice the triangular arrow in the record selection button at the left edge of the record.)



You use the left set of buttons in the bar at the bottom of the datasheet window to position the current record pointer within the bounds of the table. Click the button with a left-arrow and vertical bar to position the current record pointer at the first record in the table. Click the button with a right arrow and vertical bar to move to the last or bottom record of the table. The left- and right-arrow buttons move the pointer one record at a time in the direction indicated. You can enter a record number in the Record text box or use the vertical scroll bar to position the active record pointer. The vertical scroll bar appears only if more records are present than can fit in the vertical dimension of the Datasheet window.

The horizontal scroll bar enables you to view fields that lie to the right of the far right visible field. Use the left and right arrows for small movements, or use the scroll box to make large moves to the left or right. You also can traverse several fields by clicking the region of the scroll bar between the scroll box and the arrow. If you click the right scroll arrow in the Categories table two or three times, you'll be able to see the column for the Picture field, which is an OLE Object field containing Windows Paint bitmap pictures.



To select all data cells in a field column, click the button containing the field name at the top of the datasheet; as an example, Figure 2.7 shows the Description column selected. When the mouse pointer is over a field name, the pointer changes to a down arrow. You can search for cells within the selected field that contain a particular group of characters by clicking the toolbar's Find button—a pair of binoculars. The Find dialog appears (see Figure 2.7).

Oracle Ex. 1009, p. 82
Oracle v. DeBellis
IPR2024-00221



FIG. 2.7 Locating records whose fields contain particular words or phrases with the Find dialog.

Enter the characters you want to find in the Find What text box. Then open the Match drop-down list box by clicking the down arrow, and select Any Part of Field. You can search up, down, or in both directions from the position of the current record by selecting Up, Down, or All in the Search drop-down list box. If you select All, Access searches down from the current record until it reaches the end of the database, and then continues from the beginning until the current record position is reached. When searching Up or Down, Access searches in the specified direction and stops when it reaches the beginning or end of the database records. Click the Find First button to search for the first occurrence of a match, and then click the Find Next button to locate other matches. After you find all matching records or no matching records, a dialog appears stating that you have reached the end of the table. Click the Close button to return to the Table window.

Viewing and Editing Graphics

One of the reasons for choosing the Categories table for this chapter is that it includes bitmapped graphic images stored in an OLE Object field. If you double-click one of the Bitmap Image data cells, Windows 95's Paint window appears with the bitmapped image displayed, as shown in Figure 2.8. (You may need to scroll to the right to make the Picture field column visible in the window.)

You can use Paint to display the image or to edit it. Because Windows 95's Paint is an ActiveX component (referred to in earlier versions of Microsoft Office as an OLE server), all of Paint's bitmapped image display and editing capabilities are available to you while you are using Access. (In-place editing isn't suitable in Datasheet view, so Access invokes Paint in a separate window if you edit a Bitmap Image field when the table is in Datasheet view.)

When you're finished viewing an image, click the Close Window button in the upper-right corner of Paint's window to close the image. You also can exit an ActiveX component by choosing File, Exit & Return.

Oracle Ex. 1009, p. 83
Oracle v. DeBellis
IPR2024-00221



FIG. 2.8 A bitmapped image displayed using Windows 95 Paint.

Caution

If you make changes to a Bitmap Image OLE Object field, Access automatically stores your changes when you exit Paint; your changes cannot be undone.

Using Design Mode



Design mode for tables displays the characteristics of each field in the table in a grid format similar to a spreadsheet. To view the design of the Categories table, click the Design View button, which appears as an architect's triangle and pencil, at the left end of the toolbar. Design mode appears, as shown in Figure 2.9. Alternatively, you can choose View, Table Design or select the table name in the Database window and click the Design button.

▶▶ See "Defining Access Operating Modes," p. 66



You can view and edit the properties that apply to the table object as a whole in the Table Properties window, as shown in Figure 2.10. Click the Properties button on the toolbar (a hand pointing to a window) or choose View, Properties to display t he Table Properties window. The Table Properties window enables you to enter a text description of the table and assign a value to the Validation Rule property of the table. Table-level validation rules can contain a rule that applies to more than one field in a table. The Validation Text text property is the message that appears if you attempt to violate the validation rule.

▶▶ See "Validating Data Entry," p. 170



Oracle Ex. 1009, p. 84
Oracle v. DeBellis
IPR2024-00221

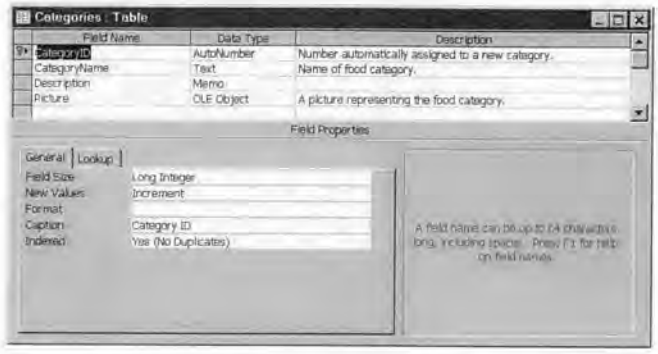


FIG. 2.9 Displaying Field Name, Data Type, and Description properties in Design view.

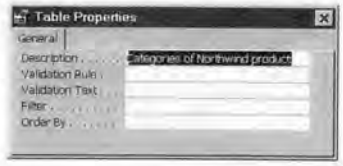


FIG. 2.10 Adding validation rules and messages that appear when the rules are violated in the Table Properties window.

Click the Indexes button or choose View, Indexes to display the Indexes window. The Indexes window, shown in Figure 2.11, displays each of the fields of the table and identifies the *primary-key field*, which is the field with the symbol of the key in its selection button column. The primary-key field is the field or combination of fields that is used to uniquely identify each record in the table. Most tables are indexed on a single primary-key field, and primary-key field indexes do not permit duplicate keys. *Indexes* are internal tables that speed the creation of query result tables by simulating the sorting of the table on the value of the key field. Key fields establish the relations by which multiple tables of a database are linked when you create a query. The term *relational database* indicates database management applications that are capable of linking tables by key fields.

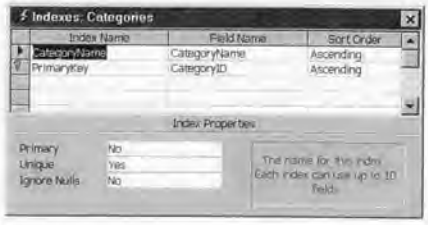


FIG. 2.11 Displaying the Index Name, Field Name, and Sort Order for each index of a table in the Indexes window.

Oracle Ex. 1009, p. 85
Oracle v. DeBellis
IPR2024-00221

▶▶ See "Using Access Indexes," p. 823



Each field in a table requires a unique name and must be assigned a field data type. Field names, data types, and an optional text description of the field are entered in the design grid. (Click the Indexes button or click the Close Window button of the Indexes window to close it so that the entire design grid is visible. Do the same to close the Properties window.) Text is the most common type of data in tables; therefore, Text is the default data type.







Many other field data types are available in Access; you already have been introduced to the OLE Object field data type. Click one of the Data Type cells, and then click the down arrow to open the Data Type drop-down list box to display the Data Type choices offered by Access. Click the arrow again to close the list. Field data types in Access include various numeric formats, date and time, and other types that you learn about in later chapters of this book. Now click the Datasheet View button to display the Datasheet view of the Categories table in Run mode.

Printing the Contents of a Table

Access enables you to print the contents of your table without creating a fully formatted report. You may want to print raw table data to proofread the new records you have created or the old ones you have edited. The Print Preview window is much like the one offered by Microsoft Word and Excel. The Print Preview window enables you to see how tables, forms, and reports will appear if printed.



To see how the Categories table will appear if printed, click the toolbar's Preview button (the page symbol with the magnifying glass). The Print Preview window appears, as shown in Figure 2.12. The title bar of the Print Preview window reflects what you're preparing to print. In this example, the title bar shows `Categories:Table`. You also can choose File, Print Preview to display the Print Preview window.



To see a magnified view of how the printed version of your table will appear, click the surface of the simulated sheet of paper in the Print Preview window. When the mouse pointer is on the paper, the pointer turns into the shape of a magnifying glass, as shown in Figure 2.12. Place the magnifying glass over the image; otherwise, when the window is magnified, the view is of a blank page and you wonder where the table went. Figure 2.13 shows the magnified (zoomed) view of the Categories table. Click again to restore the original, unreadable version of the report. The Zoom button has an effect similar to clicking the mouse on the surface of the preview image.

Oracle Ex. 1009, p. 86
Oracle v. DeBellis
IPR2024-00221



FIG. 2.12 Displaying the eight records of the Categories table in the Print Preview window.

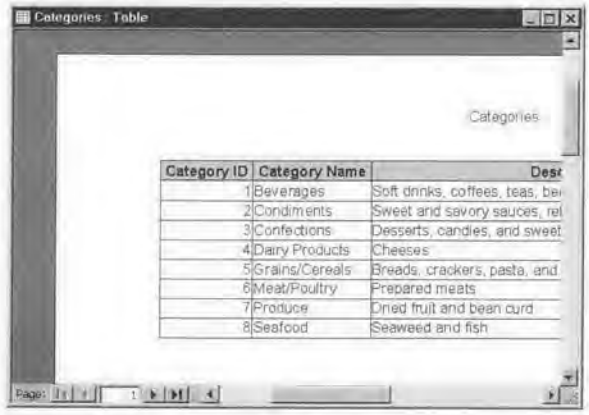


FIG. 2.13 Clicking the print preview image to show a magnified view of the Categories table.

When you are ready to print, click the Print button (with the symbol of a printer) on the toolbar—Access immediately begins printing the table, using the current printer settings for your computer.

To print in Landscape mode with your laser printer or to otherwise change your printer's settings, you must choose File, Print to display the Print dialog, shown in Figure 2.14. To change the printer settings, click the Properties button and make the appropriate changes in the *PrinterName* properties sheet for your printer, as shown in Figure 2.15 (the exact appearance of your printer's Properties sheet depends on your specific printer). Click the OK button to close the printer properties sheet, then click OK in the Print dialog to begin printing the table.

If you want to change the printing margins, choose File, Page Setup to display the Page Setup dialog shown in Figure 2.16. Click the Margins tab to display the page margin settings, and change the print margins by entering different values in the Margins text boxes. Click OK to close the Page Setup dialog and return to the Print Preview windows. Click the Print button on the toolbar to print the table with the new margin settings.

Oracle Ex. 1009, p. 87
Oracle v. DeBellis
IPR2024-00221

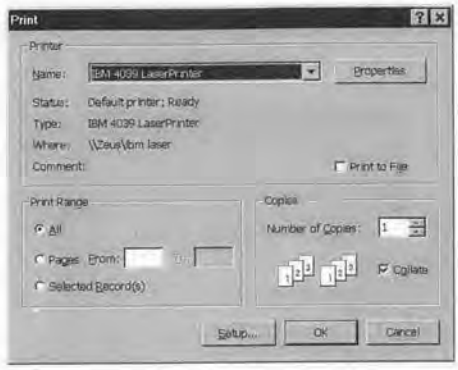


FIG. 2.14 The Print dialog for reports.



FIG. 2.15 Setting printer-specific options in the printer properties sheet.

Tip

You can also change the page orientation (portrait or landscape) on the Page sheet of the Page Setup dialog. If you only want to change printing margins or the page orientation, using the Page Setup dialog is easiest.

Now that you have run the gamut of the basic operations available for Access tables, double-click the Document Control menu box to close the Print Preview window and return to the Database window.

Now, you may open one or more of the other tables that comprise the Northwind Traders database to learn a bit more about their contents. The Employees table includes scanned images of photographs of the fictional staff of Northwind Traders. The Suppliers table lists an eclectic group of food-processing firms from many points on the globe; the

Oracle Ex. 1009, p. 88
Oracle v. DeBellis
IPR2024-00221

suppliers table also contains a Hyperlink field to connect you to a simulated Web page for some suppliers.

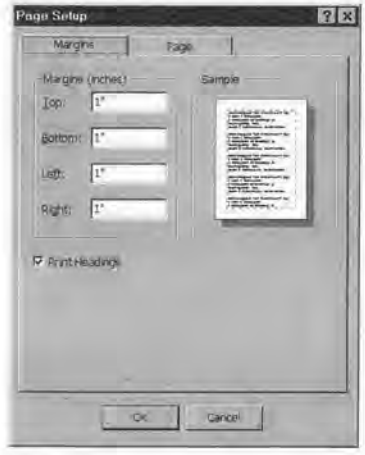


FIG. 2.16 Changing printing margins and paper orientation in the Page Setup dialog.

You can see most of the international characters, such as umlauts and tildes, in Windows' ANSI character set in the supplier name and address fields. Both Access 97 and Windows 95 are available in a wide range of languages in addition to British, Canadian, and U.S. English. Access 97 replaces 8-bit ANSI with 16-bit Unicode characters to accommodate Asian and other languages that use more than the about 200 characters accommodated by ANSI. Use of Unicode in Access 97 is transparent to the user, but requires modification of specific string operations performed with Access VBA.



To see a brief description of each table listed in the Database window, click the Details button on the toolbar. Access displays the text entered in the Description text box of the table's Table Properties dialog to the right of the table's name, along with date and time the table was last modified, and the date and time of the table's creation. You can display the contents of the Database window in any of the viewing formats available in any Windows 95 folder window: large icons, small icons, list, and details.

Oracle Ex. 1009, p. 89
Oracle v. DeBellis
IPR2024-00221

Chapter 3

Navigating Within Access

This chapter describes how Microsoft has structured and organized Access to expedite the design and use of the database objects that it offers. A substantial portion of this chapter consists of tables that list the functions of window Control-menu boxes, toolbar buttons, and an array of function-key combinations. Many function and key assignments derive from other Microsoft applications, such as Excel (F2 for editing) and Word for Windows (Shift+F4 to find the next occurrence of a match).

This chapter is a reference to which you can return when you conclude that a keystroke combination might be a better choice for an action than a mouse click, but you cannot remember the required combination. (Refer to Chapter 5, "Entering, Editing, and Validating Data in Tables," for details of the key combinations that you use to edit data in tables and queries.) This chapter also explains the structure and content of Access's Help system.

Understanding Access's Functions and Modes

Access, unlike word processing and spreadsheet applications, is a truly multifunctional program. Although word processing applications, for example, have many sophisticated capabilities, their basic purpose is to support text entry, page layout, and formatted printing. All word-processing application's primary functions and supporting features are directed to these ends. You perform all word processing operations using views that represent a sheet of paper—usually 8 1/2×11 inches. Most spreadsheet applications use the row-column metaphor for all their functions—even for writing highly sophisticated programs in the applications' macro languages. The sections that follow describe Access's basic functions and operating modes.

Defining Access Functions

To qualify as a full-fledged relational database management system, an application must perform the following four basic but distinct functions, each with its own presentation (or view) to the user:

Oracle Ex. 1009, p. 90
Oracle v. DeBellis
IPR2024-00221



- *Data organization* involves creating and manipulating tables that contain data in conventional tabular (row-column or spreadsheet) format, called *Datasheet View* by Access.



- *Table linking and data extraction* links multiple tables by data relationships to create temporary tables, stored in your computer's memory or temporary disk files, that contain the data that you choose. Access uses *queries* to link tables and to choose the data to be stored in a temporary table called a `Recordset` object. (For backward compatibility, Access 97 supports Access 1.x's `Dynaset` and `Snapshot` objects, which are both `Recordset` objects.) A `Recordset` object consists of the data that results from running the query; `Recordset` objects are called *virtual tables* because they are stored in your computer's memory rather than in database files. The capability to link tables by relations distinguishes relational database systems from simple list-processing applications, called *flat-file* managers. Data extraction limits the presentation of `Recordsets` to specific groups of data that meet criteria that you establish. *Expressions* are used to calculate values from data (for example, you can calculate an extended amount by multiplying unit price and quantity), and to display the calculated values as if they were a field in one of the tables.



- *Data entry and editing* require design and implementation of data viewing, entry, and editing forms as an alternative to tabular presentation. A form enables you, rather than the application, to control how the data is presented. For most users, forms are much easier to use for data entry than are `Recordsets` in tabular format, especially when many fields are involved. The capability to print forms, such as sales orders and invoices, is definitely a benefit to the user.



- *Data presentation* requires the creation of reports that can summarize the information in `Recordsets` that you can view and print (this is the last step in the process). The capability to provide meaningful reports is the ultimate purpose of any database management application. Also, the management of an enterprise usually lends more credence to reports that are attractively formatted and contain charts or graphs. Charts and graphs summarize the data for those officials who take the "broad brush" approach.



The four basic functions of Access that are implemented as views are organized into the application structure shown in Figure 3.1. If you are creating a new database, you use the basic functions of Access in the top-down sequence shown in Figure 3.1. You choose a function by clicking a button in the Datasheet window (except for security and printing operations, which are menu choices). In most views, you can display the Print Preview window that leads to printing operations by clicking the Print Preview button of the toolbar.

Oracle Ex. 1009, p. 91
Oracle v. DeBellis
IPR2024-00221

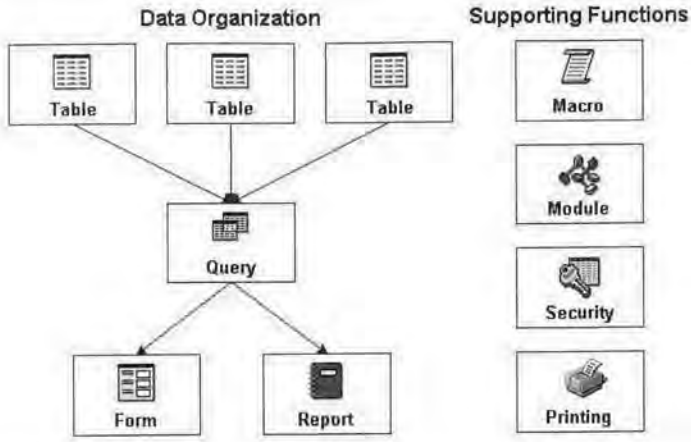


FIG. 3.1 The basic and supporting functions of Access.

Four supporting functions apply to all the basic functions of Access:

- *Macros* are sequences of actions that automate repetitive database operations. You create a macro in Access by choosing from a list of available actions, in the order in which you want Access to perform them. You can use a macro, for example, to open a report, print the report, and then close the report. Later, this section defines *open* and *close* as they are used in Access terminology. In prior versions of Access, macros were the primary means of automating database operations. In Access 97, macros are supported primarily for purposes of compatibility with databases created in earlier versions of Access. For Access 97 databases, you use the Access version of Visual Basic for Applications to automate database actions.





- *Modules* are functions and procedures written in Access's dialect of the Visual Basic for Applications (VBA) programming language. (Access's dialect of VBA was formerly known as Access Basic, and is still frequently referred to by that name.) You use Access VBA functions to make calculations that are more complex than those that can be expressed easily by a series of conventional mathematical symbols, or to make calculations that require decisions to be made. Access VBA procedures are written to perform operations that exceed the capabilities of standard macro actions—one of the reasons why macro support is gradually being dropped from Access. You run Access VBA procedures by attaching the procedure to particular events, such as clicking a command button with the mouse, that occur when a form or report is the active object. In Access 97, you can also execute Access VBA procedures directly from their module.



Oracle Ex. 1009, p. 92
Oracle v. DeBellis
IPR2024-00221

- *Security* consists of functions available as menu choices only. With security functions in a multiuser environment, you can let other people use your database. You can grant access to user groups and individuals, and you can restrict their ability to view or modify all or a portion of the tables in the database.
- *Printing* enables you to print virtually anything you can view in Access's Run mode. From the toolbar, you can print your Access VBA code, but not the macros that you write. (You can use the Documenter add-in to print the content of your macros.)

The terms *open* and *close* have the same basic usage in Access as in other Windows applications, but usually involve more than one basic function:

- Opening a database makes its content available to the application through the Database window described in Chapter 2, "Up and Running with Access 97." You can open only one database at a time during ordinary use of Access. Writing Access VBA code enables you to operate with tables from more than one database. You can achieve the equivalent of multiple open Access databases by *linking* (Access 97's term for *attaching*) tables from other databases.
- Opening a table displays a Datasheet view of its contents.
- Opening a query opens the tables involved but does not display them. Access then runs the query on these tables to create a tabular `Recordset`. Changes made to data in the `Recordset` cause corresponding changes to be made to the data in the tables associated with the query, if the `Recordset` is updatable. (Access 1.x's `Table` and `Dynaset` objects usually are updatable, but `Snapshot` objects are never updatable. The same rules apply to `Recordset` objects of the `Table`, `Dynaset`, and `Snapshot` type.)
- Opening a form or report automatically opens the table or query with which it is associated. Both forms and reports usually are associated with queries, but a query also can employ a single table.
- Closing a query closes the associated tables.
- Closing a form or report closes the associated query and its tables.

Defining Access Operating Modes

Access has three basic operating modes:

- *Startup mode* enables you to compress, convert, encrypt, decrypt, and repair a database by choosing commands from the Tools, Database Utilities and Tools, Security menu before opening a database. These commands, some of which are discussed at the end of this chapter, are available only when you *don't* have a database open.



- *Design mode* enables you to create and modify the structure of tables and queries, develop forms to display and edit your data, and format reports for printing. Access calls Design mode *Design view*.



- *Run mode* displays your table, form, and report designs in individual document windows (Run is the default mode). You execute macros by choosing one and then

Oracle Ex. 1009, p. 93
Oracle v. DeBellis
IPR2024-00221

selecting run mode. Run mode does not apply to Access VBA modules, because functions are executed when encountered as elements of queries, forms, and reports. Procedures in modules are run by macro commands, or directly from events of forms and reports. Run mode is called *Datasheet view* for tables and queries, *Form view* for forms, and *Print Preview* for reports.

You can select design or run mode by choosing command buttons in the Datasheet window, buttons on the toolbar, or commands from the View menu.

You can change the default conditions under which Access displays and prints your tables, queries, forms, and reports by choosing Tools, Options. The section "Setting Default Options" near the end of this chapter, describes options that apply to Access as a whole and those that apply only to tables.

Understanding Access's Table Display

You're probably familiar with the basic terms for many of the components that comprise the basic window in which all conventional Windows 95 applications run (these controls, although similar in function, have a somewhat different appearance and location than those used in Windows 3.x). The presentation of Access windows differs with each of the basic functions that Access performs. Because Part I of this book deals almost exclusively with tables, the examples that follow use Table view. Figure 3.2 shows Access for Windows's basic display for operations with tables. Table 3.1 describes the window's individual components.

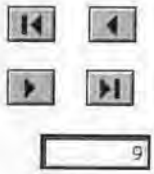
Table 3.1 Components of the Access Display for Tables

Term	Description
Active window	The window to which all mouse and keyboard actions are directed. When an application or document is active, its title bar appears in color (dark blue, unless you have changed your Windows color scheme). If both the application title bar and a document title bar are active, the document title bar receives the mouse and keyboard actions.
Application Control-menu box	The icon for the Application Control menu that controls the presentation of the Application window. You display the Application Control menu by clicking the box or pressing Alt+space bar.
Application title bar	A bar at the top of the application's window that displays its name. You can move the entire application, if it isn't maximized, by clicking the application title bar and dragging it to a new position.
Application window	The window within which Windows displays Access. Each application that you launch runs within its own application window. (Figure 3.2 shows the maximized Access 97 application window.)
Caret	A vertical flashing line that indicates the insertion point for keyboard entry in areas of a window that accept text. (The caret isn't visible in Figure 3.2.)
Current Record button	A button that indicates a single selected record in the table. When you are editing the current record, the button icon becomes a pencil rather than a triangular arrow. The Current Record button also is called the *record pointer*.

(continues)

Oracle Ex. 1009, p. 94
Oracle v. DeBellis
IPR2024-00221

Table 3.1 Continued

Term	Description
 Current Record selection	Buttons that position the record pointer to the first, next, preceding, and last record number in the table, and show the number of the currently selected record. If the table has a key field, the record number reflects the sequence of records in the primary key's sorting order; if there is no primary key in the table, the record number corresponds to the order in which records were physically added to the table.
Database window	The window that controls the operating mode of Access and selects the active document window's current function. From the database components displayed in the Database window, you select the component (such as a particular table) to display in the document window.
Document Control-menu box	The icon for the Document Control menu that controls a document window's presentation. To access the Document Control menu, click the box or press Alt+- (hyphen).
Document title bar	At the top of each document's window, a bar that displays the document's name. You can move the document, if it isn't maximized, by clicking the application title bar and dragging the document to a new position.
Document window	The window that displays an Access database component. Tables, queries, forms, reports, macros, and modules are referred to as *documents* in Windows terminology. You can have multiple Access documents of any type open simultaneously. These windows are called *multiple document interface (MDI) child windows*, because the Access application window is their *parent*.
Field scroll bar	The scroll bar that enables you to view fields of tables that are outside the bounds of the document window. Record scroll bars provide access to records located outside the document window.
Function tabs	Six tabs with which you can choose whether the active document window displays tables, queries, forms, reports, macros, or modules.
Inactive window	A window in the background, usually with a grayed title bar. Clicking the surface of an inactive window makes it the active window and brings it to the front. If an inactive window is not visible because other windows obscure it, you can make the window active by choosing the window's name from the Window menu.
Maximize button	Clicking the application's Maximize button causes Access to occupy your entire display. Clicking the document's Maximize button causes the document to take over the entire display. When a window is maximized, as shown in Figure 3.2, this button's icon changes to the Restore button. Figure 3.3 shows a maximized table document.
Menubar	A horizontal bar containing the main menu choices. The specific main menu choices, and the choices in the drop-down menus corresponding to the main menu selections change, depending on Access's status. Menubars and toolbars collectively are called *command bars*.
Minimize button	A button that enables you to collapse the application or document window to an icon at the bottom of your display.
 Mode buttons	Three buttons that determine the operating mode of Access. *Open* places Access in Run mode. *New* or *Design* puts Access in Design mode, where you can create or edit tables.

Oracle Ex. 1009, p. 95
Oracle v. DeBellis
IPR2024-00221

Term	Description
New Record button	A button with an asterisk that indicates the location of the next record to be added to a table. Entering data in the new record appends the record to the table and creates another new record.
Restore button	A double set of boxes that, when clicked, returns the window from full display to its normal size, with movable borders. When displayed, the Restore button takes the place of the Maximize button in the window's upper-right corner.
Status bar	A bar, located at the bottom of the application window, that displays prompts and indicators, such as the status of the Num Lock key.
Toolbar	A bar containing command buttons that duplicate the more commonly used menu choices. The number and type of toolbar buttons change depending on which basic function of Access you are using.

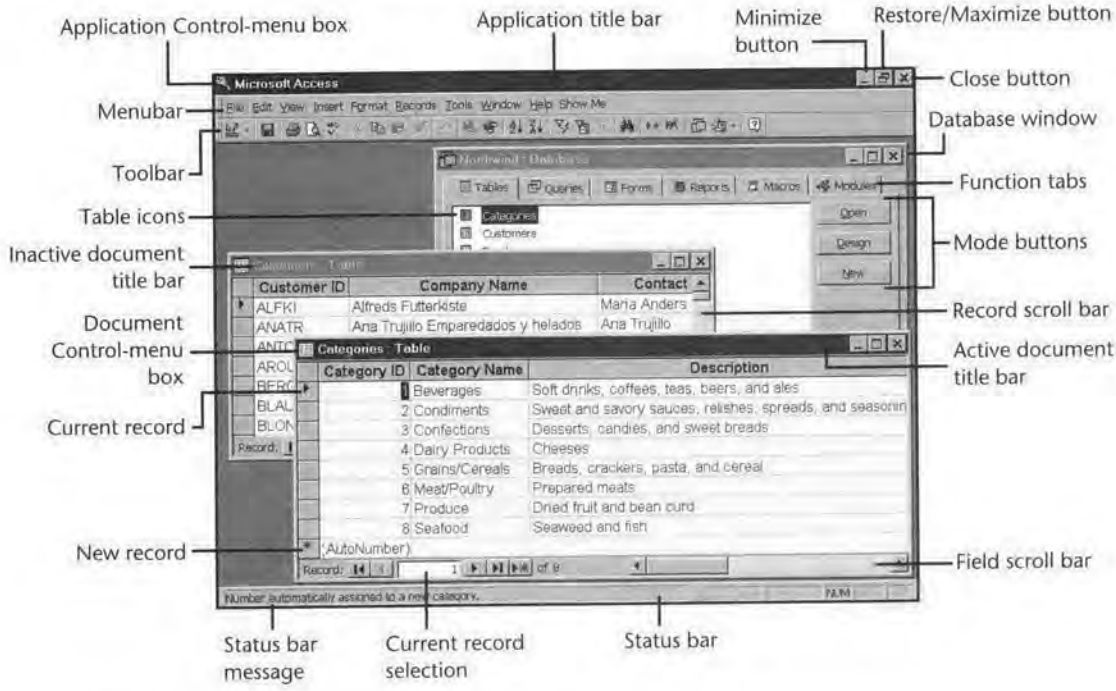


FIG. 3.2 Access 97's basic display for tables.

Maximized Document Windows

Access uses a windowing technique that you should know about; otherwise, you might accidentally minimize or close Access when you intended to minimize or close a maximized document. After you click a document window's Maximize button, the document window takes the place of the application window and occupies the entire display, except for the menu bar and toolbar (see Figure 3.3). Most other Windows applications that display multiple documents, such as Word for Windows and Excel, have a similar capability to expand a document to occupy the entire window.

Oracle Ex. 1009, p. 96
Oracle v. DeBellis
IPR2024-00221

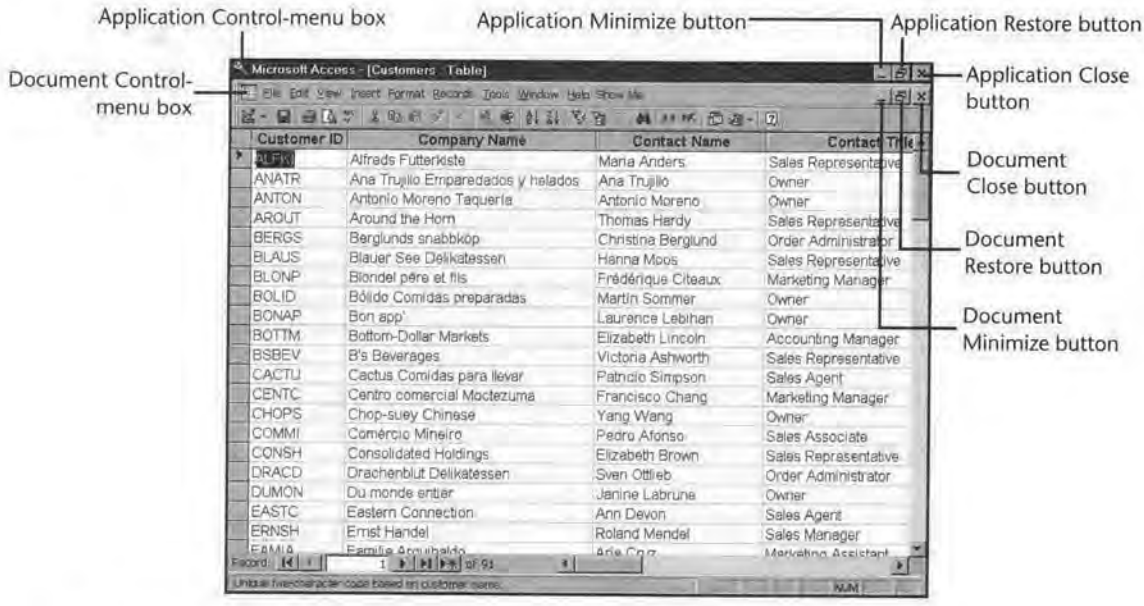


FIG. 3.3 An Access table in a maximized document window.

The Document Control-menu box and the Document Minimize, Restore, and Close buttons move to the menu bar's extreme left and right, respectively. The title of the document is added to the application title in the title bar at the top of the display. To return the document window to its original size, established when the application window was first active, click the Document Restore button; alternatively, click the Document Control-menu box and then choose Restore from the Document Control menu. You can close the document window by clicking the Document Close button, or by double-clicking the Document Control-menu box. If you accidentally click the Application Close button (or double-click the Application Control-menu box just above the Document Control-menu box), however, you close Access 97. You receive no warning that you are about to exit Access unless you have changed the design of an object.

Document Windows Minimized to Icons

Working with several overlapping windows limits each to a size that enables you to select another by clicking its surface. This overlapping might overly restrict your view of the data that the windows contain. You can minimize Access document windows and the Database window to icons that remain within the application window, as shown at the bottom of Figure 3.4. If you minimize a document window to an icon instead of closing it, you can quickly return the window to its original size by double-clicking the icon. If you single-click the icon, you can choose how the window reappears by using the Document Control menu, as shown for the Database window in Figure 3.4. You can also restore, maximize, or close a minimized icon by clicking the corresponding button within the icon.

Oracle Ex. 1009, p. 97
Oracle v. DeBellis
IPR2024-00221



FIG. 3.4 Tables, a query, a form, and the Database window minimized to icons within the application window.

If you choose to display your document window in maximized form by choosing Maximize from the Document Control menu that appears when you click the icon, the document hides the icons at the bottom of the application window. In this case, open the Window menu and choose the document that you want. If you size your document windows (like the window in Figure 3.4) by dragging their borders, you can avoid the substantial mouse movement and two-step menu-selection process to select the active document.

The Toolbars in Table View

The buttons that appear in Access's toolbar, and the number of toolbars displayed, changes according to the function that Access is currently performing. When you are working with tables in Run mode, Access 97 displays the Table Datasheet and the Datasheet Formatting toolbars, shown in Figures 3.5 and 3.6, respectively. The next two sections of this chapter describe the toolbars that appear in Table Run mode (Datasheet view).

> **Note**
>
> Even if you have some experience with previous versions of Access, you should note Tables 3.2 and 3.3. Access 97 adds many new shortcut buttons to the toolbars and has also moved or restructured several menu commands. For example, the View, Toolbars command now leads to a submenu, instead of to a dialog. The View, Toolbars submenu lists only those toolbars most pertinent to Access's current operating mode, making it easier for you to select an appropriate assortment of toolbars; a Customize menu choice leads to the Toolbars dialog, which is familiar to users of Access 95.

Oracle Ex. 1009, p. 98
Oracle v. DeBellis
IPR2024-00221

The Table Datasheet Toolbar. The Table Datasheet toolbar appears whenever you open an Access table in Datasheet view. Figure 3.5 shows the Table Datasheet toolbar, and Table 3.2 describes the buttons that appear on the toolbar.

> **Note**
>
> Toolbar buttons provide shortcuts to traditional selection methods, such as choosing menu commands or choosing command or option buttons in a particular sequence. The Alternate Method columns of Tables 3.2 and 3.3 list how you can achieve the same effect as clicking a toolbar button by using the menus or the command buttons in the Database window.

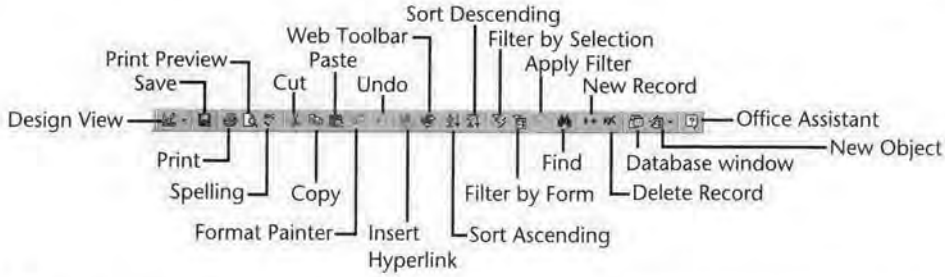


FIG. 3.5 Access's Table Datasheet toolbar displayed when a table is open in Datasheet view.

Table 3.2 Appearance and Functions of Buttons and Other Elements of the Table Datasheet Toolbar

Icon	Button	Alternate Method	Function
	Design View	View, Design View	Design mode. In Design mode, you specify the properties of the table.
	Datasheet View	View, Datasheet View	Returns the table display to Run mode (Datasheet view) from Design mode. (You cannot see this button in Figure 3.5; it appears only when the table is in Design view.)
	Save	File, Save	Saves the database.
	Print	File, Print	Prints the table.
	Print Preview	File, Print Preview	Displays the contents of a table in report format and enables you to print the table's contents.
	Spelling	Tools, Spelling	Starts the spelling checker.
	Cut	Edit, Cut	Cuts the selected information and puts it in the Windows Clipboard.

Oracle Ex. 1009, p. 99
Oracle v. DeBellis
IPR2024-00221

Icon	Button	Alternate Method	Function
	Copy	Edit, Copy	Copies selected information to the Windows Clipboard.
	Paste	Edit, Paste	Pastes information from the Windows Clipboard into Access at the current location of the caret.
	Format Painter		Copies a control's format to another control. Used only in Design view; it is enabled only when you select a control.
	Undo	Edit, Undo	Returns you to the status immediately preceding the last action that you took. The Undo button is inactive if there is no action to undo. Access provides a single-level Undo feature that repeals only one action.
	Insert Hyperlink	Insert, Hyperlink	Opens the Insert Hyperlink dialog, which lets you add a URL or UNC address to a hyperlink field in a table. Chapter 16, "Working with Hyperlinks and HTML," describes hyperlink fields in more detail.
	Web Toolbar		Displays or hides the Web toolbar. The Web toolbar lets you activate your Web browser to view HTML pages stored locally on your computer or network, or on the Internet. The Web Toolbar button is a toggle button—when in its "up" position, the Web toolbar is hidden; when in its "down" position, the Web toolbar is displayed.
	Sort Ascending	Records, Sort, Ascending	Sorts the records in ascending order, based on the current field.
	Sort Descending	Records, Sort, Descending	Sorts the records in descending order, based on the current field.
	Filter by Selection	Records, Filter, Filter by Selection	Filters records based on the selected text in a field.
	Filter by Form	Records, Filter, Filter by Form	Enables you to enter criteria in a table datasheet to establish how records are filtered.

(continues)

Oracle Ex. 1009, p. 100
Oracle v. DeBellis
IPR2024-00221

Table 3.2 Continued

Icon	Button	Alternate Method	Function
	Apply/ Remove Filter	Records, Apply Filter/Sort	Applies or removes a filter.
	Find	Edit, Find	Displays the Find dialog that locates records with specific characters in a single field or all fields.
	New Record	Edit, Go To, New	Selects the tentative append record.
	Delete Record	Edit, Delete Record	Deletes the active record.
	Database Window	Window, 1	Displays the Database window.
	New Object		Displays a drop-down list from which you choose the type of new object that you want to create: tables, forms, reports, queries, macros, or modules.
	Office Assistant	F1	Activates the Microsoft Office Assistant described in the "Using the Office Assistant" section later in this chapter.

Note

In Access 97, and in other applications in Microsoft Office 97, the toolbar buttons appear as flat icons on the toolbar. The toolbar buttons only have a raised button-like appearance when the mouse pointer is over the button. The exception to this rule is "toggle" buttons—that is, buttons that represent the on/off status of a feature, like the Web Toolbar button (refer to Table 3.2 and Figure 3.5). When a toggle button is "up," it appears as a flat icon on the toolbar until you move the mouse pointer over it; the button's "up" appearance indicates that the feature controlled by that button is off. Toggle buttons in the "down" positions are shaded so that they look as if they are sunken below the surface of the toolbar; the "down" appearance indicates that the feature controlled by that button is on.

The Datasheet Formatting Toolbar. In addition to the Table Datasheet toolbar, you can display the Datasheet Formatting toolbar whenever you view a table in Datasheet View. Choose View, Toolbars, Formatting (Datasheet) to add the toolbar. The buttons in the Datasheet Formatting toolbar provide shortcuts to various text-formatting commands. In Datasheet view, the text-formatting commands apply to the entire table; you cannot format individual cells in Datasheet view. Figure 3.6 shows the Datasheet Formatting toolbar, and Table 3.3 summarizes the action of each button on the toolbar.

Oracle Ex. 1009, p. 101
Oracle v. DeBellis
IPR2024-00221

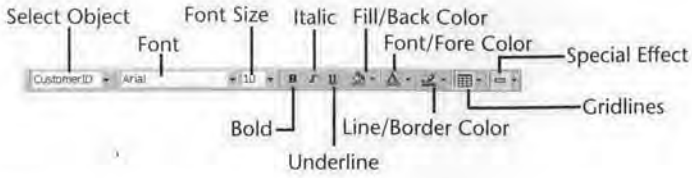


FIG. 3.6 The Datasheet Formatting toolbar in Datasheet view.

Table 3.3 Appearance and Functions of Buttons and Other Elements of the Datasheet Formatting Toolbar

Icon	Button	Alternate Method	Function
CustomerID	Select Object		Displays a drop-down list from which you can jump quickly to any field in the table.
Arial	Font	Format, Font	Enables you to select the font (typeface) for text in a table.
10	Font Size	Format, Font	Enables you to select the size of the text in a table.
B	Bold	Format, Font	Turns bold text formatting on and off for the text in a table.
I	Italic	Format, Font	Turns italic text formatting on and off for text in a table.
U	Underline	Format, Font	Turns underlining on and off for text in a table.
	Fill/Back Color	Format, Cells	Displays a palette of colors from which to choose the background color for the table's data cells.
	Font/Fore Color	Format, Font	Displays a palette of colors from which to choose the color of the text in the table.
	Line/Border Color	Format, Cells	Displays a color palette from which to choose the color of the gridlines that indicate rows and columns in the table.
	Gridlines	Format, Cells	Displays four buttons that enable you to choose which gridlines are shown: horizontal and vertical, vertical only, horizontal only, or none.
	Special Effect	Format, Cells	Displays three buttons that enable you to choose the cell display style: flat, raised, or sunken.

Manipulating Access's Toolbars

Access 97 uses the resizable, customizable, floating toolbars that have become standard in Microsoft applications such as Excel and Microsoft Word. In Microsoft Office 97, menubars and toolbars have been combined into a single object, called a *command bar*, and now share many features in common. The primary characteristic that distinguishes a

Oracle Ex. 1009, p. 102
Oracle v. DeBellis
IPR2024-00221

menubar from a toolbar in Access 97 (and other Office 97 applications) is that every application has at least one menu bar, and the menubar may not be hidden. In all other respects, menubars and toolbars are now the same.

Access 97's View, Toolbars menu choice enables you to select which toolbars are currently visible. The View, Toolbars menu lists those toolbars pertinent to Access's current operating mode. Figure 3.7 shows the View, Toolbars menu for Table Datasheet view. The currently displayed toolbars are indicated on the menu by a check mark at the left of the menu choice for that specific toolbar. To display or hide a toolbar, click its name in the submenu.



FIG. 3.7 Displaying or hiding toolbars with the View, Toolbars menu choice.



Note

In Figure 3.7, and other figures throughout this book, you may notice a menu choice that isn't discussed in the text—Show Me. The Show Me choice only appears on the menu bar when the Northwind.mdb database is open, and only if you also chose to install the Show Me help files when you installed Access 97. The Show Me menu isn't part of Access 97; instead, it is displayed by Northwind.mdb. If you're interested in viewing help topics relating to the Access VBA code that is part of the Northwind.mdb sample database, click the Show Me menu choice.

The Customize choice on the View, Toolbars menu opens the Customize dialog (see Figure 3.8) that lets you display as many toolbars at once as will fit in your display—or hide toolbars that Access would otherwise display automatically. To display a toolbar, click the Toolbars tab to display the Toolbars page (if necessary), then click the box to the left of the toolbar name in the Toolbars dialog so that the check box is selected. To hide a toolbar, click the box again to clear the check box selection. For help using the Customize dialog, click the Office Assistant button at the lower-left area of the Customize dialog to activate the Office Assistant, if it isn't already active (see "Using the Office Assistant" later in this chapter).

Note

When an Access toolbar is in its docked position, it has a fixed width, anchored at its left edge. If you reduce the width of Access's application window by dragging either vertical border inward, the buttons at the docked toolbar's extreme right begin to disappear beyond the application window's right edge. Operating Access in a maximized window with docked toolbars is usually best, because you can then easily access all the toolbar buttons when you use the default in-line horizontal toolbar.

Oracle Ex. 1009, p. 103
Oracle v. DeBellis
IPR2024-00221



FIG. 3.8 Selecting the toolbars to be displayed in the Customize dialog.

You also can use the Customize dialog to change the viewing options for toolbars. Click the Options tab of the Customize dialog to display the Options page shown in Figure 3.9; the Options page enables you to select various toolbar viewing options. If you're using SVGA 1024×768 screen resolution, you may want to select the Large Icons option, which causes the toolbar button icons to approximately double in size; this makes the icons easier to discern, and easier to click with the mouse. The Show ScreenTips on toolbars option governs whether Access displays the ScreenTips (formerly known as ToolTips) hints on the mouse cursor for toolbar buttons. The Show Shortcut Keys in ScreenTips option determines whether Access displays the keyboard shortcut (if there is one) as part of the ScreenTip text. Finally, the Menu Animations drop-down list lets you choose how Access draws menus on-screen. You may choose None (for no special effects when drawing menus), Random (Access randomly chooses an animation effect each time you open a menu), Unfold (the menu unfolds like a fan), or Slide (the menu opens like a rollershade) as the technique for displaying Access's menus.



FIG. 3.9 Selecting viewing options for toolbars and menus in the Options page of the Customize dialog.

In addition to displaying multiple toolbars, you can reshape or reposition the toolbars to suit your own taste. Click a blank area of the toolbar and hold down the left mouse

Oracle Ex. 1009, p. 104
Oracle v. DeBellis
IPR2024-00221

button to drag the toolbar to a new location. The toolbar turns into a popup floating toolbar, similar to the toolbox that you use to add control objects to forms and reports. Popup toolbars always appear on top of any other windows open in your application.

Figure 3.10 shows three floating command bars: the Table Datasheet toolbar, the Formatting Datasheet toolbar, and the Menu Bar. (These are the same toolbars—discussed in the preceding section of this chapter—that Access displays in Datasheet View mode. The Menu Bar—which is Access's main menubar—was moved from its position at the top of the Access application window to demonstrate its new features as a command bar.) Command bars in their fixed position are called *docked* command bars, while command bars in their popup window are referred to as *floating* command bars. After you change a command bar to a floating command bar (or dock it), Access always displays the command bar in that location until you again reposition the command bar.

Tip

You can also dock command bars (both menubars and toolbars) at the bottom of the Access application window, or at either the left or right edge of the application window.

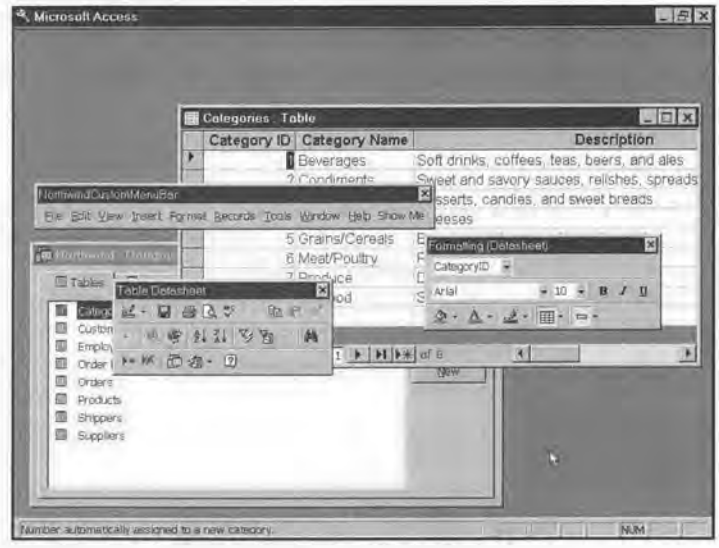


FIG. 3.10 Access's Table Datasheet toolbar, Datasheet Formatting toolbar, and main menubar dragged from their default positions at the top of the application window.

Right Mouse Button Shortcut Menus

Another feature that Access 97 shares with other Microsoft applications, plus Windows 95 and Windows NT 4.0, is the shortcut menu that appears when you right-click the surface of an Access database object. Shortcut menus (also called *popup* or *context* menus) present choices that vary depending on the type of object that you click. Figure 3.11 shows the shortcut menu for a field of a table selected by clicking the field name header.

Oracle Ex. 1009, p. 105
Oracle v. DeBellis
IPR2024-00221

Tip

Shortcut menus are quite useful, and provide shortcuts to many common tasks. If you're not sure what you can do with an object on-screen, try right-clicking it to see what shortcut menu commands are available.



FIG. 3.11 The shortcut menu for selected column of a table.

Using the Function Keys

Access assigns specific purposes to all 12 function keys of the 101-key extended keyboard. Some keys, such as Shift+F4 (which you press to find the next occurrence of a match with the Find dialog), derive from other Microsoft applications—in this case, Word for Windows. You use function keys with the Shift, Alt, and Ctrl keys to enable users to perform as many as 96 functions with the 12 function keys.

Global Function Keys

Windows, rather than Access, uses global function-key assignments, except for F11 and Alt+F1, to perform functions that are identical in all Windows applications. Table 3.4 lists the global function-key assignments.

Table 3.4 Global Function-Key Assignments

Key	Function
F1	Displays context-sensitive help related to the present basic function and status of Access. If a context-sensitive help topic isn't available, F1 starts the Office Assistant (described in the next section of this chapter).
Shift+F1	Adds a question mark to the mouse pointer. Place the mouse pointer with the question mark over an object on-screen for which you want help and then click.
Ctrl+F4	Closes the active window.
Alt+F4	Exits Access or closes a dialog if one is open.
Ctrl+F6	Selects each open window in sequence as the active window.
F11 or Alt+F1	Selects the Database window as the active window.
F12 or Alt+F2	Opens the File Save As dialog.
Shift+F12 or Alt+Shift+F2	Saves your open database; the equivalent of the File, Save menu choice.

Oracle Ex. 1009, p. 106
Oracle v. DeBellis
IPR2024-00221

Function-Key Assignments for Fields, Grids, and Text Boxes

Access assigns function-key combinations that aren't reserved for global operations to actions that are specific to the basic function that you are performing at the moment. Table 3.5 lists the function-key combinations that apply to fields, grids, and text boxes. (To present complete information, this table repeats some information that appears in the previous tables.)



▶▶ See "Using Keyboard Operations for Entering and Editing Data," p. 162

Table 3.5 Function Keys for Fields, Grids, and Text Boxes

Key	Function
F2	Toggles between displaying the caret for editing and selecting the entire field.
Shift+F2	Opens the Zoom box for entering expressions and other text.
F4	Opens a drop-down combo list or list box.
Shift+F4	Finds the next occurrence of a match of the text entered in the Find or Replace dialog, if the dialog is closed.
F5	Moves the caret to the record-number box. Enter the number of the record that you want to display and press Enter.
F6	In Table Design view, cycles between upper and lower parts of the window. In Form Design view, cycles through the header, body (detail section), and footer.
Shift+F6	In Form Design view, cycles through the footer, body (detail section), and header, moving backward.
F7	Starts the spelling checker.
F8	Turns on Extend mode. Press F8 again to extend the selection to a word, the entire field, the whole record, and then all records.
Shift+F8	Reverses the F8 selection process.
Ctrl+F	Opens the Find dialog.
Ctrl+H	Opens the Replace dialog.
Ctrl++ (plus sign)	Adds a new record to the database.
Ctrl+- (minus sign)	Deletes the current record.
Shift+Enter	Saves changes to the active record in the database.
Esc	Undoes changes in the current record or field. By pressing Esc twice, you can undo changes in both the current field and record. Also cancels extend mode.

Function Keys in the Module Window

You use the Module window when writing Access VBA code, which is the subject of this book's Part VII, "Programming, Distributing, and Converting Access." Table 3.6 lists the purposes of the Module window's function keys. The Module window shares many of the characteristics of Windows's Notepad applet, including the F3 key, which you use for searching.

Oracle Ex. 1009, p. 107
Oracle v. DeBellis
IPR2024-00221

▶▶ See "Exploring the Module Window," p. 1007



Table 3.6 Function Keys in the Module Window

Key	Function
F1	Displays context-sensitive help about the currently selected Access VBA keyword, dialog, or menu command.
F2	Opens the Object Browser dialog, which enables you to view Access objects and move among them and other applications that support VBA, including the applications' properties, controls, procedures, and methods.
Shift+F2	Goes to the procedure selected in the Module window.
F3	Finds the next occurrence of text specified in the Find or Replace dialog.
Shift+F3	Finds the preceding occurrence of text specified in the Find or Replace dialog.
F5	Continues executing code after a break condition.
Shift+F5	Resets the VBA interpreter, utilizing all variables.
F6	Cycles between upper and lower panes (if you have split the window).
F8	Traces execution one step at a time (single step or Step Into mode).
Shift+F8	Traces execution by procedure (procedure step or Step Over mode).
F9	Toggles a breakpoint at the selected line.
Shift+F9	Creates an Instant Watch for the selected variable or expression.
Ctrl+Shift+F9	Clears all breakpoints.
Ctrl+F	Opens the Find dialog.
Ctrl+G	Displays the Debug window.
Ctrl+H	Opens the Replace dialog.
Ctrl+L	Displays the Calls dialog.
Ctrl+M	Indents selected lines.
Ctrl+Shift+M	Removes indentation.
Ctrl+Y	Cuts the current line and copies it to the Clipboard.
Ctrl+Break	Halts macro execution.
Ctrl+↑	Displays the previous procedure.
Ctrl+↓	Displays the next procedure.

Setting Default Options

You can set about 100 options that establish the default settings for the system as a whole, and also those for the six functions defined by the Database window's tabs. You aren't likely to change default options until you are more familiar with Access. However, because this book is a reference as well as a tutorial guide, and options are a basic element of Access's overall structure, this section explains how to change these settings.

You set defaults by choosing Tools, Options. The Options dialog appears as shown in Figure 3.12. View, General, Keyboard, and Edit/Find Category options apply to the

Oracle Ex. 1009, p. 108
Oracle v. DeBellis
IPR2024-00221

system as a whole. Datasheet, Tables/Queries, and Forms/Reports options all apply to table views in Datasheet view, forms, and queries. The remainder of the option categories are specific to other basic functions.

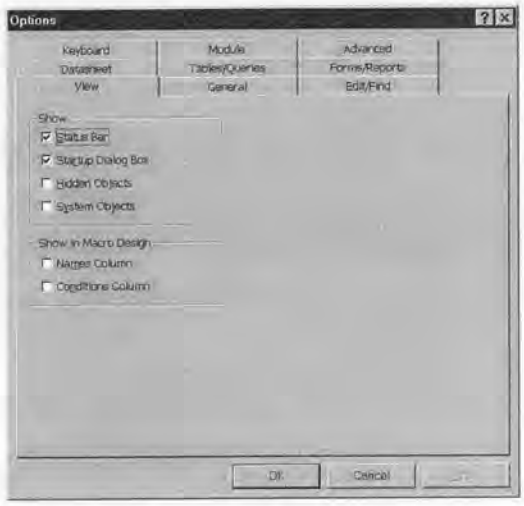


FIG. 3.12 The Options dialog displaying the View Options sheet.

You select an Options category by clicking the tab near the top of the Options dialog. When you change a category, the dialog displays the Options sheet for that category. Most of the settings are option buttons and check boxes, although many other items require multiple-choice entries that you select from drop-down lists. In some cases, you must enter a specific value from the keyboard. After you complete your changes, click OK to close the dialog. If you decide not to implement your changes, click Cancel to exit the Options dialog without making any changes. The next few sections and their tables summarize options that affect Access as a whole, and those options that affect viewing and printing data in Datasheet view.

System Defaults

Access uses a special Access database, System.mdw, to store all default properties for displaying and printing the contents of tables, queries, forms, reports, and modules for each user of Access. The .mdw extension for workgroup system files was new with Access 95, replacing the .MDA extension shared by libraries, wizards, and System.mda in Access 1.x and 2.0. Access 97, like Access 95, stores user default properties in system tables, along with other tables that determine the behavior of Access, in System.mdw. System.mdw also stores user names and passwords when you secure your Access database or use Access in a multiuser (workgroup) environment.



▶▶ See "Sharing Your Access Database Files with Other Users," p. 868

Oracle Ex. 1009, p. 109
Oracle v. DeBellis
IPR2024-00221

Note

System.mdw is vital to the proper operation of Access. You should keep a backup copy of System.mdw on a floppy disk. If you make changes to any options or implement Access's security features, you should create an updated backup after completing your changes or additions.

View Options. The View options, as described in Table 3.7, enable you to customize the appearance of Access's application window.

Table 3.7 View Options for the Access System

Option	Group	Function
Status Bar	Show	If checked, displays the status bar at the bottom of the Access application window.
Startup Dialog Box	Show	If checked, displays the startup dialog whenever you start Access. This dialog prompts you to open an existing database or to create a new one.
Hidden Objects	Show	When checked, displays hidden objects in the Database window.
System Objects	Show	If checked, displays system objects in the Database window.
Names Column	Macro Design	If checked, displays the Names column in new macros.
Conditions	Macro Design	If checked, displays the Conditions Column column in new macros.

General Options. General options, described in Table 3.8, apply to Access as a whole. The settings that you make in the General options apply to any new objects that you create (tables, forms, and reports), but don't retroactively affect existing objects. For example, changing the print margins in General options affects only any reports that you create subsequently, but not any existing reports. To change the print margins of existing objects, you must change each object's individual printing settings in Design view.

Margins usually are expressed in inches. If you are using an international version of Access, margin settings are in centimeters. You also can specify margin settings in twips, the default measurement of Windows. A *twip* is 1/20 of a printer's point. A point is 1/72 inch, so a twip is 1/1,440 inch.

The one-inch default margins are arbitrary; you might want to reset them to your preference before creating any forms or reports of your own. If you are using a laser printer, refer to its manual to determine the maximum printable area. The printable area determines the minimum margins that you can use.

Apart from the printing margins, the General option you're most likely to want to change is the default database directory. When you create your own databases, you should store them in a folder dedicated to databases, to simplify backup operations. A dedicated database folder also is a good place to keep a backup copy of System.mdw.

Oracle Ex. 1009, p. 110
Oracle v. DeBellis
IPR2024-00221

Table 3.8 General Options for the Access System

Option	Group	Function
Left Margin	Print Margins	Establishes the default left margin.
Top Margin	Print Margins	Establishes the default top margin.
Right Margin	Print Margins	Establishes the default right margin.
Bottom Margin	Print Margins	Establishes the default bottom margin.
Default Database Folder		Changes the default folder for the Open Database dialog. The default folder is the Access working folder, indicated by a period.
New Database Sort Order		Sets the alphabetical sort order used for new databases. You can change the sort order for an existing database by selecting a different sort-order setting and then compacting the database by choosing Tools, Database Utilities, Compact Database.
Already Followed	Hyperlink	Selects the color for hyperlink text which has been followed in this session.
Not Yet Followed	Hyperlink	Selects color for hyperlink text which hasn't yet been clicked this session.
Underline Hyperlinks	Hyperlink	If selected, Access displays hyperlink text with an underline.
Show Hyperlink Addresses in Status Bar	Hyperlink	If selected, Access displays the full hyperlink address in the status bar at the bottom of the application window.

Edit/Find Options. The Edit/Find options all affect the behavior of Access's Find feature for both tables in Form or Datasheet view and when working with Access VBA code in a module. Table 3.9 summarizes the Edit/Find options and their effects. The options in the Default Find/Replace Behavior group all determine the default searching method for the Edit, Find and Edit, Replace commands. Options in the Confirm group all determine which actions Access asks the user to confirm. The final option group in the Find/Replace options sheet is the Filter by Form Defaults for the current database. These options don't actually affect Access itself, but affect the defaults for the particular database that is open.

Table 3.9 Edit/Find Options for the Access System

Option	Group	Function
Fast Search	Default Find/ Replace Behavior	Sets the default search method to search in the current field, and to match the whole field.
General Search	Default Find/ Replace Behavior	Sets the default search method to search in all fields, and matches any part of a field.

Oracle Ex. 1009, p. 111
Oracle v. DeBellis
IPR2024-00221

Option	Group	Function
Start of Field Search	Default Find/ Replace Behavior	Causes the default search method to search the current field, matching only the beginning of the field.
Record Changes	Confirm	Causes Access to confirm any changes that you make to a record.
Document Deletions	Confirm	Causes Access to confirm document (table, form, or report) deletions.
Action Queries	Confirm	Causes Access to confirm an action query (such as adding or deleting records) before carrying out the query.
Local Indexed Fields	Filter by Form Defaults	Includes local indexed fields in the list of values that you can use criteria.
Local Non-Indexed Fields	Filter by Form Defaults	Includes nonindexed fields in the list of values that you can use when entering filter criteria.
ODBC Fields	Filter by Form Defaults	Includes fields from remote ODBC tables in the lists of values that you can use when entering filter criteria.
Don't Display Lists Where More Than This Number Of Records Read		Prohibits the display of filter values whenever the number of items in the list exceeds the specified number.

Keyboard Options. Keyboard options, listed in Table 3.10, are especially important if you are accustomed to a particular type of arrow-key behavior. You probably will want to change keyboard options more than any of the other categories. For example, you can make the arrow keys behave as if you are editing xBase fields, instead of using the keys' default behavior, which duplicates that of Excel. The options in the Move After Enter group affect what Access does when you press Enter after editing or entering data. The options in the Arrow Key Behavior group affect how the left- and right-arrow keys work, while the options in the Behavior Entering Field group determine what happens when the caret enters a field.

▶▶ See "Using Data Entry and Editing Keys," p. 163



Table 3.10 Keyboard Options for the Access System

Option	Group	Function
Don't Move	Move After Enter	When selected, the caret remains in the current field when you press Enter.
Next Field	Move After Enter	When selected, the caret moves to the next or previous field when you press Enter. This is the Move After Enter group's default option.

(continues)

Oracle Ex. 1009, p. 112
Oracle v. DeBellis
IPR2024-00221

Table 3.10 Continued

Option	Group	Function
Next Record	Move After Enter	When selected, the caret moves down the column to the next or previous record when you press Enter.
Next Field	Arrow Key Behavior	If selected, pressing the right- or left-arrow keys moves the caret to the next or previous field. This is the Arrow Key Behavior group's default option.
Next Character	Arrow Key Behavior	If selected, pressing the right- or left-arrow keys moves the caret to the next or previous character in the same field.
Select Entire Field	Behavior Entering Field	When this option is selected, the entire field's contents are selected when you use the arrow keys to move the caret into the field. This is the Behavior Entering Field group's default option.
Go to Start of Field	Behavior Entering Field	If selected, causes the caret to move to the beginning of the field when you use the the arrow keys to move the caret into the field.
Go to End of Field	Behavior Entering Field	If selected, causes the caret to move to the end of the field when you use the arrow keys to move the caret into the field.
Cursor Stops at First/Last Field		If selected, keeps the caret from moving to another record when the left- or right-arrow keys are pressed, and the caret is in the first or last field of the record.

Advanced Options. Access has several advanced system options that affect multiuser operations, OLE updates, DDE linking and updating, and tables attached by the Open Database Connectivity (ODBC) feature of the Jet 3.5 database engine. Table 3.11 describes the Advanced options that you can set. Options in the Default Record Locking group affect how Access locks records in a multiuser environment, and the Default Open Mode option group controls whether Access shares opened databases. The DDE Operation options group controls how Access handles DDE requests from other applications; other options control OLE updating and query updating. The Current Database Only group contains options that relate to any Access VBA code that may be stored in your database. Usually you won't need to change the Advanced options much, unless you're working in a multiuser environment with several users sharing the same database, or the database contains Access VBA code that uses command arguments and conditional compiler directives.

Oracle Ex. 1009, p. 113
Oracle v. DeBellis
IPR2024-00221

Table 3.11 Advanced Options for the Access System

Option	Group	Function
No Locks	Default Record Locking	When selected, leaves all records unlocked in the open database tables, so that other networked users can update the records. This is the group's default option.
All Records	Default Record Locking	When selected, locks all records in the open database tables. No other networked users can update the records.
Edited Records	Default Record Locking	When selected, locks only edited records. When the changes to the record are saved, Access unlocks the record.
Ignore DDE Request	DDE Operations	If enabled, ignores all Dynamic Data Exchange (DDE) requests from other Windows applications.
Enable DDE Refresh	DDE Operations	If enabled, lets Access dynamically update linked DDE data. This is the group's default option.
OLE/DDE Timeout (sec)		Specifies how long Access waits for a response from a DDE or OLE server. If the specified interval passes without a response, Access reports an error. Access's default value for this option is 30 seconds.
Shared	Default Open Mode	When selected, enables other networked users to use the open database simultaneously. This is the group's default option.
Exclusive	Default Open Mode	When selected, opens the database in an exclusive mode so that other network users cannot open the database.
Number of Update Retries		Specifies how many times that Access tries to update a query, OLE object, or DDE link before giving up. The default number of tries is 2.
ODBC Refresh Interval (sec)		Specifies how many seconds that Access waits before refreshing records that you view through an ODBC connection. The default interval is 1,500 seconds.
Refresh Interval (sec)		Specifies how many seconds that Access waits before refreshing remote data. The default Refresh Interval is 60 seconds.
Update Retry Interval (msec)		Specifies how many milliseconds (thousandths of a second) that Access waits in between attempts at updating an OLE, DDE, ODBC, or other link. The default is 25 ms.

(continues)

Oracle Ex. 1009, p. 114
Oracle v. DeBellis
IPR2024-00221

Table 3.11 Continued

Option	Group	Function
Command-Line Arguments	Current Database Only	Use this text box to enter any arguments that you want returnedby the `Command()` function.
Conditional Compilation	Current Database Only	Use this text box to specify values for conditional compilation directives.
Project Name	Current Database Only	Enter a name for your Access VBA project in this text box; this is the name by which code in your Access database is known to other applications.

Defaults for Datasheet View

You use Datasheet View options to customize the display of all query datasheets and new table and form datasheets (see Table 3.12). As with printing options, to change the display format of existing table and form datasheets, you must edit the appropriate properties of the table or form in Design view. The Datasheet View options that you set don't apply to forms and reports created with Access wizards. Each wizard has its own set of default values. The options in the Default Colors group set the background and foreground colors for text displayed in Datasheet view, while the Default Font group's options determine the typeface and text size. The Default Gridlines Showing options determine which gridlines (if any) Access displays in Datasheet view. Finally, the Default Cell Effect options enable you to select a default style for datasheet cells.

Table 3.12 Options for Datasheet Views

Option	Group	Function
Font	Default Colors	Displays a drop-down list from which you can select the color of the text in new tables, queries, and forms. Access's default Font color selection is black.
Background	Default Colors	Enables you to select the background color of cells in Datasheet view. Access's default Background color is white.
Gridlines	Default Colors	Enables you to select the color of the gridlines displayed in Datasheet view. Access's default gridline color is silver.
Font	Default Font	Displays a drop-down list from which you can select the typeface that Access uses to display text in Datasheet view. Access's default font is Arial.
Weight	Default Font	Displays a drop-down list from which you can select the weight of the text characters displayed in Datasheet view. You can select Normal (the default), Thin, Extra Light, Medium, Semi-bold, Bold, Extra Bold, or Heavy.

Oracle Ex. 1009, p. 115
Oracle v. DeBellis
IPR2024-00221

Option	Group	Function
Size	Default Font	Enables you to select the default font size, in points. Access's default font size is 10 points.
Italic	Default Font	If selected, displays all datasheet text in italics.
Underline	Default Font	If selected, displays all datasheet text with a single underline.
Horizontal	Default Gridlines Showing	If selected, displays horizontal gridlines (that is, gridlines between rows) in Datasheet view. By default, this option and the Vertical Gridlines option are turned on.
Vertical	Default Gridlines Showing	If selected, displays vertical gridlines (that is, gridlines between columns) in Datasheet view. You can display both vertical and horizontal gridlines by combining the Vertical Gridlines option with the Horizontal Gridlines option.
Default Column Width		Specifies the default column width in inches. Access's default value for this text box setting is one inch.
Flat	Default Cell Effect	When selected, displays a data cell as a "flat" cell—that is, the cell has no special shading. This is the group's default option.
Raised	Default Cell Effect	When selected, adds shadow effects to each data cell so that the cell appears to be raised above the surface of the screen, like a command button.
Sunken	Default Cell Effect	When selected, adds shadow effects to each data cell so that the cell appears to be sunken below the surface of the screen.
Show Animations		If selected, displays animated cursors and other animation effects. If you have a slow computer, you'll probably want to turn this option off to improve Access's operating speed slightly.

Note

The remaining option categories—Tables/Queries, Forms/Reports, and Module—are discussed in the chapter(s) of this book that cover the subject of the particular option category. Also, options related to multiuser, DDE, and ODBC features are described in more detail in the chapter(s) devoted to those special topics.

Using Access Help

The Access help system is extensive and easy to use. All the new help functions incorporated in Windows 95's 32-bit WinHelp engine are used by Access's Help system.

Oracle Ex. 1009, p. 116
Oracle v. DeBellis
IPR2024-00221

Context-Sensitive Help

Context-sensitive help tries to anticipate your need for information by displaying help windows related to the operating mode, function, and operation in which you are involved or attempting to perform. You can get context-sensitive help in any of the following ways:

- By using the help mouse pointer that appears after you click the What's This button (located near the top-right corner of the dialog window) in the active dialog. Move the help mouse pointer over the dialog option for which you want help, and click again to display a popup help window with information about that dialog option.
- By using the help mouse pointer that appears after you press Shift+F1 or choose Help, What's This?. Move the help mouse pointer over the item for which you want help, and click again to display a popup help window.
- By pressing the F1 key. Access displays a help window with information about the active area (dialog control, window, menu command, and so on), or displays the Office Assistant, described later in this section.
- By clicking the Help button in a dialog. Use this method, or press the F1 key, for dialogs that don't have a What's This button.

To get context-sensitive help in an open dialog, click the What's This button; the mouse pointer changes to a question mark. Move the help mouse pointer over the dialog control that you want help with, and click. You can also get context-sensitive help on the active dialog control by pressing F1 or clicking the control with the right mouse button. For example, you might want more information about the effects of the Find dialog's Match Case option. To find such information, click the What's This button, and then click the Find dialog's Match Case option. Figure 3.13 shows the resulting popup help window explaining the Match Case option.

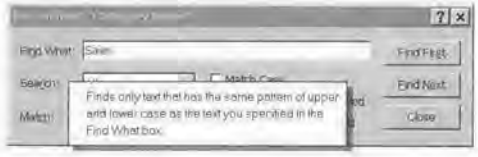


FIG. 3.13 The context-sensitive help for the Find dialog's Match Case option.

Another method for getting context-sensitive help is to press Shift+F1 (or choose Help, What's This?) and then place the mouse pointer with the question mark on the item with which you need help. When you click the mouse button, the topic related to the object appears. Figure 3.14 shows an example that explains the purpose of the Table Datasheet toolbar's Find button.

Oracle Ex. 1009, p. 117
Oracle v. DeBellis
IPR2024-00221

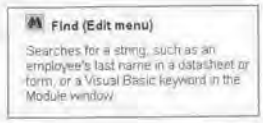


FIG. 3.14 The popup help window explaining the purpose of the toolbar's Find button.

The Help Menu

Access's Help menu provides an alternative to using context-sensitive help. Table 3.13 lists the options that the Help menu presents.

Table 3.13 Access's Help Menu Options

Option	Function
Microsoft Access Help	Activates the Office Assistant, described later in this section.
Contents and Index	Displays the Help Topics dialog, which enables you to select or search for help topics either through a table of contents, an index, or a Find utility.
What's This?	Changes the mouse pointer to the Help pointer for obtaining context-sensitive help. Move the Help mouse pointer over the object or menu choice for which you want help, and click to display a popup help window.
Microsoft on the Web	Displays a submenu list of Web sites related to Microsoft, Microsoft Access, frequently asked questions, free software, and other topics.
About Microsoft Access	Displays the copyright notice for Microsoft Access, and the name and organization that you entered during setup. The About dialog also contains two command buttons: one that displays sources of technical support for Access in North America and throughout the world, and another that displays information about your computer system, such as how much memory you have installed, whether you have a math coprocessor, and the amount of remaining disk space.

Using the Help Topics Dialog

You can get a more general form of help by choosing Help, Contents and Index. In this case, you always start from square one—the Help Topics dialog for the entire help system, shown in Figure 3.15. The Help Topics dialog contains three tabbed sheets: Contents, Index, and Find; each of these options is described in the following paragraphs.

Note

The Answer Wizard of Access 95 has been incorporated in the new Office Assistant feature of Access 97; the Office Assistant is described at the end of this section.

Oracle Ex. 1009, p. 118
Oracle v. DeBellis
IPR2024-00221

The Contents Tab. Figure 3.15 shows the Help Topics Contents tab (you might have to click the Contents tab to bring the table of contents to the front of the dialog). The Contents tab is like the table of contents in a book; it shows the structure of the topics in the Help system, based on the topic's title.

Each table of contents entry that has subheadings is indicated by a book icon to its left. To see subheadings for a topic, double-click the closed book icon. The Help system expands the topic list and changes the icon to an open book. (To hide the list of subheadings, double-click the open book icon; this hides the expanded subheading branch and changes the icon back to a closed book.)

Table of contents headings that lack subheadings have to their left an icon resembling a sheet of paper with a question mark on it. Figure 3.15 shows the "Finding and Sorting Data" heading expanded to show its subheadings; the "Sorting Data in Tables, Queries, and Forms" subheading, in turn, has been expanded, revealing a list of three help topics. To display a topic, double-click it. Figure 3.16 shows the displayed help topic for "Sort records in a table, query, form, or subform."



FIG. 3.15 The Contents tab of the Help Topics dialog, showing expanded headings and subheadings.

Understanding the Help Window. You can reposition and resize the help window by dragging its borders with your mouse. To reposition the help window, click and drag the Help title bar. If the help file on the topic that you selected has more information than can fit in the window, a vertical scroll bar appears at the right of the window. Drag the scroll box down to display additional text.

Oracle Ex. 1009, p. 119
Oracle v. DeBellis
IPR2024-00221



FIG. 3.16 You display a help topic by double-clicking a topic in the Help Topics dialog's Contents list.

Tip

You can copy text from any help window. Simply drag the mouse over the text that you want to copy to select it, and then press Ctrl+C or choose Edit, Copy. You can then paste the copied help text into any Windows application from the Clipboard.

Most of Access's help windows include hotspots that provide additional information about a topic. Hotspots with dotted underlines, such as "Form view" and "Datasheet view" in Figure 3.16, display popup windows that usually contain a definition of the term or contain more detailed information about that topic. Figure 3.17 shows the window that pops up when you click the "Datasheet view" hotspot. To close the popup window, click anywhere on the screen outside the popup window.

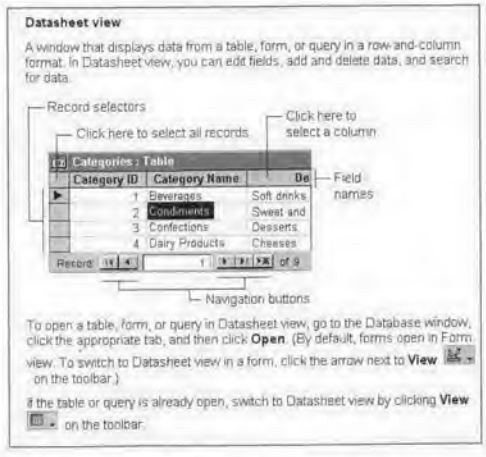


FIG. 3.17 The popup window displayed by the "Datasheet view" hotspot.

Oracle Ex. 1009, p. 120
Oracle v. DeBellis
IPR2024-00221

Note

When you click hotspot text, which usually is highlighted in green, you receive an explanation of the hotspot topic. Hotspots in green, underlined, and bold text are links to additional windows in the help file related to the hotspot's topic.

Some hotspots lead to additional help topics. These hotspots are usually shown in bold text with a solid underline (like all hotspots, they're typically displayed in green text). You click the hotspot that represents the subject about which you want to learn. This action causes a jump to the subject's first help window, which often provides several additional choices for more detailed help on a specific topic.

Tip

You can return to the Help Topics dialog at any time by clicking the Help Topics button in the Help window.

Most help windows, in addition to providing information about a particular topic, also have tutorial, step-by-step instructions for the task about which you are inquiring. In Figure 3.16, notice that the help topic has a section titled "What do you want to do?" (visible near the bottom of the window). This section lists a variety of tasks that you might be trying to accomplish if you're looking for help on sorting records (the list extends beyond the bottom edge of the dialog in Figure 3.16). Each item in the list has a button to its left. After you click this button, Access displays a help screen with step-by-step instructions for accomplishing the indicated task.

For example, Figure 3.18 shows the step-by-step tutorial help window that Access displays if you click the button next to Sort Records on Demand in Datasheet View or Form View at the bottom of the dialog in Figure 3.16.



FIG. 3.18 A step-by-step help window opened by clicking the first button in the "What do you want to do?" list from the Sort records help topic shown in Figure 3.16.

Oracle Ex. 1009, p. 121
Oracle v. DeBellis
IPR2024-00221

Tip

To return to the previously displayed topic, choose the Back button in the Help window. To move backward through more than one topic, choose the same button repeatedly. When there are no previous topics to return to, Access disables the Back button.

Notice the icon near the bottom of the step-by-step help window shown in Figure 3.18. This icon, which depicts the Sort Ascending toolbar button described earlier in this chapter, is also a hotspot. Clicking this hot spot displays a popup window describing the action of the Sort Ascending toolbar button. Many help topics throughout the Access Help system contain graphic hotspots like this one.

Tip

Whenever you place the mouse pointer over a help hotspot, the pointer turns into a pointing-hand shape.

The Index Tab. You can also look up help topics in an index much like that which is at the end of this book. Click the Index tab in the Help Topics dialog to bring the Index sheet to the front of the dialog. Figure 3.19 shows the Index sheet as it appears after you type the topic **help** in the text box.

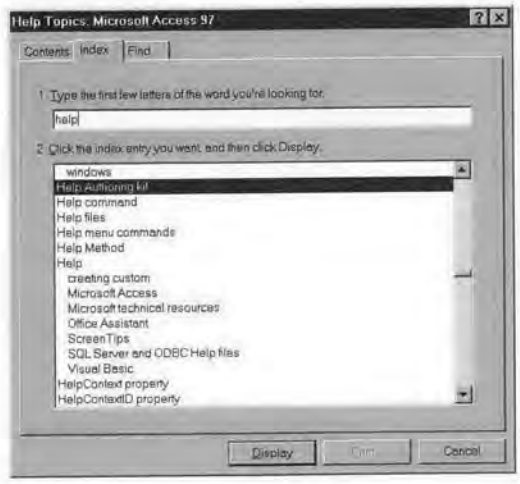


FIG. 3.19 The Index sheet of the Help Topics dialog.

Using the Help Topics Index is simple—just type in the text box at the top of the dialog the name of the topic on which you want help. As you type, the Help system adjusts the topic list in the bottom of the dialog to show the topic that most closely matches the text that you've typed so far. In Figure 3.19, the user has typed **help** in the text box at the top of the dialog, and the help topics list shows the first matching entry: "Help

Oracle Ex. 1009, p. 122
Oracle v. DeBellis
IPR2024-00221

Authoring kit." To display a topic from the list, double-click it (or click once to select it and then choose the Display button). If you want, you can also use the scroll bar to view the list of available help topics.

The Find Tab. The Help Topics Index is only an alphabetical listing of the help topics available. The Find tab provides you with a way to search quickly through the actual text in the available help topics so that you can quickly find topics even if you don't know the name of the topic.

Click the Find tab in the Help Topics dialog to display the Find sheet. Figure 3.20 shows the Find sheet, after searching for the words **getting help**.

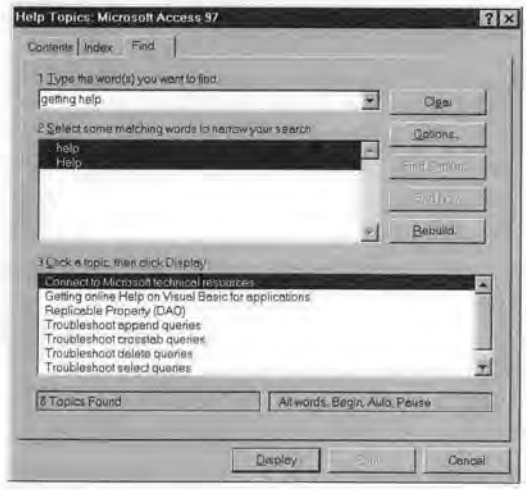


FIG. 3.20 The Find sheet of the Help Topics dialog.

> **Note**
>
> The first time that you use the Help Topics Find option, the Windows 95 WinHelp engine must build a word list to use for searching. The Help system needs to build this word list only once; thereafter, the system loads that list each time that you display the Find sheet.

The Help Topics Find sheet works essentially the same as the Index sheet. In the text box at the top of the dialog, you type words related to the topics that you want help on. Find uses its word list to locate all the topics that contain the words that you type. The list in the center of the dialog displays additional matching words, and a list at the bottom of the dialog displays matching help topics. Unfortunately, explaining all the options available in the Find sheet is beyond the scope of this book. For most searches, the default settings work just fine. To get more help about the specific controls in the Find sheet, click the What's This button.

Oracle Ex. 1009, p. 123
Oracle v. DeBellis
IPR2024-00221

The Help Window Options. The Help system provides you with several options that you can take advantage of to print, copy, or annotate the displayed help topic, or that enable you to change the help window's style or appearance. To display a menu of options, click the help window's Options button or right-click the dialog. Table 3.14 summarizes each menu choice and its effect.

Table 3.14 Help Window Options

Option	Function
Annotate	Enables you to add your own comments to the current help topic.
Copy	Copies selected text from the help window to the Windows Clipboard.
Print Topic	Prints the currently visible help topic on your printer.
Font	Enables you to select one of three predetermined font sizes for text displayed in the help window: Small, Normal, and Large.
Keep Help on Top	Enables you to choose whether the help window always displays on top of other windows, never displays on top of other windows, or displays with the default setting. (The Help system determines whether the window should be on top or not.)
Use System Colors	Tells the Help system to use the window border and background colors defined for the Windows 95 system as a whole. By default, the Help system uses its own color scheme for its windows.

Using the Office Assistant



An important new feature in Access (and all the Microsoft Office 97 applications) is the Office Assistant. The Office Assistant integrates several features found in previous versions of Access, Word, and Excel; in particular, the Office Assistant is the new embodiment of the Tip Wizard and the Answer Wizard. When the Office Assistant is on, it displays tips related to activities that you are currently performing, and provides a means for you to search for help on specific tasks.

The Office Assistant appears as an animated character in its own floating window, and displays its messages in a "speech" balloon. You may choose one of several available office assistant characters. Figure 3.21 shows Clippit, the default Office Assistant character (an animated paper clip) as it appears while asking if the user wants to save changes to a table's layout. In this case, the Office Assistant takes the place of Windows' standard message box.



FIG. 3.21 The Office Assistant's version of a conventional message box.

Oracle Ex. 1009, p. 124
Oracle v. DeBellis
IPR2024-00221

Note

All cautions, warnings, requests for confirmation, and other Access messages are displayed in a text balloon by the Office Assistant, whenever the Office Assistant is turned on. Figure 3.22 shows the same message displayed by the Office Assistant in Figure 3.21—and, generated under the exact same circumstances—as it appears when the Office Assistant is off.



FIG. 3.22 The standard message box for warnings, cautions, and requests for information.

By default, the Office Assistant is activated. You might find the Office Assistant distracting or prefer to use the Office Assistant only when you specifically want help. You can turn the Office Assistant off by either clicking the Close button on the Office Assistant's floating window, or by right-clicking the Office Assistant and then choosing Hide Office Assistant from the popup menu.

If you hide the Office Assistant, it remains hidden until you turn it on again, or until you use a wizard or some other feature of Access which automatically invokes the Office Assistant. Access "remembers" the status of the Office Assistant whenever you close Access; if the Office Assistant was hidden at the time you last exited Access, then the Office Assistant remains hidden the next time you start Access.



You can activate the Office Assistant by clicking the Office Assistant button on the toolbar, or by choosing Help, Microsoft Access Help. Access also starts the Office Assistant if you press F1 for context-sensitive help, but Access can't determine the precise context in which you're working.

To use the Office Assistant to search for help on any topic, click anywhere on the Office Assistant window; the Office Assistant displays the text balloon shown in Figure 3.23. Type a question about what you want to do in the text box, and then click Search. Figure 3.24 shows the Office Assistant's text balloon after searching for help topics that answer the question "How do I find text in a field?" The Office Assistant displays a list of topics that answer your question, and lead to step-by-step instructions on carrying out various tasks. To initiate a new search, type another question in the text box, and click Search again.

Tip

If you can't find help any other way, use the Search feature of the Office Assistant.

If the Office Assistant has a tip for you, a light bulb icon appears in the Office Assistant window. Click the Office Assistant window to view the tip. You may also view tips

Oracle Ex. 1009, p. 125
Oracle v. DeBellis
IPR2024-00221

related to recent activities you have performed, by clicking the Office Assistant and then clicking the Tips button (refer to Figure 3.23); the Office Assistant displays a speech balloon similar to the one shown in Figure 3.25. Click Back and Next to move to the previous or next available tip. When you're done with the Office Assistant, click Close to close the text balloon; the Office Assistant remains on-screen.



FIG. 3.23 The Office Assistant enables you to search for help on any task.



FIG. 3.24 The Office Assistant, after searching for help related to finding text in a field.

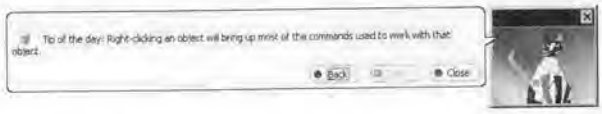


FIG. 3.25 The Office Assistant can also display tips related to activities you have performed recently.

As mentioned previously, you can select one of several Office Assistant characters. You can also modify the Office Assistant's behavior. To change the Office Assistant character, or to change the Office Assistant's behavior, click the Options button in the Office Assistant's text balloon, or right-click the Office Assistant window and then select Options from the popup menu. The Office Assistant dialog appears, as shown in Figure 3.26.

Oracle Ex. 1009, p. 126
Oracle v. DeBellis
IPR2024-00221



FIG. 3.26 The Office Assistant offers several animated characters.

To select a different Office Assistant character, click the Gallery tab to display the Gallery sheet. Click the Back and Next buttons to view the available Office Assistant characters; click OK to select the character you want. If the animation data files for the Office Assistant character you select weren't installed on your hard disk at the time Microsoft Office 97 was installed, you'll be prompted to insert your distribution disk so that Office Assistant can load the required animation data files onto your computer's hard disk.

To alter the Office Assistant's behavior, click the Options tab to display the Options sheet, shown in Figure 3.27. The options in the Assistant capabilities group affect how much help the Office Assistant provides when it is active. The Show Tips About options group contains settings that affect what kinds of tips the Office Assistant shows you, as does the Other Tip Options group. Table 3.15 lists the available Office Assistant option settings, and summarizes their effect.

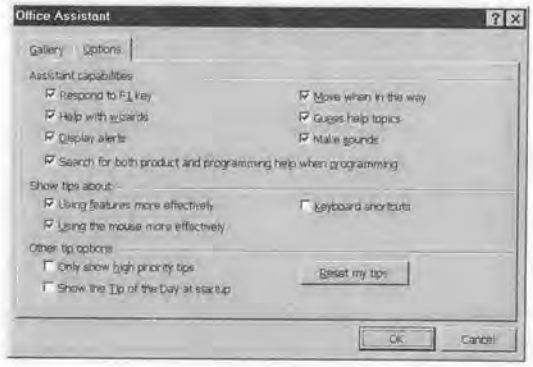


FIG. 3.27 The Office Assistant's behavior is controlled by the settings on the Options sheet.

Oracle Ex. 1009, p. 127
Oracle v. DeBellis
IPR2024-00221

Table 3.15 Office Assistant Options

Option	Group	Function
Respond to F1 Key	Assistant Capabilities	When selected, causes the Office Assistant to respond to the F1 key. If This check box is cleared, then the Help Topics dialog appears when you press F1.
Help with Wizards	Assistant Capabilities	When selected, causes the Access wizards to provide help through the Office Assistant.
Display Alerts	Assistant Capabilities	When selected, causes all Access alert messages to be displayed by the Office Assistant; if cleared, Access alert messages appear in a standard dialog, whether or not the Office Assistant is active.
Search for Both Product and Programming Help When Programming	Assistant Capabilities	If selected, causes Access to search for help topics in both the VBA programming reference and in the Access application help when you are programming in VBA.
Move When in the Way	Assistant Capabilities	If selected, causes the Office Assistant window to move out of the way on-screen if it would otherwise obscure a dialog, table view, or other on-screen object. Also causes the Office Assistant window to shrink to a smaller size if the Office Assistant isn't used in five minutes.
Guess Help Topics	Assistant Capabilities	If selected, the Office Assistant offers help based on your current activities.
Make Sounds	Assistant Capabilities	If selected, the Office Assistant plays various sound effects as you use it. You must have a sound card and speakers installed in your computer to hear sounds; this option may be selected even if your computer isn't capable of playing sounds.
Using Features More Effectively	Show Tips About	When selected, this option causes the Office Assistant to display tips to help you learn about Access features you don't know, and to more effectively use features you do know.
Using the Mouse More Effectively	Show Tips About	When selected, the Office Assistant also shows tips related to using the mouse more efficiently in Access.
Keyboard Shortcuts	Show Tips About	When selected, the Office Assistant shows tips related to using keyboard shortcuts.

(continues)

Oracle Ex. 1009, p. 128
Oracle v. DeBellis
IPR2024-00221

Table 3.15 Continued

Option	Group	Function
Only Show High Priority Tips	Other Tip Options	If selected, limits the tips displayed by the Office Assistant to only those with a high priority, such as time-saving alternatives.
Show the Tip of the Day At Startup	Other tip options	When selected, the Office Assistant starts each time you start an Office program, and displays a tip of the day.
Reset my Tips	Other tip options	Clicking this command button resets the Office Assistant's internal record of tips you have already seen; click this button if you want the Office Assistant to display tips you have seen previously.

Using the Database Utilities



Access 97 offers four utility functions, which you access from the Tools, Database Utilities menu. Three of these functions (Compact a Database, Repair a Database, and Make an MDE File) you may perform with or without a database open; the fourth utility function (Convert a database) can only be performed when there is no open database. If you have a large database, these operations take a considerable amount of time. If you select one of the utility operations described in the following sections when you do *not* already have a database open, the operation involves two dialogs. In the first dialog, you select the database in which Access is to perform the operation; in the second dialog, you enter the name of the file that the operation is to create. Default file names for new files are Db#.mdb, where # is a sequential number, beginning with 1, assigned by Access. For the three utility operations you can perform with an open database, only the Make MDE File option displays a dialog. When compacting or repairing an open database, Access assumes that you want to operate on the open database, and that you want the resulting compacted or repaired database to replace the currently open database.

Compacting Databases

After you make numerous additions and changes to objects within a database file, especially additions and deletions of data in tables, the database file can become disorganized. When you delete a record, you don't automatically regain the space in the file that the deleted data occupied. You must compact the database to optimize both its file size and the organization of data within the tables that the file contains. When you pack an Access file, you regain space only in 32K increments.

To compact a database, perform the following steps:



1. Open the database you want to compact.
2. Choose Tools, Database Utilities, Compact Database. Access immediately begins compacting the open database. When Access finishes compacting the database, it returns you to the Database Window. Your compacted database is stored with the same name it had before you compacted it.

Oracle Ex. 1009, p. 129
Oracle v. DeBellis
IPR2024-00221

If you want, you may also compact a database and save the compacted database in a different database file by following these steps:

1. Close any open database.
2. Choose Tools, Database Utilities, Compact Database. The Database to Compact From dialog appears, as shown in Figure 3.28.

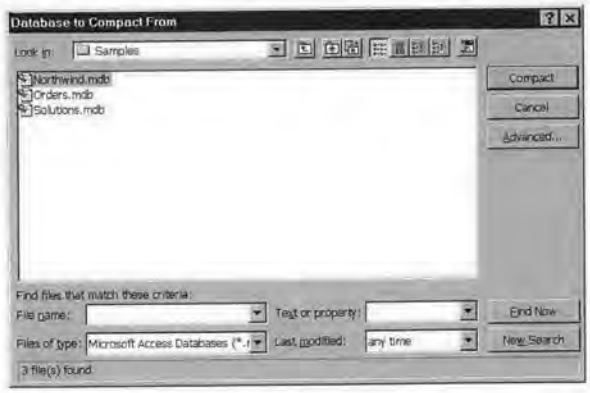


FIG. 3.28 The Database to Compact From dialog.

3. Double-click the name of the database file that you want to compact, or click the name and then click Compact. The Compact Database Into dialog appears, as shown in Figure 3.29.



FIG. 3.29 The Compact Database Into dialog.

4. In the File Name text box, enter the name of the new file that is to result from the compaction process. If you choose to replace the existing file with the compacted version, you see a message box requesting that you confirm your choice. Click Save.

Oracle Ex. 1009, p. 130
Oracle v. DeBellis
IPR2024-00221

Caution

If the compaction process fails, your database might be damaged. Databases damaged in the compaction process are unlikely to be repairable (see the "Repairing Databases" section that follows). Thus, you should not compact the database into a new database with the same name. Do so only after backing up your database with a different name, in a different folder, or on a floppy disk.

Access then creates a compacted version of the file. The progress of the compaction is shown in a blue bar in the status bar. If you decide to use the same file name, the new file replaces the preceding file after compaction.

Periodically compacting files usually is the duty of the database administrator in a multiuser environment, typically in relation to backup operations. You should back up your existing file on disk or tape before creating a compacted version. When you're developing an Access 97 database, you should compact the database frequently. Uncompacted Access 97 databases grow in size much more rapidly during modification than with prior versions of Access.

Converting Databases to Access 97 Format

To convert Access .mdb database files, .MDA library files created with Access 1.x or Access 2.0, and .mde library files created with Access 95 to the new database format of Access 97, first close any open database, then choose Tools, Database Utilities, Convert Database. The process of converting database files from earlier versions of Access database formats to that of Access 97 is almost identical to the second file-compaction process described in the preceding section. The only difference that you'll notice is that the names of the dialogs are Database to Convert From and Database to Convert Into. (Chapter 33, "Migrating Access 2.0 and Access 95 Applications to Access 97," covers this conversion process in detail.)

Caution

Although you can convert databases created with earlier versions of Access into Access 97 format, Access 97 does not enable you to convert the databases back from Access 97 format to a prior Access database format. If you attempt to open an Access 97 database or library file with the Convert Database menu choice, you receive the following message: `The database you tried to convert was either created in or was already converted to the current version of Microsoft Access.` Thus, if you want to support users of Access database applications who do not have Access 97, you must maintain two separate sets of database files. Therefore, you must have the retail versions of any earlier Access versions and Access 97 available to maintain your application.

Repairing Databases

A database can become corrupted as the result of the following problems:

- Hardware problems in writing to your database file, either locally or on a network server.

Oracle Ex. 1009, p. 131
Oracle v. DeBellis
IPR2024-00221

- Accidental restarting of the computer while Access databases are open.
- A power failure that occurs after you have made modifications to an Access object, but have not saved the object.

Access includes a database repair facility that you can use to recover a usable database file from the majority of files corrupted by one of the preceding causes. The process is the same as that for creating compacted and encrypted versions of the file.

Occasionally, a file might become corrupted without Access detecting the problem. This lack of detection occurs most frequently with corrupted indexes. If Access or your application behaves strangely when you open an existing database and display its contents, try repairing the database. Choose Tools, Database Utilities, Repair Database, and then follow the same steps as described for compacting the database.

Creating MDE Files

An .mde file is a special version of an Access .mdb file. In an .mde file, all VBA code is stored only in compiled format, and the program source code for that database is unavailable. (The file extension .mde stands for Microkernel Development Environment.) Also, users are no longer able to modify forms, reports, queries, or tables stored in that database, although those object may be exported to other databases. Typically, .mde databases are used to create libraries of add-in wizards, custom database applications intended for commercial or in-house distribution, and to provide templates for forms, reports, queries, and other objects for use in other databases.

You can convert any .mdb database to an .mde file by choosing Tools, Database Utilities, Make MDE File. If you have a database open at the time you select this command, Access assumes you want to save the current open database as an .mde file, and immediately displays a Save MDE As dialog; this dialog functions essentially the same way as any Save As dialog. If you choose Tools, Database Utilities, Make MDE File when there is no database open, Access first displays a Database to Save as MDE dialog; use this dialog to select the .mdb database file you want to convert to an .mde file. Part VIII of this book, "Programming, Distributing, and Converting Access Applications," discusses .mde files in more detail.

Oracle Ex. 1009, p. 132
Oracle v. DeBellis
IPR2024-00221

Chapter 4

Working with Access Databases and Tables

The traditional definition of a database is *a collection of related data items that are stored in an organized manner.* Access is unique among desktop database development applications for the PC because of its all-encompassing database file structure. A single Access .mdb file can contain data objects—tables, indexes, and queries—as well as application objects—forms, reports, macros, and Access Visual Basic for Applications (VBA) code modules. Thus, you can create a complete Access database application stored in a single .mdb file. Most Access developers use two .mdb files: one to contain data objects and the other to hold application objects.

This chapter and Chapter 25, "Securing Multiuser Network Applications," explain why developers use two .mdb files. Regardless of the approach that you choose, Access's all-encompassing .mdb file structure makes creating and distributing database applications simpler.

Defining the Elements of Access Databases

Access databases include the following elements in a single .mdb database file:



- *Tables* store data items in a row-column format similar to that used by spreadsheet applications. An Access database can include as many as 32,768 objects (the combination of tables, forms, reports, queries, and so on), and as many as 1,024 tables can be open at one time if you have sufficient resources available. You can import tables from other database applications (such as xBase and Paradox), client/server databases (such as Microsoft SQL Server), and spreadsheet applications (such as Microsoft Excel and Lotus 1-2-3). In addition, you can link to Access databases other types of database tables (dBASE, FoxPro, and Paradox tables, for example), formatted files (Excel worksheet and ASCII text), and other Access databases. Chapter 7 discusses linking, importing, and exporting tables.

Oracle Ex. 1009, p. 133
Oracle v. DeBellis
IPR2024-00221



- *Queries* display selected data contained in as many as 16 tables. With queries, you can specify how to present data by choosing the tables that comprise the query and as many as 255 specific fields (columns) of the chosen tables. You determine the records (rows) to display by specifying the criteria that the data items in the query data must meet to be included in the display. Part II, "Querying for Specific Information," explains how to create queries.



- *Forms* display data contained in tables or queries and enable you to add new data and update or delete existing data. You can incorporate pictures and graphs in your forms, and, if you have a sound card, include narration and music in your form. You learn how to create forms in Chapters 12 and 13, and you learn how to add graphics to forms in Chapter 20. Access 97 forms also can incorporate Access VBA code in class models to provide event-handling subprocedures for forms and the controls that appear on forms.



- *Reports* print data from tables or queries in virtually any format that you want. Access enables you to add graphics to your reports so that you can print a complete, illustrated catalog of products from an Access database. Access's report capabilities are much more flexible than those of most other relational database management applications, including those designed for mini- and mainframe computers. Like forms, you can include Access VBA event-handling subprocedures in Access 97 reports. Chapters 14 and 15 cover creating reports.



- *Macros* automate Access operations. In prior versions of Access, macros took the place of the programming code required by other database applications, such as xBase, to perform specific actions in response to user-initiated events, such as clicking a command button. Access 97 supports macros primarily for compatibility with database applications created with earlier versions of Access. Microsoft recommends that you use Access VBA program code for any automation or event-handling procedures you create in Access 97 databases; future versions of Access are likely to phase out macro support in favor of VBA programming.



- *Modules* contain Access VBA code that you write to handle events such as clicking a command button in a form, to create customized functions for use in forms, reports, and queries, and to otherwise automate database operations. By judiciously adding Access VBA code to your database, you can create complete database applications with customized menus, toolbars, and other features. Access VBA code enables you to programmatically control many database options and operations you can't control with a macro. Chapter 28 describes how to write Access VBA code in general. Chapter 30 describes the specifics of writing event-handling VBA code stored behind forms and reports.



▶▶ See "Defining the Client/Server Environment," p. 915

A better definition of an Access database is *a collection of related data items and, optionally, the methods necessary to select, display, update, and report the data.* This definition emphasizes an important distinction between Access and other database management

Oracle Ex. 1009, p. 134
Oracle v. DeBellis
IPR2024-00221

applications. Even client/server database systems, such as Microsoft SQL Server, that include all related tables within a single database do not include the equivalent of forms and reports within the database. You must use another application, called a *front-end*, to display, edit, and report data stored in client/server databases. You can use Access to create front-ends for client/server databases by linking tables from the client/server database to your Access database. Creating front-ends for client/server databases is one of the major applications for Access in medium- to large-sized firms.

Note

It is a good database application development practice to maintain tables that store your application's data in one Access database (.mdb) file and the remainder of your application's objects, such as forms and reports, in a separate .mdb file. This chapter uses the Northwind Traders sample database, which is a self-contained application with a single .mdb file. Chapter 7, "Linking, Importing, and Exporting Tables," describes how to use or create separate .mdb files to store data and application objects.

This chapter introduces you to Access databases and tables—the fundamental elements of an Access application. You will see many references in this book to the term *Access application*. An Access application is an Access database that has the following characteristics:

- It contains the queries, forms, reports, and macros necessary to display the data in a meaningful way and to update the data as necessary. This book calls these elements *application objects*. A self-contained (single-user) Access application includes tables in the application database. Multiuser Access applications usually consist of two .mdb files, one containing the tables shared by users (*database objects*) and the other consisting of the Access application that manipulates the data. (Most Access developers use separate application and data .mdb files even for single-user applications.) If you are creating a front-end application, you usually link tables from the client/server database. Some front-end applications, however, also use local tables stored in the application database file.
- It does not require the database's users to know how to design any of its elements. All elements of the database are fully predefined during the application's design stage. In most cases, you want to restrict other users from intentionally or unintentionally changing the application's design.
- It is automated by Access VBA code (and/or Access macros for applications created prior to Access 97) so that users make choices from command buttons or custom-designed menus rather than from the lists in the Database window that you used in Chapter 2, "Up and Running with Access 97."

As you progress through the chapters in this book, you create a model of an Access application called Personnel Actions. Later in this chapter, you create the Personnel Actions table. In the following chapters, you add new features to the Personnel Actions application until you have a complete, automated method of adding and editing Personnel

Oracle Ex. 1009, p. 135
Oracle v. DeBellis
IPR2024-00221

Actions data. Therefore, you should read this book sequentially, at least to Chapter 15, "Preparing Advanced Reports." Make sure to perform the example exercises for the Personnel Actions application each time that you encounter them, because succeeding examples build on your prior work.

Understanding Relational Databases

All desktop database managers enable you to enter, edit, view, and print information contained in one or more tables divided into rows and columns. At this point, the definition of a database manager doesn't differ from that of a spreadsheet application—most spreadsheets can emulate database functions. Three principal characteristics distinguish relational database management systems (RDBMS) from spreadsheet applications:

- All RDBMS are designed to deal efficiently with very large amounts of data—much more than spreadsheets can handle conveniently.
- RDBMS easily link two or more tables so that they appear to the user as if they are one table. This process is difficult or impossible to accomplish with spreadsheets.
- RDBMS minimize information duplication by requiring repetition of only those data items, such as product or customer codes, by which multiple tables are linked.

Database managers that cannot link multiple tables are called *flat-file managers* and are used primarily to compile simple lists such as names, addresses, and telephone numbers.

Because relational databases eliminate most duplicate information, they minimize data storage and application memory requirements. Figure 4.1 shows a typical relational database that a manufacturing or distributing firm might use. This database structure is similar to that of the Northwind Traders sample database provided with Access.

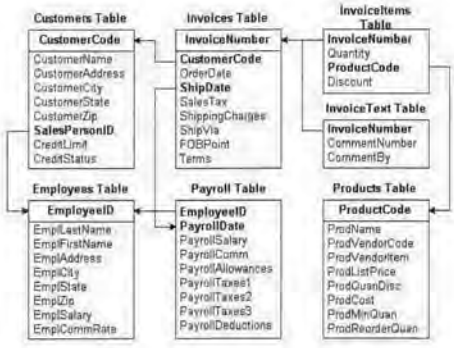


FIG. 4.1 A portion of a typical database for a manufacturing or distributing firm.

▶▶ See "The Process of Database Design," p. 790

Oracle Ex. 1009, p. 136
Oracle v. DeBellis
IPR2024-00221

If your job is to create an invoice-entry database, you don't need to enter a customer's name and address more than once. Just assign each customer a unique number or code and add to the Customers table a record containing this information. Similarly, you don't need to enter the names and prices of the standard products for each invoice. You assign unique codes to products, and then add records for them to the Products table. When you want to create a new invoice for an existing customer, you enter the customer code and type the codes and quantities for the products ordered. This process adds one record (identified by an automatically assigned sequential numeric code) to the Invoices table and one record for each different item purchased to the InvoiceItems table.

▶▶ See "Joining Tables to Create Multitable Queries," p. 322



Each table is related to the other by the customer, invoice, and product codes and numbers, shown by the connecting lines between the tables in Figure 4.1. The codes and numbers shown in boxes are unique; only one customer corresponds to a particular code, and one invoice or product corresponds to a given number. When you display or print an invoice, the Invoices table is linked (called a *join*) with both the Customers and InvoiceItems tables by their codes. In turn, the InvoiceItems table is joined with the Products table by the common value of a ProductCode in the InvoiceItems table and a ProductNumber in the Products table. Your query (view) of the desired sales orders displays the appropriate customer, invoice, items, and product information from the linked records. (The following section explains queries.) You can calculate quantity-price extensions, including discounts, by multiplying the appropriate values stored in the tables. You can add the extended items, sales taxes, and freight charges; you also can calculate the total invoice amount. These calculated values need not be included (and in a properly designed database never are included) in the database tables.

Using Access Database Files and Tables

Access has its own database file structure, similar to that used by client/server RDBMS, and uses the .mdb extension. As discussed in this chapter's introduction, Access differs from traditional PC databases in that a single file contains all the related tables, indexes, forms, and report definitions. The .mdb file even includes the programming code that you write in Access VBA. You don't need to be concerned with the intricacies of the .mdb file structure because Access handles all the details of file management for you.

All the field data types familiar to xBase and Paradox users are available in Access, as well as some new and useful field data types, such as Currency for monetary transactions. dBASE users need to learn to use the term *table* for file and *database* to indicate a group of related tables or files, the equivalent of dBASE's CATALOG. Records commonly are called *rows*, and fields often are called *columns*. This book uses the terms *records* and *fields* when referring to database tables, rows, and columns for sets of records returned by queries. Users with Paradox, Excel, or 1-2-3 experience should find the terminology of Access quite familiar. Paradox and FoxPro users will appreciate dealing with only one .mdb file rather than the myriad of files that make up a multitable Paradox or FoxPro database.

Oracle Ex. 1009, p. 137
Oracle v. DeBellis
IPR2024-00221

The Access System Database

In addition to including database files with the .mdb extension, Access includes a master database file, called a *workgroup file*, named System.mdw. (System.mdw is the equivalent of Access 2.0's SYSTEM.MDA.) This file contains information about the following:



◀◀ See "Setting Default Options," p. 81

- Names of users and groups of users who can open Access
- User passwords and a unique binary code, called a System ID (SID), that identifies the current user to Access
- Operating preferences that you establish by choosing Tools, Options
- Definitions of customized Access 97 toolbars that each user creates

Chapter 25, "Securing Networked Multiuser Applications," covers sharing database files and granting permission for others to use the files.

Access Library Databases

Another category of Access database files is *add-ins*, also called *libraries*. Add-ins are Access library databases, usually with an .mde or .mda extension to distinguish them from user databases, that you can link to Access by choosing Tools, References in the Module window, or through the Add-In Manager (which you can access by choosing Tools, Add-Ins).



▶▶ See "Creating a Transaction-Processing Form with the Form Wizard," p. 397
▶▶ See "Access's Integrated Data Dictionary System," p. 819

When you link to an Access library, all the elements of the library database are available to you after you open Access. The Access 97 wizards—which you use to create forms, reports, and graphs—are stored in a series of Access library database files: Wzlib80.mde, Wztool80.mde, and Wzmain80.mde. Another wizard enables you to create data dictionaries for Access databases. A *data dictionary* is a detailed written description of each of a database's elements. Add-in library databases are an important and unique feature of Access. Microsoft and other third-party firms provide a wide range of Access libraries to add new features and capabilities to Access.

Creating a New Database

If you have experience with relational database management systems, you might want to start building your own database as you progress through this book. In this case, you need to create a new database file at this point. If database management systems are new to you, however, you should instead explore the sample databases supplied with Access as you progress through the chapters of this book, and design your first database using the principles outlined in Chapter 23, "Exploring Relational Database Design and Implementation." Then return to this section and create your new database file.

Oracle Ex. 1009, p. 138
Oracle v. DeBellis
IPR2024-00221

To create a new database, follow these steps:

1. If you aren't already running Access, launch it and skip to step 3.
2. If Access is running and the Database window is visible, click its title bar to make the Database window the active window. If the Database window is not visible, click the Show Database Window button of the toolbar, choose Window, 1 Database, or press the F11 key.

   This action is called giving the Database window the *focus*. When a window has the focus, the background of its title bar is usually blue; when a window does not have the focus, it is inactive and its title bar is usually gray. If you have changed from the default Windows color scheme by altering the Windows 95 desktop's properties, these colors are different.
3. Click the New Database button of the toolbar, or choose File, New Database. For the Database toolbar to be visible and the New Database and other database file options to be present when you open the File menu, the Access application window must be empty or the Database window must be active. The New dialog appears as shown in Figure 4.2.



FIG. 4.2 The New dialog, in which you select the type of database to create.

The General page of the New dialog enables you to choose to create a blank database, and the Database page lets you use any one of 22 new database templates. Access 97 comes with database templates for asset tracking, book and video collections, contact management, and many other typical business and personal database uses. You choose a template that suits the purpose for which you want to create the database.

4. For this example, click the General tab, select Blank Database, and then click OK to display the File New Database dialog shown in Figure 4.3.

   Access supplies the default file name, db1.mdb, for new databases. (If you have previously saved a database file as db1.mdb, Access proposes db2.mdb as the default.)

Oracle Ex. 1009, p. 139
Oracle v. DeBellis
IPR2024-00221



FIG. 4.3 The File New Database dialog, in which you enter the new database's name.

5. In the File Name text box, enter a file name for the new database. Use conventional Windows 95 file-naming rules (you can use spaces and punctuation in the name). Don't include an extension in the file name; Access automatically supplies the .mdb extension.

6. Click Create or press Enter to create the new database.

   If a database was open when you created the new database, Access closes any windows associated with the database and the Database window. During the process of creating the database, the following message appears in the status bar:

   ```
   Verifying system objects
   ```

   Whenever you open a new or existing database, Access checks whether all the database's elements are intact. Access's main window and the Database window for the new database (named new.mdb for this example) appear as shown in Figure 4.4.



FIG. 4.4 The Database window for a newly created database.

Each new Access 97 database occupies approximately 60K of disk space when you create it—a reduction in size from the approximately 80K databases created by Access 95. Most of the 60K is space consumed by hidden system tables for adding the information necessary to specify the names and locations of other database elements that the database file contains. Because of the way that data is stored in Access tables, what appears to be an excessive amount of reserved space quickly is compensated for by Access's more efficient data storage methods.

Oracle Ex. 1009, p. 140
Oracle v. DeBellis
IPR2024-00221

Understanding the Properties of Tables and Fields

Before you add a table to a database that you have created, or to one of the sample databases supplied with Access, you need to know the terms and conventions that Access uses to describe the structure of a table and the fields that contain the data items that comprise the information stored in the table. With Access, you specify properties of tables and fields.

▶▶ See "Working with Data Dictionaries," p. 818





Properties of Access tables apply to the table as a whole. Entering table properties is optional. You enter properties of tables in text boxes of the Table Properties window (see Figure 4.5), which you display by clicking the toolbar's Properties button in Table Design view. The five basic properties of Access tables follow:

- *Description* allows you to enter an optional explanation of the table's purpose. If you choose View, Details, the Database window displays this description. This description also is useful with a data dictionary. You use data dictionaries to document databases and database applications.
- *Validation Rule* allows you to enter an optional Validation Rule property value that lets you establish domain integrity rules with expressions that refer to more than one field of the table. The validation rule that you enter here applies to the table as a whole, rather than to a single field.
- *Validation Text* allows you to enter an optional Validation Text property value that specifies the text of the message box that appears if you violate a table's Validation Rule expression.
- *Filter* allows you to enter an optional property value that specifies a filter to apply to the table whenever it is opened. Chapter 6, "Sorting, Finding, and Filtering Data in Tables," discusses filters.
- *Order By* allows you to enter an optional property value that specifies a sorting order to apply to the table whenever it is opened. Chapter 6 also explains sorting orders.

▶▶ See "Using Access Indexes," p. 823



Access 97 provides an Indexes window to specify the primary key and all table indexes. The section "Adding Indexes to Tables" later in this chapter describes how to use the Indexes window.



You assign each field of an Access table a set of properties. The first three field properties are assigned within the Table Design grid, the upper pane of the Table Design window shown in Figure 4.6. To assign the Primary Key property, select the field and click the

Oracle Ex. 1009, p. 141
Oracle v. DeBellis
IPR2024-00221

Primary Key button on the toolbar (the Order Details table shown in Figure 4.6 has two primary keys). You set the remaining property values in the Table Design window's lower pane, Field Properties.



FIG. 4.5 The Table Properties window for the Northwind Traders sample database's Order Details table.



FIG. 4.6 The Table Design window for the Northwind Traders sample database's Order Details table.

The following list summarizes the properties that you set in the Table Design grid:

- *Field Name*. You enter the name of the field in the Table Design grid's first column. Field names can be as long as 64 characters and can include embedded (but not leading) spaces and punctuation—except periods (.), exclamation marks (!), and square brackets ([ ]). Field names are mandatory, and you cannot assign the same field name to more than one field. It is a good database programming practice not to include spaces in field names. (Substitute an underscore (_) for spaces or use upper- and lowercase letters to improve the readability of field names.) Minimizing the length of field names conserves resources and saves typing when you refer to the field name in macros or Access VBA code.



- *Data Type*. You select data types from a drop-down list in the Table Design grid's second column. Data types include Text, Memo, Number, Date/Time, Currency, AutoNumber, Yes/No, OLE Object, Hyperlink, and Lookup Wizard. (AutoNumber replaces the Counter data type of Access 1.x and 2.0; Hyperlink data types are a new feature of Access 97.) Choosing a data type is the subject of the next section of this chapter.

Oracle Ex. 1009, p. 142
Oracle v. DeBellis
IPR2024-00221

- *Description*. You can enter an optional description of the field in the text box in the Table Design grid's third column. If you add a description, it appears in the status bar at the lower left of Access's window when you select the field for data entry or editing.
- *Primary Key*. To choose a field as the primary-key field, select the field by clicking the field-selection button to the left of the Field Name column, and then click the Primary Key button on the toolbar. The Order Details table has a composite primary key, consisting of the OrderID and ProductID fields. (See "Selecting a Primary Key" later in this chapter for instructions on how to create a composite primary key.)

Depending on the specific data type that you choose for a field, there are additional properties that you can set for a table field. You set these additional properties in the General tab of the Table Design window's Field Properties pane by selecting from drop-down or combo lists or by typing values in text boxes. (You use the Field Properties pane's Lookup tab to set the control type for lookup fields on forms—list box, combo list, and so on. Chapter 13, "Designing Custom Multitable Forms," describes how to use lookup fields.) The following list summarizes the General field properties:

- *Field Size*. You enter the field size for the Text data type in this text box. (See "Text Fields" later in this chapter to learn how to choose a text field size.) For Numeric data types, you choose the field size by selecting from a drop-down list. Field size does not apply to the Date/Time, Yes/No, Currency, Memo, Hyperlink, or OLE Object data type.
- *Format*. You can select a standard, predefined format in which to display the values in the field from the drop-down combo list applicable to the data type that you chose (except Text). Alternatively, you can enter a custom format in the text box (see "Custom Display Formats" later in this chapter). The Format property does not affect the data values; it affects only how these values are displayed. The Format property does not apply to OLE Object fields.
- *Decimal Places*. You can select Auto or a specific number of decimal places from the drop-down combo list, or you can enter a number in the text box. The Decimal Places property applies only to Number and Currency fields. Like the Format property, the Decimal Places property affects only the display, not the data values, of the field.
- *Input Mask*. Input masks are character strings, similar to the character strings used for the Format property, that determine how to display data during data entry and editing. If you click the Ellipsis button for a field of the Text, Currency, Number, or Date/Time field data type, Access starts the Input Mask Wizard to provide you with a predetermined selection of standard input masks, such as telephone numbers with optional area codes.

Oracle Ex. 1009, p. 143
Oracle v. DeBellis
IPR2024-00221

- *Caption.* If you want a name (other than the field name) to appear in the field name header button in Table Datasheet View, you can enter in the Caption list box an alias for the field name. The restrictions on punctuation symbols do not apply to the Caption property. (You can use periods, exclamation points, and square brackets.)
- *Default Value.* By entering a value in the Default Value text box, you specify a default value that Access automatically enters in the field when a new record is added to the table. The current date is a common default value for a Date/Time field. (See "Setting Default Values of Fields" later in this chapter for more information.) Default values do not apply to fields with AutoNumber or OLE Object field data types.
- *Validation Rule.* Validation rules test the value entered in a field against criteria that you supply in the form of an Access expression. Chapter 9, "Understanding Operators and Expressions in Access," explains expressions. The Validation Rule property is not available for fields with AutoNumber, Memo, or OLE Object field data types. Adding validation rules to table fields is one of the subjects in Chapter 5, "Entering, Editing, and Validating Data in Tables."
- *Validation Text.* You enter the text that is to appear in the status bar if the value entered does not meet the Validation Rule criteria.
- *Required.* If you set the value of the Required property to Yes, you must enter a value in the field. Setting the Required property to Yes is the equivalent of typing **Is Not Null** as a field validation rule. (You do not need to set the value of the Required property to Yes for fields included in the primary key because Access does not permit `Null` values in primary-key fields.)
- *Allow Zero Length.* If you set the value of the Allow Zero Length property to Yes, and the Required property is also Yes, the field must contain at least one character. The Allow Zero Length property applies to the Text, Memo, and Hyperlink field data types only. A zero-length string (`""`) and the `Null` value are not the same.
- *Indexed.* From the drop-down list, you can select between an index that allows duplicate values or one that requires each value of the field to be unique. You remove an existing index (except from a field that is a single primary-key field) by choosing No. The Indexed property is not available for Memo, OLE Object, or Hyperlink fields. (See "Adding Indexes to Tables" later in this chapter for more information on indexes.)
- *New Values.* This property applies only to AutoNumber fields. You select either Increment or Random from a drop-down list. If you set the New Values property to Increment, Access generates new values for the AutoNumber field by adding 1 to the highest existing AutoNumber field value. If you set the property to Random, Access generates new values for the AutoNumber field by producing a pseudo-random-long integer. Typically, Access uses randomly generated AutoNumber values for replicated databases to ensure that it assigns unique values to records in the database replicas.

Oracle Ex. 1009, p. 144
Oracle v. DeBellis
IPR2024-00221

To add your first table, Personnel Actions, to the Northwind Traders database, you must choose appropriate data types, sizes, and formats for your table's fields.

Choosing Field Data Types, Sizes, and Formats

You must assign a field data type to each field of a table, unless you want to use the Text data type that Access assigns as the default. One of the principles of relational database design is that all the data in a single field consists of one type of data. Access provides a much wider variety of data types and formats from which to choose than most PC database managers. Besides setting the data type, you can set other field properties that determine the format, size, and other characteristics of the data that affect its appearance and the accuracy with which numerical values are stored. Table 4.1 lists the field data types that you can select for data contained in Access tables.

Table 4.1 Field Data Types Available in Access

Information	Data Type	Description of Data Type
Characters	Text	Text fields are most common, so Access assigns Text as the default data type. A Text field can contain as many as 255 characters, and you can designate a maximum length less than or equal to 255. Access assigns a default length of 50 characters. A fixed-length Text data type is similar to xBase's Character field and Paradox's Alphanumeric field.
	Memo	Memo fields can contain as many as 64,000 characters. You use them to provide descriptive comments. Memo fields are similar to those of xBase, except that the Memo field's data is included in the table rather than in a separate file. Access displays the contents of Memo fields in Datasheet view. A Memo field cannot be a key field, and you cannot index a Memo field.
Numeric Values	Number	A variety of numeric data subtypes are available. You choose the appropriate data subtype by selecting one of the Field Size property settings listed in Table 4.2. You specify how to display the number by setting its Format property to one of the formats listed in Table 4.4.
	AutoNumber	An AutoNumber field is a numeric (Long Integer) value that Access automatically fills in for each new record that you add to a table. Access can increment the AutoNumber field by one for each new record, or fill in the field with a randomly generated number, depending on the New Values property setting that you choose. An AutoNumber field creates a value similar to xBase's and Paradox's record number. The maximum number of records in a table that can use the AutoNumber field is slightly more than two billion.

(continues)

Oracle Ex. 1009, p. 145
Oracle v. DeBellis
IPR2024-00221

Table 4.1 Continued

Information	Data Type	Description of Data Type
	Yes/No	Logical (Boolean) fields in Access use numeric (Logical Fields) values: –1 for Yes (**True**) and 0 for No (**False**). You use the Format property to display Yes/No fields as Yes or No, True or False, On or Off, or –1 or 0. (You also can use any nonzero number to represent **True**.) Logical fields cannot be key fields, but they can be indexed. Currency is a special fixed format with four decimal places designed to prevent rounding errors that would affect accounting operations where the value must match to the penny (similar to the Paradox Currency data type).
Dates and Times	Date/Time	Dates and times are stored in a special fixed format. The date is represented by the whole number portion of the Date/Time value, and the time is represented by its decimal fraction. You control how Access displays dates by selecting one of the Date/Time Format properties listed in Table 4.4.
Large Objects	OLE Object (BLOBs, binary, large objects)	Includes bitmapped graphics, vector-type drawings, waveform audio files and other types of data that can be created by an ActiveX component application (formerly known as an OLE server application), some of which are listed in Table 4.3. You cannot assign an OLE Object as a key field, and you cannot include an OLE Object field in an index.
Web and other HTML document addresses	Hyperlink	Hyperlink fields store Web page addresses. A Web address stored in the Hyperlink field may refer to a Web page on the Internet, or a Web page stored locally on your computer or network. Clicking a Hyperlink field causes Access to start your Web browser and display the referenced Web page; choose Insert, Hyperlink to add a new hyperlink address.

Regardless of the length that you set for Text fields in Access, the database file stores them in variable-length records. All trailing spaces are removed. This technique conserves the space wasted in xBase files, for example, which store text in fixed-length character fields. Fixed-length character fields in conventional PC RDBMS waste the bytes used to pad short text entries in long fields.

Choosing Field Sizes for Numeric and Text Data

The Field Size property of a field determines which data type a Number field uses or how many characters fixed-length text fields can accept. Field Size properties are called *subtypes* to distinguish them from the *data types* listed in Table 4.2. For numbers, you select a Field Size property value from the Field Size drop-down list in the Table Design window's lower pane, Field Properties (refer to Figure 4.6).

Subtypes for Numeric Data. The Number data type of Table 4.2 isn't a fully specified data type. You must select one of the subtypes from those listed in Table 4.3 for the Field

Oracle Ex. 1009, p. 146
Oracle v. DeBellis
IPR2024-00221

Size property to define the numeric data type properly. To select a data subtype for a Number field, follow these steps:

1. Select the Data Type cell of the Number field for which you want to select the subtype.
2. Click the Field Size text box in the Field Properties window. You also can press F6 to switch windows, and then use the arrow keys to position the caret within the Field Size text box.
3. Click the drop-down arrow to open the list of choices shown in Figure 4.7. You can also press the F4 key to open the list.



FIG. 4.7 Selecting from the Field Size list a subtype for the Number data type.

4. Select the data subtype. Table 4.3 describes data subtypes. When you make a selection, the list closes.

After you select a Field Size property, you select a Format property from those listed in Table 4.4 to determine how to display the data. Table 4.4 includes the Currency data type because it also can be considered a subtype of the Numeric data type.

Regardless of how you format your data for display, the number of decimal digits, the range, and the storage requirement remains that specified by the Field Size property. Except for the Byte and Boolean data types, these data types are available in most dialects of BASIC, including Visual Basic for Applications and Visual Basic 4.0. Access VBA includes all the data types listed in Table 4.2 as reserved words. You cannot use a reserved data type word for any purpose in Access VBA functions and procedures other than to specify a data type.

Table 4.2 Subtypes of the Number Data Type Determined by the Field Size Property

Field Size	Decimals	Range of Values	Bytes	xBase	Paradox
Double	15 places	–1.797 * 10^308 to +1.797 * 10^308	8	Numeric	Numeric
Single	7 places	–3.4 * 10^38 to +3.4 * 10^38	4	N/A	N/A

(continues)

Oracle Ex. 1009, p. 147
Oracle v. DeBellis
IPR2024-00221

Table 4.2 Continued

Field Size	Decimals	Range of Values	Bytes	xBase	Paradox
Long Integer	None	-2,147,483,648 to +2,147,483,647	4	N/A	N/A
Integer	None	-32,768 to 32,767	2	N/A	Short Number
Byte	None	0 to 255	1	N/A	N/A
Currency (a data type, not a subtype)	4 places	-922337203685477.5808 to +922337203685477.5808	4	N/A	N/A

Table 4.2's xBase and Paradox columns indicate the Access data types that correspond to data types commonly used with these two types of Windows and DOS RDBMSs.

Both xBase RDBMSs and Paradox store numbers with 15 significant-digit precision. Neither xBase nor Paradox, however, offers the full range of values of numbers having Access's Double Field Size property. All references to Paradox in this book apply to DOS 4.5 and Paradox for Windows 5. Access is compatible only with Paradox 3.5 (and later) tables.

As a rule, you select the Field Size property that results in the smallest number of bytes that encompasses the range of values you expect and that expresses the value in sufficient precision for your needs. Mathematical operations with Integer and Long Integer proceed more quickly than those with Single and Double data types (called *floating-point numbers*) or the Currency and Date/Time data types (*fixed-point numbers*).

Fixed-Width Text Fields. You can create a fixed-width Text field by setting the value of the Field Size property. By default, Access creates a 50-character-wide Text field. Enter the number, from 1 to 255, in the Field Size cell corresponding to the fixed length that you want. If you import to the field data that is longer than the selected field size, Access truncates the data; thus, you lose the far right characters that exceed your specified limit. You therefore enter a field length value that accommodates the maximum number of characters that you expect to enter in the field. Fixed-width Text fields behave identically to the Character and Alphanumeric field data types of xBase and Paradox applications.

Note

The terms *fixed-width* and *fixed-length* have two different meanings in Access. Even if you specify a fixed-width for a field of the Text field data type, Access stores the data in the field in variable-length format.

Subtypes for the OLE Object Data Type. Fields that have data types other than characters and numbers must use the OLE (object linking and embedding) Object data type. Chapter 19, "Using 32-Bit OLE Components," describes object linking and embedding,

Oracle Ex. 1009, p. 148
Oracle v. DeBellis
IPR2024-00221

and Chapter 20, "Adding Graphics to Forms and Reports," explains the OLE Object field data type. Because many subtypes of OLE data exist, this data type enables you to violate the rule of database design that requires all data in a field to be of a single data type. Table 4.3 lists typical OLE Object subtypes. To avoid breaking the rule, you should create separate OLE Object fields for different OLE data subtypes. For example, if your table stores both sound and images, use one OLE Object field for OLE sound objects, and a second OLE Object field for OLE images.

Table 4.3 Subtypes of the OLE Object Data Type and the OLE Servers that Create Them

OLE Subtype	File Format	OLE Server(s) that Create the Subtype
Bitmapped graphics	.bmp, .dib, .tif	Paint (supplied with Windows 95 and Windows NT 4.0), Adobe Photoshop
Vector-based drawings	.wmf	CorelDRAW!, PowerPoint
Diagrams	.vsd	Visio 3 and higher
Formatted text	.rtf	Wordpad (supplied with Windows 95 and Windows NT 4.0), Word for Windows, Lotus WordPro
Unformatted text	.txt	Object Packager (with Windows Notepad as the application); Memo fields are a better choice
Spreadsheet	.xls, .dif	Microsoft Excel 3.0 and later
Waveform audio	.wav	Windows 95/NT 4.0 Sound Recorder, Windows 95/NT 4.0 Media Player
MIDI music	.mid	Windows 95/NT 4.0 Media Player
Digital video	.avi	Windows 95/NT 4.0 Media Player

In the case of OLE Object fields, the data subtype is determined by the OLE server used to create the data, rather than by an entry in a text box or a selection from a list box. Windows 95's and Windows NT's Object Packager OLE server enables you to embed files created by applications that aren't OLE servers. You can, for example, embed .txt and .mid files with Object Packager.

Selecting a Display Format

You establish the Format property for the data types that you select so that Access displays them appropriately for your application. You select a format by selecting the field and then clicking the Format text box in the Field Properties window. Figure 4.8 shows the choices that Access offers for formatting the Long Integer data type. You format Number, Date/Time, and Yes/No data types by selecting a standard format or creating your own custom format. The following sections describe these two methods.

Oracle Ex. 1009, p. 149
Oracle v. DeBellis
IPR2024-00221



FIG. 4.8 Assigning a standard format to a Long Integer field.

Standard Formats for Number, Date/Time, and Yes/No Data Types. Access provides 17 standard formats that apply to the numeric values in fields of the Number, Date/Time, and Yes/No data types. The standard formats shown in Table 4.4 probably will meet most of your needs.

Table 4.4 Standard Display Formats for Access Number, Date/Time, and Yes/No Data Types

Data Type	Format	Appearance
Number	General Number	1234.5
	Currency	$1,234.50
	Fixed	12345
	Standard	1,234.50
	Percent	0.1234 = 12.34%
	Scientific	1.23E+03
Date/Time	General Date	10/1/97 4:00:00 PM
	Long Date	Thursday, October 1, 1997
	Medium Date	1-Oct-97
	Short Date	10/1/97
	Long Time	4:00:00 PM
	Medium Time	04:00 PM
	Short Time	16:00
Yes/No	Yes/No	Yes or No
	True/False	True or False
	On/Off	On or Off
	None	–1 or 0

The Short Date format is similar to the Date data type in xBase and Paradox, except that leading zeros for months and days having a value less than 10 aren't displayed. The four date formats are included in the 11 date formats offered by Paradox; you can create the other Paradox formats by using custom formats described in the next section. The Short Time format is equivalent to what you obtain from an xBase `SUBSTR(TIME(),5)` expression. A Double data type with Currency format appears identical to Paradox's Currency data type.

The *Null* Value in Access Tables. Fields in Access tables can have a special value, **`Null`**, which is a new term for most users of PC-based database management systems. The **`Null`** value indicates that the field contains no data at all. **`Null`** isn't the same as a numeric

Oracle Ex. 1009, p. 150
Oracle v. DeBellis
IPR2024-00221

value of zero, nor is it equivalent to blank text that consists of one or more spaces. **Null** is similar but not equivalent to an empty string (a string of zero length, often called a *null string*). For now, the best synonym for **Null** is *no entry*. (**Null** is set in monospace bold type because it is a reserved word in Access VBA.)

The **Null** value is useful for determining whether a value has been entered in a field, especially a numeric field in which zero values are valid. Until the advent of Access, the capability to use **Null** values in database managers running on PCs was limited to fields in the tables of client/server database systems, such as Microsoft SQL Server. Later in this chapter, the sections "Custom Display Formats" and "Setting Default Values of Fields" use the **Null** value.

Custom Display Formats. To duplicate precisely the format of xBase's Date data type, or one of the date formats offered by Paradox that is not a standard format in Access, you must create a custom format. You also want or need to create custom display formats if you have some special format in which you want data to appear. You can set a custom display format for any field type, except OLE Object. You do so by creating an image of the format using combinations of a special set of characters called *placeholders*, listed in Table 4.5. Figure 4.9 shows an example of a custom format for date and time. If you enter **mmmm dd", "yyyy - hh:nn** as the format, the date 03/01/97 displays as `March 1, 1997 - 00:00`.

Except as noted, the example numeric value that Table 4.5 uses is `1234.5`. Bold type distinguishes the placeholders that you type from the surrounding text. The resulting display is shown in `monospace` type.

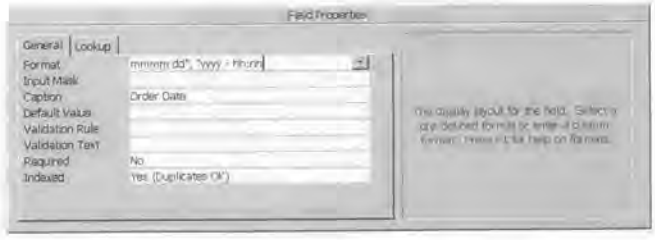


FIG. 4.9 A custom date and time format entry in the Format text box.

Table 4.5 Placeholders for Creating Custom Display Formats

Placeholder	Function
`Empty string`	Displays the number with no formatting. Enter an empty string by deleting the value in the Format Text field of the Field Properties window.
`0`	Displays a digit if one exists in the position, or a zero if not. You can use the `0` placeholder to display leading zeros for whole numbers and trailing zeros in decimal fractions. **00000.000** displays `01234.500`.
`#`	Displays a digit, if one exists in the position; otherwise, displays zeros. The # placeholder is equivalent to `0`, except that leading and trailing zeros aren't displayed. **#####.###** displays `1234.5`.

(continues)

Oracle Ex. 1009, p. 151
Oracle v. DeBellis
IPR2024-00221

Table 4.5 Continued

Placeholder	Function
$	Displays a dollar sign in the position. **$###,###.00** displays $1,234.50.
.	Displays a decimal point at the indicated position in a string of 0 and # placeholders. **##.##** displays 1234.5.
%	Multiplies the value by 100 and adds a percent sign in the position shown with 0 and # placeholders. **#0.00%** displays 0.12345 as 12.35% (12.345 is rounded to 12.35).
, (comma)	Adds commas as thousands separators in strings of 0 and # placeholders. **###,###,###.00** displays 1,234.50.
E- e-	Displays values in scientific format with the sign of exponent for negative values only. **#.####E-00** displays 1.2345E03. 0.12345 is displayed as 1.2345E-01.
E+ e+	Displays values in scientific format with the sign of exponent for positive and negative values. **#.####E+00** displays 1.2345E+03.
/	Separates the day, month, and year to format date values. **mm/dd/yy** displays 06/06/97. You can substitute hyphens to display 06-06-97.
m	Specifies how to display months for dates. **m** displays 1, **mm** displays 01, **mmm** displays Jan, and **mmmm** displays January.
d	Specifies how to display days for dates. **d** displays 1, **dd** displays 01, **ddd** displays Mon, and **dddd** displays Monday.
y	Specifies how to display years for dates. **yy** displays 97, and **yyyy** displays 1997.
: (colon)	Separates hours, minutes, and seconds in format time values. **hh:mm:ss** displays 02:02:02.
h	Specifies how to display hours for time. **h** displays 2, and **hh** displays 02. If you use an AM/PM placeholder, **h** or **hh** displays 4 PM for 16:00 hours.
n	Minutes placeholder for time. **n** displays 1, and **nn** displays 01. **hhnn "hours"** displays 1600 hours.
s	Seconds placeholder for time. **s** displays 1, and **ss** displays 01.
AM/PM	Displays time in 12-hour time with AM or PM appended. **h:nn AM/PM** displays 4:00 PM. Alternative formats include am/pm, A/P, and a/p.
@	Indicates that a character is required in the position in a Text or Memo field. You can use @ to format telephone numbers in a Text field, as in **@@@-@@@-@@@@** or **(@@@) @@@-@@@@**.
&	Indicates that a character in a Text or Memo field is optional.

Oracle Ex. 1009, p. 152
Oracle v. DeBellis
IPR2024-00221

Placeholder	Function
>	Changes all text characters in the field to uppercase.
<	Changes all text characters in the field to lowercase.
*	Displays the character following the asterisk as a fill character for empty spaces in a field. **"ABCD"*x** in an eight-character field appears as ABCDxxxx.

The Format drop-down combo list is one of the few examples in Access where you can select from a list of options or type your own entry. Format is a true drop-down combo list; lists that only enable you to select from the listed options are *drop-down lists*. You don't need to enter the quotation marks shown in Figure 4.9 surrounding the comma and space in the Format text box (**mmmm dd", "yyyy - hh:nn**) because Access does this for you. The comma is a nonstandard formatting symbol for dates (but is standard for number fields). When you create nonstandard formatting characters in the Field Properties window, Access automatically encloses them in double quotation marks.

When you change Format or any other property of a field, and then change to Datasheet view in Run mode to view the result of your work, you first must save the updated table design. The confirmation dialog shown in Figure 4.10 asks you to confirm any design changes.



FIG. 4.10 The confirmation dialog for changes to a field's format.

If you apply the custom format string **mmmm dd", "yyyy - hh:nn** (refer to Figure 4.9) to the Birth Date field of the Employees table, the Birth Date field entries appear as shown in Figure 4.11. For example, Nancy Davolio's birth date appears as `December 08, 1948 - 00:00`. The original format of the Birth Date field was Medium Date, the format also used for the Hire Date field.

You need to expand the width of the Birth Date field to accommodate the additional characters in the Long Date format. You increase the field's width by dragging the field name header's right vertical bar to the right to display the entire field. Access left-justifies date fields.

Access displays the time of birth as `00:00` because the decimal fraction that determines time is 0 for all entries in the Birth Date field.

The following is an example that formats negative numbers enclosed in parentheses and replaces a **Null** entry with text:

```
$###,###,##0.00;$(###,###,##0.00);0.00;"No Entry Here"
```

Oracle Ex. 1009, p. 153
Oracle v. DeBellis
IPR2024-00221

The entries 1234567.89, –1234567.89, 0, and a **Null** default value appear as follows:

```
$1,234,567.89
$(1,234,567.89)
0.00
No Entry Here
```

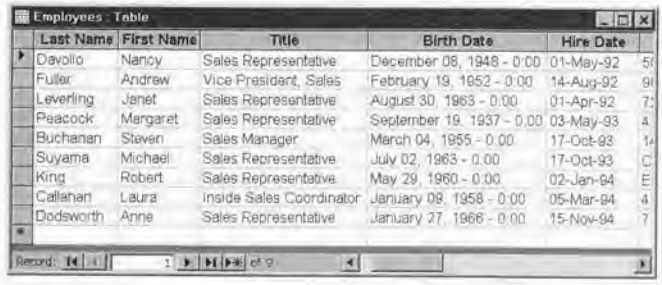


FIG. 4.11 Comparing date formats.

Using Input Masks

Access 97 enables you to restrict entries in Text fields to numbers or to otherwise control the formatting of entered data. Access 97's Input Mask property is used to format telephone numbers, Social Security numbers, ZIP codes, and similar data. Table 4.6 lists the placeholders that you can use in character strings for input masks in fields of the Text field data type.

Table 4.6 Placeholders for Creating Input Masks

Placeholder	Function
Empty string	No input mask.
0	Number (0–9) or sign (+/–) required.
9	Number (0–9) optional (a space if nothing is entered).
#	Number (0–9) or space optional (a space if nothing is entered).
L	Letter (A–Z) required.
?	Letter (A–Z) not required (a space if nothing is entered).
A	Letter (A–Z) or number (0–9) required.
a	Letter (A–Z) or number (0–9) optional.
&	Any character or a space required.
C	Any character or a space optional.

Oracle Ex. 1009, p. 154
Oracle v. DeBellis
IPR2024-00221

Placeholder	Function
. , : ; / ( )	Literal decimal, thousands, date, time, and special separators.
>	All characters to the right are converted to uppercase.
<	All characters to the right are converted to lowercase.
!	Fills the mask from right to left.
\	Precedes the other placeholders to include the literal character in a format string.

For example, typing **\(000") "000\-0000** as the value of the Input Mask property causes results in the appearance of (___) ___-____ for a blank telephone number cell of a table. Typing **000\-00\-000** creates a mask for Social Security numbers, ___-__-____. When you type the telephone number or Social Security number, the digits that you type replace the underscores.

Note

The \ characters (often called *escape characters*) that precede parentheses and hyphens specify that the character that follows is a literal, not a formatting character. If the format includes spaces, enclose the spaces and adjacent literal characters in double quotation marks.

Access 97 includes an Input Mask Wizard that appears when you place the caret in the Input Mask text box for a field of the Text or Date/Time field data type and click the Ellipsis button at the extreme right of the text box. Figure 4.12 shows the opening dialog of the Input Mask Wizard, which provides 10 common input mask formats from which you can choose.

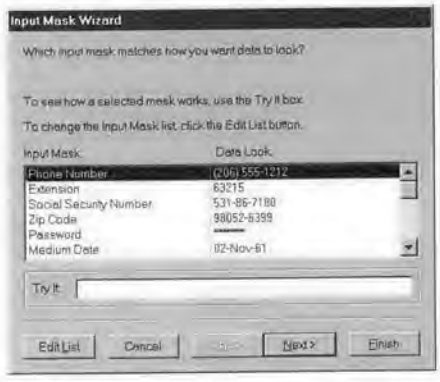


FIG. 4.12 The Input Mask Wizard for Text and Date/Time field data types.

Oracle Ex. 1009, p. 155
Oracle v. DeBellis
IPR2024-00221

Using the Northwind Traders Sample Database

One of the fundamental problems with books about database management applications is the usual method of demonstrating how to create a "typical" database. You are asked to type fictitious names, addresses, and telephone numbers into a Customers table. Next, you must create additional tables that relate these fictitious customers to their purchases of various widgets in assorted sizes and quantities. This process is unrewarding for readers and authors, and few readers ever complete the exercises.

Therefore, this book takes a different tack. Access includes a comprehensive and interesting sample database, Northwind Traders. Instead of creating a new database at this point, you create a new table as an addition to the Northwind Traders database. Adding a new table minimizes the amount of typing required and requires just a few entries to make the table functional. The new Personnel Actions table demonstrates many of the elements of relational database design. Before you proceed to create the Personnel Actions table, try the quick example of adding a new table to the Northwind Traders sample database that is given in the following section.

Using the Table Wizard to Create New Tables

Access includes a variety of wizards that provide services that simplify the creation of new database objects. Wizards lead you through a predetermined set of steps that determine the characteristics of the object that you want to create. Access 97 includes a Table Wizard that you can use to create new tables based on prefabricated designs for 77 business-oriented and 44 personal-type tables. Many of the business-oriented table designs are based on tables contained in Northwind.mdb.

The Table Wizard serves as an excellent introduction to the use of Access wizards in general. Follow these steps to create a new Access table that catalogs a video collection:



1. If the Employees table is open, close it by clicking the Close Window button to make the Database window active. Alternatively, click the Show Database Window button of the toolbar.



2. Click the Tables button of the Database window if it isn't selected, and then click the New button to display the New Table dialog shown in Figure 4.13.



FIG. 4.13 The New Table dialog.

Oracle Ex. 1009, p. 156
Oracle v. DeBellis
IPR2024-00221

3. Select Table Wizard in the list, and click OK to display the opening dialog of the Table Wizard. (If you select Datasheet view, Access creates a blank table with default fields; Design view creates a blank table and displays it in design mode ready for you to add fields. The Import Table Wizard and Link Table Wizard import databases and link external tables to a database, respectively.)

4. Click the Personal option button to display a list of tables for personal use in the Sample Tables list, and then use the vertical scroll bar to display the Video Collection entry in the list.

5. In the Sample Tables list, click the Video Collection entry to display the predetermined set of field names for the new table in the Sample Fields list.

6. Click the >> button to add all the fields from the Sample Fields list to the Fields in My New Table list. (The > button adds a single selected field from the Sample Fields list, the < button removes a single selected field from the My New Table list, and the << button deletes all the fields in the My New Table list.) The Table Wizard's dialog now appears as in Figure 4.14.

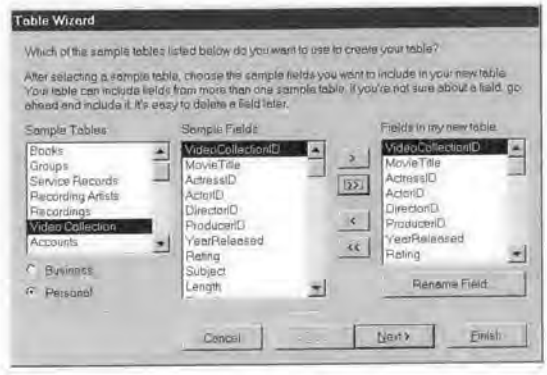


FIG. 4.14 Adding fields to the new Video Collection table.

7. Choose the Next button to display the second Table Wizard dialog in which you select the name for your new table, and select how to determine the table's primary-key field. Accept the default table name or enter a name of your choice, and then click the option button No, I'll Set the Primary Key. The Table Wizard's second dialog now appears (see Figure 4.15).

8. Choose the Next button to display the dialog shown in Figure 4.16. This dialog enables you to select the primary-key field and its data type. The VideoCollectionID field is the logical choice for a primary key, and the AutoNumber field data type, which automatically creates a sequential number for the VideoCollectionID, is appropriate in this case. (The Table Wizard's Consecutive Numbers Microsoft Access Assigns Automatically to New Records option creates an AutoNumber type field.) Thus, you can accept the default values determined by the Table Wizard.

Oracle Ex. 1009, p. 157
Oracle v. DeBellis
IPR2024-00221



FIG. 4.15 Choosing a table name and how to determine the table's primary key.

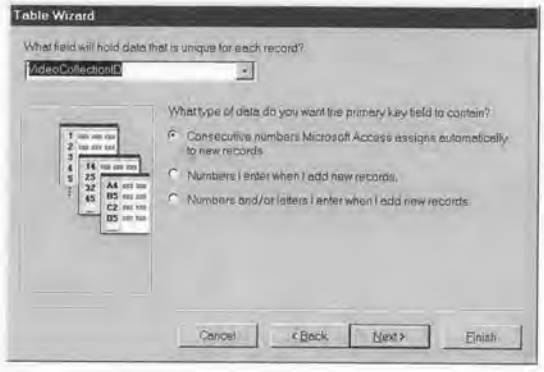


FIG. 4.16 The Table Wizard's primary-key dialog.

9. Choose the Next button to continue with the next stage of the table design definition process. The Table Wizard's relationships dialog (see Figure 4.17) appears only if other tables already exist in the database in which you're creating the new table. Because almost every database consists of two or more related tables, the Table Wizard gives you an opportunity to define the relationships between tables. By default, the new table has no relationship to other tables in the database. In this exercise, you don't add any table relationships to the new Video Collection table.

10. Choose the Next button to finish designing the table. Access displays the final step of the Table Wizard, shown in Figure 4.18.

11. Click the Modify the Table Design option, and then choose the Finish button to display your new table in Design mode, as shown in Figure 4.19.

12. When you have finished reviewing the design of your new table, click the Close Window button to close the table.

Oracle Ex. 1009, p. 158
Oracle v. DeBellis
IPR2024-00221

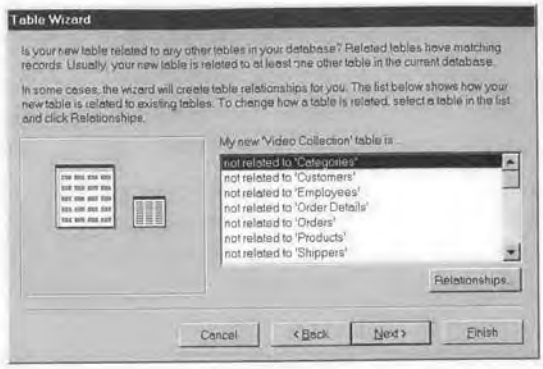


FIG. 4.17 Specifying relationships between fields in the new table and other tables.

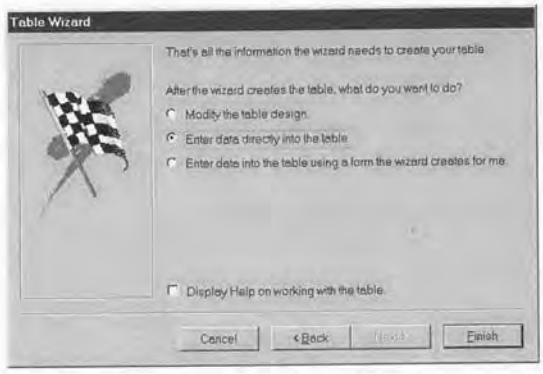


FIG. 4.18 The final Table Wizard dialog.

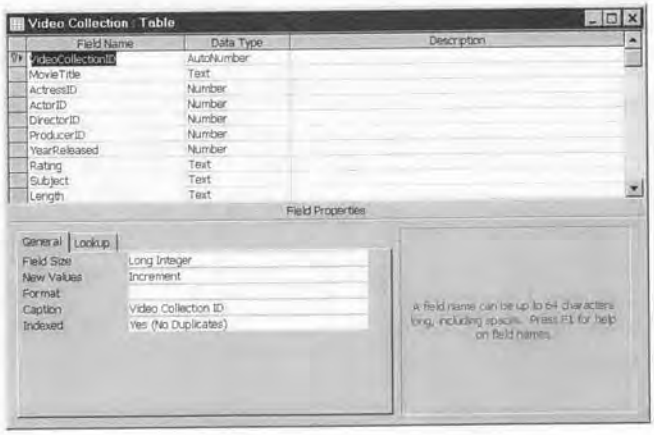


FIG. 4.19 The new Video Collection table in Design mode.

Oracle Ex. 1009, p. 159
Oracle v. DeBellis
IPR2024-00221

Tip

In any wizard, you can always redo a step by choosing the Back button until you return to the step that you want to redo.





If you want to delete the Video Collection table from Northwind.mdb, click the Show Database Window button of the toolbar, click the Table button if the Tables list is not open, and then click the Video Collection entry (or whatever you named your table) in the Table list to select (highlight) it. Press Delete, and click OK when the message box asks you to confirm the deletion. (You must close a table before you can delete it.)

Note

Creating tables based on the sample tables provided by the Table Wizard has limited usefulness in real-life business applications. In most cases, you import data from another database or spreadsheet application to create your Access tables. If you can't import the data, you probably need to define the fields of the tables to suit particular business needs. Thus, in the remainder of this chapter, you design a new database table by using the traditional method of manually adding fields to a blank table design and then specifying the properties of each field.

Creating a Table Directly in Datasheet View

If you're a complete database novice, and you're under pressure to create database tables immediately, Access enables you to create tables directly in Datasheet view. When you create a table in Datasheet view, Access creates an empty table with a default structure of 20 fields and 30 empty records. You then enter data directly into the table. When you save the table, Access analyzes the data you have entered and chooses a field type for each field that best matches the data you have entered. To create a table in Datasheet view, follow these steps:



1. Close any open tables to make the Database window active. Alternatively, click the Show Database Window button of the toolbar.





2. Click the Tables button of the Database window if it isn't selected, and then click the New button to display the New Table dialog (refer to Figure 4.13).
3. Select Datasheet View in the list, and click OK. Access displays an empty table, shown in Figure 4.20. By default, the new table contains 20 fields and 30 records.
4. Type your data into the table, pressing Enter to move to the next field. To see how creating a table in Datasheet view works, type in the following mailing address (taken from the Employees table of Northwind.mdb) for Nancy Davolio in the first record of the new table:

Ms. Nancy Davolio
Sales Representative
507 20th Ave. E.
Seattle WA 98122
USA

Oracle Ex. 1009, p. 160
Oracle v. DeBellis
IPR2024-00221

As you enter this address, divide the information into several fields. For example, type **Ms.**, then press Enter to move to the next field; type **Nancy**, then press Enter to move to the next field, and so on, until you have entered the complete address. The complete address should occupy nine fields when you are finished: title of respect, first name, last name, job title, street, city, state, ZIP code, and country.

5. When you have finished entering records into the table, click the Design View button on the toolbar to review the table's structure. Access displays the Save As dialog (shown in Figure 4.21) prompting you to enter a name for the new table. Alternatively, you can click the Close button to close the new table; in this case, Access first asks you to confirm saving changes to the table, and then asks if you want to create a primary key for the table, before prompting you for the table's name. For this exercise, answer No if asked to create a primary key; you learn more about primary keys later in this chapter.

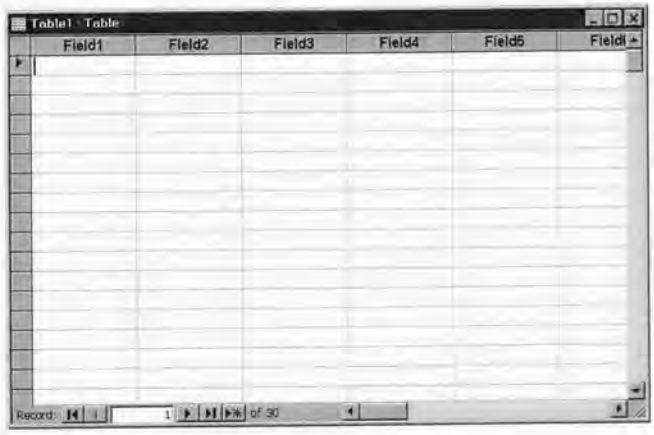


FIG. 4.20 Creating a new table in Datasheet view.



FIG. 4.21 The Save As dialog.

6. Enter a suitable name for your table in the Table Name text box of the Save As dialog, and then click OK to save the table. Access saves the table, including only fields in which you have entered data, and including only those records you entered. If you entered only one record with nine fields, then Access saves a table with nine fields and one record. After saving the table, Access displays the new table in Design view, as shown in Figure 4.22.

Oracle Ex. 1009, p. 161
Oracle v. DeBellis
IPR2024-00221

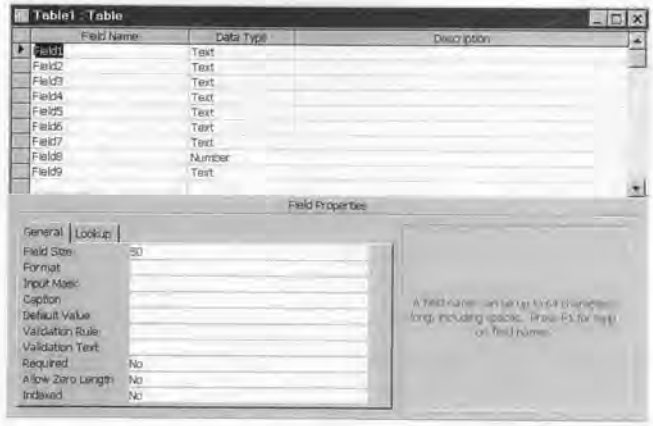


FIG. 4.22 The new table in Design view.

When you first save a table created in Datasheet view, Access analyzes the records you entered on a field-by-field basis. If all of the fields in a particular column of the table contain only numbers, then Access assigns a Number Data Type to that field column. If none of the numbers contain a decimal fraction, Access gives the Number field a Long Integer subtype, otherwise it assigns the field a Double subtype. If the fields in a single column of the table contain text, or a mixture of numbers and text (either in the same field or different fields of the same column), then Access assigns that field a Text Data Type. Text fields in the new table have the default Field Size setting of 50.

In Figure 4.22, notice that the fields are named in the format Field*N*, where *N* corresponds to the order of the field in the table from left to right. Although Access provided a default of 20 fields, only those fields in which you actually entered data are saved. Pay attention to the Data Type of Field8; this is the field in which the mailing address's ZIP code was entered. Because this field contained only digits, Access has assigned it the Number Data Type. Because the number entered in this field contained no decimal fraction, it has the Long Integer subtype.





If you want to delete the table you just created from Northwind.mdb, click the Show Database Window button of the toolbar, click the Tables tab if the Tables list isn't open, and then click the name of the table you just created (the name you entered in step 6 of the preceding instructions) to select it. Press Delete, and click OK when the message box asks you to confirm the deletion. (You must close a table before you can delete it.)

Note

Creating tables in Datasheet view has limited use. The table fields don't have meaningful names, so you'll almost always need to edit the table's structure to rename its fields. Further, as seen in the ZIP code example, Access may not always make the best determination for a field's Data Type—ZIP or postal code fields should almost always be of the Text type. Also, tables you create in Datasheet view cannot include OLE Object or Memo fields—if you want such fields in your table,

Oracle Ex. 1009, p. 162
Oracle v. DeBellis
IPR2024-00221

you need to alter its design. Because of the amount of additional field name and Data Type property editing you'll almost always need to perform for tables created in Datasheet view, you won't necessarily save much time over creating tables by using the traditional method of manually adding fields and setting their properties in a blank table design.

Adding a New Table to an Existing Database

The Northwind Traders database includes an Employees table that provides most of the information about the employees of the firm that is typical of personnel tables. This chapter explains how to add a table called Personnel Actions to the database. The Personnel Actions table is a record of hire date, salary, commission rate, bonuses, performance reviews, and other compensation-related events for employees. Because Personnel Actions is based on information in the Employees table, the first step of this process is to review the structure of the Employees table to see how you can use it with your new table. The structure of tables is displayed in Design mode. In the next chapter, "Entering, Editing, and Validating Data in Tables," you add validation rules to the Personnel Actions table and enter records in the table.

To open the Employees table in design mode, follow these steps:

1. Close any Access document windows that you have open, and then click the Tables tab button in the Database window to display the list of tables in the Northwind.mdb database.



2. Click Employees in the Database window, and then click the Design button. You also can open the Employees table by double-clicking the Database window entry and then clicking the Tables toolbar's Design View button.



3. The Design grid for the Employees table appears. Maximize the document window to the size of your Access window by clicking the document's Maximize button.

4. Close the Properties window, if it appears, by clicking its Close Window button. Alternatively, you can choose View, Properties.

   The View, Properties command toggles the visibility of the Table Properties window. The Properties icon to the left of the Properties menu choice has a sunken appearance to indicate that the Properties window is always visible in Table Design view.

At this point, your display resembles that shown in Figure 4.23.

The Table Design window displays the field names and the field data types and provides a third column for an optional description of each field in the table. This display is called a *grid* rather than a *datasheet* because the display doesn't contain data from a table. A scroll bar is provided, regardless of whether more fields exist in the table than the window can display. The Field Properties pane enables you to set additional properties of individual fields and briefly describes the purpose of each column of the grid and of the Field Properties entries as you select them. You cannot resize this pane.

Oracle Ex. 1009, p. 163
Oracle v. DeBellis
IPR2024-00221

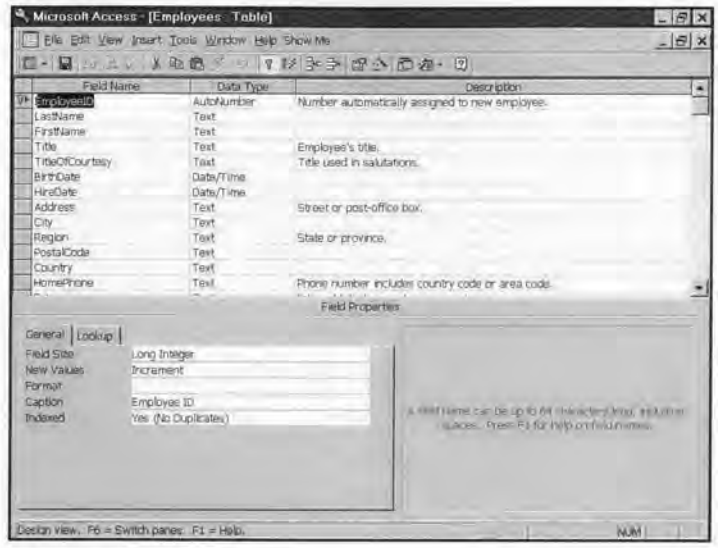


FIG. 4.23 The Table Design view of the Employees table.



One field is conspicuous by its absence: the Social Security number that most firms use in databases to identify their personnel. The EmployeeID field is an adequate substitute for the Social Security number for an example table because a unique sequential number (the AutoNumber field data type) is assigned to each employee. Click the Datasheet View button to display the data in the EmployeeID field, and then return to Design mode by clicking the Design View button.

Designing the Personnel Actions Table

Instead of adding fields for entries (such as salary, commission rate, and bonuses) to the Employees table, you should place employee remuneration data in a table of its own, for the following reasons:

- Multiple personnel actions are taken for individual employees over time. If you add these actions to records in the Employees table, you have to create many additional fields to hold an arbitrary number of personnel actions. If, for example, quarterly performance reviews are entered, you have to add a new field for every quarter to hold the review information. In this situation, flat-file managers encounter difficulties.
- You can categorize personnel actions by type so that any action taken can use a common set of field names and field data types. This feature makes the design of the Personnel Actions table simple.
- You can identify employees uniquely by their EmployeeID numbers. Therefore, records for entries of personnel actions can be related to the Employees table by an EmployeeID field. This feature eliminates the necessity of adding employee names and other information to the records in the Personnel Action table. You link the Employees table to the Personnel table by the EmployeeID field, and the two tables

Oracle Ex. 1009, p. 164
Oracle v. DeBellis
IPR2024-00221

are joined; they act as if they are a single table. Minimizing information duplication to only what is required to link the tables is your reward for choosing a relational, rather than a flat-file, database management system. (In an actual business's employee database, you would probably use the employee's Social Security number as the unique identifier for each employee, and as the link to the Personnel Actions table.)

- Personnel actions usually are considered confidential information and are made accessible only to a limited number of people. Although Access enables you to grant permission for others to view specific fields, restricting permission to view an entire table is simpler.

The next step is to design the Personnel Actions table. Chapter 23, "Exploring Relational Database Design and Implementation," discusses the theory of database design and the tables that make up databases. Because the Personnel Actions table has an easily discernible relationship to the Employees table, the theoretical background isn't necessary for this example.

Determining What Information the Table Should Include. Designing a table requires that you identify the type of information that the table should contain. Information associated with typical personnel actions might consist of the following items:

- *Important dates*. The date of hire and termination, if applicable, are important dates, but so are the dates when the employer adjusts salaries, changes commission rates, and grants bonuses. You should accompany each action with the date when it was scheduled to occur and the date when it actually occurred.
- *Types of actions*. Less typing is required if personnel actions are identified by a code character rather than a full-text description of the action. This feature saves valuable disk space, too. First-letter abbreviations used as codes, such as *H* for *hired*, *T* for *terminated*, and *Q* for *quarterly review*, are easy to remember.
- *Initiation and approval of actions*. As a rule, the employee's supervisor initiates a personnel action, and the supervisor's manager approves it. Therefore, the table should include the supervisor's and manager's EmployeeID number.
- *Amounts involved*. Salaries are assumed to be based on monthly payment, bonuses are paid quarterly with quarterly performance reviews, and commissions are paid on a percentage of sales made by the employee.
- *Performance rating*. Rating employee performance by a numerical value is a universal, but somewhat arbitrary, practice. Scales of 1 to 9 are common, with exceptional performance ranked as 9 and candidacy for termination as 1.
- *Summaries and comments*. The table should provide for a summary of performance, explanation of exceptionally high or low ratings, and reasons for adjusting salaries or bonuses.

If you are involved in personnel management, you probably can think of additional information that the table might include, such as accruable sick leave and vacation hours

Oracle Ex. 1009, p. 165
Oracle v. DeBellis
IPR2024-00221

per pay period. The Personnel Actions table is just an example; it isn't meant to add full-scale human resources development capabilities to the database. The limited amount of data described serves to demonstrate several uses of the new table in this and subsequent chapters.

Assigning Information to Fields. After you determine the types of information, called *data entities* or *entities*, to include in the table, you must assign each data entity to a field of the table. This process involves choosing a field name that must be unique within the table. Table 4.7 lists the candidate fields for the Personnel Actions table. Candidate fields are written descriptions of the fields proposed for the table. Data types have been assigned from those listed in Table 4.8 in the following section.

Note

Although the table name contains a space, the field names of the Personnel Actions table do not contain spaces (as shown in Table 4.8). As mentioned earlier in this book, including spaces in table names or field names is not a good database design practice. In this case, the table names include a space to demonstrate the special rule (enclosing the name within square brackets) that you must observe when referring to object names that include spaces. The Northwind Traders sample database includes spaces in many of its table names, so the use of spaces here is consistent with the other tables in the database.

Table 4.7 Candidate Fields for the Personnel Actions Table

Field Name	Data Type	Description
paID	Number	The employee to whom the action applies. paID numbers are assigned based on the EmployeeID field of the Employee table (to which the Personnel Actions table is linked).
paType	Text	Code for the type of action taken: H is for hired; C, commission rate adjustment; Q, quarterly review; Y, yearly review; S, salary adjustment; B, bonus adjustment; and T, terminated.
paInitiatedBy	Number	The EmployeeID number of the supervisor who initiates or is responsible for recommending the action.
paScheduledDate	Date/Time	The date when the action is scheduled to occur.
paApprovedBy	Number	The EmployeeID number of the manager who approves the action proposed by the supervisor.
paEffectiveDate	Date/Time	The date when the action occurred. The effective date remains blank if the action has not occurred.
paRating	Number	Performance on a scale of 1–9, with higher numbers indicating better performance. A blank indicates no rating; 0 is reserved for terminated employees.

Oracle Ex. 1009, p. 166
Oracle v. DeBellis
IPR2024-00221

Field Name	Data Type	Description
paAmount	Currency	The salary per month, the bonus per quarter, or commission rate as a percent of the amount of the order, expressed as a decimal fraction.
paComments	Memo	Abstracts of performance reviews and comments on actions proposed or taken. The comments can be of unlimited length. The supervisor and manager can contribute to the comments.

Note

Use distinctive names for each field. This example precedes each field name with the abbreviation *pa* to identify the field with the Personnel Actions table. A common practice is to use similar names for fields that contain identical data but are located in different tables. Because of the way that Access uses field names in expressions for validating data entry and calculating field values (discussed later in this chapter and in Chapter 9, "Understanding Operators and Expressions in Access"), the best practice is to assign related, but distinctive, names to such fields.

Creating the Personnel Actions Table

Now you can put to work what you have learned about field names, data types, and formats by adding the Personnel Actions table to the Northwind Traders database. Table 4.8 shows the field names, taken from Table 4.7, and the set of properties that you assign to the fields. The text in the Caption property column substitutes for the Field Name property that is otherwise displayed in the field header buttons.

Table 4.8 Field Properties for the Personnel Actions Table

Field Name	Caption	Data Type	Field Size	Format
paID	ID	Number	Long Integer	General Number
paType	Type	Text	1	>@ (all uppercase)
paInitiatedBy	Initiated By	Number	Long Integer	General Number
paScheduledDate	Scheduled	Date/Time	N/A	Short Date
paApprovedBy	Approved By	Number	Long Integer	General Number
paEffectiveDate	Effective	Date/Time	N/A	Short Date
paRating	Rating	Number	Integer	General Number
paAmount	Amount	Currency	N/A	#,##0.00#
paComments	Comments	Memo	N/A	(None)

You must set the paID field's Field Size property to the Long Integer data type, although you might not expect Northwind Traders to have more than the 32,767 employees that

Oracle Ex. 1009, p. 167
Oracle v. DeBellis
IPR2024-00221

an integer allows. You must use the Long Integer data type because the AutoNumber field data type of the Employees table's EmployeeID field is a Long Integer. The section "Enforcing Referential Integrity" later in this chapter explains why paID's data type must match that of the Employees table's EmployeeID number field.

To add the new Personnel Actions table to the Northwind Traders database, complete the following steps:

1. Close the Employees table, if it is open, by clicking the Close Window button to make the Database window active.



2. Click the Tables tab of the Database window, if it isn't selected, and then click the New button. Select Design View in the New Table dialog and click OK. Access enters Design mode and opens a blank grid where you enter field names, data types, and optional comments. By default, Access selects the grid's first cell.

3. Enter **paID** as the first field name. Press Enter to accept the field name. The caret moves to the Data Type column; Access adds the default field type, Text.

4. Press F4 to open the Data Type list. You use the function keys rather than the mouse because your entries are from the keyboard.

5. Use the arrow keys to select the Number data type, and press Enter to accept your selection (see Figure 4.24).

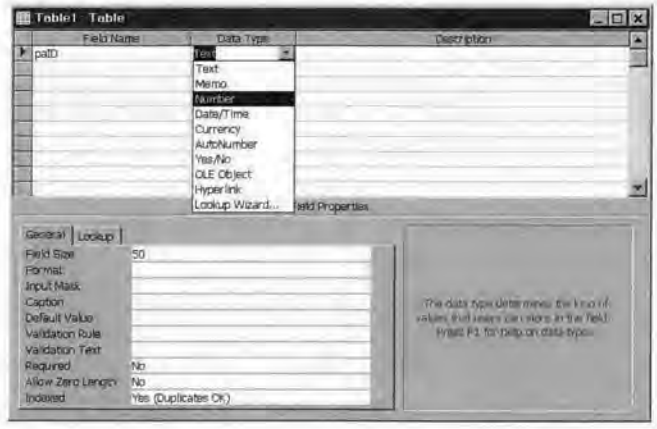


FIG. 4.24 Entering the paID field's Data Type.

6. Press F6 to move to the Field Properties window's Field Size text box. Access has already entered Long Integer as the value of the default Field Size property. To learn more about the Field Size property, press F1 for help.

   Whenever you create a new Number type field, Access enters Long Integer in the Field Size property as the default. Because the paID field should be a Long Integer, you don't need to set the Field Size property for this field, and can skip to step 8; continue with step 7 when you enter the other fields from Table 4.8.

Oracle Ex. 1009, p. 168
Oracle v. DeBellis
IPR2024-00221

7. For Number data types, press F4 to open the Field Size list. Select from the list the appropriate field size value for the field, and press Enter.
8. Press the down arrow to select the Format text box. You can press F1 for context-sensitive help on the Format property.
9. Press F4 to open the Format list, select General Number from the list, and press Enter (see Figure 4.25).
10. Press the down-arrow key three times, bypassing the Decimal Places and Input Mask properties, and select the Caption text box. You skip the Decimal Places property; Long Integers cannot have decimal fractions, so Decimal Places can remain set to Auto. You skip the Input Mask property because this field doesn't need an input mask.
11. Enter **ID** as the caption and press Enter. ID is used as the Caption property to minimize the column width necessary to display the paID number.



FIG. 4.25 Assigning the General Number format to the paID field.

12. Press F6 to return to the Table Design grid. The caret is located in the Description column. You complete the remaining properties for each field after completing the basic properties shown in Table 4.8.
13. You use descriptions to create prompts that appear in the status bar when you are adding or editing records in Run mode's Datasheet view. Although descriptions are optional, a good database design practice is to enter the field's purpose if its use isn't obvious from its Field Name or Caption property. You can skip the Caption property entries for now. After completing the basic steps described here, refer to Table 4.8 and enter the captions as a group.
14. Press Enter to move the caret to the first cell of the next row of the grid.
15. Repeat steps 3–13, entering the values shown in Table 4.8 for each of the eight remaining fields of the Personnel Action table. N/A (not applicable) means that the entry in Table 4.8 doesn't apply to the field's data type.

Oracle Ex. 1009, p. 169
Oracle v. DeBellis
IPR2024-00221

Your Table Design grid should now look similar to the one shown in Figure 4.26. You can double-check your properties entries by selecting each field name with the arrow keys and reading the values shown in the properties text boxes of the Field Properties window.



FIG. 4.26 The initial design of the Personnel Actions table.



Click the Datasheet toolbar button to return to Datasheet view in Run mode to view the results of your work. You receive the `Must save table first. Save now?` message. Click OK, and a Save As dialog appears requesting that you give your table a name and suggesting the default table name, Table1. Type **Personnel Actions**, as shown in Figure 4.27, and press Enter or click OK.



FIG. 4.27 The Save As dialog for naming the Personnel Actions table.

At this point, Access displays a dialog informing you that the new table does not have a primary key. You add primary keys to the Personnel Actions table later in this chapter, so click No in this dialog.

Your table appears in Datasheet view, with its first default record. To view all the fields of your new table, narrow the field name header buttons by dragging to the left the right vertical bar that separates each of the headers. When you finish adjusting your fields' display widths, the Personnel Actions table appears in Datasheet view as shown in Figure 4.28. Only the empty tentative append record (a new record that Access will add to your table if you enter values in the cells) is present. You have more property values to add to your Personnel Actions table, so don't enter data in the tentative append record at this point.



FIG. 4.28 The tentative append record of the Personnel Actions table.

Oracle Ex. 1009, p. 170
Oracle v. DeBellis
IPR2024-00221

Setting Default Values of Fields

Access 1.x assigned your fields' default values, such as 0 for Number and Currency fields and No for Yes/No fields. Access 2.0 did not assign default values to Number and Yes/No fields automatically. Access 95 and Access 97 cover the middle ground by assigning Number and Currency fields a default value of 0; all other field types are empty by default. (Notice that the tentative append record in Figure 4.28 has zeros entered in all the Number and Currency fields.) In all versions of Access, Text, Memo, and Date fields are empty by default. You can save data-entry time by establishing your own default values for fields; in some cases, Access 97's default values for Number and Currency fields may be inappropriate, and you'll need to change them. Table 4.9 lists the default values for the Personnel Actions table's fields.

Table 4.9 Default Field Values for the Personnel Actions Table

Field Name	Default Value	Comments
paID	No entry	0 is not a valid Employee ID number, so you should remove Access's default.
paType	Q	Quarterly performance reviews are the most common personnel action.
paInitiatedBy	No entry	0 is not a valid Employee ID number.
paScheduledDate	Date()	The expression to enter today's (DOS) date.
paApprovedBy	No entry	0 is not a valid Employee ID.
paEffectiveDate	Date()+28	Today's date plus 28 days. (The date is obtained from your computer's system clock.)
paRating	No entry	In many cases, a rating does not apply. A 0 rating is reserved for terminated employees.
paAmount	No entry	If a salary, bonus, or commission has no change, no entry should appear. 0 would indicate no salary, for example.
paComments	No change	For now, Access's default is adequate.

If you don't enter anything in the Default Value text box, you create a `Null` default value. You can use `Null` values for testing whether a value has been entered into a field. Such a test can ensure that the user has entered required data. (The following section discusses this subject.) The `Date()+28` default is an *expression* that returns the date (according to your computer's clock) plus four weeks. You use expressions to enter values in fields, make calculations, and perform other useful duties, such as validating data entries. Expressions are discussed briefly in the next section and in greater detail in Chapter 9, "Understanding Operators and Expressions in Access." Expressions that establish default values always are preceded by an equal sign.

Oracle Ex. 1009, p. 171
Oracle v. DeBellis
IPR2024-00221

To assign the new default values from those of Table 4.9 to the fields of the Personnel Actions table, complete these steps:

1. Change to design mode by choosing View, Design View. Select the paID field.
2. Press F6 to switch to the Field Properties window, and then move the caret to the Default Value text box. Press Delete to clear the text box.
3. Press F6 to switch back to the Table Design grid. Move to the next field and press F6 again.
4. Create the default values for the eight remaining fields from the entries shown in Table 4.9, repeating steps 2–4. For example, after selecting the Default Value text box for the paType field, you would type **Q** to set the default value. Enter **=Date()** for the paScheduledDate field and **=Date()+28** for the paEffectiveDate Date field. Delete any default values that might appear in the other fields that call for no entry in Table 4.9.
5. After completing your default entries, choose View, Datasheet View to return to Run mode. A dialog appears requesting that you confirm your changes. Click OK. The Personnel Actions table now appears in Datasheet view with the new default entries that you have assigned, as shown in Figure 4.29.



FIG. 4.29 The first record of the Personnel Actions table with the new default values.

Working with Relations, Key Fields, and Indexes

Your final tasks before adding records to the Personnel Actions table are to determine the relationship between Personnel Actions and an existing table in the database, assign a primary-key field, and add indexes to your table.

Establishing Relationships Between Tables

Relationships between existing tables and your new table determine the field used as the new table's primary key. The following four possibilities exist for relationships between tables:

- *One-to-one* relationships require that the key field's value in only one record in your new table matches a single corresponding value of the related field in the existing table. In this case, the key field in your new table must be unique; duplicate values aren't allowed in the key field. A key field that refers to another table's primary-key field is called a *foreign-key field*. A one-to-one relationship is the equivalent of a table that contains all the fields of the existing table and the new table. Tables with one-to-one relationships are uncommon.

Oracle Ex. 1009, p. 172
Oracle v. DeBellis
IPR2024-00221

- *Many-to-one* relationships allow your new table to have more than one value in the key field corresponding to a single value in the related field of the existing table. In this case, duplicate key field values are allowed because the field is a foreign-key field. Many-to-one relationships are the most common type; the capability to create many-to-one relationships is the principal reason for choosing a relational system, rather than a flat-file application, to manage your databases.
- *One-to-many* relationships require that your new table's primary-key field be unique, but the values in the foreign-key field of the new table can match many entries in the related field of the existing database. In this case, the related field of the existing database has a many-to-one relationship with the primary-key field of the new database.
- *Many-to-many* relationships are free-for-alls in which no unique relationship exists between the key fields in the existing table or the new table, and both of the tables' foreign-key fields contain duplicate values. Many-to-many relationships are created through *relation tables* that contain only foreign-key fields.

Keep in mind that the many-to-one and one-to-many relationships apply to how your new table relates to an existing table. When viewed from the existing table's standpoint, the relationships to your new table are one-to-many and many-to-one, respectively. Chapter 23, "Exploring Relational Database Design and Implementation," explains the four types of relations more comprehensively.

▶▶ See "Types of Relationships," p. 809



Many entries in the Personnel Actions table may apply to a single employee whose record appears in the Employees table. A record is created in Personnel Actions when the employee is hired, and a record is created for each quarterly and yearly performance review. Also, any changes made to bonuses or commissions other than as the result of a performance review are entered, and employees may be terminated. Over time, the number of records in the Personnel Actions table is likely to be greater by a factor of 10 or more than the number of records in the Employees table. Thus, the records in the new Personnel table have a many-to-one relationship with the records in the Employees table. Establishing the relationships between the new and existing tables when you create the new table enables Access to reestablish the relationship automatically when you use the tables in queries, forms, and reports.

Access requires that the two fields participating in the relationship have exactly the same data type. In the case of the Number field data type, the Field Size property of the two fields must be identical. You cannot, for example, create a relationship between an AutoNumber type field (which uses a Long Integer data type) and a field containing Byte, Integer, Single, Double, or Currency data. On the other hand, Access enables you to relate two tables by text fields of different lengths. Such a relationship, if created, can lead to strange behavior when you create queries, which is the subject of Part II, "Querying for Specific Information." As a rule, the relationships between text fields should use fields of the same length.

Oracle Ex. 1009, p. 173
Oracle v. DeBellis
IPR2024-00221

Access 97 uses a graphical Relationships window to display and create the relationships among tables in a database. To establish the relationships between two tables using Access's Relationships dialog, follow these steps:

1. Close the Personnel Actions table by clicking the Close Window button. If the Employees table is open, close it. You cannot create or modify relationships between open tables.



2. If the Database window isn't the active window (indicated by a colored title bar), click the Database window, click the Show Database Window button of the toolbar, or choose Window, 1 Northwind : Database. As many as nine of the windows for database objects that you have opened appear as numbered choices in the Windows menu. (The Database window is always number 1.) Before you can establish relationships, the Database window must be active.



3. Click the Relationships button of the toolbar or choose Tools, Relationships. The Relationships window for the Northwind Traders database appears (see Figure 4.30). This window displays all the tables for which relationships are defined.



4. Click the Show Table button of the toolbar or choose Relationships, Show Table. The Show Table dialog shown in Figure 4.31 appears.

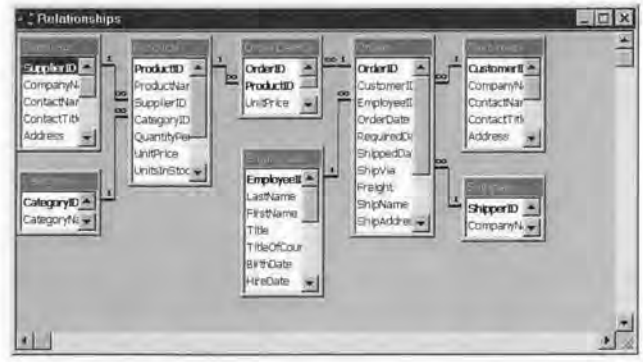


FIG. 4.30 The Relationships window.



FIG. 4.31 The Show Table dialog, used to add tables to the Relationships window.

Oracle Ex. 1009, p. 174
Oracle v. DeBellis
IPR2024-00221

5. Add the Personnel Actions table to the Relationships window by double-clicking the Personnel Actions entry in the Tables list or by clicking the entry to select it and then choosing the Add button. Choose the Close button to close the Show Table dialog.

6. The relationship of the Personnel Actions table with the Employees table is based on the Personnel Actions table's paID field and the Employees table's EmployeeID field. Click the Employees table's EmployeeID field and, holding the left mouse button down, drag the field to the Personnel Actions table's paID field. Release the left mouse button to drop the field symbol on the paID field. When you drag-and-drop a new relationship, the Relationships dialog appears (see Figure 4.32).

Note

The sequence of the drag-and-drop operation to create a new relationship is important. Drag the field from the *one* side of a one-to-many relationship and drop it on the *many* side. This sequence ensures that the primary (or base) table for the *one* side of the relationship appears in the Table/Query list and that the table for the *many* side appears in the Related Table/Query list. If you reverse the relationships (creating a many-to-one relationship) and attempt to enforce referential integrity, you receive an error message in the final step of the process when you attempt to create the relationship.

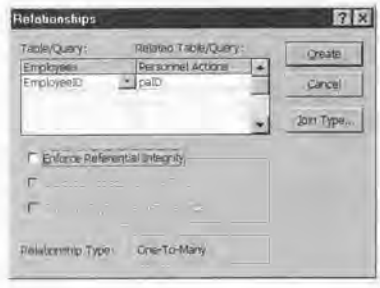


FIG. 4.32 Defining a relationship with the Relationships dialog.

7. Choose the Join Type button to display the Join Properties dialog shown in Figure 4.33. You want to create a one-to-many join between the Employees table's EmployeeID field (the *one* side) and the Personnel Actions table's paID field (the *many* side). Thus, you want to include *all* Personnel Actions records for a single employee. To do so, choose option 3 in the Join Properties dialog. Click OK to close the dialog and return to the Relationships dialog.

8. The Relationships dialog offers the Enforce Referential Integrity check box so that you can specify that Access perform validation testing and accept entries in the paID field that correspond to values for the Employees table's EmployeeID field. This process is called *enforcing* (or maintaining) referential integrity. The following section discusses referential integrity. The relationship between these two tables requires enforced referential integrity, so make sure that you select this check box. The Relationships dialog now appears as shown in Figure 4.34.

Oracle Ex. 1009, p. 175
Oracle v. DeBellis
IPR2024-00221



FIG. 4.33 Choosing the type of join for the Personnel Actions and Employees tables.

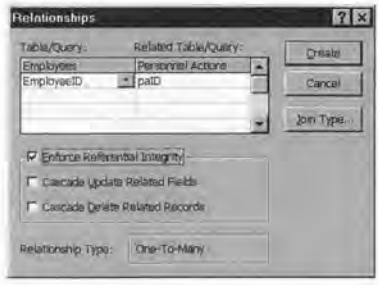


FIG. 4.34 The Relationships dialog entries for a one-to-many relationship with referential integrity enforced.

> **Note**
>
> Access 97 automatically maintains referential integrity of tables by providing check boxes that you can select to cause cascading updates to, and cascade deletions of, related records when the primary table changes. The following section discusses cascading updates and deletions. Access enables the cascade check boxes only if you elect to enforce referential integrity.

9. Click the Create button to accept the new relationship and display it in the Relationships window as shown in Figure 4.35.
10. Click the Close Window button to close the Relationships window and return to the Database window. Click Yes when asked to confirm that you want to save the layout changes to the Relationships diagram.

Access uses the relationship that you have created when you create queries and design forms and reports that require data in the Personnel Actions table. Access does not require that the related table be indexed.

Oracle Ex. 1009, p. 176
Oracle v. DeBellis
IPR2024-00221

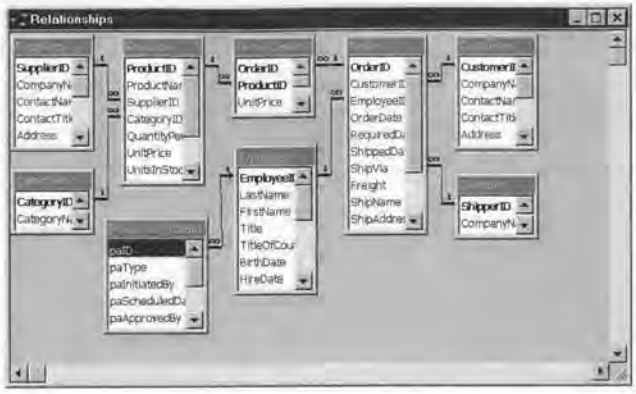


FIG. 4.35 The Relationships window with the new Personnel Actions relationship added.

Enforcing Referential Integrity. The capability to enforce referential integrity automatically is an important feature of Access; few other PC relational database managers include this feature. Referential integrity prevents the creation of orphan records that have no connection to a primary table. An example of an orphan record is a record for a personnel action for paID 10 when you have records in the Employees file for employees numbered only 1 through 9. You could not know who employee 10 is until you enter the next employee hired. Then the orphan record, intended for some other employee, is linked, improperly, to the new employee's record.

How Referential Integrity Is Enforced. Referential integrity enforcement prevents you from deleting or modifying values of a primary table's record on which related records depend. If you terminate an employee and then try to delete the employee's record from the Employees table, Access prevents you from doing so. Access displays a message box informing you that you must delete all records related to the primary table's record before you can delete the primary record. You can't change a value in the Employees table's EmployeeID field because the field data type is AutoNumber. If the data types are such that you can change the value of an EmployeeID on which related records depend, however, Access also displays a warning message.

Similarly, if you attempt to change an employee ID value in the paID field of the Personnel Actions table to a value that does not exist in the Employees table's EmployeeID field, you also incur an error message. Thus, enforcing referential integrity eliminates the need to validate entries in the paID field using the Validation Rule property. With referential integrity enforced, Access automatically ensures that the value you enter corresponds to a valid EmployeeID value when you save the new or edited record.

Cascading Updates and Deletions. Prior to Access 2.0, you had to write macros or Access VBA code to implement the series of actions required to maintain referential integrity. For example, to delete a record in the Employees table, you had to run a test to determine whether related records existed in any other table that depended on the EmployeeID field, and then delete the dependent records. If you wanted to change the CustomerID of a record in the Customers table, the situation became more complex:

Oracle Ex. 1009, p. 177
Oracle v. DeBellis
IPR2024-00221

You couldn't change the related records to a new CustomerID that wasn't yet in the table, nor could you change the CustomerID in the primary table because it had dependent records. You could resolve this dichotomy by a variety of methods, none of which were simple.

With Access 97's cascading deletion and cascading update options for tables with enforced referential integrity, maintaining referential integrity is a simple process: Just check the Cascade Update Related Fields and Cascade Delete Related Records check boxes. Access 97 does all the work for you.

Note

Automatically enforcing referential integrity is usually, but not always, a good database design practice. An example of where you would *not* want to employ cascade deletions is between the EmployeeID fields of the Orders and Employee tables. If you terminate an employee and then attempt to delete the employee's record, you might accidentally choose to delete the dependent records in the Orders table. Deleting records in the Orders table could have serious consequences from a marketing and accounting standpoint. (In the real world, however, you probably would not delete a terminated employee's record.)

Selecting a Primary Key

You do not need to designate a primary-key field for a table that is never used as a primary table. A *primary table* contains information representing a real-world object, such as a person or an invoice, and only one record uniquely associated with that object. The Personnel Actions table can qualify as a primary table because it identifies an object—in this case, the equivalent of a paper form representing the outcome of two actions: initiation and approval. Personnel Actions, however, probably would not be used as a primary table in a relationship with another table.

Using a key field is a simple method of preventing the duplication of records in a table. Access requires that you specify a primary key if you want to create a one-to-one relationship or to update two or more tables at the same time. Chapter 10, "Creating Multitable and Crosstab Queries," covers this subject.

The primary table participating in relations that you set with the Relationships dialog must have a primary key. Access considers a table without a primary-key field to be an oddity; therefore, when you make changes to the table and return to Design view, you might see a message stating that you haven't created a key field. (Access 97 asks you only once whether you want to add a primary-key field.) Related tables can have primary-key fields and often do. A primary-key field is useful to prevent the accidental addition of duplicate records.

You can create primary keys on more than one field. In the case of the Personnel Actions table, a primary key that prevents duplicate records must consist of more than one field, because more than one personnel action for an employee can be scheduled or approved on the same date. If you establish the rule that no more than one type of personnel action for an employee can be scheduled for the same date, you can create a primary key that consists of the paID, paType, and paScheduledDate fields. When you create a

Oracle Ex. 1009, p. 178
Oracle v. DeBellis
IPR2024-00221

primary key, Access creates an index based on the primary key. The next section and Chapter 23, "Exploring Relational Database Design and Implementation," discuss indexes in detail.

To create a multiple-field primary key and index for the Personnel Actions table, follow these steps:

1. Open the Personnel Actions table from the Database window in Design view.



2. Click the selection button for the paID field.
3. Hold down the Ctrl key and click the selection button for the paType field. In most instances, when you hold down Ctrl and click a selection button, you can make multiple selections.
4. Hold down Ctrl and click the selection button for the paScheduledDate field.

   If you accidentally select one of the other fields, click the field's selection button again to deselect it.

5. Click the Primary Key button on the toolbar. Symbols of keys appear in each of the selected fields, indicating their inclusion in the primary key (see Figure 4.36).



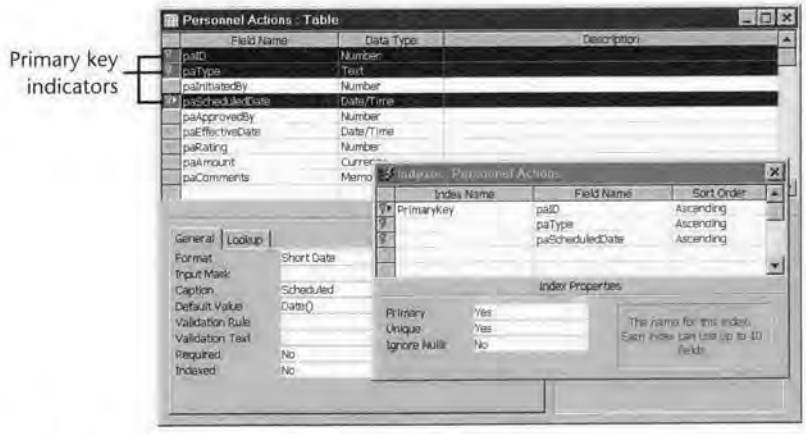


FIG. 4.36 Setting a multiple-field primary key for the Personnel Actions table.

6. To determine the sequence of the fields in the primary key, click the toolbar's Index button to display the Indexes window as shown in Figure 4.36.



   In Access, you can create multiple-field primary keys and indexes with fields of different data types. The capability to concatenate different data types to form an index instruction or a string is the result of Access's Variant data type, which is discussed in Chapter 28, "Writing Visual Basic for Applications Code."

You now have a multiple-field primary key and a corresponding index to the Personnel Actions table that precludes the addition of records that duplicate records with the same primary key.

Oracle Ex. 1009, p. 179
Oracle v. DeBellis
IPR2024-00221

Adding Indexes to Tables

Although Access creates an index on the primary key, you might want to create an index on some other field or fields in the table. Indexes speed searches for records that contain specific types of data. You might want to find all personnel actions that occurred in a given period and all quarterly reviews for all employees in paScheduledDate sequence, for example. If you have many records in the table, an index speeds up the searching process. A disadvantage of multiple indexes is that data entry operations are slowed by the time that it takes to update the additional indexes. You can create as many as 32 indexes for each Access table, and five of those can be of the multiple-field type. Each multiple-field index can include as many as 10 fields.

To create a single-field index for the Personnel Actions table based on the paEffectiveDate field, and a multiple-field index based on the paType and the paScheduledDate fields, follow these steps:

1. Select the paEffectiveDate field by clicking its selection button.
2. Select the Indexed text box in the Field Properties window.
3. Open the Indexed drop-down list by clicking the arrow button or pressing F4. The list appears as shown in Figure 4.37.

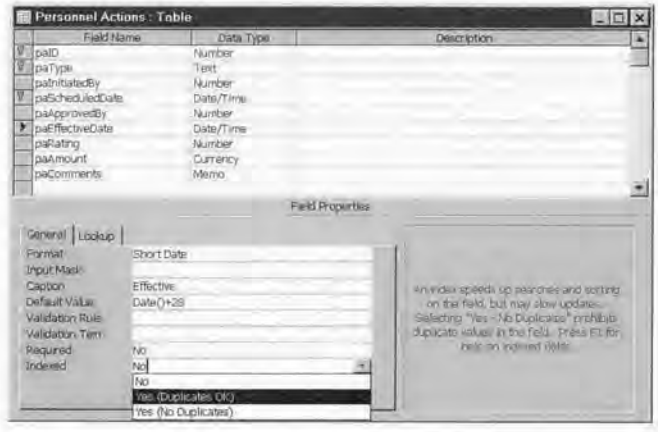


FIG. 4.37 Creating a single-field index on the paEffectiveDate field.

4. In this case, duplicate entries are acceptable, so click Yes (Duplicates OK) and close the list. You can create only a single-field index by using this method.



5. Click the Indexes button if the Indexes window is not open. The Primary Key and paEffectiveDate indexes already created appear in the list boxes. Enter **paType/Date** as the name of the composite index, and then select **paType** in the Field Name drop-down list; move the caret to the next row of the Field Name column and select **paScheduledDate** to create a multiple-field index on these two fields, as shown in Figure 4.38.

Oracle Ex. 1009, p. 180
Oracle v. DeBellis
IPR2024-00221

6. Click the Datasheet View button to return to Run mode. Click OK when the message box asks whether you want to save your design changes. A message in the status bar indicates that Access is creating the new indexes as you leave Design mode.

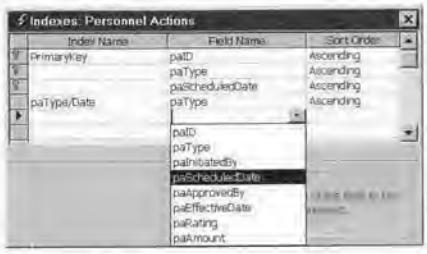


FIG. 4.38 Creating a multiple-field index.

You now have three indexes for the Primary Key table: the index automatically created for the primary key, the single-key index on paEffectiveDate, and the multiple-key index on paType and paScheduledDate.

Altering Fields and Relationships

When you are designing your own database, you often discover that you must alter the original choices that you made for the sequence of fields in a table, data types, or relationships between tables. One of the reasons for adding substantial numbers of records to tables during the testing process is to discover any changes that are necessary before putting the database into daily use.

You can change formats, change validation rules and text, change lengths of Text fields, and make other minor modifications to the table by changing to Design mode, selecting the field to modify, and making the changes in the Properties boxes. Changing data types can cause a loss of data, however, so be sure to read the section "Changing Field Data Types and Sizes" later in this chapter before you attempt to make such changes. Changing relationships between tables is considered a drastic action if you have entered a substantial amount of data, so this subject also is covered in a later section, "Changing Relationships Between Tables."

Note

Avoid changing a field name if you have created data entry forms or reports that use the data in the field. Although Access performs many operations automatically, it does not change the field names that you have assigned to text boxes and other objects in forms or to the groups in reports. The time to finalize field names is while creating your tables; a bit of extra thought at this point saves hours of modification after you are well into creation of a complex application.

Rearranging the Sequence of Fields in a Table

If you are manually entering historical data in Datasheet view, you might find that the sequence of entries isn't optimum. You might, for example, be entering data from a

Oracle Ex. 1009, p. 181
Oracle v. DeBellis
IPR2024-00221

printed form with a top-to-bottom, left-to-right sequence that doesn't correspond to the left-to-right sequence of the corresponding fields in your table. Access makes rearranging the order of fields in tables a matter of dragging and dropping fields where you want them. You can choose whether to make the revised layout temporary or permanent when you close the table.

To rearrange the fields of the Personnel Actions table, follow these steps:



1. Click the Datasheet View button. This is the only table design change that you can implement in Access's Datasheet view.
2. Click the field name button of the field that you want to move. This action selects the field name button and all the field's data cells.
3. Hold down the left mouse button while over the field name button. The mouse pointer turns into the drag-and-drop symbol, and a heavy vertical bar marks the field's far-left position. Figure 4.39 shows the paScheduledDate field being moved to a position immediately to the left of the paEffectiveDate field.

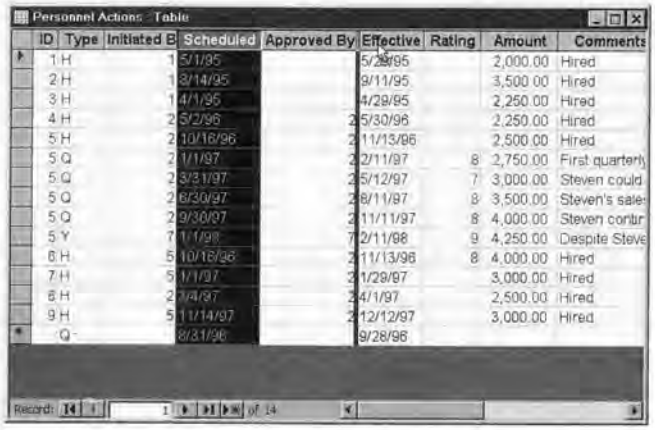


FIG. 4.39 Dragging a field to a new position in Datasheet view.

4. Move the mouse pointer and vertical bar combination to the new position for the selected field and release the mouse button. The field assumes the new position shown in Figure 4.40.
5. When you close the Personnel Actions table, you see the familiar Save Changes message box. To make the modification permanent, click OK; otherwise, click No.

You can reposition fields in Design View by clicking the select button of the row of the field that you want to move and then dragging the row vertically to a new location. Changing the position of a table's field doesn't change any of the field's other properties.

Changing Field Data Types and Sizes

You might have to change a field data type as the design of your database develops, or if you import tables from another database, a spreadsheet, or a text file. If you import tables, the data type automatically chosen by Access during the importation process

Oracle Ex. 1009, p. 182
Oracle v. DeBellis
IPR2024-00221

probably won't be what you want, especially with Number fields. Chapter 7, "Linking, Importing, and Exporting Tables," discusses importing and exporting tables and data from other applications. Another example of altering field properties is changing the number of characters in fixed-length Text fields to accommodate entries that are longer than expected, or converting variable-length Text fields to fixed-length fields.

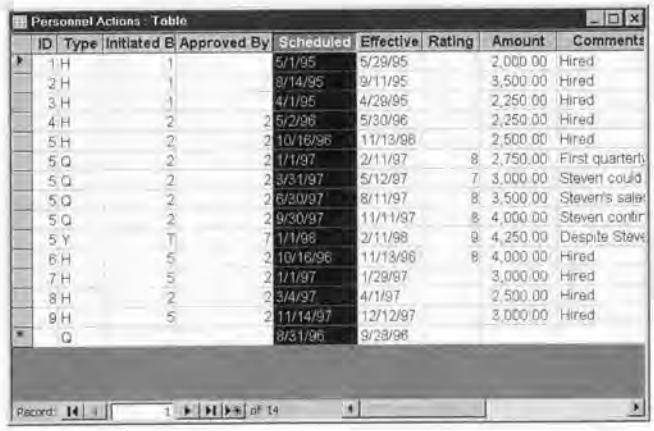


FIG. 4.40 The paScheduledDate field dropped into a new position.

Caution

Before making changes to the field data types of a table that contains substantial amounts of data, back up the table by copying or exporting it to a backup Access database. If you accidentally lose parts of the data contained in the table, such as decimal fractions, while changing the field data type, you can import the backup table to your current database. Chapter 7, "Linking, Importing, and Exporting Tables," covers the simple and quick process of exporting Access tables. After creating a backup database file, you can copy a table to Windows Clipboard and then paste the table to the backup database. The section "Copying and Pasting Tables" later in this chapter discusses copying and pasting tables to and from the Clipboard.

Numeric Fields. Changing a data type to one that requires more bytes of storage is, in almost all circumstances, safe. You do not sacrifice your data's accuracy. Changing a numeric data type from Byte to Integer to Long Integer to Single and, finally, to Double does not affect your data's value because each change, except for Long Integer to Single, requires more bytes of storage for a data value. Changing from Long Integer to Single and Single to Currency involves the same number of bytes and decreases the accuracy of the data only in exceptional circumstances. The exceptions can occur when you are using very high numbers or extremely small decimal fractions, such as in some scientific and engineering calculations.

On the other hand, if you change to a data type with fewer data bytes required to store it, Access might truncate your data. If you change from a fixed-point format (Currency) or floating-point format (Single or Double) to Byte, Integer, or Long Integer, any decimal fractions in your data are truncated. *Truncation* means reducing the number of digits in a number to fit the new Field Size property that you choose. If you change a numeric data

Oracle Ex. 1009, p. 183
Oracle v. DeBellis
IPR2024-00221

type from Single to Currency, for example, you might lose your Single data in the fifth, sixth, and seventh decimal places (if any exists) because Single provides as many as seven decimal places and Currency provides only four.

You cannot convert any type of field to an AutoNumber-type field. You can use the AutoNumber field only as a record counter; the only way that you can enter a new value in an AutoNumber field is by appending new records. You cannot edit an AutoNumber field. When you delete a record in Access, the AutoNumber values of the higher-numbered records are *not* reduced by 1. Sequential Access AutoNumber field values are assigned to records in the order in which the records were entered, not the order of the primary key. Access also provides a random assignment of AutoNumber field values, primarily for use with replicated databases. The randomly assigned AutoNumber values help ensure that, when replicated database copies are reconciled, there will be no duplication of values in AutoNumber fields.

Text Fields. You can convert Text fields to Memo fields without Access truncating your text. Converting a Memo field to a Text field, however, truncates characters beyond the 255-character limit of Text fields. Similarly, if you convert a variable-length Text field to a fixed-length field, and some records contain character strings that exceed the length that you chose, Access truncates these strings.

Conversion Between Number, Date, and Text Field Data Types. Access makes many conversions between Number, Date, and Text field data types for you. Conversion from Number or Date to Text field data types does not follow the Format property that you assigned to the original data type. Numbers are converted using the General Number format, and dates use the Short Date format. Access is quite intelligent in the methods it uses to convert suitable Text fields to Number data types. For example, it accepts dollar signs, commas, and decimals during the conversion, but ignores trailing spaces. Access converts dates and times in the following Text formats to internal Date/Time values that you then can format the way that you want:

```
1/4/97 10:00 AM

04-Jan-97

January 4

10:00

10:00:00
```

Changing Relationships Between Tables

Adding new relationships between tables is a straightforward process, but changing relationships might require you to change data types so that the related fields have the same data type. To change a relationship between two tables, complete the following steps:

Oracle Ex. 1009, p. 184
Oracle v. DeBellis
IPR2024-00221

1. Close the tables that are involved in the relationship.
2. If the Database window is not active, click the Show Database Window button, or choose Window, 1 Database.
3. Display the Relationships window by clicking the Relationships button of the toolbar or choosing Tools, Relationships.
4. Click the join line that connects to the field whose data type you want to change. When you select the join line, the line becomes darker (wider) as shown in Figure 4.41.

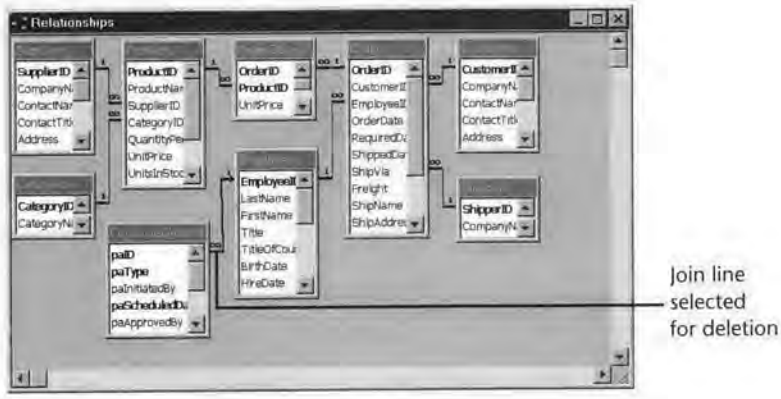


FIG. 4.41 Selecting a relationship to delete or modify.

5. Press Delete to clear the existing relationship. Click Yes when the message box asks you to confirm your deletion.
6. If you are changing the data type of a field that constitutes or is a member field of the primary table's primary key, delete all other relationships that exist between the primary table and every other table to which it is related.
7. Change the data types of the fields in the tables so that the data types match in the new relationships.
8. Re-create the relationships, using the procedure described in the earlier section, "Establishing Relationships Between Tables."

Copying and Pasting Tables

You can copy a complete table or records of a table to the Windows Clipboard by using the same methods that apply to most other Windows applications. (Using the Clipboard to paste individual records or sets of records into a table is one of the subjects of the next chapter.) You use Clipboard operations extensively when you reach Part V of this book, "Integrating Access with Other Office 97 Applications." You can copy tables into other databases, such as a general-purpose backup database, by using the Clipboard; however, exporting a table to a temporary database file, described in Chapter 7, "Linking, Importing, and Exporting Tables," is a more expeditious method.

Oracle Ex. 1009, p. 185
Oracle v. DeBellis
IPR2024-00221

To copy a table to another Access database, a destination database must exist. To create a backup database and copy the contents of the Personnel Actions table to the database, follow these steps:



1. Make the Database window active by clicking it, if it is accessible, or by choosing Window, 1 Database.



2. Click the Tables Tab, if necessary, to display the list of tables.
3. Select the table that you want to copy to the new database.



4. Click the Copy button on the toolbar, press Ctrl+C, or choose Edit, Copy.

   If you plan to copy the table to your current database, skip to step 7.

5. If you have created a destination backup database, choose File, Open Database to open the database; then skip to step 7.
6. To create a backup database, choose File, New Database; choose a blank database and name it backup.mdb or another appropriate file name. Access creates your Backup.mdb database, which occupies approximately 60K without any tables (this is called 60K of *overhead*). Your new database is now active.



7. Click the Paste button on the toolbar, press Ctrl+V, or choose Edit, Paste. The Paste Table As dialog shown in Figure 4.42 appears.



FIG. 4.42 The Paste Table As dialog.

8. You have three options for pasting the backup table to the destination database. You can create a new table or replace the data in a table of the name that you enter in the Table Name text box by selecting Structure and Data. This is the most common choice. You also can paste the structure only and then append data to the table later by selecting Structure Only, or append the records to an existing table of the name that you enter. For this example, accept the default: Structure and Data.
9. Your current or backup database now has a copy of the table that you selected, and the name that you entered appears in the backup's Database window. You can save multiple copies of the same table under different names if you are making a series of changes to the table that might affect the integrity of the data that it contains.

To delete a table from a database, select the table name in the Database window and then press Delete. A confirmation message box appears. Click Yes to delete the table forever. You cannot choose Edit, Undo after deleting a table.

Oracle Ex. 1009, p. 186
Oracle v. DeBellis
IPR2024-00221

Chapter 5

Entering, Editing, and Validating Data in Tables

Ease of data entry is a primary criterion for an effective database development environment. Most of your Access database applications probably use forms for data entry. In many instances, however, entering data in Table Datasheet view is more expeditious than using a form, especially during the database development cycle. For example, it is a good idea to test your proposed database structure before you commit to designing the forms and reports because changing table and field names or altering relationships between tables after you create a collection of forms and reports, involves a substantial amount of work.

In order to test the database design, you often need to enter test data. In this instance, using Table Datasheet view to enter data makes more sense than using a form. Even if you import data from another database type or from a worksheet, you will likely need to edit the data to make it compatible with your new application. The first part of this chapter concentrates on data entry and editing methods.

Another important factor in a database development environment is the capability to maintain the domain integrity of your data. *Domain integrity rules* limit the values you enter in fields to a range or set

Like Access 95, Access 97 enables you to enforce domain integrity rules (often called *business rules*) at the field and table levels. You enforce domain integrity by entering expressions as the value of the Validation Rule property of fields and tables. This chapter teaches you how to use simple expressions for domain integrity validation rules. After you master Access operators and expressions (the subject of Chapter 9, "Understanding Operators and Expressions in Access"), you can write complex validation rules that minimize the possibility of erroneous data in your tables.

Oracle Ex. 1009, p. 187
Oracle v. DeBellis
IPR2024-00221

Using Keyboard Operations for Entering and Editing Data

Although Access is oriented to using a mouse to make selections, keyboard equivalents are provided for the most common actions. One reason for providing keyboard commands is that constant shifting of the hand from a keyboard to mouse and back can reduce data entry rates by more than half. Shifting between a keyboard and mouse also can lead to or aggravate repetitive stress injury (RSI), of which the most common type is carpal tunnel syndrome (CTS).

Keyboard operations are as important in a data entry environment as they are in word processing applications. Consequently, the information concerning key combinations for data entry appears here rather than being relegated to fine print in an appendix. The data entry procedures you learn in the sections that follow prove quite useful when you come to the "Entering Personnel Actions Table Data and Testing Validation Rules" section near the end of the chapter.

Creating an Experimental Copy of Northwind.mdb

If you want to experiment with the various keyboard operations described in the following sections, you are wise to work with a copy of the Northwind.mdb database. When you use a copy, you don't need to worry about making changes that affect the sample database. Experimenting also gives you the opportunity to try the Access database-compacting operation described in Chapter 3, "Navigating Within Access."

Tip

If you're not short on fixed disk space, open a new database and copy the Northwind.mdb Customers and Orders tables to your new database as described in the "Copying and Pasting Tables" section of Chapter 4, "Working with Access Databases and Tables."

To compact Northwind.mdb to a new copy of Northwind.mdb, follow these steps:

1. Close all open document windows.
2. Choose File, Close, or click the Close Window button on the Database window to close the database. Access reverts to a blank window.
3. Choose Tools, Database Utilities, Compact Database to open the Database to Compact From dialog. In this case, the file is compacted to make a copy of the Northwind.mdb file.
4. Double-click the Northwind.mdb item in the Database to Compact From dialog's list box. The Compact Into dialog appears.
5. You can accept the default filename, db1.mdb, in the Filename text box, or you can enter a more creative name, such as **Illwind.mdb**, in the Filename text box, then click Save. Compacting a database file with a new name creates a new, compacted database that you can use for testing.

Oracle Ex. 1009, p. 188
Oracle v. DeBellis
IPR2024-00221

6. Choose File, Open Database, and double-click db1.mdb or the name of your file from step 5.
7. Open the Customers table by double-clicking its entry in the Database window.

Note

Most keyboard operations described in this section apply to tables and updatable queries in Datasheet view, text boxes on forms, and text boxes used for entering property values in Properties windows and in the Field Properties grid of Table Design view. In the examples in this section, the `Arrow Key Behavior` property is set to the `Next Character` value rather than the `Next Field` value (the default). See the "Setting Data Entry Options" section that follows for instructions on how to change the value of the `Arrow Key Behavior` property. When the `Arrow Key Behavior` property is set to Next Field, the arrow keys move the caret from field to field. Data entry operators accustomed to DOS or mainframe database applications usually prefer the Next Character approach.

Using Data Entry and Editing Keys

Arrow keys and key combinations in Access are, for the most part, identical to those used in other Windows applications. Little resemblance exists between these combinations and the key combinations used by DOS database managers, however. The F2 key, used for editing cell contents in Excel, has a different function in Access—it toggles between editing and select mode. *Toggle* means to alternate between two states. In the editing state, the caret indicates the insertion point in the field; the key combinations shown in Table 5.1 are active. If the field is selected (indicated by a black background with white type), the editing keys behave as indicated in Table 5.2.

Note

In the tables that follow, the term *field* is used in place of the more specific description, *data cell* or *cell*, to maintain consistency with Access's documentation and the Help windows. A *field*, in conventional database terminology, indicates the collection of data consisting of the contents of a certain category of information in every record of the table.

The term *grid* in the tables that follow indicates a display in tabular form that doesn't represent fields and records. The list of fields and their descriptions in Table Design mode is an example of a grid.

Table 5.1 Keys for Editing Fields, Grids, and Text Boxes

Key	Function
F2	Toggles between displaying the caret for editing and selecting the entire field. The field must be deselected (black type on a white background), and the caret must be visible for the keys in this table to operate as described.
Ctrl+End	Moves the caret to the end of the line.

(continues)

Oracle Ex. 1009, p. 189
Oracle v. DeBellis
IPR2024-00221

Table 5.1 Continued

Key	Function
Ctrl+End	Moves the caret to the end of a multiple-line field.
←	Moves the caret one character to the left until you reach the first character in the line.
Ctrl+←	Moves the caret one word to the left until you reach the first word in the line.
Home	Moves the caret to the beginning of the line.
Ctrl+Home	Moves the caret to the beginning of the field in multiple-line fields.
Backspace	Deletes the entire selection or the character to the left of the caret.
Delete	Deletes the entire selection or the character to the right of the caret.
Ctrl+Z	Undoes typing, a replace operation, or any other change to the record since the last time it was saved. An edited record is saved to the database when you move to a new record or close the editing window.
Alt+Backspace	Same as Ctrl+Z.
Esc	Undoes changes to the current field. Press Esc twice to undo changes to the current field and to the entire current record, if you edited other fields.

Table 5.2 Keys for Selecting Text in Fields, Grids, and Text Boxes

Selection	Key	Function
Text within a field	F2	Toggles between displaying the caret for editing and selecting the entire field. The field must be selected (white type on a black background) for the keys in this table to operate as described.
	Shift+→	Selects or deselects one character to the right.
	Ctrl+Shift+→	Selects or deselects one word to the right. Includes trailing spaces.
	Shift+←	Selects or deselects one character to the left.
	Ctrl+Shift+←	Selects or deselects one word to the left.
Next field	Tab or Enter	Selects the next field. The "Setting Default Data Entry Options" section later in this chapter tells you how to change the effect of the Enter key.
Record	Shift+spacebar	Selects or deselects the entire current record.

Oracle Ex. 1009, p. 190
Oracle v. DeBellis
IPR2024-00221

Selection	Key	Function
	↑	Selects the preceding record when a record is selected.
	↓	Selects the next record when a record is selected.
Column	Ctrl+spacebar	Toggles selection of the current column.
	←	Selects the column to the left (if a column is selected and there is a column to the left).
Fields and records	F8	Turns on Extend mode. You see EXT in the status bar. In Extend mode, pressing F8 extends the selection to the word, field, record, and all records.
	Shift+F8	Reverses the last F8.
	Esc	Cancels Extend mode.

Operations that select the entire field or a portion of the field, as listed in Table 5.2, generally are used with the Windows Clipboard operations. Selecting an entire field and then pressing Delete or typing a character is a quick way of ridding the field of its original contents.

Using Key Combinations for Windows Clipboard Operations

In Table Datasheet view, the Windows Clipboard is used primarily for transferring Access data between applications. However, you also can use the Clipboard for repetitive data entry. Access 97 enables you to select a rectangular block of data cells in a table and copy the block to the Clipboard. To select a block of cells, follow these steps:

1. Position the mouse pointer at the left edge of the top left cell of the block you want to select. The mouse pointer (shaped like an I-beam until this point) turns into a cross similar to the mouse pointer for Excel worksheets.
2. Hold the left mouse button down and drag the mouse pointer to the right edge of the bottom right cell of the desired block.
3. The selected block appears in reverse video (white on black). Release the left mouse button when the selection meets your requirement.

A selected block of data in the Customers table appears in Figure 5.1. You can copy data blocks but cannot cut them.

Table 5.3 lists the key combinations for copying or cutting data to and pasting data from the Clipboard. When you paste data from the Clipboard, all the data in the Clipboard is pasted to a single cell if the Clipboard data is of the correct data type and fits within the size of the field. You also can use the Cut, Copy, and Paste buttons on the toolbar as a substitute for the key combinations. Another alternative is to choose Edit, Cut, Copy, or Paste.



Oracle Ex. 1009, p. 191
Oracle v. DeBellis
IPR2024-00221

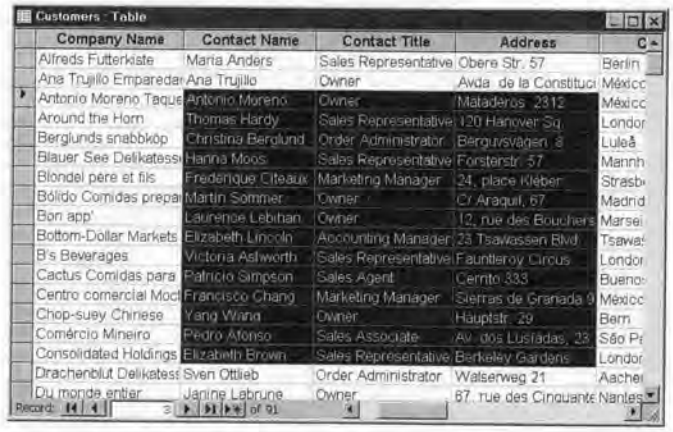


FIG. 5.1 Selecting a rectangular data block in Table Datasheet view.

Table 5.3 Key Combinations for Windows Clipboard Operations

Key	Function
Ctrl+C	Copies the selection to the Clipboard.
Ctrl+Insert	Copies the selection to the Clipboard.
Ctrl+V	Pastes the contents of the Clipboard at the location of the caret.
Shift+Insert	Pastes the contents of the Clipboard at the location of the caret.
Ctrl+X	Copies the selection to the Clipboard, and then deletes it. This operation also is called a *cut*. You can cut only the content of a single cell you select with the caret.
Shift+Delete	Copies the selection to the Clipboard, and then deletes it. This operation also is called a *cut*. You can cut only the content of a single cell you select with the caret.
Ctrl+Z	Undoes your last Cut, Delete, or Paste operation.
Alt+Backspace	Undoes your last Cut, Delete, or Paste operation.

Note

If you attempt to paste a rectangular block into a cell, you receive a `Data too long for field` error message. Access then creates a Paste Errors table that contains the contents of the rectangular block. This is a quick way of creating a new table from a block selection. If you create a table this way, rename it immediately, or Access overwrites the table with a new Paste Errors table the next time a paste error occurs.

Oracle Ex. 1009, p. 192
Oracle v. DeBellis
IPR2024-00221

Using Shortcut Keys for Fields and Text Boxes

You use shortcut keys to minimize the number of keystrokes required to accomplish common data entry tasks. Most shortcut key combinations use the Ctrl key with other keys. Ctrl+C, Ctrl+V, and Ctrl+X for Clipboard operations are examples of global shortcut keys in Windows 95. Table 5.4 lists shortcut keys applicable to field and text box entries.

Table 5.4 Shortcut Keys for Text Boxes and Fields in Tables

Key	Function
Ctrl+; (semicolon)	Inserts the current date.
Ctrl+: (colon)	Inserts the current time.
Ctrl+' (apostrophe) or Ctrl+" (quote)	Inserts the value from the same field in the preceding record.
Ctrl+Enter	Inserts a newline character (carriage return plus line feed) in a text box.
Ctrl++ (plus)	Adds a new record to the table.
Ctrl+– (minus)	Deletes the current record from the table.
Shift+Enter	Saves all changes to the current record.

Tip

Emulating the data entry key behavior of DOS or mainframe RDBMS can make a major difference in the acceptance of your database applications by data entry operators who have years of experience with DOS or mainframe database applications.

Setting Data Entry Options

You can modify the behavior of the arrow keys and the Tab and Enter keys by choosing Tools, Options and clicking the Keyboard tab to display the keyboard options settings. Table 5.5 lists the available options with the default values shown in bold type. (This table also appears in Chapter 3, "Navigating Within Access.") These keyboard options enable you to make the behavior of the data entry keys similar to that of DOS database managers, such as dBASE and Paradox.

◀◀ See "Keyboard Options," p. 85



Oracle Ex. 1009, p. 193
Oracle v. DeBellis
IPR2024-00221

Table 5.5 Keyboard Options for the Access System

Option	Group	Function
Don't Move	Move After Enter	When this option is selected, the caret remains in the current field when you press Enter.
Next Field	Move After Enter	When this option is selected, the caret moves to the next field when you press Enter. Use this setting to duplicate xBase behavior.
Next Record	Move After Enter	When this option is selected, the caret moves down the column to the next record when you press Enter.
Next Field	Arrow Key Behavior	If this option is selected, pressing the right or left arrow keys moves the caret to the next field.
Next Character	Arrow Key Behavior	If this option is selected, pressing the right or left arrow keys moves the caret to the previous or next character in the same field. Use this setting if you want to duplicate the behavior of xBase or mainframe databases.
Select Entire Field	Behavior Entering Field	When this option is selected, the entire field's contents are selected when you use the arrow keys to move the caret into the field.
Go to Start of Field	Behavior Entering Field	Selecting this option causes the caret to move to the beginning of the field when you use the arrow keys to move the caret into the field.
Go to End of Field	Behavior Entering Field	Selecting this option causes the caret to move to the end of the field when you use the arrow keys to move the caret into the field. Use this setting to duplicate xBase and mainframe database behavior.
Cursor Stops at First/Last	None	If this option is selected, it keeps the caret from moving to Field another record when the left or right arrow keys are pressed,and the caret is in the first or last field of the record.

Adding Records to a Table



When you create a new table in Datasheet view, it contains 30 empty records with an asterisk (*) in the record selection button of the last (31st) row. Record selection buttons are the gray buttons in the leftmost column of Table Datasheet view. A similar blank record also appears at the end of an existing table if the table is updatable. (An *updatable table* is one whose data you can add to or edit.) If you open a database for read-only

Oracle Ex. 1009, p. 194
Oracle v. DeBellis
IPR2024-00221

access by marking the Exclusive check box of the Open dialog, this blank record does not appear. Tables attached from other databases also can be read-only: the updatability of attached tables is discussed in Chapter 7, "Linking, Importing, and Exporting Tables."

▶▶ See "Modifying Linked and Imported Tables," p. 251



This book refers to the blank record as the *tentative append record*. The term *tentative* is used because the record is appended to the table only after you enter data in one of the fields and then save the changes you make to the record. You can save changes to a record by moving the record pointer to a different record or by choosing Records, Save Record. The location of the record pointer is indicated by an arrow symbol in the record selection button. The record with the arrow symbol is called the *selected record*.



Tip

To go to the tentative append record quickly, click the New Record button on the toolbar.

◀◀ See "Using Input Masks," p. 128



When you place the caret in a field of the tentative append record, the record selection button's asterisk symbol turns into the selected record symbol. When you add data to a field of the selected tentative append record, the selected record symbol changes to the edit symbol (a pencil), and a new tentative append record appears in the row after your addition. Figure 5.2 shows a new record in the process of being added to the Customers table. The CustomerID field has an input mask that requires you to enter five letters, which are capitalized automatically as you enter them. The input mask changes the caret from an I-beam to a reverse-video block.



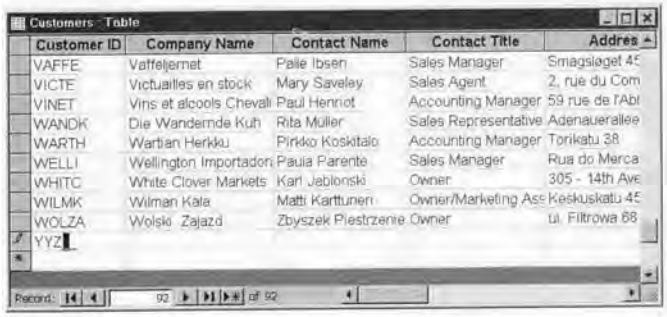


FIG. 5.2 Adding a new record to the Customer table.

Oracle Ex. 1009, p. 195
Oracle v. DeBellis
IPR2024-00221

You can cancel the addition of a new record by deleting all the entries you made in the record before moving the record pointer. If you edited only the first field and did not move the record pointer, you can press the Esc button to cancel the addition.

Selecting, Appending, Replacing, and Deleting Table Records

You can select a single record or a group of records to copy or cut to the Clipboard, or to delete from the table, by the following methods:

- To select a single record, click its record selection button.
- To select a contiguous group of records, click the first record's selection button and then drag the mouse pointer along the record selection buttons to the last record of the group.
- Alternatively, to select a group of records, click the first record's selection button, hold down the Shift key, and then click the last record to include in the group. Alternatively, you can hold down the Shift key and press the down arrow to select a group of records.

> **Note**
>
> You can cut groups of records to the Clipboard, deleting them from the table, but you cannot cut data blocks. A *group of records* includes all fields of one or more selected records. A *data block* consists of a selection in a table datasheet that does not include all fields of the selected rows. The Edit, Cut menu choice is enabled for groups of records and disabled for data blocks.

You can cut or copy and append duplicate records to the same table (if appending the duplicate records does not cause a primary-key violation) or to another table. You cannot cut records from a primary table that have dependent records in a related table if you enforce referential integrity. The following methods are applicable to appending or replacing the content of records with records stored in the Clipboard:

- To append records from the Clipboard to a table, choose Edit, Paste Append. (No shortcut key exists for Paste Append.)



- To replace the content of a record(s) with data from the Clipboard, select the record(s) whose content you want to replace and then press Ctrl+V or choose Edit, Paste or click the Paste button on the toolbar. Only the number of records you select or the number of records stored in the Clipboard, whichever is fewer, is replaced.

To delete one or more records, select the records you want to delete and press Delete. If deletion is allowed, a message box asks you to confirm your deletion. You cannot undo deletions of records.

Oracle Ex. 1009, p. 196
Oracle v. DeBellis
IPR2024-00221

Validating Data Entry

The data entered in tables must be accurate if the database is to be valuable to you or your organization. Even the most experienced data entry operators occasionally enter incorrect information. You can add simple tests for the reasonableness of entries by adding short expressions to the Validation Rule text box. If the data entered fails to conform to your validation rule, a message box informs the operator that a violation occurred. Validating data maintains the domain integrity of your tables.

Expressions are the core element of computer programming. Access enables you to create expressions without requiring that you be a programmer, although some familiarity with a programming language is helpful. Expressions are statements used to calculate values using the familiar arithmetic symbols, `+`, `-`, `*` (multiply), and `/` (divide). These symbols are called *operators* because they operate on (use) the values that precede and follow them. The symbols are printed in bold monospace type because they are reserved symbols in Access VBA. The values operated on by operators are called *operands*.

You can also use operators to compare two values; the `<` (less than) and `>` (greater than) symbols are examples of *comparison operators*. **`And`**, **`Or`**, **`Is`**, **`Not`**, `Between`, and `Like` are called *logical operators*. Comparison and logical operators return only **`True`**, **`False`**, and unknown (the **`Null`** value). The **`&`** operator combines two text entries (*character strings* or just *strings*) into a single string. (You can use `+` in Access to concatenate strings, but **`&`** is the preferred symbol because it leaves no doubt as to the intended operation; using the `+` operator with numbers and strings may be ambiguous to both Access and users as to whether string concatenation or arithmetic addition is intended.) To qualify as an expression, at least one operator must be included. You can construct complex expressions by combining the different operators according to rules that apply to each of the operators involved. The collection of these rules is called *operator syntax*.

▶▶ See "Understanding the Elements in Expressions," p. 288



Data validation rules use expressions that result in one of two values: **`True`** or **`False`**. Entries in a data cell are accepted if the result of the validation is true and rejected if it is false. If the data is rejected by the validation rule, the text you enter in the Validation Text text box appears in a message box. Chapter 9, "Understanding Operators and Expressions in Access," explains the syntax of Access validation expressions.

Adding Field-Level Validation Rules

Validation rules that restrict the values entered in a field based on only one field are called *field-level validation rules*. Table 5.6 lists the simple field-level validation rules used for some of the fields of the Personnel Actions table you created in Chapter 4, "Working with Access Databases and Tables."

Oracle Ex. 1009, p. 197
Oracle v. DeBellis
IPR2024-00221

Table 5.6 Validation Criteria for the Fields of the Personnel Actions Table

Field Name	Validation Rule	Validation Text
paID	`>0`	Please enter a valid employee ID number.
paType	`"H" Or "S" Or "Q" Or "Y" Or "B" Or "C"`	Only H, S, Q, Y, B, and C codes can be entered.
paInitiated By	`>0`	Please enter a valid supervisor ID number.
paScheduledDate	`Between Date() -3650 And Date() + 365`	Scheduled dates cannot be more than 10 years ago or more than one year from now.
paApprovedBy	`>0 Or Is Null`	Enter a valid manager ID number or leave blank if not approved.
paRating	`Between 0 And 9 Or Is Null`	Rating range is 0 for terminated employees, 1 to 9, or blank.
paAmount	No rule	No text
paComments	No rule	No text

The validation rules for fields that require employee ID numbers are not, in their present form, capable of ensuring that a valid ID number is entered. You could enter a number greater than the total number of employees in the firm. The validation rule for the paID field tests the EmployeeID number field of the Employees table to determine whether the paID number is present. You don't need to create this test because the rules of referential integrity, discussed in the "Enforcing Referential Integrity" section in Chapter 4, "Working with Access Databases and Tables," perform this validation for you. Validation rules for paInitiatedBy and paApprovedBy require tests based on entries in the Employees table.

To add the validation rules to the Personnel Actions table, follow these steps:

1. Open the Personnel Actions table, if it isn't already open, by double-clicking the table name in the Database window.



2. Return to Design mode by clicking the Design View button. The paID field is selected.
3. Press F6 to switch to the Field Properties window, and then move to the Validation Rule text box.
4. Enter **>0**. Press Enter to accept the entry and move to the Validation Text text box.
5. Type **Please enter a valid employee ID number** in the Validation Text text box. The text scrolls to the left when it becomes longer than can be displayed in the text box. To display the beginning of the text, press Home. Press End to position the caret at the last character. Press Enter to complete the operation.

Oracle Ex. 1009, p. 198
Oracle v. DeBellis
IPR2024-00221

6. Move to the Required text box and enter **Yes**, or open the drop-down list and click Yes. Figure 5.3 shows your entries in the Field Properties text boxes.
7. Press F6 to switch back to the Table Design grid. Move to the next field, and press F6.
8. Enter the validation rule and validation text for the six remaining fields listed in Table 5.6 that use data entry validation, repeating steps 2 through 5. Square brackets ([]) enclose field names that include punctuation or spaces. Enter **Yes** in the Required text box for the paType, paInitiatedBy, and paScheduledDate fields.



FIG. 5.3 The Field Properties text boxes, showing your first validation entries.

You test your validation rule entries in the "Entering Personnel Actions Table Data and Testing Validation Rules" section later in this chapter.

Adding Table-Level Validation Rules and Using the Expression Builder

One of the fields, paEffectiveDate, requires a validation rule that depends on the value of paScheduledDate. The effective date of the personnel department's action should not be prior to the scheduled date for the review that results in the action. Access 1.x enabled you to refer to one or more field names when creating a field-level validation rule expression. Like Access 2.0 and Access 95, however, you cannot refer to other field names in a validation rule expression in Access 97; instead, you enter the validation rule in the Table Properties window. Validation rules in which the value of one field depends on a previously entered value in another field of the current record are called *table-level validation rules*.

The following steps create a table-level validation rule for the paEffectiveDate field:

1. Click the Properties button on the toolbar to display the Table Properties window (see Figure 5.4).



2. Enter **Personnel Department Actions** in the Description text box, as shown in Figure 5.4.
3. Move the caret to the Validation Rule text box. Click the ellipsis button (...) that appears to right of the Validation Rule text box (see Figure 5.5) to display the Expression Builder dialog. The current table, Personnel Actions, is selected in the left list, and the fields of the table appear in the center list.
4. Double-click paEffectiveDate in the center list to place [paEffectiveDate] in the Expression text box at the top of the dialog.

Oracle Ex. 1009, p. 199
Oracle v. DeBellis
IPR2024-00221

5. Enter >= in the expression text box and double-click paScheduledDate in the center list to add [paScheduledDate] to the expression.
6. You also want to accept a blank entry if the effective date of the personnel action is not scheduled, so add **Or [paEffectiveDate] Is Null** to the expression. Your expression appears as shown in Figure 5.5.
7. Click OK to add the table-level validation rule and close the Expression Builder dialog.
8. Move to the Validation Text text box and enter **Effective date must be on or after scheduled date**. Your Table Properties dialog appears as shown in Figure 5.6.



9. Click the Close Window button of the Table Properties window or click the Properties button on the toolbar to close the window.



FIG. 5.4 Adding a table description in the Table Properties dialog.

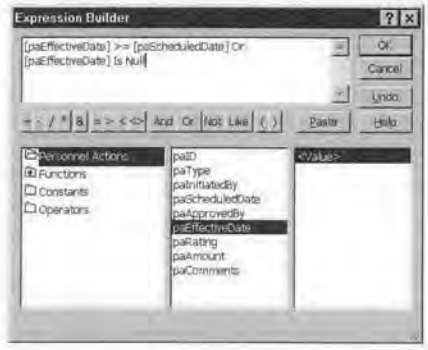


FIG. 5.5 Creating a validation rule with the Expression Builder.

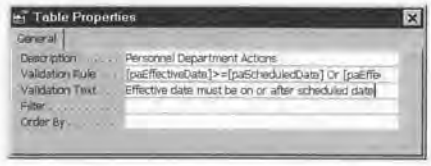


FIG. 5.6 Adding Validation Text to a table-level validation rule.

Oracle Ex. 1009, p. 200
Oracle v. DeBellis
IPR2024-00221

Note

You can create more than one table-level validation rule by using the And reserved word and then adding another expression that involves the relationship between other fields of the table. However, Access only provides a single Validation Text message that appears when you violate any of the table-level validation rules. You can also use an Access VBA function procedure to supply a table- or field-level validation rule. By using an Access VBA function procedure, you can display different validation text messages—in this case, your validation messages are generated by your VBA function procedure, rather than the `Validation Text` property. Chapter 28, "Writing Visual Basic for Applications Code," describes how to use Access VBA.

Adding Records to the Personnel Actions Table

Now you have a chance to test your work in creating the Personnel Actions table and to check whether Access is enforcing referential integrity. The initial entries for each of the nine employees of Northwind Trading are shown in Table 5.7. The entries for paScheduledDate and paEffectiveDate are taken from the HireDate field of the Employees table. The HireDate field of the Employees table now is superfluous because it duplicates the data in the Personnel Actions table. You delete the HireDate field in a later chapter. Feel free to be as generous or as parsimonious as you want with the monthly salaries shown in the paAmount field.

Table 5.7 First Nine Entries for the Personnel Actions Table

ID	Type	Initiated By	Scheduled	Approved By	Effective Date	New Amount	Comments
1	H	1	01-May-92		01-May-92	2,000	Hired
2	H	1	14-Aug-92		14-Aug-92	3,500	Hired
3	H	1	01-Apr-92		01-Apr-92	2,250	Hired
4	H	2	03-May-93	2	03-May-93	2,250	Hired
5	H	2	17-Oct-93	2	17-Oct-93	2,500	Hired
6	H	5	17-Oct-93	2	17-Oct-93	4,000	Hired
7	H	5	02-Jan-94	2	02-Jan-94	3,000	Hired
8	H	2	05-Mar-94	2	05-Mar-94	2,500	Hired
9	H	5	15-Nov-94	2	15-Nov-94	3,000	Hired

Entering historical information in a table in Datasheet view is a relatively fast process for an experienced data entry operator. This process also gives you a chance to test your

Oracle Ex. 1009, p. 201
Oracle v. DeBellis
IPR2024-00221

default entries and Format properties for each field. You can enter bogus values that don't comply with your validation rules to verify that your rules are operational. To add the first nine historical records to the Personnel Actions table using the data from Table 5.7, follow these steps:



1. Click the Datasheet button on the toolbar to return to Datasheet view in Run mode, if necessary. The caret is positioned in the paID field of the default first record.
2. Enter the paID of the employee. Press Enter, Tab, or the right-arrow key to move to the next field. When you do this, a new default blank record is added to the view but not to the content of the table. A new record is added to the table only when a value is entered in one of the fields of the default blank record.
3. Type **H** in the Type field, and then press Enter, Tab, or the right-arrow key to move to the next field.
4. Type the numeric value for the paInitiatedBy field. (You need a value in this field for each employee because of the field's validation rule.) Press Enter, Tab, or the right-arrow key to move to the next field.
5. Type the paScheduledDate entry. You don't need to delete the default date value. When you type a new date, it replaces the default value. Then press Enter, Tab, or the right-arrow key.
6. If a value is in the table for paApprovedBy, type the value. Then press Enter, Tab, or the right-arrow key.
7. Type the paEffectiveDate entry. Press Enter, Tab, or the right-arrow key twice to skip the Rating field, which is inapplicable to newly hired employees.
8. Enter the paAmount of the monthly salary at the time of hiring. Press Enter, Tab, or the right-arrow key.
9. Type **Hired** in the paComments field, or any other comment you care to make. Press Enter, Tab, or the right-arrow key. The caret moves to the paID field of the next default blank record.
10. Repeat steps 2–9 for eight more employees in Table 5.7. (You can add similar records for employees 10–15 if you want.)

When you complete your entries, the Personnel Actions table appears as shown in Figure 5.7.

Troubleshooting

Error messages appear when I attempt to enter data in fields with validation rules.

Edit or reenter the data to conform to the data types and validation rules for the field. Error messages that appear when you enter the data correctly indicate that something is amiss with your

Oracle Ex. 1009, p. 202
Oracle v. DeBellis
IPR2024-00221

validation rules. In this case, change to design mode and review your validation rules for the offending fields against those listed in Table 5.6. You may want to remove the validation rule temporarily by selecting the entire expression and cutting the rule to the Clipboard. (You can paste the expression back into the text box later.) Return to Run mode to continue with your entries.

Personnel Actions : Table

ID	Type	Initiated B	Scheduled	Approved By	Effective	Rating	Amount	Comments
1	H	1	5/1/92		5/1/92		2,000.00	Hired
2	H	1	8/14/92		8/14/92		3,500.00	Hired
3	H	1	4/1/92		4/1/92		2,250.00	Hired
4	H	2	5/3/93	2	5/3/93		2,250.00	Hired
5	H	2	10/17/93	2	10/17/93		2,500.00	Hired
6	H	5	10/17/93	2	10/17/93	8	4,000.00	Hired
7	H	5	1/2/94	2	1/2/94		3,000.00	Hired
8	H	2	3/5/94	2	3/5/94		2,500.00	Hired
9	H	5	11/15/94	2	11/15/94		3,000.00	Hired
	Q		9/2/96		9/30/96			

Record: 1 of 9

FIG. 5.7 The first nine records of the Personnel Actions table.

Entering Personnel Actions Table Data and Testing Validation Rules

You can experiment with entering table data and testing your validation rules at the same time. Testing database applications often requires much more time and effort than creating them. The following basic tests are required to confirm your validation rules:

- *Referential integrity*. Type **25** in the paID field and **2** in the paInitiatedBy field of the default blank record, record number 10, and then press the up-arrow key. Pressing the up-arrow key tells Access that you are finished with the current record and to move up to the preceding record with the caret in the same field. Access then tests the primary-key integrity before enabling you to leave the current record. The message box shown in Figure 5.8 appears. Click OK, or press Enter.



FIG. 5.8 The message box indicating an entry that violates referential integrity rules.

- *No duplicates restriction for primary key*. In the record just added, attempt to duplicate exactly the entries for record number 9, and then press the up-arrow key. You see the message box shown in Figure 5.9. Click OK, or press Enter.
- *paType validation*. Type **x** and press the right-arrow key to display the message box with the validation text you entered for the paType field, shown in Figure 5.10. Click OK, or press Enter.

Oracle Ex. 1009, p. 203
Oracle v. DeBellis
IPR2024-00221



FIG. 5.9 The message box alerting you that a record has a duplicate key.



FIG. 5.10 The message box displayed when an entry violates a validation rule.

Type **q**, and move to the paInitiatedBy field. When the caret leaves the paType field, the **q** changes to Q because of the > format character used. Type **0** (an invalid employee ID number), and press the right-arrow key to display the message box shown in Figure 5.11. Click OK, or press Enter.



FIG. 5.11 The message box that appears in response to an invalid employee ID number entry.

Continue with the testing. Type a date, such as **1/31/97**, for the paScheduledDate, and type a date one day earlier (such as **1/30/97**) for the paEffectiveDate to display the error message boxes with the validation text you entered. (You must move the caret to a different record to cause the table-level validation rule to be applied.) Enter a valid date after the test. To edit a field rather than retype it, press F2 to deselect the entire field and display the caret for editing. F2 toggles selection and editing operations.

When you finish your testing, click the selection button of the last record you added, and then press Delete. The confirmation message box shown in Figure 5.12 appears.

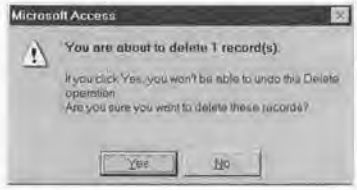


FIG. 5.12 The confirmation dialog for deleting one or more records.

Oracle Ex. 1009, p. 204
Oracle v. DeBellis
IPR2024-00221

Chapter 6

Sorting, Finding, and Filtering Data in Tables

Microsoft Access provides a variety of sorting and filtering features that make customizing the display data in Table Datasheet view a quick and simple process. Sorting and filtering records in tables is quite useful when you use data in a table to create a mailing list or print a particular set of records.

Access also includes versatile search (find) and replace facilities that enable you to locate every record that matches a value you specify, and then, optionally, change that value. If you have a large table, Access's find facility enables you to quickly locate the needles in the haystacks. Search and replace often is needed when you import data from another database or a worksheet, which is the subject of the next chapter.

The sorting, filtering, searching, and replacing features of Access actually are implemented by behind-the-scenes queries that Access creates for you. When you reach Part II of this book, which deals exclusively with queries, you'll probably choose to implement these features with Access's graphical query-by-example (QBE) methods. Learning the fundamentals of these operations with tables, however, makes understanding queries easier. You also can apply filters to query result sets, use the find feature with queries in Datasheet view, and use search and replace on the result sets of updatable queries. For readers who are xBase users or know SQL, statements in these two languages equivalent to the operation being performed are given where applicable.

Sorting Table Data

A fundamental requirement of a database development environment is the capability to sort records quickly so that they appear in the desired sequence. Early desktop database managers, such as dBASE II and III/III+, required that you create a new copy of a table if you physically wanted to sort the table's records in a new order. Creating and specifying an index on a field table enabled you to display or print the table in the order of the index. If you

Oracle Ex. 1009, p. 205
Oracle v. DeBellis
IPR2024-00221

wanted to sort the data by two or more fields, however, you either had to create a composite index on the fields, or presort the data in the order of one or more fields, and then apply the single-field index.

Modern desktop database development systems, such as Access, never require that you physically sort the table. Instead, the physical location of the records in the file is the order in which the records were entered. By default, Access displays records in the order of the primary key. This behavior is similar to that of Borland's Paradox. If your table doesn't have a primary key, the records display in the order you enter them. Unlike dBASE and its clones, you cannot choose a specific Access index to alter the order in which the records display in Table Datasheet view of the user interface (UI). You can, however, specify an index to order records of tables you manipulate with Access VBA code. Access uses sorting methods to display records in the desired order. If an index exists on the field in which you sort the records, the sorting process is much quicker. Access automatically uses indexes, if indexes exist, to speed the sort. This process is called *query optimization*. Access's indexes and query optimization methods are discussed in Chapter 23, "Exploring Relational Database Design and Implementation."

The following sections show you how to use Access's sorting methods to display records in the sequence you want. The Customers table of Northwind.mdb is used for the majority of the examples in this chapter because it is typical of a table whose data you can use for a variety of purposes.

Freezing Display of a Table Field

If the table you are sorting contains more fields than you can display in Access's Table Datasheet View, you can freeze one or more fields to make viewing the sorted data easier. Freezing a field makes the field visible at all times, regardless of which other fields you display by manipulating the horizontal scroll bar. To freeze the Customer ID and Company Name fields of the Customers table, follow these steps:

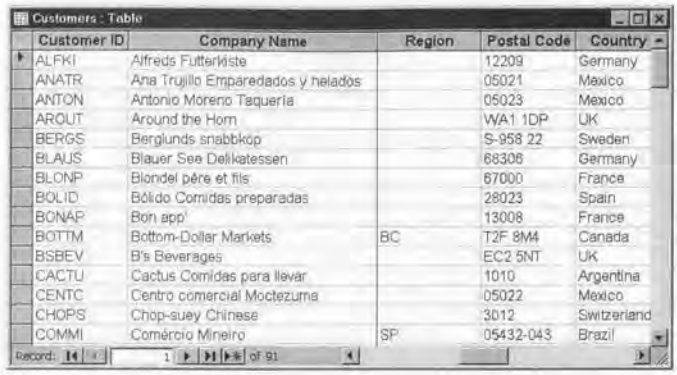
1. Open the Customers table in Datasheet view.
2. Click the field header button of the Customer ID field to select the first column.
3. Hold the Shift key down and click the Company Name field header button. Alternatively, you can drag the mouse from the Customer ID field to the Company Name field to select the first and second columns.
4. Choose Format, Freeze Columns.

When you scroll to fields to the right of the frozen columns, your Datasheet view of the Customers table appears as illustrated in Figure 6.1. A solid vertical line replaces the halftone grid line between the frozen and thawed (selectable) field columns.

Note

If you frequently freeze columns, you can add the Freeze Columns button from the Datasheet collection to your Datasheet toolbar. See Chapter 13, "Designing Custom Multitable Forms," to learn how to customize your toolbars.

Oracle Ex. 1009, p. 206
Oracle v. DeBellis
IPR2024-00221



Customers : Table

Customer ID	Company Name	Region	Postal Code	Country
ALFKI	Alfreds Futterkiste		12209	Germany
ANATR	Ana Trujillo Emparedados y helados		05021	Mexico
ANTON	Antonio Moreno Taquería		05023	Mexico
AROUT	Around the Horn		WA1 1DP	UK
BERGS	Berglunds snabbköp		S-958 22	Sweden
BLAUS	Blauer See Delikatessen		68306	Germany
BLONP	Blondel père et fils		67000	France
BOLID	Bólido Comidas preparadas		28023	Spain
BONAP	Bon app'		13008	France
BOTTM	Bottom-Dollar Markets	BC	T2F 8M4	Canada
BSBEV	B's Beverages		EC2 5NT	UK
CACTU	Cactus Comidas para llevar		1010	Argentina
CENTC	Centro comercial Moctezuma		05022	Mexico
CHOPS	Chop-suey Chinese		3012	Switzerland
COMMI	Comércio Mineiro	SP	05432-043	Brazil

Record: 1 of 91

FIG. 6.1 The Northwind.mdb Customers table with the CustomerID and CompanyName columns frozen.

Sorting Data on a Single Field

When creating a mailing list, a standard practice in the United States is to sort the records in ascending ZIP Code order. This practice often is observed in other countries that use postal codes, also. To sort the Customers table in the order of the Postal Code field, follow these steps:

1. Select the Postal Code field by clicking the field header button of the Postal Code field.
2. Click the Sort Ascending (A-Z) button of the toolbar or choose Records, Sort, Ascending.

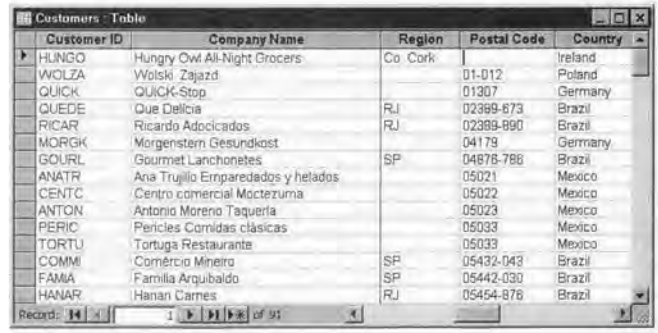


▶▶ See "Writing Select Queries in SQL," p. 847



Your Customers table quickly is sorted into the order shown in Figure 6.2. Sorting a table is equivalent to specifying the selected field as the table name of the `ORDER BY` clause of an SQL statement, as in:

```
SELECT * FROM Customers ORDER BY [Postal Code]
```



Customers : Table

Customer ID	Company Name	Region	Postal Code	Country
HUNGO	Hungry Owl All-Night Grocers	Co. Cork		Ireland
WOLZA	Wolski Zajazd		01-012	Poland
QUICK	QUICK-Stop		01307	Germany
QUEDE	Que Delícia	RJ	02389-673	Brazil
RICAR	Ricardo Adocicados	RJ	02389-890	Brazil
MORGK	Morgenstern Gesundkost		04179	Germany
GOURL	Gourmet Lanchonetes	SP	04876-786	Brazil
ANATR	Ana Trujillo Emparedados y helados		05021	Mexico
CENTC	Centro comercial Moctezuma		05022	Mexico
ANTON	Antonio Moreno Taquería		05023	Mexico
PERIC	Pericles Comidas clásicas		05033	Mexico
TORTU	Tortuga Restaurante		05033	Mexico
COMMI	Comércio Mineiro	SP	05432-043	Brazil
FAMIA	Familia Arquibaldo	SP	05442-030	Brazil
HANAR	Hanari Carnes	RJ	05454-876	Brazil

Record: 1 of 91

FIG. 6.2 Applying an ascending sort order to the Postal Code field.

Oracle Ex. 1009, p. 207
Oracle v. DeBellis
IPR2024-00221

Sorting Data on Multiple Fields

Although the sort operation in the preceding section accomplishes exactly what you specify, the result is less than useful because of the vagaries of postal code formats used in different countries. What's needed here is a multiple-field sort on the Country field first and then the Postal Code field. Thus, you might select both the Country and the Postal Code fields to perform the multi-column sort. The Quick Sort technique, however, automatically applies the sorting priority to the leftmost field you select, Postal Code. Access offers two methods of handling this problem: reorder the field display or specify the sort order in a Filter window. Filters are discussed in the "Filtering Table Data" section later in this chapter, so follow these steps to use the reordering process:

1. Select the Country field by clicking its field header button.
2. Hold the left mouse button down and drag the Country field to the left of the Postal Code field. Release the left mouse button to drop the field in its new location.
3. Press the Shift key and click the header button of the Postal Code field to select both the Country and Postal Code columns.
4. Click the Sort Ascending button of the toolbar or choose Records, Sort, Ascending.

The sorted table, shown in Figure 6.3, now makes much more sense. Applying a multi-field sort on a table (sometimes called a *composite sort*) is the equivalent of the following SQL statement:

```
SELECT * FROM Customers ORDER BY Country, [Postal Code]
```

Customers : Table

Customer ID	Company Name	Region	Country	Postal Code
RANCH	Rancho grande		Argentina	1010
OCEAN	Océano Atlántico Ltda.		Argentina	1010
CACTU	Cactus Comidas para llevar		Argentina	1010
PICCO	Piccolo und mehr		Austria	5020
ERNSH	Ernst Handel		Austria	8010
MAISD	Maison Dewey		Belgium	B-1180
SUPRD	Suprêmes délices		Belgium	B-6000
QUEDE	Que Delícia	RJ	Brazil	02389-673
RICAR	Ricardo Adocicados	RJ	Brazil	02389-890
GOURL	Gourmet Lanchonetes	SP	Brazil	04876-786
COMMI	Comércio Mineiro	SP	Brazil	05432-043
FAMIA	Família Arquibaldo	SP	Brazil	05442-030
HANAR	Hanari Carnes	RJ	Brazil	05454-876
QUEEN	Queen Cozinha	SP	Brazil	05487-020
TRADH	Tradição Hipermercados	SP	Brazil	05634-030

Record: 1 of 91

FIG. 6.3 The effect of a multiple-field sort on the Country and Postal Code fields of the Customers table.

Removing a Table Sort Order and Thawing Columns

After you freeze columns and apply sort orders to a table, you might want to return the table to its original condition. To do so, Access offers you the following choices:

- To return to Datasheet view of an Access table with a primary key to its original sort order, select the field(s) that comprise the primary key (in the order of the primary key fields).

Oracle Ex. 1009, p. 208
Oracle v. DeBellis
IPR2024-00221

- To return to the original order when the table has no primary key field, close the table, *don't* save the changes, and reopen the table.
- To thaw your frozen columns, choose Format, Unfreeze All Columns.
- To return the sequence of fields to its original state, either drag the fields you moved back to their prior position or close the table *without* saving your changes.

If you make substantial changes to the layout of the table and apply a sort order, it is usually quicker to close and reopen the table. (*Don't* save your changes to the table layout.)

Removing the sort order from a field is the equivalent of issuing the following SQL statement:

```
SELECT * FROM Customers
```

Finding Matching Records in a Table

To search for and select records with field values that match (or partially match) a particular value, use Access's Find feature. To find Luleå—a relatively large city in northern Sweden close to the Arctic Circle—in the City field, follow these steps:

1. Select the field—City—you want to search. You can select the City field by either clicking the header button or placing the caret in the field.
2. Click the Find button of the toolbar or choose Edit, Find to display the Find in field dialog shown in Figure 6.4. You can also display the Find dialog by pressing Ctrl+F.



FIG. 6.4 The opened Find in Field dialog with the City field selected.

3. Type the name of the city, **Lulea**, in the Find What text box (see Figure 6.5). When you make an entry in the Find What text box, the Find First and Find Next command buttons are enabled.
4. Select Whole Field from the Match drop-down list. (The other choices, Start of Field and Any Part of Field, are just as effective in this case.)

   The default value of the Search option button is satisfactory, and matching case or format is not important here.

Oracle Ex. 1009, p. 209
Oracle v. DeBellis
IPR2024-00221

5. Click the Find First button. If you do not have a Scandinavian keyboard, Access displays the "finished searching" message box shown in Figure 6.6.



FIG. 6.5 Attempting to find Luleå in the City field.



FIG. 6.6 The message box that appears when Access cannot find a match to the content of the Find What text box.

The "finished searching" message indicates that the Find feature did not locate a match between the present position of the record pointer and the last record of the table. The reason Access missed your entry is that the Scandinavian diacritical ° is missing over the *a* in *Lulea*. In the ANSI character set, "a" has a value of 97, and "å" has a value of 229.

> **Note**
>
> You can enter international (extended) characters in the Find What text box by typing the English letters and then using Windows 95's or Windows NT 4.0's Character Map (Charmap.exe) applet to find and copy the extended character to the Clipboard. (Don't worry about choosing the correct font.) Paste the character into the Find What text box at the appropriate location.

If the letters preceding an extended character are sufficient to define your search parameter, follow this set of steps to find Luleå:

1. Type **Lule**, deleting the "a" from Lulea, in the Find What text box.
2. Select Start of Field from the Match drop-down list.
3. Click the Find First button. Access finds and highlights Luleå in the City field, as shown in Figure 6.7.

You also can find entries in any part of the field. If you type **ule** in the Find What text box and choose Any Part of Field from the Match drop-down list, you get a match on Luleå. However, you would also match Thule, the location of the Bluie West One Airfield (also known as Thule AFB) in Greenland. (There is no actual entry for Thule in the Customers table.)

Oracle Ex. 1009, p. 210
Oracle v. DeBellis
IPR2024-00221



FIG. 6.7 Finding a record that contains a special character.

Note

Searching all fields for an entry is usually much slower than searching a single field, especially if you have an index on the field being searched. Unless you specify the Any Part of Field option, Access uses the index to speed the searching operation.

Following is a list of the options available in the Find in Field dialog:

- To specify a case-sensitive search, mark the Match Case check box.
- To search using the field's format, mark the Search Fields as Formatted check box. This enables you to enter a search term that matches the formatted appearance of the field rather than the native (unformatted) value if you applied a Format property value to the field. Using the Search Fields as Formatted option slows the search operation; indexes are not used.
- To find additional matches, if any, click the Find Next button. If the Search option is set to Down, then clicking the Find First button starts the search at the first record in the table, regardless of the current position of the record pointer. If the Search option is set to All, then clicking the Find First button starts the search at the current record pointer.
- To start the search at the last record of the table, select Up in the Search drop-down list.

SQL has no direct equivalent to a Find First or Find Next operation. SQL is a set-oriented language, so the following SQL statement

```
SELECT * FROM Customers WHERE City = "Luleå"
```

returns the set of all records that meet the criterion. The SQL `CURSOR` construct enables you to move between records of a set, but the `CURSOR` reserved word is not supported in Access SQL and its use is beyond the scope of the SQL discussion in this chapter.

Oracle Ex. 1009, p. 211
Oracle v. DeBellis
IPR2024-00221

Replacing Matched Field Values Automatically

A variation on the Find in field dialog's theme—the Replace in field dialog—enables you to replace values selectively in fields that match the entry in the Find What text box. To display the Replace in Field dialog, choose Edit, Replace, or press Ctrl+H. (No button on the standard Table Datasheet toolbar exists for the search and replace feature.) The derivation of the shortcut key combination for Edit, Replace—Ctrl+H—is a mystery.

The entries to search for *Luleå* and replace with *Lulea* appear in Figure 6.8. To replace entries selectively, click the Find Next button, and then click the Replace button for those records in which you want to replace the value. You can do a bulk replace in all matching records by clicking the Replace All button.



FIG. 6.8 The Replace in Field dialog.

▶▶ See "Updating Values of Multiple Records in a Table," p. 383

In SQL, the statement to replace all occurrences of *Luleå* with *Lulea* is as follows:

```
UPDATE Customers SET City = 'Lulea' WHERE City = 'Luleå'
```

▶▶ See "Specifying Action Query Syntax," p. 860

Filtering Table Data

Access enables you to apply a filter to specify the records that appear in Datasheet view of a table or a query result set. For example, if you wanted to view only those customers located in Germany, you would use a filter to limit the displayed records to only those whose Country field contains the text `Germany`. Access gives you three different ways to apply filters to the data in a table:

- *Filter by Selection.* This is the fastest and simplest way to apply a filter. In a filter by selection, you establish the filter criteria by selecting all or part of the data in one of the table's fields; Access displays only records that match the selected sample. You can only filter records based on criteria in a single field of the table with a filter by selection.

Oracle Ex. 1009, p. 212
Oracle v. DeBellis
IPR2024-00221

- *Filter by Form.* This is the second-fastest way to apply a filter. In a filter by form, you enter the filter criteria into a blank datasheet form of the table; Access displays records that match the combined criteria in each field. Use a filter by form to quickly filter records based on criteria in more than one field.
- *Advanced Filter/Sort.* This is the the most powerful, and also the most difficult type of filter to use. With an advanced filter/sort you can make an Access filter do double-duty because you also can add a sort order on one or more fields.

Each of the three available types of filters is described in the next few sections.

Filtering by Selection

Creating a filter by selection is as easy as selecting (highlighting) text in a field. When you apply the filter, Access uses the selected text to determine which records to display. Table 6.1 summarizes what records are displayed, depending on how you select text in the field. In all cases, Access applies the filter criteria only to the field in which you have selected text. Filter by selection only allows you to establish filter criteria for a single field at one time.

Table 6.1 How Selected Text Affects Filter by Selection

Selected Text	Filter Effect
Entire field	Displays only records whose fields contain exactly matching values.
Beginning of field	Displays records where the text at the beginning of the field matches the selected text.
End of field	Displays records where the text at the end of the field matches the selected text.
Characters anywhere in field, except beginning or end	Displays records where any part of the field matches anywhere in the selected text.

To create a Filter by Selection on the Customers table (displaying only those customers located in Germany), follow these steps:

1. If necessary, open the Customers table in Datasheet view, and use the scroll bars to make the Country field column visible in the table window.
2. Click the First Record button to make the first record in the table the active record.



3. Select all of the text in the Country field of the first record in the Customers table. (This should be Germany.)
4. Click the Filter by Selection button on the toolbar, or choose Records, Filter, Filter by Selection. Access applies the filter, as shown in Figure 6.9.



Notice that the Apply Filter button on the toolbar is now displayed in a "down" position, indicating that a filter is being applied to the table and the ScreenTip changes to `Remove Filter`. The legend (`Filtered`) is also added to the record selection and information bar at the bottom of the table window.



Oracle Ex. 1009, p. 213
Oracle v. DeBellis
IPR2024-00221

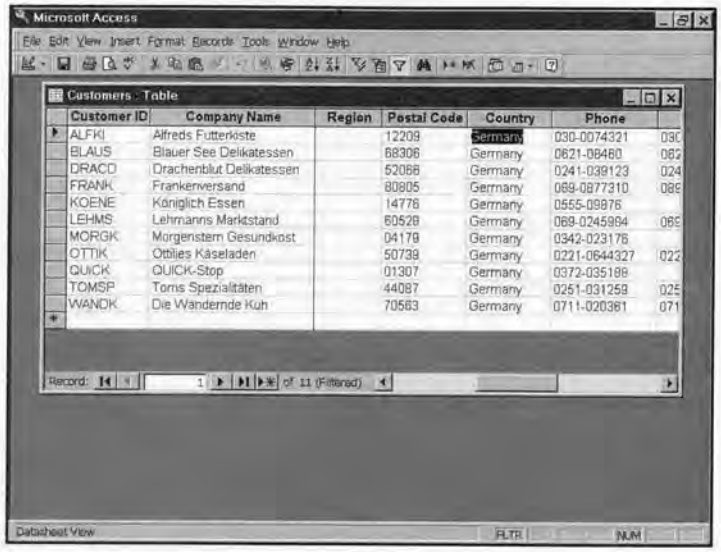


FIG. 6.9 A Filter by Selection applied to the Customers Table to display only those customers in Germany.

Tip

Use the Find dialog to quickly locate the first record of a group you're interested in filtering, and then apply a filter by selection.

As mentioned previously, you can also apply a filter by selection based on partially selected text in a field. Figure 6.10 shows the Customers table with a different filter by selection applied—this time, only the letters *er* in the Country field were selected.



FIG. 6.10 The Customers table, this time filtered by selecting the letters "er" in the Country field.

Oracle Ex. 1009, p. 214
Oracle v. DeBellis
IPR2024-00221

Tip

You can apply a filter by selection to more than one field at a time. For example, after applying a filter by selection to display only those customers located in Germany, you could then move to the City field and apply a second filter by selection for Berlin. The resulting table display will then include only those customers in both Germany and in Berlin. An easier way to apply filters based on more than one field value is to use a Filter by Form, described in the next section of this chapter.

Note

Run

To remove a filter, click the Remove Filter button on the toolbar. This is really the same as the Apply Filter button—the button is "down" whenever a filter is in effect, and "up" otherwise.

Filtering by Form

Filtering by form is slightly more complex than filtering by selection, but allows you to filter records based on criteria in more than one field at a time. For example, you saw in the preceding section how to use a filter by selection to view only those customers located in Germany. To further limit the displayed records to those customers located in Berlin, Germany, use a Filter by Form.

▶▶ See "Understanding the Elements in Expressions," p. 288
▶▶ See "Creating Access Expressions," p. 309
▶▶ See "Expressions for Query Criteria," p. 310



In a Filter by Form, Access presents you with a blank form for the table (see Figure 6.11). This window is called a *form* to distinguish it from the Table Design grid and the table datasheet windows, although it is not the same as the data-entry forms discussed later in this book. You can use any query expression you want in the fields of the Filter by Form window. Query expressions are described Chapter 9, "Understanding Operators and Expressions in Access." You can combine criteria in a Filter by Form with either a logical **Or** condition or a logical **And** condition. For example, you can filter the Customers table to display only those customers in the United States or in Canada. As another example, you could filter the Customers table to display only those customers in the United States and in ZIP codes beginning with the digit 9 (such as 94609 or 90807).

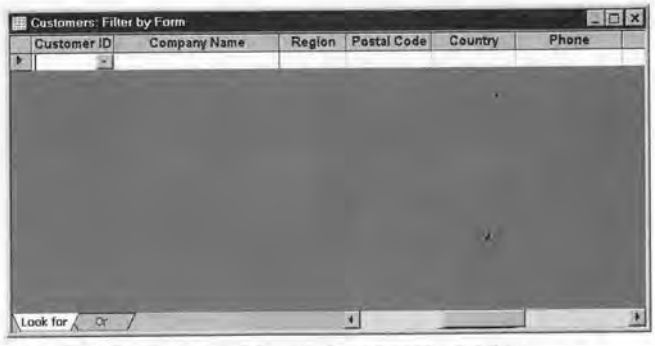


FIG. 6.11 An empty Filter by Form window for the Customers table.

Oracle Ex. 1009, p. 215
Oracle v. DeBellis
IPR2024-00221

To create a Filter by Form on the Customers table (displaying only those customers located in the United States or Canada), follow these steps:

1. If necessary, open the Customers table in Datasheet view, and use the scroll bars to make the Country field column visible in the table window.



2. Click the Filter by Form button on the toolbar, or choose Records, Filter, Filter by Form. Access displays the Filter by Form window (refer to Figure 6.11).
3. Use the scroll bars to make the Country field visible in the Filter by Form window, if necessary. (The Customer ID and Company Name columns in the figures have been frozen, as described previously in this chapter.)
4. Click inside the Country field to move the caret to that field, and then click the arrow to open the Country list box, or press F4. The drop-down list contains a list of all the unique values in the Country field (see Figure 6.12).
5. Select Canada in the list box, as shown in Figure 6.12. Access automatically adds the quotation marks around the value you select, and enters it into the Country field form box.
6. Click the Or tab at the bottom of the Filter by Form window. Access combines criteria that you enter on separate tabs in the Filter by Form window with a logical Or condition.
7. Click the arrow to open the Country list box, or press F4, and select USA from the drop-down list (see Figure 6.13).
8. Click the Apply Filter button, or choose Filter, Apply Filter/Sort. Access applies the new filter to the table, displaying the records shown in Figure 6.14.

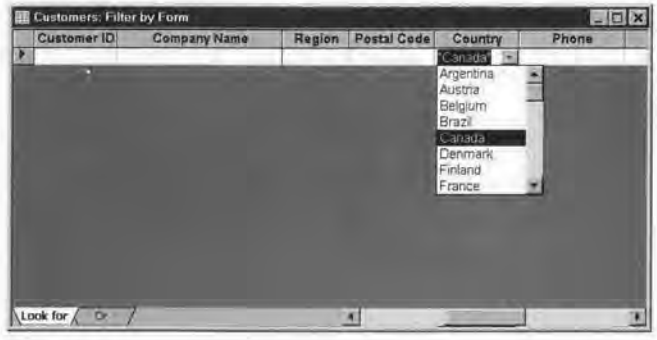


FIG. 6.12 Selecting the first country to filter for in the Customers table.

You can also combine filter criteria in a logical **And** condition by entering criteria in more than one field on the same tab of the form window. For example, say you wanted to

Oracle Ex. 1009, p. 216
Oracle v. DeBellis
IPR2024-00221

filter the Orders table to find all orders handled by Nancy Davolio and shipped to France. You could easily use a Filter by Form to do so, as the following example shows:

1. Open the Orders table, if necessary, and freeze the Order ID, Customer, and Employee columns, then use the field scroll bars to position the Ship Country field so that it is visible in the window (see Figure 6.15). (Freezing the columns isn't an essential step, but it makes setting up the filter and viewing the filtered data easier.)
2. Click the Filter by Form button on the toolbar, or choose Records, Filter, Filter by Form. Access displays the Filter by Form window.
3. Click the Clear Grid button on the toolbar, or choose Edit, Clear Filter to clear any previous filter criteria from the Filter by Form grid.
4. Use the drop-down list in the Employee field to select `Davolio, Nancy`, and then use the drop-down list in the Ship Country field to select `France` (see Figure 6.16).
5. Click the Apply Filter button, or choose Filter, Apply Filter/Sort. Access applies the new filter to the table, displaying the records shown in Figure 6.17.

   This filter only shows those records for orders that were both handled by Nancy Davolio and were shipped to France.



FIG. 6.13 Selecting the Or condition for the Filter by Form in the Customers table.



FIG. 6.14 The result of the Filter by Form displays only those records for customers in Canada or the USA.

Oracle Ex. 1009, p. 217
Oracle v. DeBellis
IPR2024-00221

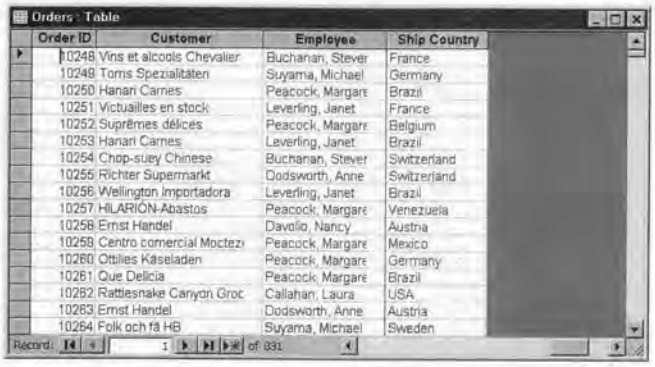


FIG. 6.15 The Orders table with the Order ID, Customer, and Employee columns frozen, and the Ship Country scrolled into visibility.



FIG. 6.16 Combining criteria in two fields in a logical And condition.

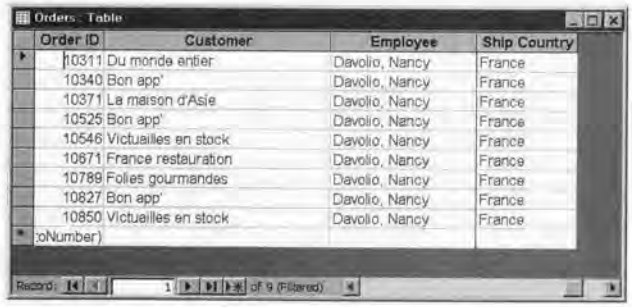


FIG. 6.17 The Orders table records displayed by the Filter by Form shown in Figure 6.16.

Troubleshooting

My Filter by Form filters don't produce the records I expect. I'm seeing either too few records, or records extraneous to the filter I've selected.

Access keeps your last filter settings for a table until you close the table. If you've applied a different filter, whether through Filter by Selection or Filter by Form earlier in your current work session, Access may be applying additional filter criteria that you're not expecting. Use the Clear Grid button or choose Edit, Clear Filter in Filter by Form to clear all previous filter criteria and ensure that the new filter criteria you enter are the only ones in effect.

Oracle Ex. 1009, p. 218
Oracle v. DeBellis
IPR2024-00221

Advanced Filters and Sorts

Filters in Access, as mentioned previously, are queries in disguise. Creating an Advanced Filter/Sort is very much like creating a query, with some basic differences. The basic differences between the Filter Design window and the Query design window are as follows:

- The Show Table dialog does not appear.
- The SQL button is missing from the toolbar, so you can't display the underlying SQL statement.
- The Show row is missing from the Filter Design grid.

Filters are limited to using one table or query that Access automatically specifies when you enter Filter Design mode. You can save a filter you create as a query, or load a filter from a query, but Access has no provision for saving a filter as a filter. The sections that follow describe how to add criteria to filter records and to add a sort order in the Filter Design window.

Adding a Multi-Field Sort and Compound Filter Criteria

In its default configuration, the Datasheet toolbar doesn't have an Advanced Filter/Sort button. Instead, you start the Advanced Filter/Sort operation by choosing Records, Filter, Advanced Filter/Sort. To create a filter on the Orders table (which provides more records to filter than the Customers table), follow these steps:

1. Open the Orders table in Datasheet view.
2. Choose Records, Filter, Advanced Filter/Sort to display the Filter window. The Filter window appears as shown in Figure 6.18. The default filter name, `Filter1`, is concatenated with the table name to create the default name of the first filter, OrdersFilter1. The field list window for the Orders table appears in the upper pane of the Filter window.

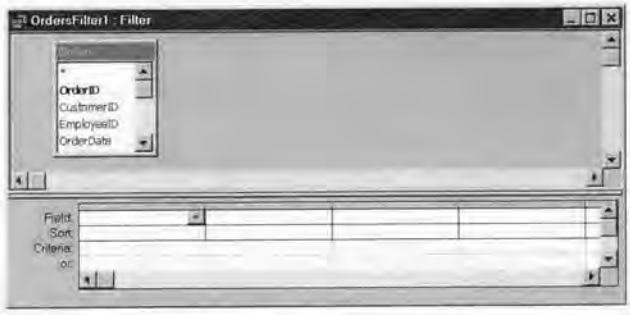


FIG. 6.18 The Filter Design window opened by choosing Records, Filter, Advanced Filter/Sort.

3. One of the fields that you might want to use to sort or limit displayed records is OrderID. Click the OrderID field in the Orders field list window in the upper pane and drag the field to the first column of the Fields row of the Filter Design grid in the lower pane and drop it. (When your mouse pointer reaches the lower pane, it turns into a field symbol.)

Oracle Ex. 1009, p. 219
Oracle v. DeBellis
IPR2024-00221

4. Repeat step 3 for other fields on which you want to sort or establish criteria. Candidates are CustomerID, ShipName, ShipCountry, ShipPostalCode, OrderDate, and ShippedDate fields.

5. Add an ascending sort to the ShipCountry and ShipPostalCode fields to check the sorting capabilities of your first advanced filter. Your Filter Design window appears as shown in Figure 6.19.

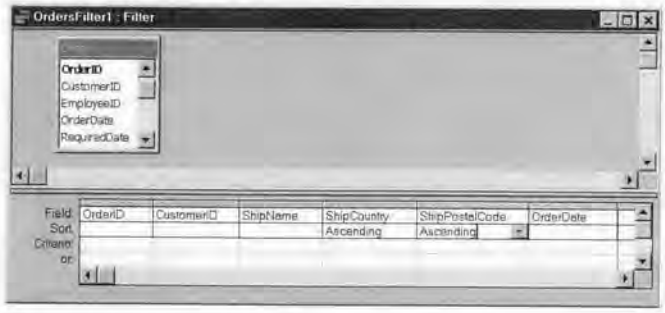


FIG. 6.19 Adding fields and sort orders to the Filter design window.



6. Click the Apply Filter button of the toolbar, or choose Filter, Apply Filter/Sort.

7. Use the horizontal scroll bar of the datasheet to reveal the ShipCountry and ShipPostalCode fields. Your sorted table appears as shown in Figure 6.20.



FIG. 6.20 The Orders table, ordered by the ShipCountry and ShipPostalCode fields.

8. Choose Records, Filter, Advanced Filter/Sort to display the Filter Design window so you can edit the filter criteria. Access displays the Filter Design window with all of the criteria from the preceding filter already entered.

9. Type **USA** in the Criteria row of the ShipCountry column to limit records to those orders shipped to an address in the United States. Access automatically adds double-quotes (") around "USA", indicating that the entry is text, not a number.

Oracle Ex. 1009, p. 220
Oracle v. DeBellis
IPR2024-00221

10. Click the Apply Filter button of the toolbar or choose Filter, Apply, Filter/Sort and scroll to display the sorted fields. Only records with destinations in the United States appear, as shown in Figure 6.21. (The first three columns of the table have been frozen, and the Ship Country and Ship Postal Code columns have been scrolled into visibility to achieve the table appearance in Figure 6.21.)



FIG. 6.21 The result of applying a "USA" criterion to the Ship Country field.

Tip

Although the Datasheet toolbar doesn't have an Advanced Filter/Sort command button in its default configuration, you can customize the Datasheet toolbar to add an Advanced Filter/Sort button to it.

Using Composite Criteria

You can apply composite criteria to expand or further limit the records that Access displays. Composite criteria are applied to more than one field. To display all orders received on or after 1/1/95 with destinations in North America, try the following:

1. Choose Records, Filter, Advanced Filter/Sort to display the Filter Design window.
2. Type **Canada** in the second criteria line of the ShipCountry column and **Mexico** in the third line, and then move the caret to a different cell. When you add criteria under one another, the effect is to make the criteria alternative—that is, combined in a logical Or condition. (Adding criteria in successive rows is the equivalent of using the OR operator in SQL.)
3. Type **>=#1/1/95#** in the first criteria line of the OrderDate field. When you add criteria on the same line as another criterion, the criteria is additive (a logical And condition); that is, orders for the United States placed on or after 1/1/95. (Adding criteria in the same row is the equivalent of using the SQL AND operator.) The # symbols indicate to Access that the enclosed value is of the Date/Time data type.

Oracle Ex. 1009, p. 221
Oracle v. DeBellis
IPR2024-00221

4. Press F2 to select the entry of step 2, and then press Ctrl+C to copy the expression to the Clipboard. Position the caret in the second row of the Order Date column, and press Ctrl+V to add the same expression for Canada. Repeat this process to add the date criterion for Mexican orders. Your Filter Design grid now appears as shown in Figure 6.22. You need to repeat the date criterion for each country criterion because of a limitation in constructing SQL statements from QBE grids, which is discussed shortly.



5. Click the Apply Filter button to display your newly filtered datasheet (see Figure 6.23).

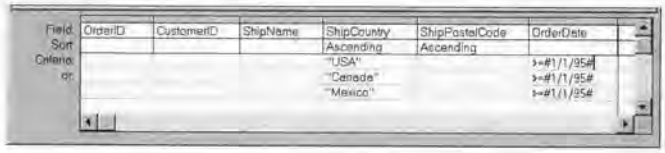


FIG. 6.22 The Filter grid with composite criteria added.

The SQL statement equivalent to this filter/sort combination is as follows:

```
SELECT * FROM Orders
   WHERE ([Ship Country] = 'USA' AND [Order Date] >= #1/1/95#)
      OR ([Ship Country] = 'Canada' AND [Order Date] >= #1/1/95#)
      OR ([Ship Country] = 'Mexico' AND [Order Date] >= #1/1/95#)
   ORDER BY [Ship Country], [Ship Postal Code]
```

The following is a more efficient SQL statement that accomplishes the same objective:

```
SELECT * FROM Orders
   WHERE ([Ship Country] = 'USA'
      OR [Ship Country] = 'Canada'
      OR [Ship Country] = 'Mexico')
      AND [Order Date] >= #1/1/95#
   ORDER BY [Ship Country], [Ship Postal Code]
```



FIG. 6.23 The result of the filter of Figure 6.22 applied to the Orders datasheet.

Oracle Ex. 1009, p. 222
Oracle v. DeBellis
IPR2024-00221

A statement using the SQL IN predicate is even simpler:

```
SELECT * FROM Orders
  WHERE [Ship Country] IN('USA', 'Canada', 'Mexico')
    AND [Order Date] >= #1/1/94#
  ORDER BY [Ship Country], [Ship Postal Code]
```

You can't generate either of the more efficient forms of the SQL statement with QBE because Access has to take into consideration that you might want a different range of order dates for each country.

Saving Your Filter as a Query and Loading a Filter

Access does not have a persistent Filter object. A *persistent* database object is an object you create that is stored as a component of your database's .mdb file. All persistent database objects appear as items in one of the list boxes of the Database window. A filter is the equivalent of a single-table query, so Access lets you save your filter as a QueryDef object. Access saves the names of the filters associated with each table in the system tables of your database when you save a filter as a query. This is the principle advantage of using a filter rather than a query when only a single table is involved.

To save your filter and remove the filter from the Orders table, follow these steps:

1. Choose Records, Filter, Advanced Filter/Sort to display the Filter design window, if it is not already displayed.
2. Choose File, Save As Query to display the Save As Query dialog, or click the Save As Query toolbar button.
3. Enter a descriptive name for your filter in the Query Name text box. Using the **flt** prefix distinguishes the filters you save from conventional queries (see Figure 6.24).



FIG. 6.24 Naming the QueryDef object that contains a filter.

4. Click OK to save the filter.
5. Click the Close Window button to close the Filter window.
6. Click the Remove Filter on the toolbar or choose Records, Remove Filter/Sort to remove the filter from the Orders datasheet.



7. A filter remains in memory while the table to which it applies is open. To close the filter, click the Close Window button to close the Orders table.

Re-creating a filter from the filter you saved as a query requires the following steps:

1. Reopen the Orders table in Datasheet view.
2. Choose Records, Filter, Advanced Filter/Sort to open the Filters window with an empty filter.

Oracle Ex. 1009, p. 223
Oracle v. DeBellis
IPR2024-00221

3. Click the Load from Query toolbar button, or choose File, Load from Query to display the Applicable Filter dialog, shown in Figure 6.25.



FIG. 6.25 A saved filter listed in the Applicable Filter dialog.

4. Double-click the fltOrdersNorthAmerica filter to load the saved query into the Filter window.



5. Click the Apply Filter button on the toolbar to display the resulting filter set in the Orders datasheet.

You can save the preceding steps by simply executing the saved query. You execute a query the same way you open a table:

1. Close the Orders table.



2. Click the Queries tab of the Database window to list the saved queries.
3. Double-click the fltOrdersNorthAmerica item. The datasheet of the fltOrdersNorthAmerica: Select Query window that appears is identical to the datasheet you created in step 5 of the preceding operation.



4. Click the Design View button of the toolbar to display the query design, shown in Figure 6.26. Notice that the columns of your original filter, in which no selection criteria or sort order were entered, don't appear in the query's grid.
5. Choose View, SQL to display the SQL statement behind the query, illustrated by Figure 6.27.

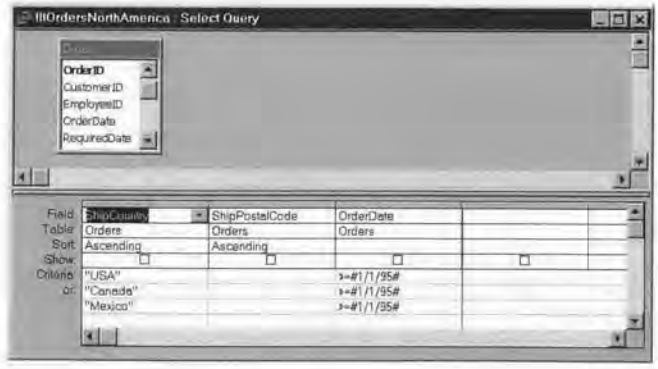


FIG. 6.26 The Query Design view of a saved filter.

Oracle Ex. 1009, p. 224
Oracle v. DeBellis
IPR2024-00221

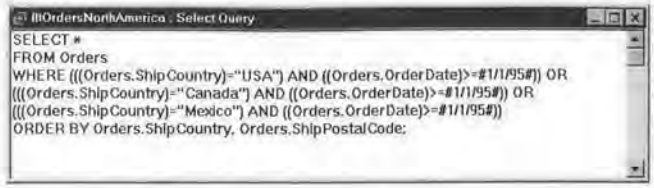


FIG. 6.27 The SQL statement for the filter saved as a query.

Access adds a multitude of parentheses, plus table name qualifiers, to the field names of the statements created from QBE grids. Most of the parentheses are superfluous. (They are present to help the Jet database engine's query parser execute more complex queries.) Table name qualifiers are not necessary in an SQL statement when only one table is included in the `FROM` clause.

Customizing Datasheet View

You can customize the appearance of Datasheet view by hiding the fields you don't want to appear in your datasheet, changing the height of the record rows, eliminating the grid lines, and selecting a different font for your display. The following list describes each of the options for customizing Table and Query Datasheet view:

- To hide a field, select the field by clicking the field header or placing the caret in the column for the field and then choose Format, Hide Columns. Alternatively, you can use the Unhide Columns method described in the next option.
- To show a hidden field, choose Format, Unhide Columns to display the Unhide Columns dialog shown in Figure 6.28. Columns that appear in Datasheet view are indicated by a check mark next to the field name in the Column list. Click the box to the left of the field name to toggle between hiding and showing the column.



FIG. 6.28 The Unhide Columns dialog, which allows you to show and hide datasheet fields.

- To change the font used to display and print the datasheet, use the Font drop-down list on the Formatting toolbar (if it is displayed), or choose Format, Font to display the Font dialog, shown in Figure 6.29. (The Font dialog is one of the common dialogs of Windows 95 and Windows NT 4.0. Other common dialogs include the Open and Save dialogs.)

Oracle Ex. 1009, p. 225
Oracle v. DeBellis
IPR2024-00221



FIG. 6.29 Choosing a display and printing font for datasheets.



- To remove gridlines from the display and printed versions of the datasheet, click the Gridlines Shown drop-down list button on the Formatting toolbar, or choose Format, Cells. If you choose the Gridlines Shown drop-down list, Access displays a palette of four gridline display choices: Both, Horizontal, Vertical, and None; click the button corresponding to the gridline display you want. If you choose Format, Cells, Access displays the Cells Effects dialog, which contains check boxes for the horizontal and vertical gridlines; select or clear the check boxes to obtain the desired gridline display.
- To change the height of the rows as displayed and printed, position the mouse pointer at the bottom edge of one of the record selector buttons. The mouse pointer turns into a double-headed arrow, as shown in Figure 6.30. Drag the bottom edge of the button to adjust the height of all the rows. Alternatively, choose Format, Row Height and set the height in points in the Row Height dialog. (Multiply the size of your font by about 1.25 to obtain normal row spacing; printers call 10-point type with 12-point spacing 10 on 12.)
- To change the width of the columns to accommodate a larger font, choose Format, Column Width and then click the Best Fit button to let Access determine the size of your columns. You might need to adjust individual column widths by dragging the right edge of the field header with the mouse.

Figure 6.30 shows the Orders datasheet with several columns hidden, gridlines off, 9-point Century Schoolbook TrueType font, and the height of the rows adjusted to accommodate the larger font.

Tip

For the greatest printing speed, choose a typeface family native to your printer, such as Helvetica for PostScript or Swiss for LaserJet printers. (Native fonts are indicated by a printer and page symbol next to the typeface family name in the Font list.) Alternatively, choose a TrueType face, such as the default Arial, for both display and printing.

Oracle Ex. 1009, p. 226
Oracle v. DeBellis
IPR2024-00221



FIG. 6.30 The Orders datasheet in a customized view.

Copying, Exporting, and Mailing Sorted and Filtered Data

A primary use for filters and customized datasheets is so that you can export the filtered records to another application, such as Microsoft Excel or Word. A variety of methods for exporting filtered and custom-formatted records is available, each of which is described in the following list:

▶▶ See "Exporting Data from Access Tables," p. 253



- Copy the entire datasheet to the Clipboard, and then paste it into the other application. Hidden columns don't appear, but formatting (font, font attributes, and row height) is preserved.
- Use the Save As/Export feature to export the datasheet to an Excel worksheet (.xls) or a rich text format (.rtf) file for Word or other Windows word processing applications. (Choose File, Save As/Export, select To an external File or Database in the Save As dialog, and then select the file type you want in the Save As Type drop-down list of the Save Table In dialog.) Save As/Export preserves the attributes you use to customize the filtered and sorted data when you choose Excel format. Hidden columns, however, appear when you open the resulting file in any version of Excel.
- Choose Tools, Office Links, Analyze It with MS Excel to save the filtered or sorted data in an Excel worksheet, or choose Tools, Office Links, Publish It with MS Word to save the data as an .rtf document. Whether you choose to Analyze It or Publish It, Access starts Excel or Word (whichever is appropriate) with the exported document displayed.
- Choose Tools, Office Links, Merge It with MS Word to create form letters with Microsoft Word. Using Mail Merge with Microsoft Word is discussed in Chapter 22, "Using Access with Microsoft Word and Mail Merge."
- Send the file as an attachment to a Microsoft Mail or Exchange message. Hidden columns don't appear, but formatting is not preserved in Microsoft Mail messages. (The attached file is in Excel BIFF format.)

Oracle Ex. 1009, p. 227
Oracle v. DeBellis
IPR2024-00221

If you make the Database window the active window and choose File, Save As/Export, the entire content of the table is exported without regard to the filter you added.

Troubleshooting

When attempting to use the Merge It With Word feature, a `Must save object first` *or* `Name not found in this collection` *message box appears.*

You applied a filter or created a query for your Mail Merge operation that you did not save before clicking the Merge It button. The filter or query object (filters are saved as QueryDef objects) must be a member of the `QueryDefs` collection before the DDE conversation between Access and Word commences. Save the filter or the query, and then try clicking the Merge It button again. Chapter 29, "Understanding the Data Access Object Class," provides detailed information on database objects and object collections.

Oracle Ex. 1009, p. 228
Oracle v. DeBellis
IPR2024-00221

Chapter 7

Linking, Importing, and Exporting Tables

Undoubtedly, more than 90 percent of personal computer users have data that can be processed by database management techniques. Any data a computer can arrange in tabular form, even tables in word processing files, can be converted to database tables. The strength of a relational database management system (RDBMS) lies in its capability to handle large numbers of individual pieces of data stored in tables, and to relate the pieces of data in a meaningful way.

PC users acquire RDBMSs when the amount of data created exceeds the application's capability to manipulate the data effectively. A common example of this situation is a large mailing list created in a word processing application. As the number of names in the list increases, using the word processor to make selective mailings and maintain histories of responses to mailings becomes increasingly difficult. A PC RDBMS is the most effective type of application for manipulating large lists.

One Access strong point is its capability to transform existing database tables, spreadsheets, and text files created by other DOS and Windows applications into the Access .mdb format—a process known as *importing* a file. Access can *export* (create) table files in any format in which it can import the files. Most PC RDBMSs share this capability, but Access can import and export Borland Paradox files—many other systems cannot. Most client/server RDBMSs can import and export only text-type files.

Access can link a database table file created by Access or another RDBMS to your current Access database; Access then acts as a database front end. Because Access has a linking capability, it can use a file created by another RDBMS in its native form. This capability is far less common in other PC and client/server RDBMSs. When you link a database table from a different RDBMS, you can display and update the linked table as if it were an Access table contained in your .mdb file. If the file that contains the table is shared

Oracle Ex. 1009, p. 229
Oracle v. DeBellis
IPR2024-00221

on a network, other users can employ the file with their applications, while it is linked to your database. This capability to link files is an important feature for two reasons: you can have only one Access database open at a time, and you can create new applications in Access that can coexist with applications created by other database managers.

This chapter deals primarily with what are known as *desktop database-development applications*—a term used to distinguish them from client/server RDBMSs, such as Microsoft and Sybase SQL Server, ORACLE, Informix, and Ingres databases. (Client/server RDBMSs are designed specifically for use with networked PCs and—except for Microsoft SQL Server for Windows NT—require you to set aside a PC for use as a database application and file server to run the RDBMS and store the database files.) Desktop RDBMSs, such as dBASE, FoxPro, and Paradox, are more widely used than client/server systems. The majority of desktop RDBMSs can share files on a network, but several publishers of desktop RDBMSs require that you purchase a special multiuser version of the RDBMS to do so. Multiuser desktop RDBMSs—while accommodating the workstation-server configuration required by conventional networks such as Novell NetWare, Windows NT Server, or IBM LAN Server—are especially well-suited to the peer-to-peer networks discussed in Chapter 25, "Securing Multiuser Network Applications," such as Windows 95 peer networking, NetWare Lite, and LANtastic. Chapter 25 explains how to use Access with shared database files in general, and Chapter 26, "Connecting to Client/Server Databases," deals with client/server databases in particular.

Learning How Access Handles Tables in Other Database File Formats

Conventional desktop database development applications maintain each table in an individual file. Each file contains a header followed by the data. A *header* is a group of bytes that provides information on the structure of the file, such as the names and types of fields, number of records in the table, and length of the file. When you create a table file in dBASE, FoxPro, or Paradox, for example, the file contains only a header. As you add records to the file, the file grows by the number of bytes required for one record, and the header is updated to reflect the new file size and record count.

Desktop RDBMSs create a variety of supplemental files, which are shown in the following list—some of which are required to import, link, or export RDBMSs:

- Paradox stores information about the primary-key index file (.px) in the associated table (.db) file; the .px file for the .db file must be available for Access to open a Paradox .db file for updating. Access links the .px file automatically if it exists.
- dBASE and FoxPro store memo-type data in a separate .dbt file. If a dBASE table file contains a memo field, the .dbt file must be available. If the .dbt file is missing, you cannot import or link dBASE or FoxPro tables that contain a memo field.
- Use of .ndx (dBASE III), .mdx (dBASE IV), or .idx or .cdx (FoxPro) index files is optional. You always should use index files when you have them. If you don't link

Oracle Ex. 1009, p. 230
Oracle v. DeBellis
IPR2024-00221

the index files when you link an indexed .dbf table file, modifications you make to the linked tables aren't reflected in the index, which causes errors to occur when you try to use the indexed tables with dBASE or FoxPro.

All supplemental files must be in the same folder as the related database file to be used by Access.

The header of an Access 97 .mdb file differs from conventional PC RDBMS files in that an .mdb header consists of a collection of system tables that contain information on all the tables, indexes, macros, and Access VBA functions and procedures stored in a single Access file. The Access system tables also contain information on the location and characteristics of other PC RDBMS files that you linked to your Access database. Access's system tables are similar to the tables used in client/server databases that maintain information on the content of database devices (files), plus the databases and tables contained in the devices.

Note

You can view the Access 97 system tables by choosing Tools, Options. Select the View tab, and in the Show group, select System Objects.

Never modify anything in these tables (most of them are read-only). Some database developers have used the data and values in these tables to aid in referencing items in the database. This is not a good practice, because the design of these tables is not guaranteed to remain consistent from version to version and could result in substantial rework to convert a database application to a new version of Access.

Identifying PC Database File Formats

Access can import, link, and export the following types of database table files used by the most common PC database managers:

- *dBASE .dbf table and .dbt memo files, and dBASE III .ndx and dBASE IV .mdx index files.* dBASE III and IV files and indexes are the standard language of the PC RDBMS industry. The majority of PC RDBMSs, and also all common spreadsheet applications, can import and export .dbf files. Most of these RDBMSs can update existing .ndx and .mdx index files, and some RDBMSs can create these index files.

  The .dbf file structure is native to other xBase clone applications such as FoxPro, but not all these RDBMSs create fully compatible dBASE file structures. Compilers like CA-Clipper have their own native index-file structures, but they can use .ndx indexes when necessary. The capability to use .mdx multiple-index files is less widespread.

  Access can link and create .ndx and .mdx files. Access updates both types of dBASE index files when you edit or add records to a linked .dbf file. The section "Setting Primary Keys and Using Index Files" later in this chapter, discusses index files in linked tables. When you export an Access table with a memo field, a .dbt memo file with the same name you assign to the dBASE file is created.

Oracle Ex. 1009, p. 231
Oracle v. DeBellis
IPR2024-00221

- *FoxPro 2+ .dbf table files.* You can import, export, and link FoxPro 2+ .dbf files and files created by earlier versions of FoxPro. The procedures for handling FoxPro 2+ .dbf files are the same as the procedures used for dBASE III and IV. Access maintains the currency of FoxPro 2+ .idx (single) and .cdx (multiple) index files.
- *Paradox 3.x, 4.x, and 5.0 .db table and .px primary key files.* You can link Paradox for DOS 3.x and 4.x table and index files, and those files created by Paradox for Windows 5.0. The following section presents the specific limitations applicable to Paradox files.

The majority of applications that use table and index files also use the standard file extensions presented in the preceding paragraphs. The dBASE memo file, for example, requires a standard extension—.dbt. Using the standard extensions for all types of files, however, is not a requirement. Some developers of xBase applications disguise their files by using arbitrary extensions. You may have to do some detective work to determine which files contain data and which are indexes.

Note

If you work in a multiuser environment, you must have exclusive access to the file you intend to import. No other user can have this file open when you initiate the importing process, and other users are denied access to the file until you close the Import dialog.

Caution

Make sure that you work on a backup, not on the original copy of the linked file, until you are certain that your updates to the data in the linked table are valid for the existing database application.

Linking and Importing External Tables

To link or import an xBase or Paradox file as a table in an open Access database, such as Northwind.mdb, follow these steps:

Note

Linking an external file to an Access database was referred to as *attaching* a table in versions of Access prior to Access 95. Don't confuse linking an external file to an Access database with OLE links; when you link an external file, you just give Access information about the external file so it knows how to open, read, and modify the data in that file.



1. Click the Show Database Window button on the toolbar or choose Window, 1 Database. Access doesn't require that all open tables are closed before you link or import a table.



2. If you have a test database that you can use for this procedure, click the Open Database button of the toolbar or choose File, Open Database; then select the test database, open it, and skip to step 5.

Oracle Ex. 1009, p. 232
Oracle v. DeBellis
IPR2024-00221

3. If you don't have a test database, create a sample to use throughout this chapter. Click the New Database button of the toolbar or choose File, New Database to display the New dialog.



4. Double-click the Blank Database icon in the New dialog; Access displays the File New Database dialog. Type a name, such as **Mdb Test.mdb**, in the File Name text box and click Create. You must wait while Access creates and tests the new database.

5. In this example, you link an external table to the database. Choose File, Get External Data, Link Tables. The Link dialog appears (see Figure 7.1); the Link dialog is a variation of the common Windows 95 and Windows NT 4.0 Open dialog. If you choose File, Get External Data, Import, the Import dialog appears.

6. Use the Files of Type drop-down list to select the type of file you want to link. (If you have a suitable Paradox table to link, select Paradox. Otherwise, select dBASE III, dBASE IV, or another database type as appropriate to the format of your table file.)

7. Double-click the name of the table you want to link or import (or click the name to select it, and then click the Link button). Access supplies the standard extensions for dBASE, FoxPro, and Paradox table files.

   If you're linking a .dbf file that has indexes, the Select Index Files dialog appears. The section "Setting Primary Keys and Using Index Files" later in this chapter explains how to use the Select Index Files dialog.

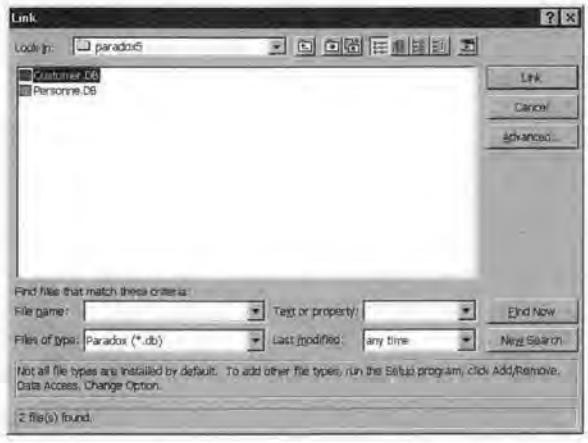


FIG. 7.1 The Link dialog used to select external files for linking to an Access database.

> **Note**
>
> You can link Access 1.x, Access 2.0, and Access 95 files to an Access 97 database. Linking Access 1.x, 2.0, and 95 files solves the problem of not being able to convert an Access 97 mdb file to any previous Access version .mdb format for backward compatibility, which is another reason to always use one .mdb file for tables and another .mdb file for your application database objects.

Oracle Ex. 1009, p. 233
Oracle v. DeBellis
IPR2024-00221

8. If the file you choose is encrypted (coded) and requires a password to decrypt it, the Password Required dialog appears. Type the password in the box and press Enter.

9. After you successfully link or import the file, a dialog appears, confirming this operation (see Figure 7.2). If you link more than one table with the same name, Access automatically appends a sequential digit to the table name.

   The Link (or Import) dialog remains open; if you have additional external tables you want to link or import to this database—most Paradox and xBase databases consist of several separate table files—repeat steps 6–9 for all the files you want to link or import. If you are linking external Access tables, you may select at once all the tables you want to link by simply clicking each one.



FIG. 7.2 Access displays this message after successfully linking an external table to a database.

10. In the Link dialog, click Close. The table(s) you linked or imported now are listed in the Database window. If you linked a file, Access adds an icon that indicates the type of database table and an arrow that indicates that the table is linked (see Figure 7.3).

11. Select the table you linked, and then click the Open button to display the records in Table Datasheet view (see Figure 7.4). Alternatively, you can double-click the table name.



FIG. 7.3 The special icons displayed in the Database window indicating linked tables.

After you link an external file as a table, you can use it almost as if it were a table in your own database. The only general limitation is that you cannot change the structure of a linked table: field names, field data types, or the Field Size properties. In linked Paradox files, Access prevents you from changing a table's primary-key field that previously was defined because this property determines the contents of the associated .px index file.

Oracle Ex. 1009, p. 234
Oracle v. DeBellis
IPR2024-00221

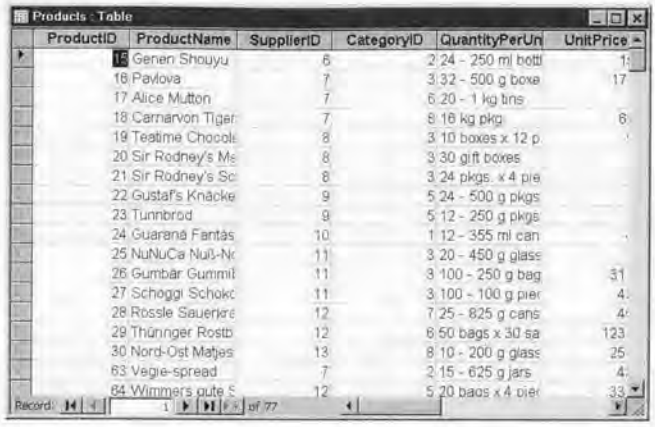


FIG. 7.4 The Datasheet view of a linked Paradox file.

> **Note**
>
> Although you cannot change field properties for linked tables, you can change the name of the attached table, *within this database only*. Choose Edit, Rename and type the new name for the table. The name for the table (called an *alias*) is changed only in the current Access database and not in the native database.

Solving Problems with Importing or Linking Files

Access detects problems with linked or imported tables that may cause errors when you try to use the tables with Access. The following sections describe these problems and how to overcome most of them.

The Incorrect Password Dialog. If you enter a wrong password or just press Enter, Access informs you that it cannot decrypt the file. You do, however, get another opportunity to enter the password or click Cancel to terminate the attempt (see Figure 7.5).



FIG. 7.5 The message indicating an incorrect password entry to a Paradox 5 table.

The Null Value in Index Dialog. Occasionally, older Paradox .px index files don't have an index value for a record; when this situation occurs, you see a warning dialog with the message, `Can't have Null value in Index`. Usually, you can disregard the message and continue linking or importing the file. The offending record, however, may not appear in the table; fixing the file in Paradox and starting over is better than ignoring the message.

Oracle Ex. 1009, p. 235
Oracle v. DeBellis
IPR2024-00221

The Missing Memo File Dialog. Both dBASE and Paradox use additional memo files to store the data from memo fields in a particular table. dBASE memo files have the .dbt file type, and Paradox memo files have the .mb file type. Access correctly decides that it can't import or link an external table if it can't open the table's associated memo file—either because the memo file doesn't exist, isn't in the same folder as the table with which it is associated, or contains nontext data.

If the table you're trying to link or import is a dBASE table, Access displays the error dialog shown in Figure 7.6. If the table you're trying to link is a Paradox table, however, you receive the less informative error message shown in Figure 7.7. For assistance in solving the problem, click the Help button. To close the error dialog, click OK; Access then cancels the linking or importation.



FIG. 7.6 The error message when Access can't open a required dBASE memo file.



FIG. 7.7 The error message when Access can't open a required Paradox memo file.

The Graphics Field Type Restriction. If the successful-link dialog doesn't appear, your table or its accompanying dBASE or Paradox memo file probably contains a graphics field type. The following section discusses how to modify files with graphics content so that you can link or import them as Access tables.

Dealing with Graphics in External Files

Most database managers designed for Windows include some form of graphics field data type. Superbase, Paradox 4.x, and Paradox for Windows provide a special field data type for graphics. Although dBASE IV lacks a field data type for graphics, third-party software firms publish applications that enable you to store graphic images in dBASE memo fields. A variety of add-on applications enables CA-Clipper programmers to display and edit graphic images. The images usually are in individual files but a few third-party applications place images in memo files. CA-dbFast, for example, can display—but not edit—images stored in Windows bitmap (.bmp) files. CA-dbFast doesn't add a graphics field type to store bitmapped data within tables.

When you try to import or link Paradox 4.x, Paradox for Windows files, or dBASE .dbt files that contain images or other binary data, you may receive an error message that the memo file is corrupted or a message that you cannot import the .db or .dbf file that

Oracle Ex. 1009, p. 236
Oracle v. DeBellis
IPR2024-00221

contains the offending memo or graphics field. In rare cases—usually involving tiny images—you can import the .dbf and .dbt files, but you see random characters in the Access memo field. With Paradox tables, the graphic or binary fields simply disappear from the table.

If a dialog appears that reports a problem during importing or linking a file, the linking or importing process is canceled.

The following recommendations can help you deal with graphic images processed with other RDBMSs and add-on applications:

- Use add-on applications for xBase clones and compilers that operate with the original graphics files in their native format, such as .tif, .pcx, .gif, or .tga. In nearly all cases, the original graphics file is on your computer's fixed disk or on a file server. You can link or embed the graphics file in an Access OLE Object field by using the techniques described in Chapter 20, "Adding Graphics to Forms and Reports."
- Do not use add-on applications that incorporate graphics in .dbt files. If you are committed to this approach, use the method that follows to place the offending memo file in a new file.
- If you use Paradox 4.x or Paradox for Windows with application development in Access, maintain files with graphics fields (as well as any OLE fields in Paradox for Windows tables) separate from files containing conventional data types.
- Use an OLE server that can process the graphics file type of the original image. Windows Paint is limited to Windows bitmap files (.bmp and .dib) and can only read—not save—.pcx files. To display the image in a form or report, you can create a reduced-size, 16-color or 256-color, Windows bitmap file to be displayed as a bound object. Chapter 20 discusses methods of handling images in this way.

To link or import an xBase file that contains a memo field or a Paradox file that contains graphics fields, you must be familiar with file-restructuring methods for dBASE or Paradox. To restructure an xBase file with a memo file that contains graphic images, follow these steps:

1. Make a copy of the file and give it a new name.
2. Modify the structure of the original file by deleting all but the related fields and the memo or graphics field of the original file. Modifying the new file with Modify Structure creates a backup of the original file with a .bak extension.
3. Modify the structure of the new file by deleting the memo or graphics field.
4. Add a field for the path and file name of the original graphics file, if it isn't already included. Access then can use the location of the original graphics file to pass the file name to an OLE server. You must write some Access VBA code, however, to handle this process. See Chapter 28, "Writing Visual Basic for Applications Code," and Chapter 30, "Responding to Events with VBA 5.0," for examples of writing Access VBA code.

Oracle Ex. 1009, p. 237
Oracle v. DeBellis
IPR2024-00221

5. Modify the source code of your original application, establishing a one-to-one relationship between the new files.

Converting Field Data Types to Access Data Types

When you import or link a file, Access reads the header of the file and converts the field data types to Access data types. Access usually is quite successful in this conversion; Access offers a greater variety of data types than most of the other widely used PC RDBMSs. Table 7.1 shows the correspondence of field data types between dBASE, Paradox, and Access files.

Table 7.1 Field Data Type Conversion Between Access and Other RDBMSs

dBASE III/IV/5	Paradox 3.x, 4.x, 5.0	Access
Character	Alphanumeric	Text (Specify Size property)
Numeric, Float*	Number, Money, BCD*	Number (Double) Number (Single) Number (Byte)
	Short Number Long Number AutoIncrement	Number (Integer) Number (Long) AutoNumber
Logical	Logical	Yes/No
Date	Date, Time, Timestamp*	Date/Time
Memo	Memo, Formatted Memo, Binary*	Memo
	OLE	OLE

**Sometimes two types of field data, separated by commas, are shown within a single column in Table 7.1. When Access exports a table that contains a data type that corresponds with one of the two field data types, the first of the two data types is assigned to the field in the exported table. The Float data type is available only in dBASE IV and 5.*



◀◀ See "Choosing Field Data Types, Sizes, and Formats," p. 119
◀◀ See "Adding Indexes to Tables," p. 154

If you are importing tables, you can change the field data type and the Field Size property to make them more suitable to the type of information contained in the field. When you change a data type or Field Size, however, follow the precautions noted in Chapter 4, "Working with Access Databases and Tables." Remember that you cannot change the field data type or Field Size property of linked tables. You can, however, use the Format property with imported or linked tables to display the data in any format compatible with the field data type of imported or linked files. You can change any remaining properties applicable to the field data type, such as validation rules and text. By using the Caption property, you can give the field a new and more descriptive name.

Oracle Ex. 1009, p. 238
Oracle v. DeBellis
IPR2024-00221

Setting Primary Keys and Using Index Files

Methods of setting primary keys and creating indexes differ according to the type of file you use to link a table. Tables based on Paradox and client/server RDBMS files usually have predefined primary-key fields and are indexed on the key fields. Files based on dBASE structures, however, don't have fields specified as primary-key fields and use separate index files. The following two sections discuss the effects these differences have on the tables you create.

Establishing Key Fields in Linked Paradox Tables. When you link or import a Paradox or Btrieve table that has a primary-key index, Access establishes this primary key as the primary key for the new table. To verify that Access establishes a primary key for a linked Paradox table, click the Design View button on the toolbar. A message box states that you cannot modify some properties of the table and asks if you want to open the table anyway (see Figure 7.8). Click Yes.



FIG. 7.8 The message box that appears when you open a linked table in Design view.

The table appears in Design mode. The primary key of the linked table is shown with the key icon next to the field name, as with primary keys for Access tables you create yourself.

When you link a Paradox .db table file that has a primary key, Access uses the Paradox .px file to establish the primary key. If you modify the values in a key field, the .db and .px files simultaneously reflect this modification.

Linking dBASE Index Files. Key-field indexing is not automatic with dBASE files linked as tables; dBASE file headers do not include data about the indexes used by the application that has the .dbf file. The first time you link the .dbf file as an Access table, you must manually link the index files associated with a dBASE file. Then, when you open the Access database with the linked table again, Access links the indexes you specify.

Note

Access cannot open or update index files in proprietary formats of xBase clones and compilers. You cannot, for example, link CA-Clipper .ntx files. CA-Clipper (5+), however, can create and maintain .ndx files as an addition to, or a substitute for, their original index structures. Access cannot use secondary indexes of Paradox tables. If you create or commission custom database applications that use nonstandard or secondary indexes and you plan to use Access to update these files while they are linked, you must modify your applications so that they use only .ndx or .px indexes. You probably don't need to make a major revision of the source code, but you may find that your present applications run more slowly with .ndx indexes.

Oracle Ex. 1009, p. 239
Oracle v. DeBellis
IPR2024-00221

When you link a dBASE file as a table in your database and select the file name for the table, Access displays a Select Index Files dialog (see Figure 7.9). When you import a dBASE file, this dialog doesn't appear.

If you select a dBASE III source file, Access supplies the default .ndx file extension in the Select Index Files dialog. Your dBASE III file may have one or more .ndx index files associated with it. The five indexes that appear in Figure 7.9 are for the Customer.dbf table (a dBASE III file, created by exporting the Customers table from the Northwind.mdb sample database).

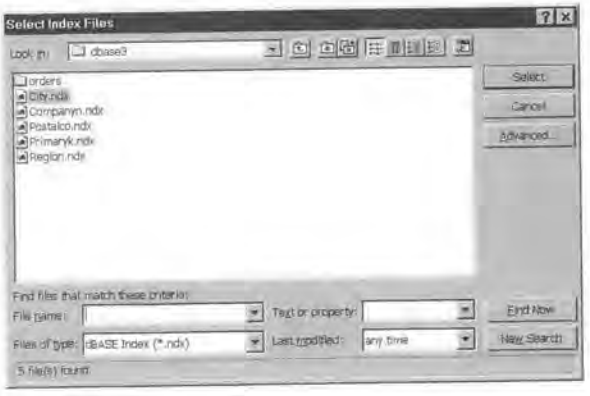


FIG. 7.9 Selecting dBASE II, III, IV, and 5 index files.

dBASE IV and 5 can create a multiple-index (.mdx) file that includes all the indexes associated with the .dbf file. The .mdx file usually has the same file name as the .dbf file. Access supplies both .mdx and .ndx as default index-file extensions when you select a dBASE IV or 5 file. The Select Index Files dialog appears whenever you select an xBase file type for linking or importing into an Access database, as described in the section "Linking and Importing External Tables," earlier in this chapter.

To link index files to tables created from dBASE files, follow these steps:

1. From the file list in the Select Index Files dialog, select the index file name, and then click Select. Access adds the index and displays a message dialog, confirming that the index was added (see Figure 7.10).

Caution

Access does not test to determine whether the table's .ndx or .mdx index file matches the structure of its associated Access .dbf file. If the index file doesn't match the Access file, Access doesn't update the index file. Unfortunately, Access does not advise you that it is ignoring the nonconforming index. If you add records to the table in Access, and then attempt to use the table with an xBase application that uses the proper index, you receive the error message, `Index does not match database`. You then must re-index the .dbf file.

Oracle Ex. 1009, p. 240
Oracle v. DeBellis
IPR2024-00221



FIG. 7.10 Confirming the addition of a dBASE index to a linked table.

Caution

When you link a dBASE file that is used by another application, you must select all indexes associated with this file if your Access application modifies fields in the index expressions of each index file. If you do not update all the associated index files while updating the .dbf file with Access, the other application may display the error message, `Index does not match file`. Worse, the application may produce erroneous or unpredictable results. If the message or errors occur, you must re-index the file in the other application.

2. If more than one index file is associated with the dBASE .dbf file, you must repeat step 1 for each .ndx file until you add the names of all the indexes needed for your Access table, and then click Close in the Select Index Files dialog. Access displays the Select Unique Record Identifier dialog (see Figure 7.11).



FIG. 7.11 The dialog for selecting a unique index for a linked dBASE file.

3. If you want Access to use one of the linked dBASE indexes to ensure that records have a unique value in the indexed field (that is, to use the dBASE index as a primary key in the table), double-click this index's name in the Indexes list of the Select Unique Record Identifier dialog.
4. Click the Close button in the Select File dialog.
5. Click the Design button in the Database window to display the structure of your linked dBASE table.



6. If the Indexes window is not visible, click the Indexes button on the toolbar or choose View, Indexes. The field names that are the basis for any linked single- or multiple-field indexes appear in the Index Name text boxes of the Indexes window (see Figure 7.12).



Oracle Ex. 1009, p. 241
Oracle v. DeBellis
IPR2024-00221

Caution

Access does not distinguish between dBASE indexes created with `SET UNIQUE ON` and `SET UNIQUE OFF`. If there are multiple identical entries in the field you select as the primary key, Access rejects your choice of that field in the Select Unique Record Identifier dialog. You must copy the dBASE file with `SET UNIQUE ON` to another file, re-index the file, and then repeat the attachment process.

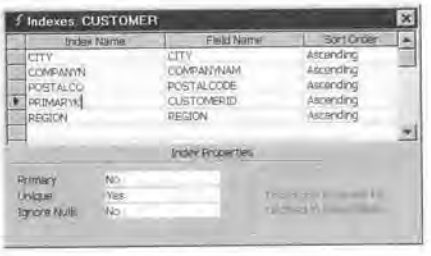


FIG. 7.12 The Indexes window of a linked dBASE table.

If you try to modify the primary-key field or change the data type or the Field Size property of a linked table, and then click the Datasheet button in Run mode, Access displays a dialog (see Figure 7.13). Click Yes. (Clicking No restores the original values.) Access doesn't change the properties in the linked file, nor do the Design and Datasheet windows display the changes.



FIG. 7.13 The message to remind you that design changes to linked tables are not permitted.

Creating Access .inf Files for dBASE Indexes. When you link one or more indexes to a table created from a dBASE file, Access creates a file with the same file name as the .dbf file but with an .inf extension. The *Filename*.inf file is a text file that contains the path and the file name of the index you linked, in the following format (the following .inf file is for the Customer dBASE III indexes shown in Figure 7.12, and shows UNC file names to a network location):

```
[dBase III]
NDX1=\\Zeus\zeusd\DOC\TESTDATA\dbase3\CITY.NDX
NDX2=\\Zeus\zeusd\DOC\TESTDATA\dbase3\COMPANYN.NDX
NDX3=\\Zeus\zeusd\DOC\TESTDATA\dbase3\POSTALCO.NDX
NDX4=\\Zeus\zeusd\DOC\TESTDATA\dbase3\PRIMARYK.NDX
NDX5=\\Zeus\zeusd\DOC\TESTDATA\dbase3\REGION.NDX

[UIDX1 PRIMARYK]
NDX5=\\Zeus\zeusd\DOC\TESTDATA\dbase3\REGION.NDX
```


Oracle Ex. 1009, p. 242
Oracle v. DeBellis
IPR2024-00221

The .inf file is located in the same folder as the .dbf file you are linking. If you create an invalid .inf file or if one or more of the indexes listed in the .inf file isn't found in the specified location, Access terminates the current link operation, and you must link the dBASE file again. Figure 7.14 shows the error message you see if Access can't find an index specified in the .inf file. When you relink the dBASE file to correct a prior error, a message box indicates that an .inf file already exists (see Figure 7.15). If the indexes changed or you moved the indexes, click Yes to create a new .inf file.

Note

If you try to find an .inf file by selecting the folder in which it should be located, the .inf extension may be missing. This is because .inf files are registered by their type, which is Setup Information.



FIG. 7.14 The warning that a linked index for a dBASE file wasn't found.

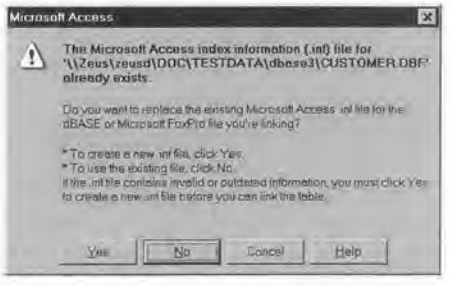


FIG. 7.15 The warning that an .inf file for a linked dBASE file already exists.

Linking Tables from Other Access Databases

The procedure for linking a table from one Access database to another Access database is like the procedure for linking other tables. To link a table from Northwind.mdb to your test database, for example, follow these steps:

1. Choose File, Get External Data, Link tables, select Northwind.mdb in the Link dialog, and click the Link button. The Link Tables dialog appears (see Figure 7.16).
2. Select the name of the table to link from the Tables list that displays the names of tables in the other Access database. You can select more than one table in this list; just click each table in the list you want to link, or click Select All to link all the tables.
3. Click OK. Access adds the linked table(s) and closes all dialogs. The name(s) of your linked Access table(s) appears in the Database window.

Oracle Ex. 1009, p. 243
Oracle v. DeBellis
IPR2024-00221

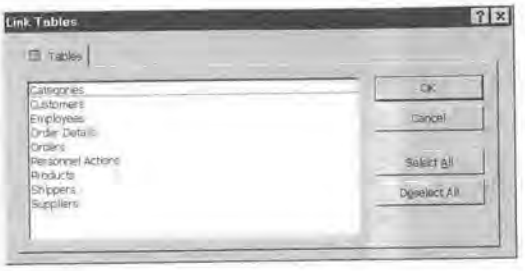


FIG. 7.16 The Link Tables dialog displaying the tables in the external .mdb file.

Access maintains a record of the drive and folder containing the files responsible for your linked tables. If you rename or change the location of a file that you linked as a table, Access no longer can find the file and displays an error dialog (see Figure 7.17).



FIG. 7.17 The dialog indicating that Access cannot find a linked file.

Using the Linked Table Manager Add-In to Relink Tables

Before Access 2.0, if you moved a file that was linked to or contained objects linked to an Access database, you had to delete the linked tables, and then relink the tables from their new location. Access 97 provides an add-in assistant known as the *Linked Table Manager*, which simplifies relinking tables. (The Linked Table Manager was known as the Attachment Manager in Access 2.0.)

If you move an Access, dBASE, FoxPro, or Paradox file that provides a table linked to an Access 97 database, choose Tools, Add-Ins, Linked Table Manager. The Linked Table Manager's window lists all the linked tables (except linked ODBC tables). Access also displays the path to the database containing the linked table(s) at the time the link was created. (You also can view the path to the database containing a linked table by opening the linked table in Design view and opening the Table Properties window.) Click the check box of the file(s) whose location(s) changed (see Figure 7.18).

Click OK to display the Select New Location of the *TableName* dialog shown in Figure 7.19. Select the folder and file where the table is located, and then click Open to change the link reference and close the dialog. If Access successfully refreshes the table links, it displays a dialog saying so; click OK to close the success message dialog. Click the Close button of the Linked Table Manager to close the add-in.

Oracle Ex. 1009, p. 244
Oracle v. DeBellis
IPR2024-00221

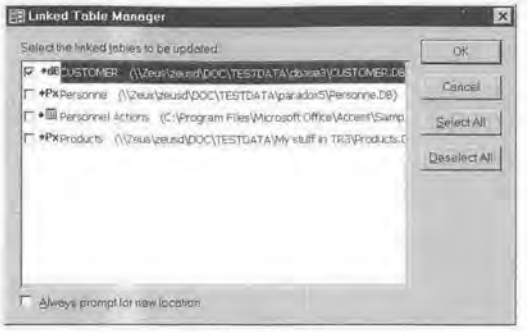


FIG. 7.18 The Linked Table Manager add-in, which enables you to update the path to linked tables.

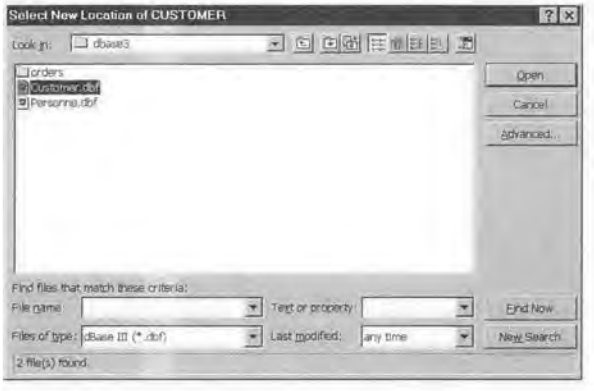


FIG. 7.19 Changing the location of a table with the Linked Table Manager add-in.

Note

The Linked Table Manager is capable of only refreshing links for tables that have been moved to another disk or folder—the table must have the same name. If the linked table's file was renamed, you must delete the table link from your Access database and relink the table under its new name.

Importing versus Linking Database Files as Tables

The preceding examples demonstrate the differences between the behavior of Access with linked and imported database files. You should link tables contained in another database file if any of the following conditions exist:

- You share the file with other users who are allowed to update the file, or you make updates of the file available to other users.
- You use another RDBMS to modify the file in any way.
- The file is resident on another computer, such as a server, and its size is larger than fits comfortably on your fixed disk.

Oracle Ex. 1009, p. 245
Oracle v. DeBellis
IPR2024-00221

- You observe the recommended database application development practice of maintaining separate .mdb files for tables and your application's objects.

You should import a table when one of the following conditions exists:

- You are developing an application and want to use data types or Field Size properties different from those chosen for you by Access.
- You or the users of your application do not have online access to the required database files and cannot link them.
- You want to use a key field different from the field specified in a Paradox or client/server table. This situation can occur when the structure of one or more of the files you plan to use seriously violates one or more of the normalization rules described in Chapter 23, "Exploring Relational Database Design and Implementation."
- You need Access to allow duplicate values in your table when a primary-key field precludes duplicate values.

If you decide to use a temporarily imported table in an application that, when completed, also will use a linked table, make sure that you do not change any field names, field data types, or Field Size properties after you import the table. If you change Field Name properties, you may have to make many changes to forms, reports, macros, and Access VBA code when you change to a linked table. If your application involves Paradox and client/server database tables, do not change the primary-key fields of these tables. With dBASE tables, make sure that the indexes you create correspond to the indexes of the associated .ndx or .mdx files.

Importing and Linking Spreadsheet Files

Access can import files created by spreadsheet and related applications, such as project management systems, in the following formats:

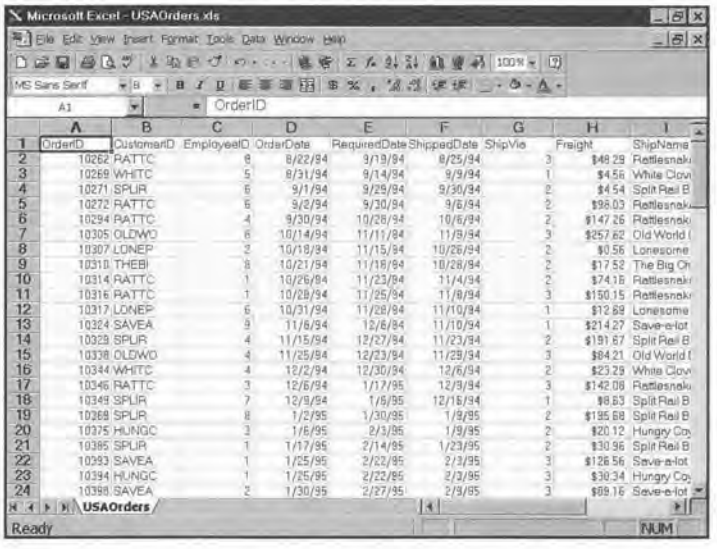
- Excel 2.x, 3.0, 4.0, 5.0, and 95 .xls files, and task and resource files created by Microsoft Project in .xls format.
- Lotus 1-2-3 .wks (Release 1 and Symphony), .wk1 (Release 2), and .wk3 (Release 3 and later) files. Most spreadsheet applications can export files to at least one of these Lotus formats.

You can use OLE to embed or link charts created by Microsoft Excel that are stored in files with an .xlc extension. Copy the contents of the file to the Windows Clipboard from Excel. Then choose Edit, Paste to embed or link (via OLE) the chart in fields of the OLE Object type, and display the chart on a form or print it on a report as an unbound object. Similarly, you can embed or link views displayed in Project for Windows 3.0 or 4.0, which also uses the Microsoft Chart applet, except task and resource forms and the Task PERT chart. Chapters 19–22 describe OLE linking and embedding techniques.

Oracle Ex. 1009, p. 246
Oracle v. DeBellis
IPR2024-00221

Creating a Table by Importing an Excel Worksheet

Figure 7.20 illustrates the preferred format for exporting data from Excel and other spreadsheet applications to Access tables. Most spreadsheet applications refer to the format as a *database*. The names of the fields entered in the first row and the remainder of the database range consist of data. The type of data in each column must be consistent within the database range you select.



	A	B	C	D	E	F	G	H	I
1	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipName
2	10262	RATTC	8	8/22/94	9/19/94	8/25/94	3	$48.29	Rattlesnak
3	10269	WHITC	5	8/31/94	9/14/94	9/9/94	1	$4.56	White Clov
4	10271	SPLIR	6	9/1/94	9/29/94	9/30/94	2	$4.54	Split Rail B
5	10272	RATTC	6	9/2/94	9/30/94	9/6/94	2	$98.03	Rattlesnak
6	10294	RATTC	4	9/30/94	10/28/94	10/6/94	2	$147.26	Rattlesnak
7	10305	OLDWO	8	10/14/94	11/11/94	11/8/94	3	$257.62	Old World
8	10307	LONEP	2	10/18/94	11/15/94	10/26/94	2	$0.56	Lonesome
9	10310	THEBI	8	10/21/94	11/18/94	10/28/94	2	$17.52	The Big Ch
10	10314	RATTC	1	10/26/94	11/23/94	11/4/94	2	$74.16	Rattlesnak
11	10316	RATTC	1	10/28/94	11/25/94	11/8/94	3	$150.15	Rattlesnak
12	10317	LONEP	6	10/31/94	11/28/94	11/10/94	1	$12.69	Lonesome
13	10324	SAVEA	9	11/6/94	12/6/94	11/10/94	1	$214.27	Save-a-lot
14	10329	SPLIR	4	11/15/94	12/27/94	11/23/94	2	$191.67	Split Rail B
15	10338	OLDWO	4	11/25/94	12/23/94	11/29/94	3	$84.21	Old World
16	10344	WHITC	4	12/2/94	12/30/94	12/6/94	2	$23.29	White Clov
17	10346	RATTC	3	12/6/94	1/17/95	12/9/94	3	$142.08	Rattlesnak
18	10349	SPLIR	7	12/9/94	1/6/95	12/15/94	1	$8.63	Split Rail B
19	10369	SPLIR	8	1/2/95	1/30/95	1/9/95	2	$195.68	Split Rail B
20	10375	HUNGC	3	1/6/95	2/3/95	1/9/95	2	$20.12	Hungry Co
21	10385	SPLIR	1	1/17/95	2/14/95	1/23/95	2	$30.96	Split Rail B
22	10393	SAVEA	1	1/25/95	2/22/95	2/3/95	3	$126.56	Save-a-lot
23	10394	HUNGC	1	1/25/95	2/22/95	2/3/95	3	$30.34	Hungry Co
24	10398	SAVEA	2	1/30/95	2/27/95	2/3/95	3	$89.16	Save-a-lot

FIG. 7.20 Data from the Orders table in a Microsoft Excel 97 worksheet.

Caution

All the cells that comprise the range of the worksheet to be imported into an Access table must have frozen values. *Frozen values* substitute numeric results for the Excel expressions used to create the values. When cells include formulas, Access imports the cells as blank data cells. Freezing values causes Access to overwrite the formulas in the spreadsheet with the frozen values. If the range to import includes formulas, save a copy of your .xls file with a new name. By using the worksheet window with the new file name, select the range to import and freeze the values by choosing Edit, Copy, or by pressing Ctrl+C. Choose Edit, Paste Special, select the Values option, and click OK. Save the new spreadsheet by its new name and use this file to import the data. The section "Using the Clipboard to Import Data" that follows in this chapter presents an alternative to this procedure.

Tip

You get an opportunity to assign field names to the columns in the worksheet during the importation process, although it's easier if you add field names as column headings first.

Oracle Ex. 1009, p. 247
Oracle v. DeBellis
IPR2024-00221

To prepare the data in an Excel spreadsheet for importation into an Access table, follow these steps:

1. Launch Excel, and then open the .xls file that contains the data you want to import.
2. Add field names above the first row of the data you plan to export (if you haven't done so). Field names cannot include periods (.), exclamation points (!), or square brackets ([ ]). You cannot have duplicate field names. If you include improper characters in field names or use duplicate field names, you see an error message when you attempt to import the worksheet.
3. If your worksheet contains cells with data you don't want to include in the imported table, select the range that contains the field names row and all the rows of data needed for the table. In Excel, choose Insert, Name, Define and name the range.
4. If the worksheet cells include expressions, freeze the values as described in the caution that precedes these steps.
5. Save the Excel file (using a different file name if you froze values), and exit Excel to conserve Windows resources for Access if your computer has less than 8M of memory.

   Now you are ready to import worksheets from the Excel workbook file.

To import the prepared data from an Excel spreadsheet into an Access table, follow these steps:

1. Launch Access, if it's not running, and open the database to which you want to add the new table. The Database window must be active (with a dark title bar, usually blue) before you can import a file.
2. Choose File, Get External Data, Import in Access. The Import dialog appears (see Figure 7.21). The Files of Type drop-down list provides several more formats for importing tables than it provides for linking files.
3. Select Microsoft Excel (*.xls) in the Files of Type drop-down list, and then double-click the name of the Excel workbook that contains the spreadsheet you want to import (you also can click the file name to select it, and then click Import). Access now invokes the Import Spreadsheet Wizard, which displays the dialog shown in Figure 7.22.
4. If you're importing an entire worksheet, select the Show Worksheets option; if you're importing a named range, select the Show Named Ranges option. The Import Spreadsheet Wizard lists the worksheets or named data ranges, depending on the option you select in the list box in the upper-right corner of the wizard's opening dialog (refer to Figure 7.22).
5. Select the worksheet or the named data range that you want to import in the list box. The Import Spreadsheet Wizard shows a sample view of the data in the worksheet or named range in the scrollable area at the bottom of the dialog.

Oracle Ex. 1009, p. 248
Oracle v. DeBellis
IPR2024-00221

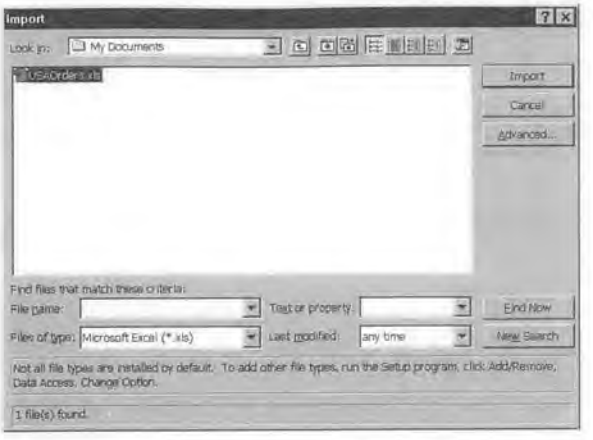


FIG. 7.21 The Import dialog is used to import Excel and Lotus 1-2-3 spreadsheets.

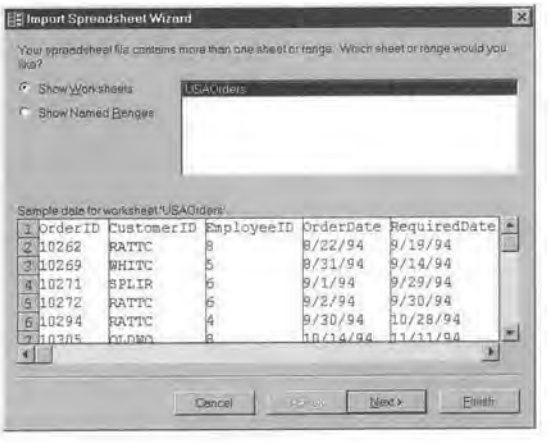


FIG. 7.22 The opening dialog of the Import Spreadsheet Wizard.

6. Click the Next button to move to the next step of the Spreadsheet Import Wizard. The wizard displays the dialog shown in Figure 7.23.

7. If the first row of your spreadsheet data contains the field names for the imported table, select the First Row Contains Column Headings check box. Click Next to continue with the third step; the Import Spreadsheet Wizard displays the dialog shown in Figure 7.24.

8. If you want to create a new table to hold the imported spreadsheet data, select the In a New Table option. To add the imported data to an existing table, select the In an Existing Table option and select the table you want to add the imported data to in the drop-down list. Click Next to continue with the fourth step; the Import Spreadsheet Wizard displays the dialog shown in Figure 7.25.

Oracle Ex. 1009, p. 249
Oracle v. DeBellis
IPR2024-00221

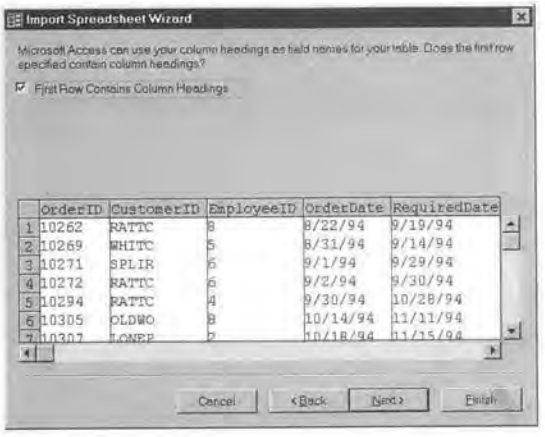


FIG. 7.23 Selecting whether the first row of imported data contains field names.

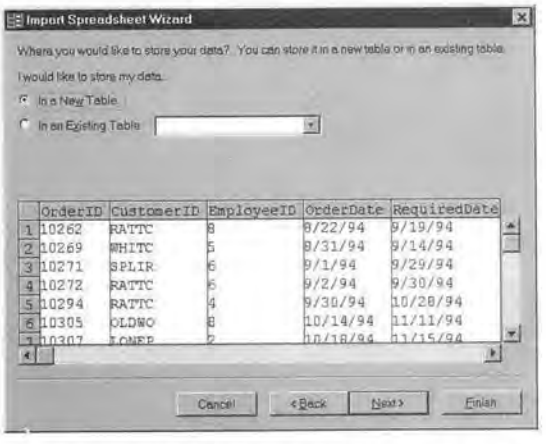


FIG. 7.24 Choosing whether to create a new table from the imported data, or to add it to an existing table.

> **Note**
>
> If you elect to add the imported data to an existing table, the Import Spreadsheet Wizard skips over all intervening steps, and goes immediately to its final dialog, described in step 16 of these instructions.

9. If you want to exclude a column from the imported database, select the column by clicking it, and then select the Do Not Import Field (Skip) check box, and skip to step 12.

10. The Import Spreadsheet Wizard enables you to edit or add the field names for the spreadsheet columns; click the column whose name you want to edit or add, and then type the name in the Field Name text box.

Oracle Ex. 1009, p. 250
Oracle v. DeBellis
IPR2024-00221

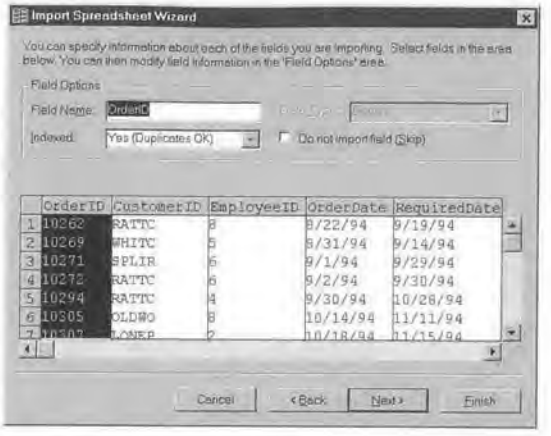


FIG. 7.25 Selecting field names, indexes, and the field's data type.

11. If you want Access to index this field, choose the appropriate index type in the Indexed list box; you may choose No index, Yes (Duplicates OK), or Yes (No Duplicates).

12. If the data in the spreadsheet column is unformatted or is formatted as text, Access enables you to select the data type for the field in the Data Type drop-down list. The Data Type control is disabled in Figure 7.25 because the cells in the selected column of the Excel worksheet have a Double number format; Access recognizes the Double format, and automatically selects a Double data type for this field.

13. Repeat steps 9–12 for each column in the worksheet or data range that you import. When you are satisfied with your options for each column, click Next to move to the next dialog in the Import Spreadsheet Wizard (see Figure 7.26).

Tip

Use an existing field column in the worksheet for a primary key field if the column contains unique values only. In Figure 7.26, the OrderID field is known to contain unique values.

14. Select the Let Access add Primary Key option to have Access add an AutoNumber field to the imported table; Access fills in a unique number for each existing row in the worksheet that you're importing. Select the Choose my own Primary Key option and select the primary key field in the drop-down list if you know there is a column in the worksheet or data range that you can use as a primary key for the imported table. If this imported table doesn't need a primary key, select the No Primary Key option.

▶▶ See "Using Access 97's Table Analyzer Wizard," p. 813



Oracle Ex. 1009, p. 251
Oracle v. DeBellis
IPR2024-00221

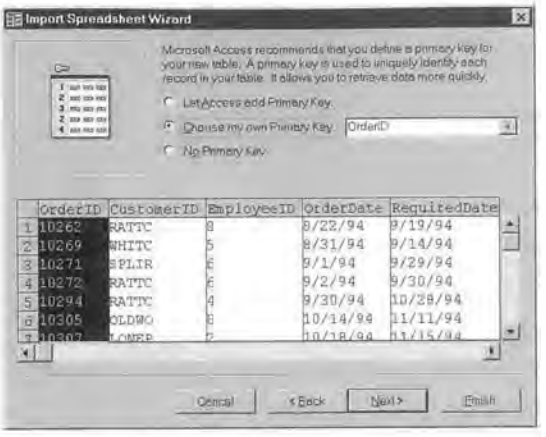


FIG. 7.26 Selecting a primary key for the new table.

15. Click Next to move to the final dialog of the Import Spreadsheet Wizard (see Figure 7.27). Type the name of the new table in the Import to Table text box; Access uses the name of the worksheet or data range as the default table name. If you want to use the Table Analyzer Wizard to split the imported table into two or more related tables, select the check box labeled, I Would Like the Wizard to Analyze the Structure of My Table After it Finishes Importing the Data. (You can use the Table Analyzer Wizard at any time, on any table by choosing Tools, Analyze, Table.)

16. Click Finish to complete the importing process. Access closes the Import Spreadsheet Wizard and imports the data. When Access completes the import process without errors, it displays the dialog shown in Figure 7.28.

Access analyzes, approximately, the first 20 rows of the spreadsheet you are importing, and assigns data types to the imported fields based on this analysis. If every cell in a column has a numeric or date value, the columns convert to Number and Date/Time field data types, respectively. If a column contains mixed text and numbers, Access converts the column as a text field. If, however, a column contains numeric data in the first 20 rows (the rows that Access analyzes), and then has one or more text entries, Access does not convert these rows.

If Access encounters cell values that it cannot convert to the data type that it assigned to the imported field, Access creates an Import Errors table with one record for each error. You can review this table, select the records in which the errors are reported, and fix them. A better approach, however, is to correct the cells in the spreadsheet, resave the file, and import the corrected data.

Note

The Import Spreadsheet Wizard doesn't display an error message when it encounters inconsistent field data types; it just creates the Import Errors table. You must look in the Database window to

Oracle Ex. 1009, p. 252
Oracle v. DeBellis
IPR2024-00221

see if the Import Errors table is present. After you resolve the import errors, make sure that you delete the Import Errors table so that you can more easily tell whether or not there are errors the next time you import a spreadsheet or other external file.

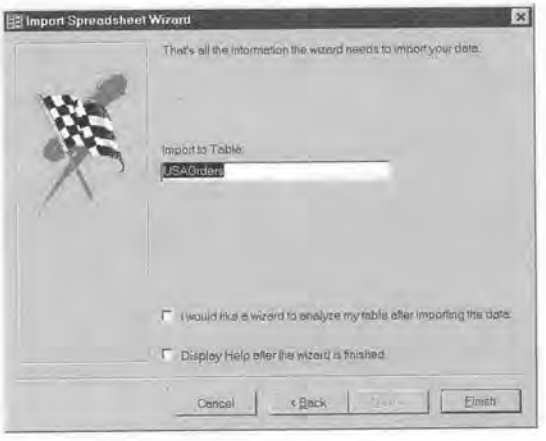


FIG. 7.27 Giving the new table a name in the Import Spreadsheet Wizard's final dialog.



FIG. 7.28 This is the message displayed by the Import Spreadsheet Wizard after successfully importing a spreadsheet.

The Database window now contains a new table with the name you entered in the final step of the Import Spreadsheet Wizard. If you import another file with the same name as your spreadsheet file name, Access adds the number 1 to the file name.

To verify that you obtained the desired result, double-click the name of the imported table in the Database window to display the new table in Datasheet view. Figure 7.29 illustrates a portion of the Access table created from the USA Orders worksheet in the USAOrders.xls spreadsheet file of Figure 7.20.

To display the .xls file data types that Access chose, click the Design View button on the toolbar. Figure 7.30 shows the structure of the new USA Orders table.

After you successfully import the table, you may want to change the properties of the fields. Unlike the procedure with linked files, Access places no restrictions on altering the field properties of imported files. The section, "Modifying Linked and Imported Tables," that follows in this chapter discusses how to change the default field data types of tables created by imported files.

Oracle Ex. 1009, p. 253
Oracle v. DeBellis
IPR2024-00221

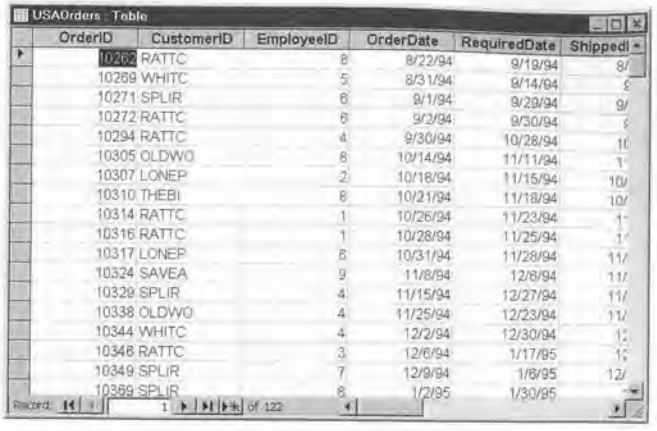


FIG. 7.29 The imported Excel worksheet data in an Access table.

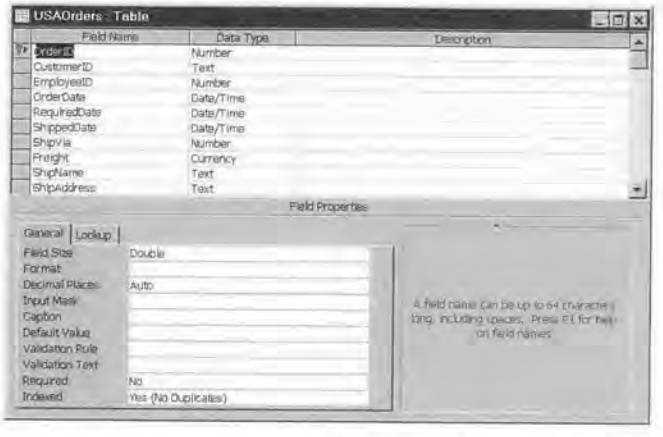


FIG. 7.30 The structure of the imported USA Orders table based on Excel data types.

Linking an Excel Worksheet by Using the ODBC API

Office 97 installs 32-bit Excel, FoxPro, Paradox, Access, dBASE, and Text file ODBC drivers, plus the SQL Server driver. ODBC and the ODBC drivers are installed if you choose a Complete Microsoft Office 97 or Access 97 setup, but you must explicitly specify installation of the SQL Server driver during the setup process. If you don't have the ODBC drivers on your system, you can rerun Setup to install them. Additional ODBC drivers are available from various third-party vendors.

You can use the ODBC drivers provided with Office 97 to link Excel worksheets to your Access databases. Linking an Excel worksheet has the advantage of providing up-to-date information if the data in the worksheet is subject to periodic updates. You can use the 32-bit ODBC Manager application supplied with Access 97 (and installed into the Windows 95 and Windows NT 4.0 Control Panel) to create an ODBC data source from your Excel worksheet. You also can link Excel worksheets directly to a database by using

Oracle Ex. 1009, p. 254
Oracle v. DeBellis
IPR2024-00221

ODBC directly from Access. *Data source* is a synonym for *database* when you use the ODBC API to link tables. (In Windows NT, the ODBC Manager is referred to as the *ODBC Administrator*.)

▶▶ See "Understanding ODBC Drivers," p. 918



Note

To link data in an Excel worksheet, you must create a named data range for the worksheet data you want to link. Usually, you should make sure that the worksheet you link via ODBC meets all the requirements for a worksheet that you intend to import, as described in the preceding section of this chapter. That is, make sure that the data in the worksheet is laid out in a tabular format. You don't, however, have to freeze values for an ODBC-linked worksheet.

Linking an Excel worksheet via ODBC is similar to the spreadsheet importing and linking you learned about previously in this chapter. Assuming that you installed the ODBC Manager and ODBC drivers, follow these steps to link an Excel worksheet to an Access database via ODBC:

1. Launch Access and open a database, if necessary. Choose File, Get External Data, Link Table to display the Link dialog (refer to Figure 7.1).
2. Select ODBC Databases in the Files of Type drop-down list. Access closes the Link dialog and displays the Select Data Source dialog (see Figure 7.31).

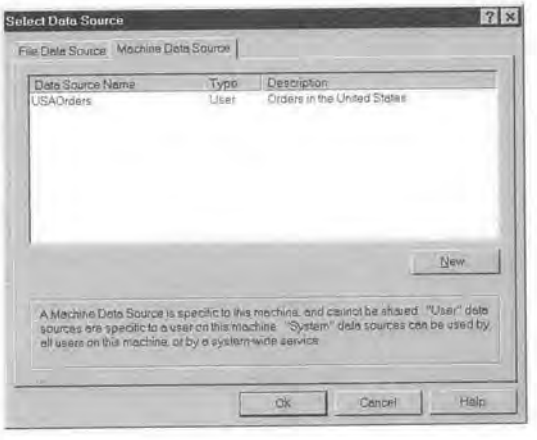


FIG. 7.31 Selecting an ODBC data source in the Select Data Source dialog.

3. Click the Machine Data Source tab to select a previously defined ODBC data source, or to connect to another data source. (You use the File Data Source page to specify a data source file that you can share with other users.)

▶▶ See "Creating a File Data Source Within Access 97," p. 931



Oracle Ex. 1009, p. 255
Oracle v. DeBellis
IPR2024-00221

4. If you previously defined an ODBC data source for the worksheet you want to link, select its name in the Select Data Source list. Otherwise, click New to display the first dialog of the Create New Data Source Wizard (see Figure 7.32).



FIG. 7.32 The opening dialog of the Create New Data Source Wizard.

5. Select the User Data Source option if you want the new data source to be available only to you. If you want anyone who logs on to your computer to be able to use your new data source, select the System Data Source option. Click Next to continue with the second dialog of the Create New Data Source Wizard (see Figure 7.33).



FIG. 7.33 Selecting a database driver for an Excel worksheet.

6. Select the database driver appropriate for the database you want to link. For an Excel worksheet, select Microsoft Excel Driver (*.xls)—refer to Figure 7.33. Click Next to display the Create New Data Source Wizard's third dialog.

7. The final dialog of the Create New Data Source Wizard displays summary information about the type of data source you have created (system or user) and the name of the database driver you selected (see Figure 7.34). Click Finish to complete the creation of your new data source.

Oracle Ex. 1009, p. 256
Oracle v. DeBellis
IPR2024-00221

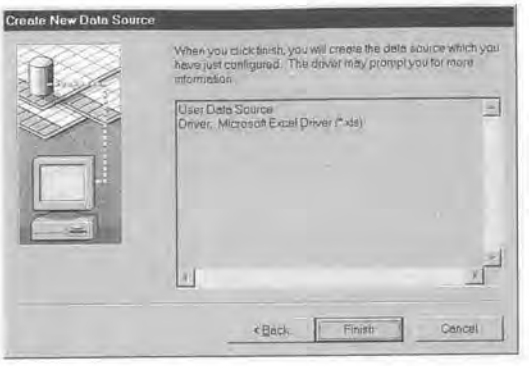


FIG. 7.34 The Create New Data Source Wizard's final dialog.

8. After creating a new data source, you must configure it. Access now displays the ODBC Microsoft Excel Setup dialog (see Figure 7.35). Enter a name for your data source in the Data Source Name text box, and enter a description of the data source in the Description text box.

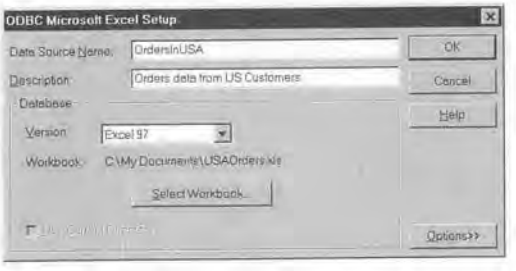


FIG. 7.35 Configuring a new Excel ODBC data source.

9. Click Select Workbook to display the Select Workbook dialog, shown in Figure 7.36. The Select Workbook dialog is a Windows 3.x-style file opening dialog. If needed, use the Drives and Directories lists to navigate to the correct folder in which your workbook is stored and select its name in the Database Name list. Click OK; Access closes the Select Workbook dialog and adds the new data source to the list in the Select Data Source dialog (refer to Figure 7.31).

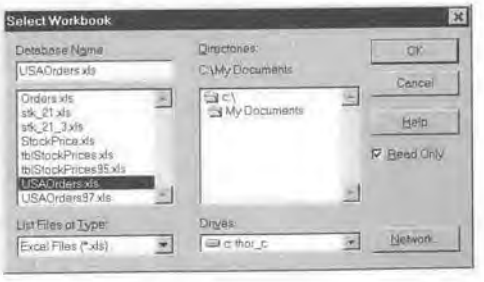


FIG. 7.36 Choosing the workbook for the new data source in the Select Workbook dialog.

Oracle Ex. 1009, p. 257
Oracle v. DeBellis
IPR2024-00221

10. Select your new data source in the Select Data Source dialog's list (refer to Figure 7.31) and then click OK. Access opens the Link Tables dialog (see Figure 7.37).

11. The Link Tables dialog lists the named data ranges and worksheets in the workbook. Click a worksheet or range name to select it. After you select all the worksheets or ranges that you want to link, click OK. Access closes the Link Tables dialog, and displays the Select Unique Record Identifier dialog (see Figure 7.38).



FIG. 7.37 Selecting the worksheet or named range to link to your database.



FIG. 7.38 Selecting a field in the linked table to create a primary key for the linked spreadsheet.

12. If you want to be able to update the data in the ODBC-linked worksheet, you must select a field (or combination of fields) that creates a unique record identification for each row in the worksheet—essentially, you create a primary key for the linked table. To select a field, click it with the mouse.

13. After you are satisfied with your key field selection, click OK. At this point, Access formulates the SQL query to establish indexes for the linked worksheet and analyzes the data in the worksheet. Depending on the outcome of this analysis, Access either immediately links the worksheet, or invokes the Link Spreadsheet Wizard.

    Using the Link Spreadsheet Wizard is the same as using the Import Spreadsheet Wizard described in the preceding section of this chapter, except that Access doesn't allow you to edit the field names it assigns to the linked data.

14. Your linked worksheet table appears in the Database window (with the ODBC globe turned to display Africa), as shown in Figure 7.39.

Oracle Ex. 1009, p. 258
Oracle v. DeBellis
IPR2024-00221



FIG. 7.39 Identifying tables linked with ODBC drivers by the globe icon.

Click the Open button of the Database window to display your linked worksheet. The look of the worksheet as a linked table is identical to a table imported from a worksheet with one major difference—you cannot update a linked worksheet. When you link a table to an Access database by using the ODBC API and the Microsoft ODBC drivers, you can update the table only if the table has a designated primary-key field. (The linked worksheet table is represented by a read-only `Recordset` object of the `Snapshot` type.)

Note

Some third-party ODBC drivers, such as the Intersolve ODBC drivers in the ODBC Driver Pack, have an option that enables you to edit Excel worksheet and text files that you link to Access tables. For more information, see Chapter 26, "Connecting to Client/Server Databases."

Importing Text Files

If the data you want to import into an Access table was developed in a word processor, database, or other application that cannot export the data as a .dbf, .wk?, or .xls file, you need to create a text file in one of the text formats supported by Access. (A *text file* is a file with data consisting of characters that you can read with a text editor, such as Windows Notepad or the DOS Edit.com text editor.) Most DOS- and Windows-compatible data files created from data in mainframe computers and files converted from nine-track magnetic tapes are text files, and Access imports these files in various formats.

Access refers to the characters that separate fields as *delimiters* or *separators*. In this book, the term *delimiter* refers to characters that identify the end of a field; the term *text identifiers* refers to the single and double quotation marks that you can use to distinguish text from numeric data.

Note

EBCDIC (*Extended Binary-Coded-Decimal Interchange Code*) is a proprietary format used by IBM to encode data stored on nine-track tape and other data interchange media. EBCDIC is similar to the

(continues)

Oracle Ex. 1009, p. 259
Oracle v. DeBellis
IPR2024-00221

(continued)

ANSI (American National Standards Institute) and ASCII (American Standard Code for Information Interchange) codes. You need to convert EBCDIC-encoded data to ANSI or ASCII code before you can import the data into an Access table. Nine-track tape drives designed for PC applications and service bureaus who provide tape-to-disk conversion services handle the EBCDIC-ASCII conversion. The printable (text) characters with values 32 through 127 are the same in ANSI and ASCII, so conversion from ASCII to ANSI, the character set used by Windows and Access, is seldom necessary.

Table 7.2 details the formats that Access supports.

Table 7.2 Text File Formats Supported by Access 97

Format	Description
Comma-delimited text files (also called *CSV* [Comma-Separated Value] files)	Commas separate (delimit) fields. The newline pair, carriage return (ASCII character 13), and line feed (ASCII character 10) separate records. Some applications enclose all values within double quotation marks; this format often is called *mail-merge* format. Other applications enclose only text (strings) in quotation marks to differentiate between text and numeric values, the standard format for files created by the xBase command, `COPY TO FILENAME DELIMITED`.
Tab-delimited text files (also called ASCII files)	These files treat all values as text and separate fields with tabs. Records are separated by newline pairs. Most word-processing applications use this format to export tabular text.
Space-delimited files	Access can use spaces to separate fields in a line of text. The use of spaces as delimiter characters is uncommon because it can cause what should be single fields, such as names and addresses, to be divided inconsistently into different fields.
Fixed-width text files	Access separates (parses) the individual records into fields based on the position of the data items in a line of text. Newline pairs separate records; every record must have exactly the same length. Spaces pad the fields to a specified fixed width. Using spaces to specify field width is the most common format for data exported by mainframes and minicomputers on nine-track tape.

Using the Text Import Wizard

To import any of the text file types listed in Table 7.2, you follow a procedure similar to the procedure for importing any external data into Access. To import a text file, follow these steps:

1. Open the database into which you want to import the text file and make the Database window active, if necessary.
2. Choose File, Get External Data, Import. Access displays the Import dialog.
3. In the Import dialog, select Text Files (*.txt, *.csv, *.tab, *.asc) in the Files of Type drop-down list. Use the Look In drop-down list to select the folder that contains the text file you want to import, and double-click the text file's name. Access now starts the Text Import Wizard (see Figure 7.40).

Oracle Ex. 1009, p. 260
Oracle v. DeBellis
IPR2024-00221

4. Select the Delimited option to import a delimited text file, or select the Fixed Width option to import a fixed-width text file. The Text Import Wizard displays a sample of the text file's contents in the lower portion of the dialog to help you determine the correct file type. Figure 7.40 shows a comma-delimited text file being imported. Click Next to proceed to the next step in the Text Import Wizard. If you selected Delimited as the file type, the Text Import Wizard displays the dialog shown in Figure 7.41; if you selected the Fixed Width option, the wizard displays the dialog in Figure 7.42.

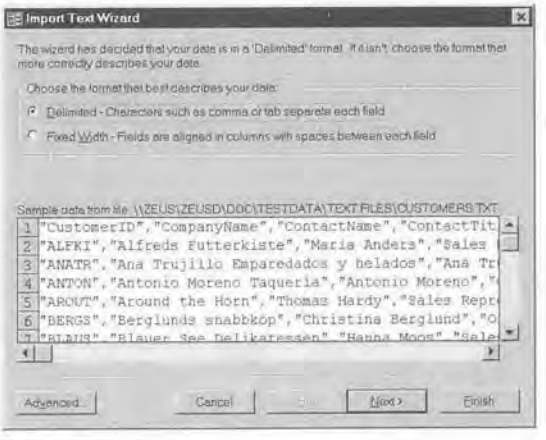


FIG. 7.40 Choosing whether the text file you're importing is delimited or fixed-width text.



FIG. 7.41 The second step of the Text Import Wizard for delimited text files.

Oracle Ex. 1009, p. 261
Oracle v. DeBellis
IPR2024-00221

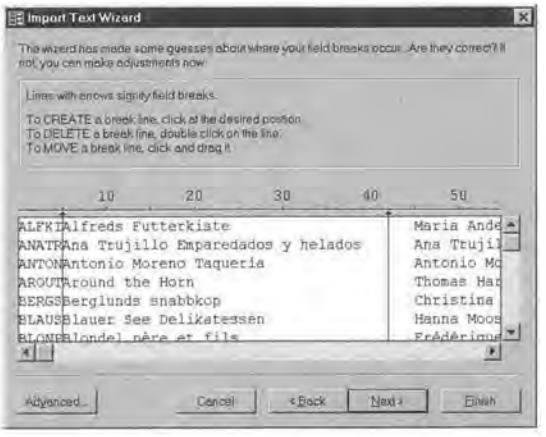


FIG. 7.42 The second step of the Text Import Wizard for fixed-width text files.

> **Note**
>
> In almost every case, you will select the Delimited option. Use the Delimited option for comma-delimited, tab-delimited, and all other types of delimited text files. Use fixed-width for space-delimited text files and also text files actually formatted as fixed-width.

5. If you're importing a fixed-width text file, skip to step 6. Otherwise, select the delimiter character that separates fields in the table (most delimited files use the default comma separator). If the text file you're importing uses a text qualifier other than the double quote, enter it in the Text Qualifier text box. If the first line in the text file contains field names (such as the column headings in a spreadsheet file), select the First Row Contains Field Names check box. Click Next to move to the next step of the Text Import Wizard.

6. If you're importing a delimited text file, skip to step 7. In a fixed-width text file, the Text Import Wizard analyzes the columns and makes an approximation about where the field breaks lie. Scan through the sample data at the bottom of the dialog; if the field breaks aren't in the right place, there are too many field breaks, or there aren't enough field breaks. You can add, delete, or move the field breaks that the Text Import Wizard suggests. To move a field break, drag it with the mouse. To remove a field break, double-click it. To add a field break, click at the location where you want the field break added. When you're satisfied with the field break arrangement, click Next to continue with the Text Import Wizard.

7. The Text Import Wizard now displays the dialog shown in Figure 7.43. Choose the In a New Table option to create a new Access table for the imported text file. Choose the In an Existing Table option to add the data in the text file to an existing database table, then select the table to which you want the data added in the accompanying drop-down list. Click Next to continue with the next step in the Text Import Wizard. (If you selected the In an Existing Table option, the Text Import Wizard skips directly to its final step, step 10 of this procedure.)

Oracle Ex. 1009, p. 262
Oracle v. DeBellis
IPR2024-00221

Caution

Access matches fields from left to right when you import a text file into an existing table. You must make sure that the data types of the fields in the imported text file match those in the Access table; otherwise, the added data values aren't inserted into the correct fields. In most cases, you end up with many import errors in the Import Errors table. If you're not absolutely certain that the format of your input data matches the format of the desired table exactly, you can choose the In a New Table option, and then place your data in the existing table with an Append query, as discussed in Chapter 11, "Using Action Queries."

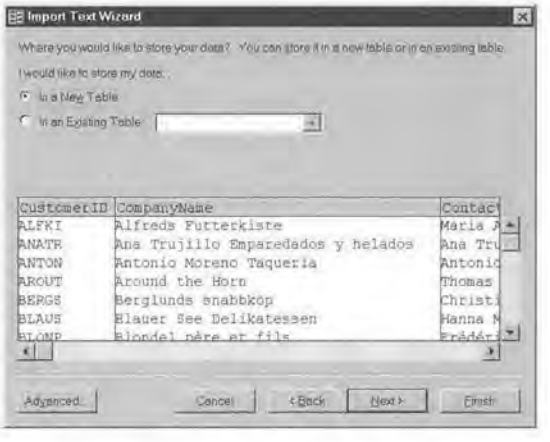


FIG. 7.43 Choosing whether to create a new table, or add the imported text data to an existing table.

8. The Text Import Wizard now displays the dialog shown in Figure 7.44. The Text Import Wizard enables you to edit field names, choose whether and what kind of index to use for each field, and to adjust each field's data type. To set the options for each field, click the field column at the bottom of the dialog to select it; you then may edit the field name, select an index method in the Indexed drop-down list, and select the data type for the field in the Data Type drop-down list. Select the Do Not Import Field (Skip) check box if you don't want to import the select field column. When you're satisfied with your field settings, click Next.

9. Now the Text Import Wizard displays the dialog in Figure 7.45. Choose the appropriate option for the primary keys you want—whether to allow Access to add a new field with an automatically generated primary key, select an existing field to use as a primary key, or import the table without a primary key. After selecting the primary key options, click Next.

10. The Text Import Wizard now displays its final dialog, shown in Figure 7.46. You now must enter the name for the new imported table. The Text Import Wizard displays this dialog even if you chose to import the text file into an existing table. Access enters either a default table name that is the same as the original name of

Oracle Ex. 1009, p. 263
Oracle v. DeBellis
IPR2024-00221

the text file or the table name you selected for importing the data into. Edit the table name, or type a different table name, if you want. Click Finish to import the text file.

The Text Import Wizard imports the text file, and displays a success message. As with other import operations, Access creates an Import Errors table to document any errors that occurred during the import process, and displays a message informing you that errors occurred.



FIG. 7.44 Editing field names and selecting the index type and the data types for fields.



FIG. 7.45 Selecting a primary key for the imported text data.

The specific values you should select for field delimiters and text qualifiers are described in following sections, along with some tips on importing fixed-width text files.

Oracle Ex. 1009, p. 264
Oracle v. DeBellis
IPR2024-00221

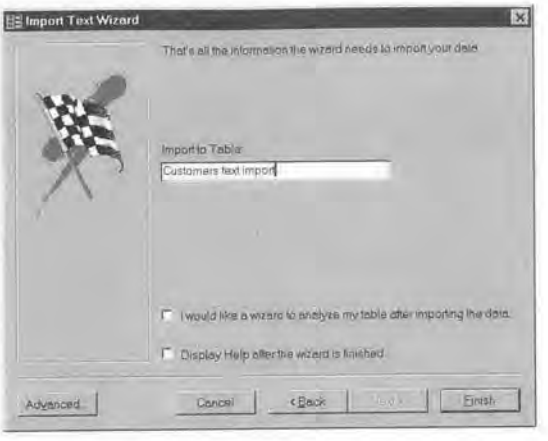


FIG. 7.46 Giving the new table a name in the Text Import Wizard's final dialog.

The Text Import Wizard Advanced Options

Occasionally, you may find that you import text data from the same text file more than once, or that you have several text files that all have the same format. A typical situation in many corporations is that data from the company's mainframe computer system is provided to desktop computer users in the form of a text file report. Frequently, reports are delivered over the network to users in a text file, using the same name for the text file each time. You can use the Text Import Wizard's advanced options to configure Access to import a text file, and save these options so that you don't have to go through every step in the wizard every time you import the text file.

Every step of the Text Import Wizard has an Advanced button. Clicking this button displays a special dialog that shows all the Text Import Wizard settings in a single dialog and allows you to select a few options, such as date formatting, that don't appear in the regular Text Import Wizard dialogs. If you select the Delimited option, the Advanced dialog has the options and field grid shown in Figure 7.47. If you select the Fixed-Width option, the Advanced dialog has the options and field grid shown in Figure 7.48.

Tip

For text files you intend to import only once, it's much easier to import the file by going step-by-step through the Text Import Wizard.

The following list summarizes the options you can select in the Advanced dialog:

- *File Format.* Use these option buttons to choose which type of text file format you're importing: Delimited, or Fixed Width. Depending on the file format you select, the specific options available to you change.
- *Field Delimiter.* Use this drop-down list to select the symbol that delimits fields in the text file. This option is disabled for fixed-width text files.

Oracle Ex. 1009, p. 265
Oracle v. DeBellis
IPR2024-00221

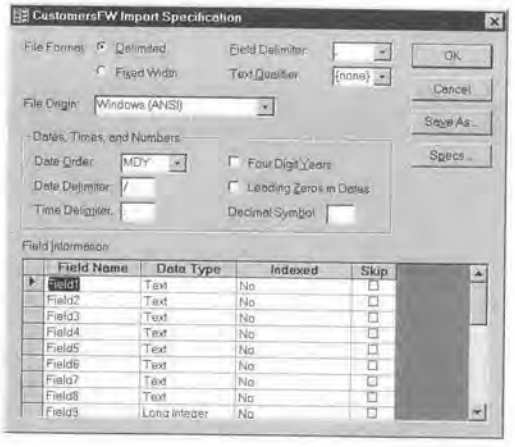


FIG. 7.47 The Advanced options for delimited text files.

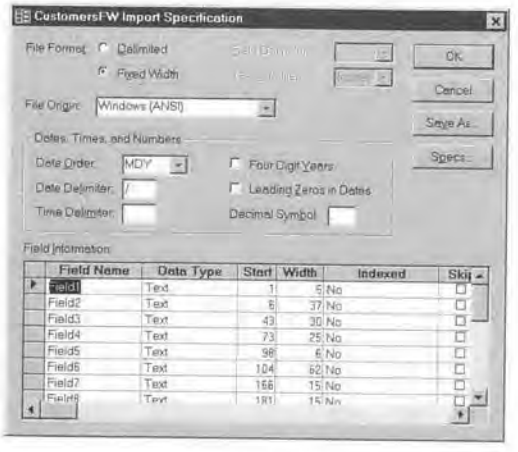


FIG. 7.48 The Advanced options for fixed-width text files.

- *Text Qualifier.* Use this drop-down list to select the symbol that marks the beginning and end of text strings in the text file. This option is disabled for fixed-width text files.
- *File Origin.* Select the character set used for the text file in this drop-down list. If you're importing a text file that originated on a Macintosh computer, for example, select Macintosh in this list.
- *Date Order.* If the data in the text file uses a European or other date format that differs from the month-day-year format typical in the United States, select the appropriate date order in the Date Order drop-down list.

Oracle Ex. 1009, p. 266
Oracle v. DeBellis
IPR2024-00221

- *Date Delimiter* and *Time Delimiter*. Type the symbol used to separate the month, day, and year in a date in the Date Delimiter text box, and type the symbol used to separate hours, minutes, and seconds in the Time Delimiter text box. For example, in the United States, the Date Delimiter is the slash (/) character, and the Time Delimiter is the colon (:).
- *Four Digit Years*. Mark this check box if the dates in the text file use four digits for the year, such as 8/28/1997.
- *Leading Zeroes in Dates*. Mark this check box if the dates in the text file have leading zeroes, such as 08/09/97.
- *Decimal Symbol*. Type the symbol used for the decimal separator in numeric values in the text box. In the United States, the decimal symbol is the period (.), but many European nations use a comma (,).
- *Field Information*. The appearance of this grid varies, depending on the file format that you select. For a delimited text file, the Field Information grid allows you to edit field names, select the field's data type and indexing, and specify whether to skip the field in importing (refer to Figure 7.47). For a fixed-width text file, the Field Information grid allows you to perform the same operations, but adds specifications for the starting column and width of each field (refer to Figure 7.48).
- *Save As*. Click this button to display the Save Import/Export Specification dialog. By entering a name for the specification and clicking OK, you can save the file import settings for later use.
- *Specs*. Click this button to display the Load Import/Export Specification dialog. Select a previously saved specification and click OK to use import settings that you defined previously.

The following sections on using delimited and fixed-width text files discuss the application of some of these Advanced options in greater detail.

Using Delimited Text Files

Delimited files can accept a wide variety of field- and record-delimiting characters. The native format (used by files created within WordPerfect) of WordPerfect secondary merge files, for example, uses control characters to separate fields and records. Access provides commas, tabs, and spaces as standard field delimiters. You can type any printable character, such as an asterisk or a dollar sign, in the text box as a delimiter (refer to Figure 7.41). Because spaces and other special-character delimiting are seldom used, only comma-delimited (.csv) and tab-delimited files are presented in this chapter.

Word-processing applications use both commas and tabs as delimiters in the mail-merge files that they or other applications create for personalized documents. The newline pair is universally used to indicate the end of a record in mail-merge files; a record always consists of a single line of text.

Oracle Ex. 1009, p. 267
Oracle v. DeBellis
IPR2024-00221

Comma-Delimited Text Files Without Text-Identifier Characters. Comma-delimited files come in two styles: with or without quotation marks surrounding text fields. The quotation marks, usually the standard double quotation marks ("), identify the fields within them as having the Text data type. Fields without quotation marks are processed as numeric values if they contain numbers only. Not all applications offer this capability; .csv files, for example, exported by Excel don't enclose text within quotation marks. Figure 7.49 shows a typical Excel .csv file opened in the Windows Notepad applet.

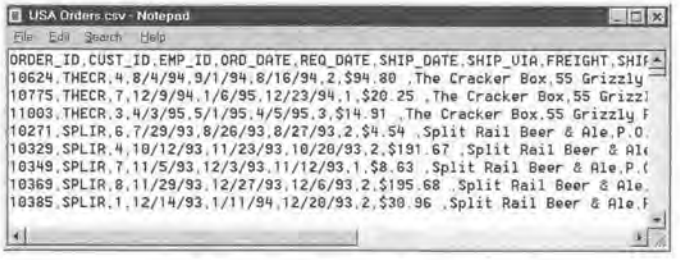


FIG. 7.49 A .csv text file exported by Excel, displayed in Windows Notepad.

Note

Using Notepad to view files that fit within its 60K file-size limitation is a quick way to determine the type of text file with which you are dealing. If the file is longer than 60K, you can use Windows WordPad to view the file. Make sure, however, that you do not save the file as a .doc file after you view or edit it. If you used WordPad to edit the file, choose File, Save As and specify a text (.txt) file.

Caution

If you export the Orders table from Northwind.mdb to Excel, and then use Excel to create an Orders.csv file and import the file into an Access table, you may receive an extraordinary number of errors. Most of these errors are due to a mixture of numeric values for United States ZIP codes and alphanumeric values used for Canadian, U.K., and European postal codes.

The first data cell in the Postal Code column is a number, so the Access import procedure determines that the field should have the Number field data type. Therefore, mixed value fields (letters and numbers) cause import errors. A wise policy is to not import delimited text files without text-identification characters to Access tables.

Comma-Delimited Text Files with Text-Identifier Characters. The default delimited text file type of dBASE, named .sdf for Standard Data Format, created by the `COPY TO FILENAME DELIMITED` command, creates comma-delimited files with text fields surrounded by double quotation marks. (Date and Numeric field types do not have quotation marks.) This type of delimited file is standard in many other database systems, as well as project and personal information management applications. Figure 7.50 shows an example of a delimited text file that contains text qualifiers.

Oracle Ex. 1009, p. 268
Oracle v. DeBellis
IPR2024-00221

Note

When you create a DOS dBASE III+ file from an Access file, Access translates characters with ANSI values 128 through 255 to the PC-8 character set used by character-based DOS and OS/2 applications. Letters with diacriticals and special characters of Scandinavian and romanized Slavic languages do not have the same values in the PC-8. These letters and characters appear on-screen as black rectangles because the PC-8 values do not correspond to printable ANSI values. For text files that don't use the Windows ANSI character set, use the Text Import Wizard's advanced options to select the appropriate character set.

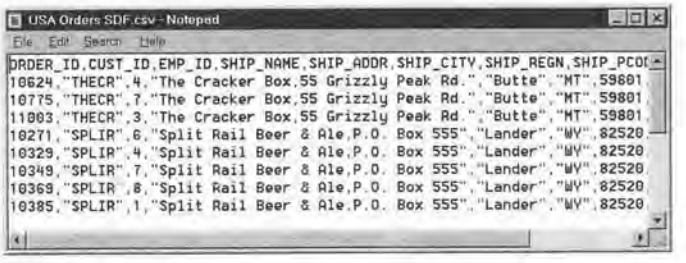


FIG. 7.50 A dBASE .sdf file displayed in Windows Notepad.

Tab-Delimited Text Files. Word-processing applications often use tab characters to separate fields in mail-merge files. These tab characters usually define the fields of tables when you convert conventional text to tabular format (and vice versa) in word processors such as Word 97. Tab-delimited text files rarely use text-qualifier characters. When exporting tab-delimited text files (described in a following section of this chapter), Access adds double quotation marks as text-identifier characters so that Word doesn't interpret embedded newline pairs (carriage return and line feed, or CR/LF) in text fields as the end of the record. (Allowing embedded newline pairs in text fields isn't a recommended database design practice.)

Many organizations acquire RDBMSs because the amount of their data is too large for their word-processing application to maintain mailing lists for direct-mail advertising and other promotional and fundraising purposes. RDBMSs also enable you to create specialized merge data files for specific types of customers, ranges of ZIP codes, and other parameters you select.

Fortunately, Access has a simple process for converting the merge data files used by most word processors to text files that can be imported and maintained by an Access database application. In Word 97, for example, you simply open the merge data file in whatever format you use (usually the native .doc) and save the document in Text Only (.txt) format under a different file name. WordPerfect 5+ (using CONVERT.EXE) and WordPerfect for Windows offer a variety of export formats for their secondary merge files. Unless you have a specific file type in mind, select the tab-delimited format for these files.

Oracle Ex. 1009, p. 269
Oracle v. DeBellis
IPR2024-00221

Handling Fixed-Width Text Files

If you have a choice of text file formats for the data you plan to import into Access, avoid using fixed-width text files by choosing a delimited file format. If you are importing data created by a mainframe or minicomputer, however, the data probably is in fixed-width format. In a fixed-width text file, you must name the fields, rather than relying on the first line of the text file to provide names for you. Fixed-width text files seldom come with a field name header in the first line.

A fixed-width text file resembles the file in Figure 7.51. Fixed-width text files often contain more spaces than data characters. In a tab-delimited file, the spaces between fields aren't included in the file itself, but are added by the text editor when tab characters (ASCII or ANSI character 9) are encountered. In a fixed-width file, each space is represented by ASCII or ANSI character 32.

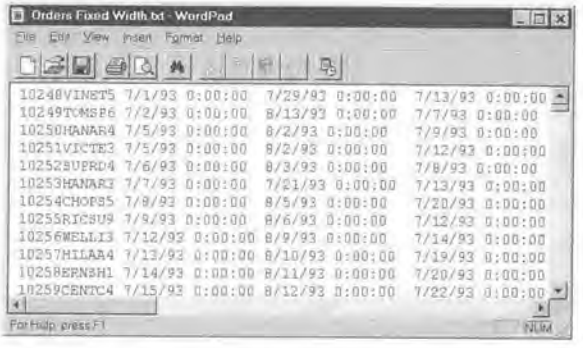


FIG. 7.51 Padding fixed-width records to the same length with spaces.

Note

The fields of fixed-width text files often run together; the first four fields of the Orders Fixed Width.txt file shown in Figure 7.51 are Order ID (five digits), Customer ID (four letters), Employee ID (one digit), and Order Date (16 characters). The Order Date field is padded with spaces to a width of 16 characters. The appearance of the data in Figure 7.51 is typical of COBOL "text dumps" from mainframe and minicomputer tables. If you have the COBOL file description for the fixed-width table, it's far easier to complete the import specification.

Troubleshooting

When importing tables created from fixed-width text files, a large number of errors occur.

You probably miscalculated one or more of the starting positions of a field. Locate the first field name with a problem; the names following it usually have problems, too. Close all open tables. From the Database window, select the table you imported and press the Delete key. If you have an Import Errors table, delete it, too. You cannot delete an open table. Perform the importation process again, and reposition the field breaks in the Text Import Wizard. Remember that Access only analyzes the first 20 lines of the text file, so its guesses about where to position the field breaks may be incorrect, and may not allow enough room for the actual width of a field.

Oracle Ex. 1009, p. 270
Oracle v. DeBellis
IPR2024-00221

Appending Text Data to an Existing Table

If the data you import is provided on a floppy disk, you may need several disks to store the data. Fixed-width text files usually require more floppy disks because in most cases, they are derived from data on mini- and mainframe computers, and the fixed-width format is quite inefficient. If you ever imported multiple-disk data into dBASE files, for example, you probably learned that you must concatenate all the files on your hard disk into one large text file. You concatenate files by using the DOS `COPY file1+file2 file3` command or by appending each file to a separate dBASE file and then appending the remaining dBASE files to the file created from the first disk. This process is very tedious.

Access enables you to append data from text files to an existing table. Besides simplifying the process described in the preceding paragraph, you can update an imported file with new text data by appending it directly from the source text file—rather than by creating a new Access table—and then appending the new table to the existing one.

You can append a text file to an existing table by following these steps:

1. Make a backup copy of your table in the same database or another database, in case an error occurs during the importing operation.
2. Choose File, Get External Data, Import, and select a text file as though you are going to import it to a new table.
3. Make sure that you select the In an Existing Table option and specify a table name in step 3 of the Text Import Wizard (refer to Figure 7.43).
4. If you used the Advanced button to create and save an import specification, click Advanced to display the advanced options dialog, and then click Specs to load the previously saved import specification.
5. Proceed with the importation process as you would for any other text file.

At the end of the appending process, a message box appears. The Import Errors table displays any errors made.

Note

Maintaining a backup copy of the table to which you are appending files is important. If you have a problem with field delimiters or field lengths, or accidentally select the wrong import specification for the file, you can end up with one error for each appended record. Then, if you do not have a backup file, you must delete the appended records and start over. To use a backup table file, close and delete the damaged table. Then choose Edit, Rename to change the name of the backup table to the name of the damaged table.

Using the Clipboard to Import Data

If another Windows application generates the data you want to import, you can use the Windows Clipboard to transfer the data without creating a file. This technique is useful for making corrections to a single record with many fields or for appending new records

Oracle Ex. 1009, p. 271
Oracle v. DeBellis
IPR2024-00221

to a table. This process requires a table with the proper field structure so that you can paste the data copied to the Clipboard into the other application. Pasting rows from an Excel spreadsheet, for example, requires a table with fields containing data types that correspond to those of each column that you copy to the Clipboard. Other Windows applications that can copy tabular data to the Clipboard use similar techniques.

Pasting New Records to a Table

To import data from the Clipboard, and then append the data to an existing table or table structure, use the following procedure:

1. Open the application you are using to copy the data to the Clipboard—in this case, Microsoft Excel. Then, open the file that contains the data.
2. Select the range to be appended to the table (see Figure 7.52). The Excel columns you select must start with the column that corresponds to the first field of your Access table. You do not, however, need to copy all the columns that correspond to fields in your table. Access supplies blank values in the columns of your appended records that are not included in your Excel range. Remember that if any of the columns you select contain formulas, the values must be frozen.

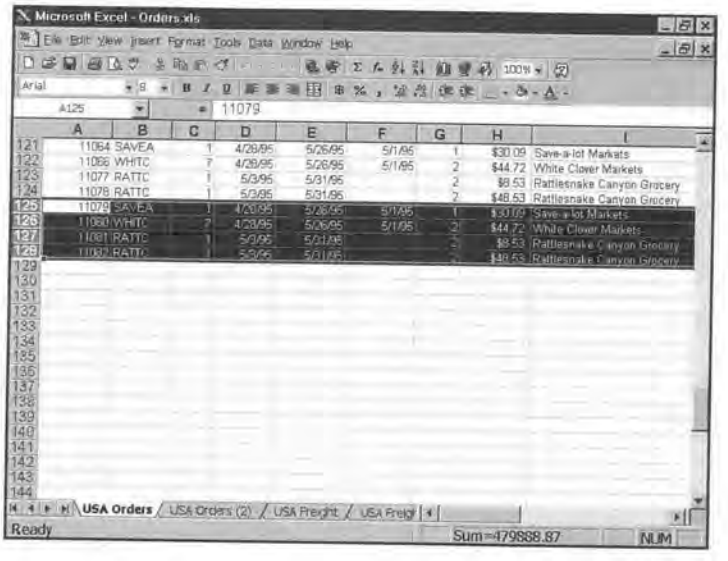


FIG. 7.52 Selecting cells in Excel to be appended to an Access table by copying to the Clipboard.

3. To copy the selected cells to the Clipboard, press Ctrl+C or choose Edit, Copy.
4. Launch Access (if necessary), and open the table to which you are appending the records in Datasheet view.
5. Choose Edit, Paste Append in Access. If no errors occur during the pasting process, a message box reports how many new records you are about to add (see Figure

Oracle Ex. 1009, p. 272
Oracle v. DeBellis
IPR2024-00221

7.53). Click Yes. The records are appended to the bottom of your Access table (see Figure 7.54). Choose Records, Remove Filter/Sort to place the appended records in the correct order.

Note

The fields you add to a table by using Paste Append must correspond (in left to right order) to the fields in the table you are pasting the data into. You cannot, therefore, Paste Append records into a table that has an AutoNumber field, unless the records you are appending have already been assigned unique numeric values greater than the highest AutoNumber value currently existing in the table you are appending the records to. Otherwise, the fields of the pasted data don't match the left to right order of the fields in the table you're pasting into, and all of the pasted records will end up in the Paste Errors table.



FIG. 7.53 The message that appears when Access successfully appends records pasted from the Clipboard.

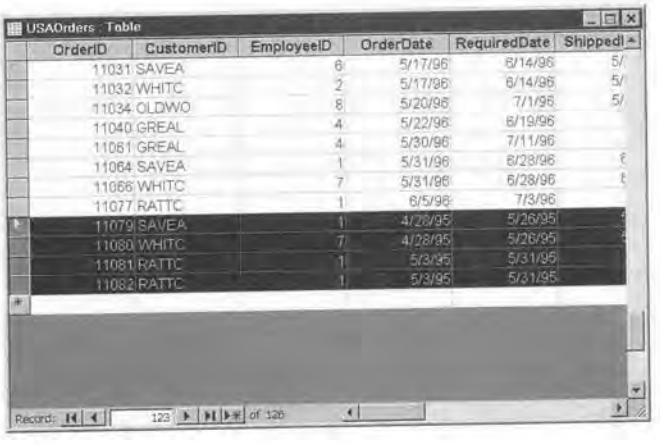


FIG. 7.54 Records appended from the Clipboard to an Access table.

Troubleshooting

Paste errors occur when pasting spreadsheet cells into a table.

Errors usually occur during the Paste/Append process for one of two reasons—the data types in the Excel cells don't match those in the corresponding fields of your Access table, or you attempted to

(continues)

Oracle Ex. 1009, p. 273
Oracle v. DeBellis
IPR2024-00221

(continued)

paste records with data that duplicates information in key fields of the table. Both types of errors result in Access creating a Paste Errors table that contains information on the records with the errors. The Paste Errors table for field-type mismatches is similar in purpose and appearance to the Import Errors table described previously in this chapter.

Errors caused by duplicate primary-key violations result in the following series of cascading message boxes:

1. Figure 7.55 shows the first message you receive that indicates a primary-key violation. Access does not offer a Cancel option; therefore, you must click OK.
2. Next, a message box appears enabling you to suppress further error messages (see Figure 7.56). To cancel the append operation, click Cancel. Otherwise, click Yes to try to paste the remaining records without reporting further errors. If you want to see which errors occur as they are encountered, click No.

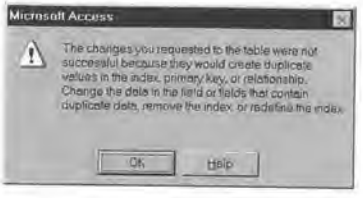


FIG. 7.55 The message informing you that a pasted record duplicates a primary key value.



FIG. 7.56 Choosing whether to view error messages for each pasted record.

3. Finally, a message box reports where the records that couldn't be pasted were placed (see Figure 7.57). Click OK.

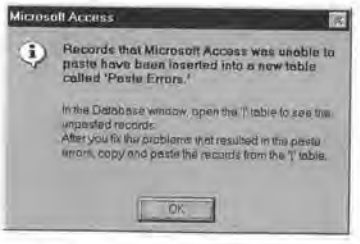


FIG. 7.57 The message reporting that some records could not be pasted.

Oracle Ex. 1009, p. 274
Oracle v. DeBellis
IPR2024-00221

Figure 7.58 illustrates the result of this Pandora's box of messages. The set of four records copied to the Clipboard from Excel had all been previously pasted into the Access table; all four records duplicated key-field values in the existing table. Access pasted records without problems into the table and inserted the four records with duplicate key values into a Paste Errors table.

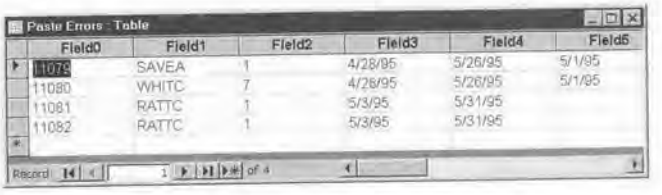


FIG. 7.58 The Paste Errors table showing records that Access couldn't paste from the Clipboard.

If you specified one or more primary-key fields for your table, records that duplicate key field values are not appended. Tables without primary-key fields do not preclude adding duplicate records. The capability to index a nonkey field with the condition, "no duplicates allowed," is useful when you make new entries into a spreadsheet or word-processing document and you want to append the new entries as records to a table. You preserve the uniqueness of the records by preventing the addition of records that duplicate records already in your table.

Note

When pasting or importing large numbers of records to a table, you must specify primary-key fields or a no-duplicates-allowed index for fields that later may become the primary key before you import any data. If you import the data before you create the primary-key fields index, you might find many duplicate records in the table. Then, when Access tries to create a no-duplicates index on the key fields, you see the message, `Can't have duplicate key`. You must manually review all the added records for duplicates because Access doesn't create an Errors table in this case.

If, however, the data you are importing contains redundant information that you ultimately will remove to one or more secondary tables, you must import every record. Do not assign key fields or no-duplicates indexes in this case. The section, "Deleting Redundant Fields in Imported Tables," that follows in this chapter, discusses the requirement to import every record when records contain one-to-many relations.

Tip

If you do encounter the `Can't have duplicate key` error message when trying to establish a primary key, there is a quick way to find the duplicates. Open a new query and select the Find Duplicates Query Wizard. This Wizard creates a query that you can use to quickly find exactly where the duplicates are, without having to search record-by-record through your data.

Replacing Records by Pasting from the Clipboard

You can replace existing records in an Access table by pasting data in the Clipboard over the records. This process is useful when you are updating records with data from another

Oracle Ex. 1009, p. 275
Oracle v. DeBellis
IPR2024-00221

Windows application. The data you paste must have a one-to-one column-to-field correspondence and must begin with the first field of the table. You need not, however, paste new data in all the fields. If no data is pasted in a field that is not included in the copied data's range, that field retains its original values.

To use data in the Clipboard to replace existing records in a table, follow this procedure:

1. Select and copy the data from the other application that you want to paste to the Clipboard, using the method previously described for appending records from Clipboard data.

   If you choose more than one row of data in Excel, for example, the rows must have a one-to-one correspondence with the records to be replaced in the Access table. The one-to-one correspondence is likely to occur only if the table is indexed and the source data you are copying is sorted in the same order as the index. You can paste only contiguous rows from Excel.

2. Open your Access table. To select the records to be replaced by the Clipboard data, click the selection button for a single record or drag the mouse across the buttons for multiple records.

   If you are replacing multiple records, the number of records you select must be equal to or exceed the number of rows you copied to the Clipboard. If the number of records selected is less than the number of rows, the remaining rows are ignored.

   If you are replacing records in a table with key fields or a no-duplicates index, the columns of the replacement data corresponding to the key or indexed fields of the table must match exactly the key fields of the selected records. Otherwise, you see the key duplication error message sequence.

3. Choose Edit, Paste in Access. In this case, rather than appending the records, the contents of the existing records are overwritten, and you see a dialog that tells you how many records will be replaced.

When you use Paste for a replacement record rather than Paste Append for a new record with an identical key field value, Access suppresses the key violation error messages.

Note

If you do not select one or more records to be replaced by the Pasting operation, and the caret is located within a data cell of one of your records or a data cell is selected, Access attempts to paste all the data in the Clipboard to this one cell, rather than to the records. If the data doesn't create a mismatch type error or exceed 255 characters (if the caret is in a Text field), you do not receive a warning message. If you notice unexpected data values in the cell, Access pasted all the data to a single cell. Press Esc before selecting another record; Access restores the original value of the data cell.

Oracle Ex. 1009, p. 276
Oracle v. DeBellis
IPR2024-00221

Modifying Linked and Imported Tables

Access provides a great deal of flexibility in the presentation of the Datasheet view of tables. You can rearrange the sequence of fields in Datasheet view without changing the underlying structure of the database. You also can alter the Caption property of the fields to change the names of the field name buttons. Although you cannot modify the field names, field data types, or Field Length properties of linked tables, you can use the Format property to display the data in linked or imported tables in various ways, as described in the sections that follow.

Restructuring the Presentation of Tables

The basic structure of the tables you link or import is controlled by the structure of the files from which the table was created. The original structure, unfortunately, may not be in the sequence you want to use for data entry. Database design, for example, often displays the key fields in a left-to-right sequence, starting with the first field. This sequence may not be the best sequence for entering data. You can change the order of the fields displayed by dragging and dropping the field name buttons of linked or imported tables to new locations, using the method described in Chapter 4, "Working with Access Databases and Tables."

◀◀ See "Rearranging the Sequence of Fields in a Table," p. 155



Adding Fields to Tables with Imported Data

You can add fields to tables that use imported data. During the development of your database application, you may need to repeatedly import data into your table. Therefore, you should append new fields to the end of the field list in Design view rather than insert them between existing fields. The imported data fills the fields in the sequence of the rows in the field list, and the added fields contain null values. You then can rearrange the display position of the new fields in Datasheet view, as described in the preceding section.

Changing Field Data Types of Imported Files

Access converts all numeric data within imported files to the Double data type. The following list contains recommendations for field data type changes for data imported from any type of source file:

- Use *Integer* or *Long Integer* field data types for numeric fields that do not require decimal values. Your database files consume less disk space, and your applications run faster.
- Change the data type of values that represent money to the data type, *Currency*, to eliminate rounding errors.
- Assign Field Length property values to text-type fields. You should assign Field Length values no greater than the longest text entry you expect to make. If you assign a Field Length value less than the number of characters in the field, characters in positions beyond the Field Length value are lost irretrievably.

Oracle Ex. 1009, p. 277
Oracle v. DeBellis
IPR2024-00221

Because Paradox versions prior to Paradox 4.x do not provide a memo field data type, you often can find comments in large, fixed-width alphanumeric fields. If you import Paradox files that contain fields of comments, you should convert the fields to memo fields unless you plan to export the data back to a Paradox 3.x file. (You cannot export a table that contains a memo-type field to Paradox 3.x files, although you can export tables with memo fields to Paradox 4.x and Paradox for Windows 5.)

Adding or Modifying Other Field Properties

The following table includes properties of both linked and imported tables that you can change:

Property	Description
Caption	Use to change the caption of the field name buttons when using linked files. You can assign any `FieldName` property to imported tables.
Decimal Places	Use to specify the number of decimal places to be displayed for numeric values other than Byte, Integer, and Long Integer.
Default Value	Default values are substituted for data elements missing in the imported or linked file. The default values, however, do not replace zeroes and blank strings that represent "no entry" in the fields of most PC database files.
Format	Use to create custom formats to display the data in the most readable form.
Validation Rule	Validation rules do not affect importing data; they affect only editing the data or appending new records in Access. You cannot use a validation rule to filter imported records. *Filtering*, in this case, means importing only records that meet the validation rule.
Validation Text	Prompts created from validation text assist data entry operators when they begin using a new table.

Deleting Redundant Fields in Imported Tables

The purpose of a relational database is to eliminate redundant data in tables. If you design a database that doesn't use data imported or linked from existing databases, you can eliminate data redundancy by following the normalization rules of relational database design described in Chapter 23, "Exploring Relational Database Design and Implementation."

When you link files, however, you are at the mercy of the database designer who originally created the files. Any redundant data the linked file contains is likely to remain a permanent characteristic of the file. Existing database structures are substantially inert; developers of applications that work usually are reluctant to make changes. Changes may introduce new problems that can later return to haunt the developers.

If you import table data from a file or append records by pasting from the Clipboard, you can eliminate data redundancy by restructuring the resulting tables. An Invoice table, for example, may contain one record for each invoice. When a customer makes more than one purchase, the customer information is duplicated in the invoice file for each purchase. In this situation, you need to create separate tables for customers and invoices.

Oracle Ex. 1009, p. 278
Oracle v. DeBellis
IPR2024-00221

The Customer table should contain one record per customer, with name and address information. The Invoice table should contain one record per customer with date, amount, and other information specific to the transaction.

Removing redundancy from existing tables by dividing them into two or more related tables is not a simple task, even if you use the Table Analyzer Wizard (accessed by choosing Tools, Analyze, Table). You must either manually (or with the help of the Table Analyzer) create a query that establishes the relationship between two tables that contain one record for each invoice based on a new primary-key field. Then, you delete the duplicate records from the Customer table. Fortunately, the Import Errors table can do much of the work; import the data into a new table with primary-key fields or a no-duplicate index on the customer name and address.

Another type of redundancy is the presence of fields in tables that have values calculated from other fields, either within the table or in the fields of related tables. Any field with values derived from combinations of values within the table or accessible in any other related table within the database or linked to the database contains redundant data. In this case, just remove the redundant field and perform the calculation with a query. Do not remove redundant fields, however, until you verify that your calculated values match those in the redundant field to be replaced.

You need to learn about queries and joining tables—the subject of the next four chapters—to know how to eliminate redundancy in imported tables. The point of this discussion is to let you know that you can perform much of the restructuring by using specialized Access operations and to caution you not to try to remove duplicate data from imported tables prematurely.

Exporting Data from Access Tables

You can export data contained in Access tables in any format you can use to import data. The most common export operation with PC RDBMSs is the creation of mail-merge files, used with word-processing applications and spreadsheet files, for data analysis. In most cases, you might want to export the result of a query, enabling you to select specific records to export rather than the entire contents of a file. Exporting tables created from action queries is discussed in Chapter 11, "Using Action Queries."

▶▶ See "Joining Tables to Create Multitable Queries," p. 322



Exporting Data Through the Windows Clipboard

If a Windows application is to use your exported data, the simplest method to export data is to copy the records you want to export to the Windows Clipboard, and then paste the data into the other application. You can copy specific groups of contiguous records to the Clipboard by using the techniques described in Chapter 5, "Entering, Editing, and Validating Data in Tables."

Oracle Ex. 1009, p. 279
Oracle v. DeBellis
IPR2024-00221

Tables To create a merge data file from the Customers table of the Northwind sample database for use with Word, follow these steps:

1. Open the Northwind.mdb database.
2. Open the Customers table from the table list.
3. Select the records to copy by selecting the upper-left corner of the block with the F2 key, and then holding down the Shift key and using the arrow keys to define the records you want to copy to the Clipboard. Alternatively, you can select the area to copy by placing the cursor at the upper left corner of the block you want to copy; the cursor turns into a big plus symbol. Hold down the left mouse button and drag to the lower-right corner of the block. Datasheet view should look like the window in Figure 7.59. (If you want to select all fields of all records in the table, choose Edit, Select All Records.)
4. Press Ctrl+C or choose Edit, Copy to copy the selected records to the Clipboard.
5. Open Word and choose File, New to create a new document for your merge data file.
6. Press Ctrl+V or choose Edit, Paste to paste the records from the Clipboard into your new document. Access 97 pastes the records as a fully formatted table in Word, as illustrated by Figure 7.60. The column widths you select in Access are used to define the column widths of the Word table.

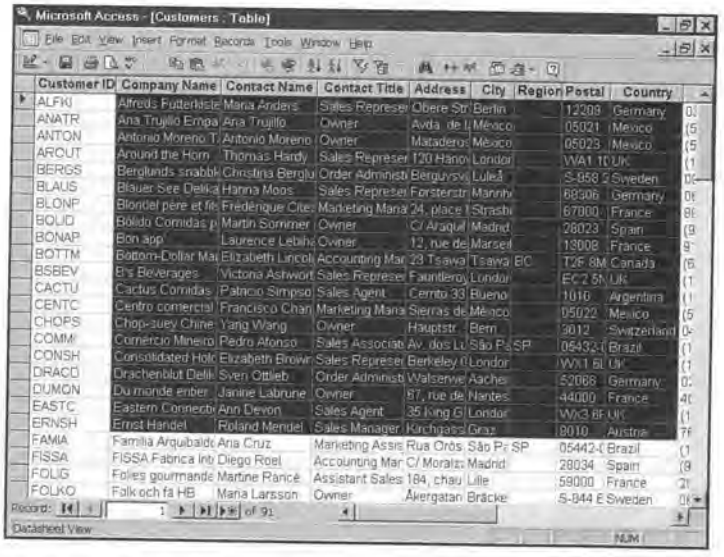


FIG. 7.59 Selecting records in an Access table to copy to the Clipboard.

Oracle Ex. 1009, p. 280
Oracle v. DeBellis
IPR2024-00221