

Contents

| | |
|--|-------------|
| Introduction | xxxi |
| Chapter 1: RDBMS Basics: What Makes Up a SQL Server Database? | 1 |
| An Overview of Database Objects | 1 |
| The Database Object | 2 |
| The master Database | 3 |
| The model Database | 4 |
| The msdb Database | 4 |
| The tempdb Database | 5 |
| ReportServer | 5 |
| ReportServerTempDB | 5 |
| AdventureWorks2008 | 5 |
| AdventureWorksLT2008 | 6 |
| AdventureWorksDW2008 | 6 |
| The pubs Database | 6 |
| The Northwind Database | 6 |
| The Transaction Log | 6 |
| The Most Basic Database Object: Table | 7 |
| Indexes | 7 |
| Triggers | 8 |
| Constraints | 8 |
| Filegroups | 8 |
| Diagrams | 8 |
| Views | 9 |
| Stored Procedures | 10 |
| User-Defined Functions | 10 |
| Users and Roles | 11 |
| Rules | 11 |
| Defaults | 11 |
| User-Defined Data Types | 11 |
| Full-Text Catalogs | 12 |
| SQL Server Data Types | 12 |
| NULL Data | 18 |
| SQL Server Identifiers for Objects | 18 |
| What Gets Named? | 18 |
| Rules for Naming | 19 |
| Summary | 19 |

Contents

| | |
|--|-----------|
| Chapter 2: Tools of the Trade | 21 |
| Books Online | 21 |
| SQL Server Configuration Manager | 23 |
| Service Management | 23 |
| Network Configuration | 24 |
| The Protocols | 25 |
| Named Pipes | 26 |
| TCP/IP | 26 |
| Shared Memory | 27 |
| On to the Client | 27 |
| SQL Server Management Studio | 28 |
| Getting Started with the Management Studio | 29 |
| Server Type | 29 |
| Server Name | 29 |
| Authentication | 31 |
| Query Window | 33 |
| Getting Started | 33 |
| Results to Text | 35 |
| Results to Grid | 36 |
| Results to File | 36 |
| sqlcmd Mode | 36 |
| Show Execution Plan | 37 |
| The Available Databases Combo Box | 38 |
| The Object Explorer | 38 |
| SQL Server Integration Services (SSIS) | 39 |
| Bulk Copy Program (bcp) | 39 |
| SQL Server Profiler | 40 |
| sqlcmd | 40 |
| PowerShell | 41 |
| Summary | 41 |
| Chapter 3: The Foundation Statements of T-SQL | 43 |
| Getting Started with a Basic SELECT Statement | 44 |
| The SELECT Statement and FROM Clause | 44 |
| The WHERE Clause | 48 |
| ORDER BY | 51 |
| Aggregating Data Using the GROUP BY Clause | 54 |
| Aggregates | 56 |
| Placing Conditions on Groups with the HAVING Clause | 62 |
| Outputting XML Using the FOR XML Clause | 64 |

 Contents

| | |
|---|--------------|
| Making Use of Hints Using the OPTION Clause | 64 |
| The DISTINCT and ALL Predicates | 64 |
| Adding Data with the INSERT Statement | 67 |
| Multirow Inserts | 72 |
| The INSERT INTO . . . SELECT Statement | 73 |
| Changing What You've Got with the UPDATE Statement | 75 |
| The DELETE Statement | 77 |
| Summary | 78 |
| Exercises | 79 |
| 81 | JOINS |
| JOINS | 81 |
| INNER JOINS | 83 |
| How It Works | 88 |
| How an INNER JOIN Is Like a WHERE Clause | 88 |
| How It Works | 90 |
| OUTER JOINS | 91 |
| The Simple OUTER JOIN | 91 |
| How It Works | 94 |
| How It Works | 94 |
| Finding Orphan or Non-Matching Records | 94 |
| Dealing with More Complex OUTER JOINS | 97 |
| Seeing Both Sides with FULL JOINS | 101 |
| How It Works | 103 |
| CROSS JOINS | 103 |
| Exploring Alternative Syntax for Joins | 104 |
| An Alternative INNER JOIN | 105 |
| An Alternative OUTER JOIN | 105 |
| An Alternative CROSS JOIN | 106 |
| The UNION | 107 |
| How It Works | 109 |
| Summary | 111 |
| Exercises | 112 |
| Chapter 5: Creating and Altering Tables | 113 |
| Object Names in SQL Server | 113 |
| Schema Name (aka Ownership) | 114 |
| A Little More About Schemas | 114 |
| The Default Schema: dbo | 115 |
| The Database Name | 116 |
| Naming by Server | 116 |
| Reviewing the Defaults | 117 |

Contents

| | |
|--|------------|
| The CREATE Statement | 117 |
| CREATE DATABASE | 117 |
| ON | 118 |
| NAME | 119 |
| FILENAME | 119 |
| SIZE | 119 |
| MAXSIZE | 119 |
| FILEGROWTH | 120 |
| LOG ON | 120 |
| COLLATE | 120 |
| FOR ATTACH | 120 |
| WITH DB CHAINING ON OFF | 121 |
| TRUSTWORTHY | 121 |
| Building a Database | 121 |
| CREATE TABLE | 123 |
| Table and Column Names | 124 |
| Data Types | 125 |
| DEFAULT | 126 |
| IDENTITY | 126 |
| NOT FOR REPLICATION | 126 |
| ROWGUIDCOL | 127 |
| COLLATE | 128 |
| NULL/NOT NULL | 128 |
| Column Constraints | 128 |
| Computed Columns | 128 |
| Table Constraints | 130 |
| ON | 130 |
| TEXTIMAGE_ON | 130 |
| Creating a Table | 130 |
| The ALTER Statement | 136 |
| ALTER DATABASE | 136 |
| Option and Termination Specs | 139 |
| ALTER TABLE | 140 |
| The DROP Statement | 143 |
| Using the GUI Tool | 144 |
| Creating a Database Using the Management Studio | 145 |
| Backing into the Code: The Basics of Creating Scripts with the Management Studio | 150 |
| Summary | 151 |
| Exercises | 152 |

Contents

| | |
|---|------------|
| Chapter 6: Constraints | 153 |
| Types of Constraints | 154 |
| Domain Constraints | 154 |
| Entity Constraints | 154 |
| Referential Integrity Constraints | 155 |
| Constraint Naming | 156 |
| Key Constraints | 157 |
| PRIMARY KEY Constraints | 157 |
| Creating the Primary Key at Table Creation | 158 |
| Creating a Primary Key on an Existing Table | 159 |
| FOREIGN KEY Constraints | 160 |
| Adding a Foreign Key to an Existing Table | 161 |
| Making a Table Self-Referencing | 162 |
| Cascading Actions | 164 |
| Other Things to Think About with Foreign Keys | 169 |
| UNIQUE Constraints | 170 |
| Creating UNIQUE Constraints on Existing Tables | 171 |
| CHECK Constraints | 172 |
| DEFAULT Constraints | 173 |
| Defining a DEFAULT Constraint in Your CREATE TABLE Statement | 174 |
| Adding a DEFAULT Constraint to an Existing Table | 175 |
| Disabling Constraints | 175 |
| Ignoring Bad Data When You Create the Constraint | 176 |
| Temporarily Disabling an Existing Constraint | 178 |
| Rules and Defaults — Cousins of Constraints | 180 |
| Rules | 180 |
| Dropping Rules | 182 |
| Defaults | 182 |
| Dropping Defaults | 182 |
| Determining Which Tables and Data Types Use a Given Rule or Default | 183 |
| Triggers for Data Integrity | 183 |
| Choosing What to Use | 183 |
| Summary | 185 |
| Chapter 7: Adding More to Our Queries | 187 |
| What Is a Subquery? | 188 |
| Building a Nested Subquery | 189 |
| Nested Queries Using Single-Value SELECT Statements | 189 |
| Nested Queries Using Subqueries That Return Multiple Values | 190 |
| Using a Nested SELECT to Find Orphaned Records | 191 |

Contents

| | |
|---|------------|
| Correlated Subqueries | 192 |
| How Correlated Subqueries Work | 192 |
| Correlated Subqueries in the WHERE Clause | 192 |
| Correlated Subqueries in the SELECT List | 195 |
| Dealing with NULL Data — the ISNULL Function | 196 |
| Derived Tables | 197 |
| The EXISTS Operator | 200 |
| Using EXISTS in Other Ways | 202 |
| Mixing Data Types: CAST and CONVERT | 203 |
| The MERGE Command | 206 |
| A Brief Look at BY TARGET versus BY SOURCE | 209 |
| The Output Clause | 210 |
| Performance Considerations | 212 |
| JOINS versus Subqueries versus ? | 212 |
| Summary | 214 |
| Exercises | 214 |
| | |
| Chapter 8: Being Normal: Normalization and Other Basic Design Issues | 215 |
| | |
| Tables | 216 |
| Keeping Your Data “Normal” | 216 |
| Before the Beginning | 217 |
| The First Normal Form | 260 |
| The Second Normal Form | 260 |
| The Third Normal Form | 260 |
| Other Normal Forms | 260 |
| Relationships | 260 |
| One-to-One | 260 |
| Zero or One-to-One | 260 |
| One-to-One or Many | 260 |
| One-to-Zero, One, or Many | 260 |
| Many-to-Many | 260 |
| Diagramming | 260 |
| Tables | 260 |
| Adding and Deleting Tables | 260 |
| Editing Table Properties and Objects That Belong to the Table | 260 |
| Properties Window | 260 |
| Relationships | 260 |
| Indexes/Keys | 260 |
| Check Constraints | 260 |
| Relationships | 260 |
| Adding Relationships in the Diagramming Tool | 260 |

 Contents

| | |
|---|------------|
| De-Normalization | 260 |
| Beyond Normalization | 260 |
| Keep It Simple | 260 |
| Choosing Data Types | 260 |
| Err on the Side of Storing Things | 260 |
| Drawing Up a Quick Example | 260 |
| Creating the Database | 260 |
| Adding the Diagram and Our Initial Tables | 260 |
| Adding the Relationships | 260 |
| Adding Some Constraints | 260 |
| Summary | 260 |
| Exercises | 260 |
| | |
| Chapter 9: SQL Server Storage and Index Structures | 261 |
| | |
| SQL Server Storage | 261 |
| The Database | 261 |
| The Extent | 262 |
| The Page | 262 |
| Page Splits | 263 |
| Rows | 263 |
| Sparse Columns | 263 |
| Understanding Indexes | 264 |
| B-Trees | 265 |
| Page Splits — A First Look | 267 |
| How Data Is Accessed in SQL Server | 268 |
| Use of Table Scans | 268 |
| Use of Indexes | 269 |
| Index Types and Index Navigation | 269 |
| Clustered Indexes | 270 |
| Non-Clustered Indexes on a Heap | 272 |
| Non-Clustered Indexes on a Clustered Table | 273 |
| Creating, Altering, and Dropping Indexes | 276 |
| The CREATE INDEX Statement | 277 |
| ASC/DESC | 277 |
| INCLUDE | 278 |
| WITH | 278 |
| PAD_INDEX | 278 |
| FILLFACTOR | 279 |
| IGNORE_DUP_KEY | 279 |
| DROP_EXISTING | 279 |
| STATISTICS_NORECOMPUTE | 280 |

Contents

| | |
|---|------------|
| SORT_IN_TEMPDB | 280 |
| ONLINE | 281 |
| ALLOW ROW/PAGE LOCKS | 281 |
| MAXDOP | 281 |
| ON | 282 |
| Creating XML Indexes | 282 |
| The Primary XML Index | 282 |
| Secondary XML Indexes | 283 |
| Implied Indexes Created with Constraints | 283 |
| Creating Indexes on Sparse and Geospatial Columns | 283 |
| Choosing Wisely: Deciding What Index Goes Where and When | 283 |
| Selectivity | 284 |
| Watching Costs: When Less Is More | 284 |
| Choosing That Clustered Index | 285 |
| The Pros | 286 |
| The Cons | 286 |
| Column Order Matters | 287 |
| ALTER INDEX | 288 |
| Index Name | 289 |
| Table or View Name | 289 |
| REBUILD | 289 |
| DISABLE | 289 |
| REORGANIZE | 290 |
| Dropping Indexes | 290 |
| Take a Hint from the Query Plan | 290 |
| Use the Database Engine Tuning Advisor | 290 |
| Maintaining Your Indexes | 291 |
| Fragmentation | 291 |
| Identifying Fragmentation vs. Likelihood of Page Splits | 292 |
| ALTER INDEX and FILLFACTOR | 294 |
| Summary | 296 |
| Exercises | 297 |
| | |
| Chapter 10: Views | 299 |
| | |
| Simple Views | 299 |
| Views as Filters | 324 |
| How It Works | 324 |
| More Complex Views | 324 |
| The DATEADD and CAST Functions | 324 |
| Using a View to Change Data — Before INSTEAD OF Triggers | 324 |
| Dealing with Changes in Views with Joined Data | 324 |

Contents

| | |
|---|------------|
| Required Fields Must Appear in the View or Have Default Value | 324 |
| Limit What's Inserted into Views — WITH CHECK OPTION | 324 |
| How It Works | 324 |
| Editing Views with T-SQL | 324 |
| Dropping Views | 324 |
| Creating and Editing Views in the Management Studio | 324 |
| Editing Views in the Management Studio | 324 |
| Auditing: Displaying Existing Code | 324 |
| Protecting Code: Encrypting Views | 324 |
| About Schema Binding | 324 |
| Making Your View Look Like a Table with VIEW_METADATA | 324 |
| Indexed (Materialized) Views | 324 |
| Summary | 324 |
| Exercises | 324 |
| | |
| Chapter 11: Writing Scripts and Batches | 325 |
| | |
| Script Basics | 325 |
| The USE Statement | 326 |
| Declaring Variables | 327 |
| Setting the Value in Your Variables | 327 |
| Reviewing System Functions | 329 |
| Using @@IDENTITY | 331 |
| How It Works | 333 |
| Using @@ROWCOUNT | 334 |
| Batches | 335 |
| A Line to Itself | 335 |
| Each Batch Is Sent to the Server Separately | 336 |
| GO Is Not a T-SQL Command | 336 |
| Errors in Batches | 337 |
| When to Use Batches | 337 |
| Statements That Require Their Own Batch | 337 |
| Using Batches to Establish Precedence | 338 |
| sqlcmd | 340 |
| How It Works | 344 |
| Dynamic SQL: Generating Your Code On the Fly with the EXEC Command | 344 |
| The Gotchas of EXEC | 346 |
| The Scope of EXEC | 346 |
| Security Contexts and EXEC | 348 |
| Use of Functions in Concatenation and EXEC | 348 |
| EXEC and UDFs | 349 |

Contents

| | |
|---|------------|
| Control-of-Flow Statements | 349 |
| The IF ... ELSE Statement | 350 |
| The ELSE Clause | 351 |
| Grouping Code into Blocks | 352 |
| The CASE Statement | 354 |
| A Simple CASE | 354 |
| A Searched CASE | 356 |
| Looping with the WHILE Statement | 359 |
| The WAITFOR Statement | 360 |
| The DELAY Parameter | 361 |
| The TIME Parameter | 361 |
| TRY/CATCH Blocks | 361 |
| Summary | 364 |
| Exercises | 365 |
| | |
| Chapter 12: Stored Procedures | 367 |
| | |
| Creating the Sproc: Basic Syntax | 368 |
| An Example of a Basic Sproc | 368 |
| Changing Stored Procedures with ALTER | 369 |
| Dropping Sprocs | 370 |
| Parameterization | 370 |
| Declaring Parameters | 370 |
| Supplying Default Values | 372 |
| Creating Output Parameters | 373 |
| Confirming Success or Failure with Return Values | 376 |
| How to Use RETURN | 376 |
| More On Dealing with Errors | 378 |
| The Way We Were ... | 379 |
| Handling Inline Errors | 379 |
| Making Use of @@ERROR | 380 |
| Using @@ERROR in a Sproc | 381 |
| Handling Errors Before They Happen | 384 |
| Manually Raising Errors | 390 |
| Message ID/Message String | 390 |
| Severity | 390 |
| State | 391 |
| Error Arguments | 391 |
| WITH <option> | 393 |
| Adding Your Own Custom Error Messages | 393 |
| @lang | 394 |

Contents

| | |
|---|------------|
| @with_log | 394 |
| @replace | 394 |
| Using sp_addmessage | 394 |
| Removing an Existing Custom Message | 395 |
| What a Sproc Offers | 395 |
| Creating Callable Processes | 395 |
| Using Sprocs for Security | 396 |
| Sprocs and Performance | 396 |
| When a Good Sproc Goes Bad | 398 |
| Extended Stored Procedures (XPs) | 399 |
| A Brief Look at Recursion | 399 |
| Debugging | 402 |
| Starting the Debugger | 402 |
| Parts of the Debugger | 402 |
| The Locals Window | 403 |
| The Watch Window | 404 |
| The Call Stack Window | 405 |
| The Output Window | 405 |
| The Command Window | 405 |
| Using the Debugger Once It's Started | 406 |
| .NET Assemblies | 407 |
| Summary | 409 |
| | |
| Chapter 13: User-Defined Functions | 411 |
| | |
| What a UDF Is | 411 |
| UDFs Returning a Scalar Value | 412 |
| UDFs That Return a Table | 416 |
| Understanding Determinism | 423 |
| Debugging User-Defined Functions | 425 |
| .NET in a Database World | 425 |
| Summary | 426 |
| Exercise | 426 |
| | |
| Chapter 14: Transactions and Locks | 427 |
| | |
| Transactions | 427 |
| BEGIN TRAN | 428 |
| COMMIT TRAN | 429 |
| ROLLBACK TRAN | 429 |
| SAVE TRAN | 429 |

Contents

| | |
|---|------------|
| How the SQL Server Log Works | 430 |
| Failure and Recovery | 431 |
| Transaction 1 | 432 |
| Transaction 2 | 432 |
| Transaction 3 | 432 |
| Transaction 4 | 433 |
| Transaction 5 | 433 |
| Implicit Transactions | 433 |
| Locks and Concurrency | 433 |
| What Problems Can Be Prevented by Locks | 434 |
| Dirty Reads | 434 |
| Non-Repeatable Reads | 435 |
| Phantoms | 436 |
| Lost Updates | 437 |
| Lockable Resources | 437 |
| Lock Escalation and Lock Effects on Performance | 438 |
| Lock Modes | 438 |
| Shared Locks | 438 |
| Exclusive Locks | 439 |
| Update Locks | 439 |
| Intent Locks | 439 |
| Schema Locks | 440 |
| Bulk Update Locks | 440 |
| Lock Compatibility | 440 |
| Specifying a Specific Lock Type — Optimizer Hints | 441 |
| Determining Locks Using the Management Studio | 441 |
| Setting the Isolation Level | 442 |
| READ COMMITTED | 443 |
| READ UNCOMMITTED | 443 |
| REPEATABLE READ | 444 |
| SERIALIZABLE | 444 |
| SNAPSHOT | 445 |
| Dealing with Deadlocks (aka “a 1205”) | 445 |
| How SQL Server Figures Out There’s a Deadlock | 446 |
| How Deadlock Victims Are Chosen | 446 |
| Avoiding Deadlocks | 446 |
| Use Objects in the Same Order | 446 |
| Keeping Transactions as Short as Possible | 447 |
| Use the Lowest Transaction Isolation Level Possible | 447 |
| No Open-Ended Transactions | 448 |
| Summary | 448 |

| | |
|---|------------|
| Chapter 15: Triggers | 451 |
| What Is a Trigger? | 452 |
| ON | 453 |
| WITH ENCRYPTION | 454 |
| The FOR AFTER vs. the INSTEAD OF Clause | 454 |
| FOR AFTER | 456 |
| WITH APPEND | 456 |
| NOT FOR REPLICATION | 457 |
| AS | 457 |
| Using Triggers for Data Integrity Rules | 457 |
| Dealing with Requirements Sourced from Other Tables | 457 |
| Using Triggers to Check the Delta of an Update | 459 |
| Using Triggers for Custom Error Messages | 461 |
| Other Common Uses for Triggers | 461 |
| Other Trigger Issues | 462 |
| Triggers Can Be Nested | 462 |
| Triggers Can Be Recursive | 462 |
| Triggers Don't Prevent Architectural Changes | 463 |
| Triggers Can Be Turned Off Without Being Removed | 463 |
| Trigger Firing Order | 464 |
| Controlling Firing Order for Logic Reasons | 465 |
| Controlling Firing Order for Performance Reasons | 465 |
| INSTEAD OF Triggers | 465 |
| Performance Considerations | 466 |
| Triggers Are Reactive Rather Than Proactive | 466 |
| Triggers Don't Have Concurrency Issues with the Process That Fires Them | 466 |
| Using IF UPDATE() and COLUMNS_UPDATED | 467 |
| The UPDATE() Function | 467 |
| The COLUMNS_UPDATED() Function | 468 |
| Keep It Short and Sweet | 469 |
| Don't Forget Triggers When Choosing Indexes | 469 |
| Try Not to Roll Back Within Triggers | 470 |
| Dropping Triggers | 470 |
| Debugging Triggers | 470 |
| Summary | 471 |

Contents

| | |
|---|------------|
| Chapter 16: A Brief XML Primer | 473 |
| XML Basics | 474 |
| Parts of an XML Document | 475 |
| The Document | 475 |
| Declaration | 476 |
| Elements | 477 |
| Nodes | 478 |
| Attributes | 479 |
| No Defects — Being Well Formed | 479 |
| An XML Example | 480 |
| Determining Elements vs. Attributes | 483 |
| Namespaces | 484 |
| Element Content | 485 |
| Being Valid vs. Being Well Formed — Schemas and DTDs | 486 |
| What SQL Server Brings to the Party | 486 |
| Defining a Column as Being of XML Type | 487 |
| XML Schema Collections | 488 |
| Creating, Altering, and Dropping XML Schema Collections | 490 |
| CREATE XML SCHEMA COLLECTION | 490 |
| ALTER XML SCHEMA COLLECTION | 491 |
| DROP XML SCHEMA COLLECTION | 491 |
| XML Data Type Methods | 492 |
| .query (SQL Server's Implementation of XQuery) | 492 |
| .value | 493 |
| .modify | 494 |
| .nodes | 495 |
| .exist | 497 |
| Enforcing Constraints Beyond the Schema Collection | 498 |
| Retrieving Relational Data in XML Format | 498 |
| The FOR XML Clause | 498 |
| RAW | 500 |
| AUTO | 502 |
| EXPLICIT | 503 |
| PATH | 504 |
| PATH 101 | 505 |
| OPENXML | 509 |
| A Brief Word on XSLT | 514 |
| Summary | 516 |

| | |
|--|------------|
| Chapter 17: Reporting for Duty, Sir! A Look At Reporting Services | 517 |
| Reporting Services 101 | 518 |
| Building Simple Report Models | 518 |
| Data Source Views | 523 |
| Manipulating Relationships Between Objects In Data Source Views | 525 |
| Building the Data Model | 526 |
| Deploying Our Model | 531 |
| Report Creation | 531 |
| A Few Last Words on Report Models | 536 |
| Report Server Projects | 536 |
| How It Works | 541 |
| Deploying the Report | 542 |
| Summary | 542 |
| Getting Integrated with Integration Services | 543 |
| Understanding the Problem | 543 |
| Using the Import/Export Wizard to Generate Basic Packages | 544 |
| Examining Package Basics | 551 |
| Connection Managers | 552 |
| The Package Editor Pane | 552 |
| Control Flow | 552 |
| Data Flow | 554 |
| Event Handlers | 556 |
| Package Explorer | 557 |
| Executing Packages | 558 |
| Executing a Package Inside the Development Studio | 558 |
| Using the Execute Package Utility | 558 |
| General | 559 |
| Configurations | 559 |
| Command Files | 559 |
| Connection Managers | 560 |
| Execution Options | 560 |
| Reporting | 560 |
| Logging | 560 |
| Set Values | 560 |
| Verification | 560 |
| Command Line | 560 |
| Executing the Package | 561 |
| Executing Using the SQL Server Agent | 561 |
| Executing a Package from Within a Program | 561 |
| A Final Word on Packages | 561 |
| Summary | 561 |

Contents

| | |
|--|------------|
| Chapter 19: Playing Administrator | 563 |
| Scheduling Jobs | 564 |
| Creating an Operator | 565 |
| Creating an Operator Using Management Studio | 565 |
| Creating Jobs and Tasks | 566 |
| Creating Jobs and Tasks Using Management Studio | 567 |
| Backup and Recovery | 575 |
| Creating a Backup: a.k.a. "A Dump" | 575 |
| Backup Type | 576 |
| Backup Set | 577 |
| Destination | 577 |
| Options | 577 |
| Schedule | 578 |
| Recovery Models | 578 |
| Recovery | 579 |
| Restoring to a Different Location | 580 |
| Recovery Status | 580 |
| Index Maintenance | 580 |
| ALTER INDEX | 581 |
| Index Name | 581 |
| Table or View Name | 581 |
| REBUILD | 581 |
| DISABLE | 582 |
| REORGANIZE | 582 |
| How It Works | 583 |
| Archiving Data | 583 |
| Policy Based Management | 584 |
| Summary | 584 |
| Exercises | 585 |
| Appendix A: System Functions | 587 |
| Appendix B: Very Simple Connectivity Examples | 649 |
| Index | 657 |