Accelerated SQL Server 2008

Robert E. Walters, Michael Coles, Robert Rae,
Fabio Ferracchiati, and Donald Farmer

Apress®

Accelerated SQL Server 2008

ISBN-13 (pbk): 978-1-59059-969-3

ISBN-10 (pbk): 1-59059-969-1

ISBN-13 (electronic): 978-1-4302-0606-4

ISBN-10 (electronic): 1-4302-0606-3

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Lead Editor: Jonathan Gennick
Technical Reviewer: Fabio Ferracchiati
Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Kevin Goff, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Denise Santoro Lincoln
Copy Editor: Marilyn Smith
Associate Production Director: Kari Brooks-Copony
Production Editor: Ellie Fountain
Compositor and Artist: Kinetic Publishing Services, LLC
Proofreader: April Eddy
Indexer: Broccoli Information Management
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit http://www.springeronline.com.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit http://www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at http://www.apress.com/info/bulksales.

This book is dedicated to Jim Gray, whose early work with SQL Server paved the way for the enterprise-ready data platform it is today.

Contents at a Glance

About the Authors xxi
Acknowledgments xxiii
Introduction xxv

PART 1 Overview of SQL Server

CHAPTER 1 SQL Server 2008 Overview 3
CHAPTER 2 SQL Server Installation and Configuration 9

PART 2 Enterprise Data Platform

CHAPTER 3 Policy Management 27
CHAPTER 4 High Availability 47
CHAPTER 5 Performance 81
CHAPTER 6 Security 109
CHAPTER 7 SQL Server Encryption 141
CHAPTER 8 Automation and Monitoring 165
CHAPTER 9 Service Broker 221
CHAPTER 10 Integrated Full-Text Search 263

PART 3 Development in SQL Server

CHAPTER 11 New Datatypes in SQL Server 2008 287
CHAPTER 12 T-SQL Enhancements for Developers 315
CHAPTER 13 T-SQL Enhancements for DBAs 367
CHAPTER 14 .NET Integration 403
CHAPTER 15 Programming Assemblies 427

CHAPTER 16 SQL Server and XML 473
CHAPTER 17 SQL Server XML and XQuery Support 515
CHAPTER 18 LINQ to SQL 553

PART 4 Business Intelligence in SQL Server

CHAPTER 19 Reporting Services 585
CHAPTER 20 Analysis Services 641
CHAPTER 21 Integration Services 687

INDEX 751

Contents

About the Authors xxi
Acknowledgments xxiii
Introduction xxv

PART 1 Overview of SQL Server

CHAPTER 1 SQL Server 2008 Overview 3

The Vision of SQL Server 2008 3
Enterprise Data Platform 4
Beyond Relational 4
Dynamic Development 4
Pervasive Insight 5
SQL Server 2008 Editions 5
A Word on Server Consolidation 6
Summary 7

CHAPTER 2 SQL Server Installation and Configuration 9

SQL Server Requirements 9
Upgrading to SQL Server 2008 10
Planning the Upgrade 10
Using Upgrade Advisor 12
Performing the Upgrade 17
Summary 23

PART 2 Enterprise Data Platform

CHAPTER 3 Policy Management 27

Needs Addressed by Policy Management 27
Policy Management Components 28
Managed Targets 28
Facets 28
Conditions 31
Policies 34

Sample Policies 38
Policy Management Administration 42
Policy Status 42
PM Security 44
Summary 45

CHAPTER 4 **High Availability** 47

High Availability Defined 48
Database Mirroring 49
How Database Mirroring Works 49
Managing Database Mirroring Using T-SQL 54
Managing Database Mirroring Using Management Studio 61
Full-Text Indexing and Mirroring 65
Service Broker and Database Mirroring 66
Client Applications and Database Mirroring 66
Monitoring Database Mirroring 67
Performance Considerations for Database Mirroring 70
Limitations of Database Mirroring 70
Database Snapshots and Mirroring 71
How Database Snapshots Work 71
Managing Snapshots Using T-SQL 73
Performance Considerations When Using Snapshots on Mirrors 74
Using and Monitoring Database Snapshots 74
Limitations of Database Snapshots 75
Windows Clustering in SQL Server 76
SQL Server Replication 76
Snapshot Replication 77
Merge Replication 77
Transaction Replication 78
Replication and Mirroring 79
Reducing Planned Downtime 79
Adding a CPU to a Running System 79
Adding Memory to a Running System 80
Performing Online Index Operations 80
Partitioning Tables and Indexes 80
Summary 80

CHAPTER 5 **Performance** 81

Managing and Monitoring Resources 81
Data Collector 81
Resource Governor 87

Optimizing Storage 94
Backup Compression 94
Data Compression 95
Improving Query Performance 101
Plan Guide Support 101
Sparse Columns 104
Column Sets 105
Summary 108

CHAPTER 6 **Security** 109

Disabled Database Engine Features 109
Remote Connections 110
Dedicated Administrator Connection 111
.NET Framework 111
Database Mail 111
SQLMail 112
Service Broker, HTTP Connectivity, and Database Mirroring 112
Web Assistant 112
xp_cmdshell XP 113
Ad Hoc Remote Queries 113
OLE Automation XPs 113
SMO and DMO XPs 113
Principals and Securables 114
Principals 114
Securables 121
Permissions 124
Types of Permissions 124
Managing Permissions 125
Code Access Security 128
Imperative and Declarative CAS 128
Using CAS with SQL Server 129
Auditing in SQL Server 2008 132
Where to Write Audit Data 133
What to Audit 135
An Auditing Example 137
Managing Audits 139
Summary 140

CHAPTER 7 SQL Server Encryption 141

Encryption Keys 142
Service Master Key 143
Database Master Keys 144
Asymmetric Keys 146
Certificates 150
Symmetric Keys 152
Transparent Data Encryption 156
Enabling TDE 157
Choosing Between TDE and Column-Level Encryption 158
Extensible Key Management 159
Encryption Without Keys 159
Hashing and Signing Data 160
Security Catalog Views 161
Query Efficiency 162
Summary 163

CHAPTER 8 Automation and Monitoring 165

SQL Server Agent 166
Scheduling Agent Jobs 166
Permissions for Executing Agent Jobs 171
Proxy Accounts 175
Job Schedule Sharing 177
Logging Agent Job-Step Output 180
WMI Events and Agent Alerts 180
Agent Performance Counters 182
Agent Upgrade 183
Maintenance Plans 184
Scheduling Maintenance Subplans 187
Managing Maintenance Plan Connections 187
Reporting and Logging Maintenance Plans 188
Defining Maintenance Plan Tasks 189
SQLCMD 192
Connecting to SQL Server 192
Passing Variables 193
Using the Dedicated Administrator Connection 194
Creating Scripts 194
PowerShell for SQL Server 195
Introducing PowerShell 196
Using SQL Server PowerShell 198

Database Mail 202
Configuring Database Mail 203
Sending Mail 207
SQL Profiler 208
Performance Monitor Correlation 210
Showplan 212
Deadlock Visualization 214
Extended Events 214
Extended Events Components 215
Extended Events Example: Detecting Deadlocks 217
Summary 219

CHAPTER 9 **Service Broker** 221

What Is Service Broker? 222
Service Broker Architecture 222
Service Broker Scenarios 224
Creating Service Broker Applications 225
Enabling Service Broker 226
Creating Message Types 226
Creating Contracts 226
Creating Queues 227
Creating Services 227
Creating Service Broker Stored Procedures 228
A Simple Service Broker Example 231
Service Broker Routing and Security 236
Creating Distributed Service Broker Applications 236
Distributed Service Broker Example 239
Message Priorities 258
Troubleshooting Service Broker Using SSBDiagnose 261
Summary 262

CHAPTER 10 **Integrated Full-Text Search** 263

Creating Full-Text Catalogs and Indexes 263
Using the GUI to Create a Full-Text Catalog and Index 264
Using T-SQL to Create a Full-Text Catalog and Index 272
Querying with iFTS 273
FREETEXT Predicate Searches 274
CONTAINS Predicate Searches 275
FREETEXTTABLE and CONTAINSTABLE Function Searches 277
Managing Thesaurus Files 278
Editing Thesaurus Files 279
Reloading a Thesaurus 281

Using Stoplists 281
Searching Documents 282
Creating a Full-Text Index for Documents 282
Querying Documents 283
Managing iFTS 283
Summary 284

PART 3 Development in SQL Server

CHAPTER 11 New Datatypes in SQL Server 2008 287

Spatial Support in SQL Server 2008 287
The GEOMETRY Type 288
The GEOGRAPHY Type 292
How Time Has Changed in SQL Server 293
New Date and Time Datatypes 293
New Date and Time System Functions 297
A New Hierarchical Datatype 299
Filestream Support 304
Enabling Filestream Functionality 305
A Filestream Example 306
Summary 314

CHAPTER 12 T-SQL Enhancements for Developers 315

Notable DML Features 315
Old-Style Outer Joins Deprecated 316
Common Table Expressions 316
TOP 325
Extensions to the FROM Clause 328
OUTPUT 337
Ranking Functions 338
EXCEPT and INTERSECT 344
Synonyms 346
MERGE 347
General Development 350
Error Handling 350
.WRITE Extension to the UPDATE Statement 357
EXECUTE 359
Code Security Context 359
.NET Declarations 362
Declaring and Setting Variables 363
Passing Table-Valued Parameters 364
Summary 366

CHAPTER 13 T-SQL Enhancements for DBAs 367

Locking Enhancements 367
Metadata Views 368
Compatibility Views 369
Catalog Views 369
Dynamic Management Views and Functions 371
Notable SQL Server Performance Monitor Counters 374
DDL Triggers 374
Creating and Altering DDL Triggers 375
Dropping DDL Triggers 376
Enabling and Disabling DDL Triggers 376
Enumerating DDL Triggers Using Catalog Views 376
Programming DDL Triggers with the eventdata() Function 377
Indexing and Performance Enhancements 378
Online Indexing 379
Controlling Locking During Index Creation 379
Creating Indexes with Additional Columns Included 380
Altering Indexes 381
Using Filtered Indexes 383
Using Filtered Statistics 385
Using Statistics for Correlated Datetime Columns 385
Improving Performance of Ordering for Tertiary Collations 386
Table and Index Partitioning 387
Using Indexed Views 393
Using Partition-Aligned Indexed Views 394
Persisting Computed Columns 394
Snapshots 395
SNAPSHOT Isolation Level 395
Database Snapshots 398
Data-Integrity Enhancements 400
Verifying a Database's Pages 400
Putting a Database into an Emergency State 400
Summary 401

CHAPTER 14 .NET Integration 403

Introduction to SQL Server .NET Integration 404
Why Does SQL Server Host the CLR? 404
When to Use CLR Routines 404
When Not to Use CLR Routines 405
How SQL Server Hosts .NET: An Architectural Overview 405

SQL Server .NET Programming Model 406
Enhancements to ADO.NET for SQL Server Hosting 406
Overview of the .NET Namespaces for SQL Server 406
Programming a CLR Stored Procedure 407
Starting a Visual Studio 2008 SQL Server Project 408
Anatomy of a Stored Procedure 411
Adding Parameters 412
Defining the Problem 412
Using the SqlPipe 414
Putting It All Together: Coding the Body of the Stored Procedure 416
Testing the Stored Procedure 418
Debugging the Procedure 420
Throwing Exceptions in CLR Routines 421
Deploying CLR Routines 425
Summary 426

CHAPTER 15 **Programming Assemblies** 427

CLR User-Defined Types 428
Applications for User-Defined Types 428
Adding a User-Defined Type to a SQL Server Project 428
Parts of a User-Defined Type 430
A Simple Example: The PhoneNumber Type 433
Another Example: The StringArray Type 440
Managing User-Defined Types 447
CLR User-Defined Functions 448
Adding a User-Defined Function to a Visual Studio Project 449
The Visual Studio 2008 User-Defined Function Template 449
The SqlFunction Attribute 450
Scalar User-Defined Functions 450
Table-Valued User-Defined Functions 453
Managing CLR User-Defined Functions 457
CLR User-Defined Aggregates 457
Adding a User-Defined Aggregate to a SQL Server Project 458
Parts of a User-Defined Aggregate 459
CLR User-Defined Triggers 466
Adding a CLR User-Defined Trigger to a SQL Server Project 466
Programming CLR Triggers 466
Managing User-Defined Triggers 470
Managing Assemblies 471
Summary 471

CHAPTER 16 SQL Server and XML 473

What Is XML? 473
What Are XPath and the XMLDOM? 474
XPath Syntax 476
XPath Functions 477
The XMLDOM: XML Document Object Model 477
The XPathDocument, XPathNavigator, and XPathExpression Classes . . . 478
Getting XML into the Database 479
SQL Server Configuration for SOAP 480
OPENXML 481
XML Views Using Annotated XML Schemas 486
SQLXML Updategrams 491
XML Bulk-Loading 493
Getting XML Out of the Database 495
FOR XML 495
Templates to Improve Performance 502
Working with XML Data 503
Validation for "Any" Types 503
Date and Time Support 504
Union and List Types 505
Programming SQLXML from .NET and COM 506
SQLXML Classes 506
SQLXML Coding Examples 508
Summary 513

CHAPTER 17 SQL Server XML and XQuery Support 515

Using the XML Datatype 516
Understanding How XML Is Stored by SQL Server 517
Creating XML Columns 518
Setting Permissions for Schema Creation 522
Constraining XML Columns 523
Examining the XML Datatype Limitations 524
Inserting Data into XML Columns 524
Using SSIS with XML Data 524
Bulk-Loading XML 526
Writing a Custom Query or Application 526
Querying XML Data 527
XQuery 101 527
Basic XML Query Methods 533
Cross-Domain Queries 535

Modifying XML Data 536
Inserting an Element 536
Deleting an Element 536
Changing a Node Value 537
Limitations of XML Modification 537
Indexing XML for Performance 537
Understanding How XML Indexing Works 538
Examining Secondary XML Indexes 539
Full-Text Search and the XML Datatype 540
Catalog Views and XML 541
Applications and XML 541
XML Web Services Support 542
Creating an Endpoint 543
Using Advanced Web Services 547
Monitoring Performance of XML Web Services 551
Summary 552

CHAPTER 18 **LINQ to SQL** 553

Object/Relational Mapping 553
Entity-Generation Tools 555
Using SQLMetal 555
Using the Visual Studio LINQ to SQL Classes Designer 557
Analyzing the Generated Code 558
Running LINQ Queries 564
The Select Operator 564
The Where Operator 566
The Join Operator 567
The OrderBy Operator 568
The DataContext Class 570
Inserting, Updating, and Deleting Records 570
Concurrency Conflict Detection 573
Deferred Query Execution 576
Deferred Loading 579
Executing Stored Procedures and User-Defined Functions 580
Summary 582

PART 4 Business Intelligence in SQL Server

CHAPTER 19 Reporting Services 585

Reporting Services Components 586
- Report Server Service 587
- Metadata Catalog 592
- Report Designer in BIDS 592
- Report Designer Preview 594
- SQL Server Management Studio Integration 596
- Reporting Services Configuration Manager 596
- Reporting Services Security 598

Building a Basic Report 599
- Launching the Designer 599
- Working with Data Sources and Datasets 600
- Laying Out and Previewing the Report 601
- Working with Expressions 602
- Deploying Your Report 603

Report Design Advanced Features 603
- Multivalued Parameters 604
- DatePicker for Date Values 605
- Interactive Sorting 606
- Analysis Services Integration 607
- Reporting Services Data Sources 612
- Custom Report Items 614

Visual Studio Integration and ReportViewer Controls 614
- Using WinForm Controls 615
- Working with the ReportViewer Control Programmatically 617
- LocalReport and ServerReport Objects 619

SharePoint Integration 620

End-User Ad Hoc Query and Reporting 621
- The Report Builder Client 621
- The Report Model and Semantic Model Definition Language 622

Report Rendering 623
- Exporting Reports 623
- Rendering Large Reports 625

Data Regions 625
- Tablix Data Region 625
- Gauge Data Region 634
- Updated Chart Data Region 636

Summary 640

CHAPTER 20 **Analysis Services** 641

New Analysis Service Features in SQL Server 2008 642
Improvements in Design Tools 642
Improvements in Monitoring Tools 642
Runtime Improvements 643
Analysis Service Fundamentals 643
Architecture 643
Development Environment 645
Analysis Services Objects and Concepts 645
OLAP, OLTP, and Data Warehouses 646
OLAP and OLTP Configuration 647
OLAP Concepts 648
Analysis Services Projects 650
Starting a New Analysis Services Project 650
Defining Data Sources 651
Editing a Data Source 655
Defining Data Source Views 655
Defining Cubes 658
Configuring Dimensions 661
Deploying Projects 666
Working with Cubes 668
Viewing Cube Structure 668
Browsing Cubes 669
Managing Displayed Data 674
Performing Calculations 676
Working with Key Performance Indicators 679
Using Analysis Services Scripting Language 682
Creating a DSV with a Named Query 683
Viewing the ASSL 684
Summary 685

CHAPTER 21 **Integration Services** 687

An Overview of SSIS Features 687
When Can You Use SSIS? 688
What's New in SSIS? 689
The SSIS Integrated Development Environment 689
Connecting to SSIS in Management Studio 690
Creating a New SSIS Project in BIDS 690
SSIS Fundamentals 691
An Overview of the SSIS Designer 692
A Data Flow Example 697
Event Handlers Design Surface 708

Control Flow and Data Flow Designer Tasks . . . 710
Control Flow Containers and Tasks . . . 711
Data Flow Designer Tasks . . . 723
Change Data Capture . . . 731
Enabling Change Data Capture . . . 732
Extracting Change Data with SSIS . . . 733
Logging . . . 733
Dynamic Package Configuration . . . 737
Variables . . . 740
Configuring Variables . . . 741
Creating Variables . . . 741
Precedence Constraints . . . 742
Checkpoints . . . 743
Transactions . . . 743
Debugging . . . 744
Control Flow and Data Flow Visual Debugging . . . 744
Data Viewers . . . 745
Breakpoints . . . 746
Other Debug Windows . . . 747
SSIS Package Handling . . . 747
The SSIS Package Deployment Utility . . . 747
Migrating SQL Server 2000 DTS Packages . . . 748
Scheduling an SSIS Package . . . 748
Summary . . . 749

INDEX . . . 751

About the Authors

ROBERT E. WALTERS is a data platform technology specialist with Microsoft. He specializes in navigating customers through the powerful features and functionality of relational databases. Rob's extensive experience with Microsoft SQL Server started more than 8 years ago, when he worked as a consultant for Microsoft Consulting Services in Denver, Colorado. Shortly after the dot-com bubble burst, Rob returned to Microsoft's headquarters and worked as a program manager in the SQL Server product unit. There, he owned various features within SQL Server, including SQL Server Agent, various management features, and the security for the database engine.

Rob coauthored *Programming Microsoft SQL Server 2005* (Microsoft Press) and *Pro SQL Server 2005* (Apress). He holds a Bachelor of Science in Electrical Engineering from Michigan State University and a Master of Business Administration from Seattle University.

When not thinking about databases, Rob enjoys spending time with his wife, children, and two Saint Bernard dogs.

MICHAEL COLES has worked in the information technology industry for more than a decade, with an emphasis on database-enabled applications. Previously, he worked in a wide range of industries, including retail, manufacturing, and insurance, to name a few. He currently serves as a database architect and applications developer for a consulting firm specializing in business intelligence solutions. Michael lives in New Jersey, and spends his spare time commuting to and from New York City.

FABIO CLAUDIO FERRACCHIATI is a prolific writer on cutting-edge technologies. Fabio has contributed to more than a dozen books on .NET, C#, Visual Basic, and ASP.NET. He is a .NET Microsoft Certified Solution Developer (MCSD) and lives in Rome, Italy. You can read his blog at `http://www.ferracchiati.com`. Fabio also was the technical reviewer for this book.

ROBERT RAE works as a senior technology specialist in enterprise architecture, focusing on large enterprise accounts for Microsoft. In this role, Robert helps customers better understand how to leverage Microsoft application platform capabilities within their enterprise architectures. Robert spends the vast majority of his time focused on database solutions for business intelligence, data quality, high availability, disaster recovery, and development. Prior to joining Microsoft Robert spent 12 years as president of a consulting firm that focused on enterprise integration and enabling software as a service.

DONALD FARMER has been a member of the Microsoft Business Intelligence team for 7 years. He has worked on both the Analysis Services and Integration Services product teams. Donald is now a principal program manager for SQL Server Analysis Services, working to build the new generation of analytic technology, including predictive analysis, within the Microsoft business intelligence offering. Donald is a popular speaker at international events for both business and technical audiences, with a wide range of interests including data integration, information quality, metadata intelligence, and master data management. He is the author of a number of books and articles. Prior to joining Microsoft, Donald worked not only on business intelligence projects, but also in fields as varied as medieval archaeology and fish farming.

Acknowledgments

As most of you know, writing a technical book requires hundreds of hours of researching, outlining, writing, editing, and reviewing content. I could not have done it without the support of many people. I would like to give a special thanks to the following folks, who gave me valuable feedback and provided timely answers to questions: Dan Jones, Bill Ramos, Richard Waymire, Euan Garden, Steven Gott, Peter Saddow, Srini Acharya, Rick Negrin, Dom Arvisais, and Michiel Wories.

Most of all, I would like to thank my wife and family for giving me the support I needed to get this book done.

Thanks!

Robert E. Walters (*lead author*)

Introduction

Before I describe the contents of this book and why I think you should just take it over to the counter and buy it, I would like to give you an insider's look at the SQL Server 2008 product development cycle. I believe this insight will provide you with a deeper understanding of how SQL Server is continuing to evolve. The rest of this book will show you why SQL Server is enterprise-ready.

For the past 5 years, I was a program manager at Microsoft in the SQL Server product unit. During this time, I owned various features within the product, including SQL Server Agent, SQL Server Express, and most recently, database security.

When I joined SQL Server in 2002, the product team was in year 3 of planning and implementing the Yukon (SQL Server 2005) release. One of my first responsibilities was to own the Create Database/Database Properties dialog in SQL Server Management Studio. After working with the user interface (UI) design team and various UI developers, we crafted the interesting grid-based dialog that you see today in Management Studio. However, arriving at the implemented Create Database dialog was not as straightforward as we wanted.

In our organization, we had separate teams writing the UI, writing the Server Management Objects (SMO) code to support the UI, and writing the code in the database engine itself. One of the more common issues we faced was the orchestration of the three separate teams working on a particular feature. Each of the three teams didn't necessarily put the same priority on the work, and this resulted in situations like having a UI that did nothing because either the SMO or database team didn't write the code to support it at the time. In the end, when it came time to ship the product, there were some features that had no UI support in SQL Server Management Studio. For example, try to manage Service Broker in Management Studio in SQL Server 2005. I will save you the time—there isn't much there.

So why am I airing our dirty laundry? Well, it's not because I want everyone to enjoy the smell. It's because I want to tell you about the dramatic improvements in efficiency that have been made, resulting in a better product for you, the SQL Server customer.

With respect to our software development issues, the upper management in the SQL Server product unit actually cared about the problems people in the product team experienced. When SQL Server 2005 was released, the management set aside a bunch of folks, locked them away (not literally), and had them come up with solutions to the problems. What came as a result was called the SQL Engineering System (SES), which has fundamentally changed the way Microsoft develops SQL Server.

As with other versions of the product, we started with the core themes of the release. In SQL Server 2008's case, these were as follows: mission-critical platform, dynamic development, beyond relational data, and pervasive business insight. These were not just marketing buzzwords, but actually meant something in the SES process. Then another, smaller group came up with scenarios that matched each of these themes. One of the scenarios I was involved with was "secure platform for data." This scenario dealt with issues around data protection. As program managers, we helped define the various improvements that would support this scenario. My specific assignments were the security-related improvements, such as transparent database encryption, Extensible Key Management, and auditing improvements. So, everything we did in the product boiled down to an improvement based on a scenario that was part of a major theme. This kept everyone focused on the common goals for the release.

To address the issues around the mechanics of software development, the SES process defined a number of other measures. One of these measures was a globally ranked improvement list (GRIL), which numbered each improvement across the entire product. The idea was one team couldn't say it had no time to help out another team if that other team was working on a higher-ranked improvement. This ascending list helped keep the hoarding of resources within teams to a minimum and allowed for better collaboration between teams. With a single ranked list, it was also possible to ensure that when an improvement was being made, all teams affected (those dealing with management tools, the database engine, setup, and so on) were brought in and contributed resources as needed.

The end result of the SES process to you, the user of SQL Server, is the following: the quality of the Community Technical Preview (CTP) releases is very high. This is because, by the time each feature is checked in, it has full SMO, tools, and SQL Server Books Online documentation. The improvements made to the product add much more value, since they interact with more parts of the product. Take Resource Governor, for example (a topic covered in Chapter 5 of this book). That improvement affected multiple teams within the product and would have failed miserably if everyone were not in sync and did not treat the feature with the same priority. Finally, it is possible for SQL Server to ship more frequently, since the quality of the code in the main code branch is near release quality.

Who This Book Is For

SQL Server 2008 is an evolution of the third generation of the SQL Server platform. With every release of the product come new features for the database administrator and developer to explore. Because we can't possibly cover absolutely everything in SQL 2008, we focus on the key features and functionality that will rapidly boost your knowledge and skills of this great product. If you know what the acronym DBA stands for and have an interest in SQL Server 2008, then this book is for you!

Valuable Resources

As a SQL Server user, you may have thought of a suggestion to enhance SQL Server, or you may have found an issue with the product. The SQL Server team has a web site that allows you to submit feedback, as well as download the latest CTP releases of the product: `http://connect.microsoft.com/sqlserver`. Don't think that what you submit goes into some database and no one ever reads it. Well, never mind the first part of that statement—the comments actually do go into a database, but people from the product team really do read them! Feedback that is entered using the SQL Server Connect web site automatically goes into our issue-tracking database, and program managers and others from the respective feature areas periodically comb through the entries. So don't think you are wasting your time by submitting suggestions and issues. On the contrary, they are all read and responded to by SQL Server team members.

The Microsoft Developer Network (MSDN) forums provide an opportunity to post questions and have them answered by the community and those in the product team. The SQL Server forums can be found at `http://forums.microsoft.com/msdn/default.aspx?forumgroupid=19&siteid=1`. These forums are very active, with thousands of posts in each topic. The response time is quick, as members of the product team actively monitor and respond to postings.

How This Book Is Structured

This book is written in such a way that you can read through the book cover to cover or dip in and out for specific topics. It is structured into 21 chapters divided into four parts, as follows:

Part 1, Overview of SQL Server: Chapter 1 discusses the vision for SQL Server 2008, the various editions of SQL Server, and SQL Server consolidation. Chapter 2 covers SQL Server installation and configuration. The experience of installing SQL Server 2008 is completely new, and those of us who have suffered battle scars installing previous versions of SQL Server will be in for a pleasant surprise.

Part 2, Enterprise Data Platform: The eight chapters in this part cover key improvements related to relational database concepts.

- Chapter 3 covers Policy Management (PM), the new policy-based framework for SQL Server. The possibilities of PM are endless. Examples of use include allowing administrators to lock down server configurations and enforce that developers use proper naming conventions when creating their objects in the database.
- Chapter 4 is about the key high availability (HA) features in SQL Server 2008, including database snapshots, Windows clustering, SQL Server replication, and other ways to reduce downtime. However, its focus is database mirroring, the newest of the HA technologies.
- Chapter 5 explores the enhancements in SQL Server 2008 as they relate to managing and monitoring resources, increasing performance by optimizing storage, and improving query performance. Specific features covered include the Data Collector, Resource Governor, backup and data compression, and sparse column support, to name a few.
- Chapter 6 covers the core security concepts included in SQL Server, as well as the new auditing feature in SQL Server 2008.
- Chapter 7 discusses encryption capabilities in SQL Server, which have been expanded enough to make encryption a topic for its own chapter! This chapter covers encrypting data using SQL Server, as well as the new transparent database encryption and extensive key management features of SQL Server 2008.
- Chapter 8 covers automation and monitoring. The plethora of tools available in SQL Server contributes to its ease of use compared with other relational database products on the market. SQL Server 2008 includes a new PowerShell provider, as well as a new event framework called Extended Events. This chapter covers these topics, as well as others, including SQL Server Agent, maintenance plans, and SQLCMD.
- Chapter 9 is about Service Broker, which is in its second release with SQL Server 2008. This chapter provides an overview of Service Broker and discusses the key improvements in SQL Server 2008, including message priorities and the SSBDiagnose diagnostic utility.
- Chapter 10 explores the Full-Text Search (FTS) feature in SQL Server 2008, which is more integrated into the database engine than in previous versions of SQL Server.

Part 3, Development in SQL Server: The eight chapters in this part cover topics important to developers, such as Transact-SQL (T-SQL) changes and LINQ to SQL.

- Chapter 11 introduces new datatypes. SQL Server 2008 comes with a bunch of new datatypes, including types for dates and times that are time-zone aware, hierarchical types, and spatial types. You'll also learn about the new filestream feature, which allows for large objects to be stored directly on the file system, while still having the transactional consistency of the database engine.

- Chapter 12 covers T-SQL for developers. T-SQL continues to be evolved in SQL Server 2008. Investments were made in new syntax, including the `MERGE` statement, which is an ISO/ANSI standard-specified statement that allows users to express multiple Data Manipulation Language (DML) actions (`INSERT`, `UPDATE`, and `DELETE`) against a specified target table based on join conditions with a source table. This and other T-SQL enhancements are discussed in depth in this chapter.
- Chapter 13 covers T-SQL for DBAs. Locking enhancements, filtered indexes, and table partitioning are among the many features that the database administrator should be aware of and utilize in SQL Server 2008.
- Chapter 14 discusses the role of .NET inside SQL Server. It also walks through programming, debugging, and deploying a common language runtime (CLR) stored procedure.
- Chapter 15 expands on the .NET discussion in the previous chapter and includes coverage of user-defined datatypes, functions (both scalar and table-valued), aggregates, and triggers.
- Chapter 16 provides an overview of the XML technology as it relates to SQL Server. It takes a broad look at XPath and XML Schema support in SQL Server 2008, and then drills down into how to get XML into and out of the database.
- Chapter 17 investigates native XML support in SQL Server 2008, via the XML datatype. You'll learn how to create XML columns, insert data into those columns, and then retrieve that XML data using XQuery.
- Chapter 18 covers Language Integrated Query (LINQ), a Microsoft .NET Framework component that adds native data-querying capabilities to .NET languages. This chapter explores the relationship between LINQ and SQL Server.

Part 4, Business Intelligence in SQL Server. The three chapters in this part discuss the tools and features that are the business intelligence offering of Microsoft.

- Chapter 19 covers Reporting Services, an extremely popular feature within the SQL Server product. Investments in the Reporting Services engine were made in SQL Server 2008, allowing it to handle massive amounts of reporting. This chapter covers the core concepts of Reporting Services, as well as the many enhancements to Reporting Services in SQL Server 2008.
- Chapter 20 focuses on Analysis Services. Databases store data, but they become truly profitable when the data can be used and interpreted to provide business intelligence. Powered by a robust Business Intelligence Development Studio (BIDS) environment, SQL Server Analysis Services is a major player in the business intelligence market. This chapter covers the advancements in Analysis Services in SQL Server 2008.
- Chapter 21 covers SQL Server Integration Services, Microsoft's Extract, Transform, and Load (ETL) tool. This chapter guides you through all of the Integration Services concepts, including data flow, control flow, and transformation tasks, using plenty of examples. You'll learn about the new Integration Services tasks, including an enhanced lookup operator that will support more flexible levels of caching. There is also new profiling data quality functionality, which will provide advanced algorithms for identifying patterns within data values.

Errata

Apress makes every effort to make sure that there are no errors in the text or code. However, mistakes happen, and we recognize the need to keep you informed of any mistakes as they're discovered and corrected. An errata sheet will be made available on the book's main page at `http://www.apress.com`. If you find an error that hasn't already been reported, please let us know.

Contacting the Authors

You can contact the book's lead author, Rob Walters, at `Robert.Walters@Microsoft.com`.

Robert E. Walters

PART 1

Overview of SQL Server

CHAPTER 1

SQL Server 2008 Overview

The previous release of SQL Server, SQL Server 2005, was a major release. It contained a ton of new functionality, including the revision of major query-processing components within the database engine. With SQL Server 2008, the development cycle was much shorter, and the changes are not quite as dramatic. Even though the overall quantity of changes is less than those in SQL Server 2005, the improvements that were made in this new release are specific and significant, and they will absolutely be of value to your organization.

In recent years, one of the trends has been the explosion of data. This massive increase in the quantity of data can be attributed to changes in behavior by consumers and businesses. For consumers, we need to look no further than digital cameras for a great example. With digital cameras, it is now possible to take a thousand different perspectives of your favorite landmark without worrying about film or development costs. All of these digital photos take up cheap disk space and need to be managed for easy searching. As another example, consider the amount of data generated by the medical industry in its daily operations. X-rays are being created and stored in a digital format for easier portability and quicker viewing time (doctors don't need to wait for someone to fetch a patient record from the vault).

The increase of data in recent years comes with a price, and it's not the cost of hardware, as some may think. Although the hardware industry has done a great job of providing low-dollar-per-gigabyte ratios year after year, the increase in digitally born data has sweetened the deal for potential data thieves. This increased security risk, combined with the influx of regulatory compliance laws, such as the Sarbanes-Oxley Act of 2002, has influenced the security features within SQL Server. Security design has been at the core of every feature implementation in SQL Server, and the SQL Server 2008 release is no different. Most of the security-specific improvements, such as auditing and transparent data encryption, are discussed in Chapters 6 and 7 of this book.

The Vision of SQL Server 2008

SQL Server 2008 is driven by a vision consisting of four key pillars: enterprise data platform, beyond relational, dynamic development, and pervasive insight. Although some might think of these catchy names as merely marketing hype, they actually meant something to the product development team.

Microsoft changed the way SQL Server is designed and developed. Specific user scenarios were derived from these core themes, and in the end, actual improvements to the product were made relating to those scenarios. The result is a high-quality release that focuses on the key pain points and industry trends. Here, we'll take a look at each of the pillars and discuss some key features within them. These features, as well as many others, are discussed in more detail in the subsequent chapters of this book.

Enterprise Data Platform

To be classified as an enterprise-ready database means much more than being able to formulate query results really fast. An enterprise database must meet the strict service level agreements established by the organizations using SQL Server. SQL Server 2008 has made improvements in supporting high service level agreements, like the ability to hot-add CPUs. Administrators will also find installing and managing the setup of cluster nodes to be much easier. More information about SQL Server setup and configuration can be found in Chapter 2.

Being an enterprise data platform also means the data that is stored inside the database is secure. SQL Server 2008 continually builds upon its security features. For example, it enables database files to be automatically encrypted with transparent data encryption. The importance of encryption is prominent with Microsoft, opening the door for Hardware Security Module (HSM) and Enterprise Key Management (EKM) vendors to integrate natively with the encryption support in SQL Server 2008. Encrypting data with SQL Server and storing the encrypted keys within the database provide security, but a more secure solution is to store the encryption keys separately from the actual data, and that is where HSM and EKM solutions add value.

Performance has always been a key attribute of an enterprise-ready database. The Data Collector is a feature within SQL Server that gives database administrators (DBAs) the ability to collect performance-related data and store it within a database. This data can be practically anything, such as Performance Monitor counters, results from database management views, and specific queries. Having performance data reside in a database allows for easy data mining and reporting, and that is the key benefit of the Data Collector. Details on this feature are discussed in Chapter 5.

SQL Server 2008 has many new features that validate SQL Server as an enterprise data platform. These features are discussed in detail throughout Part 2 of this book.

Beyond Relational

Data managed within a data platform is more than just relational data. As data growth increases, the types of data stored are no longer the traditional integer, character, and binary values we know and love. New data structures that are important to users are movie files, audio files, and medical images, to name a few. And we not only need to store these new types of data, but we also want to be able to perform useful operations on them, such as indexing and metadata searches.

With SQL Server 2008, investments were made in a feature called *filestream*, which allows files of arbitrary size to be stored in the file system and managed from the database. This capability enables database applications to exceed the 2GB limit. We can essentially place volume-sized binaries directly into SQL Server and obtain the same seek performance we would if we were querying the file system directly, instead of through Transact-SQL (T-SQL). Filestream is discussed in detail in Chapter 11.

SQL Server 2008 also includes support for spatial datatypes. The support conforms to the OpenGIS standards and allows for easy management of global positioning system (GPS) and geographic information system (GIS) data. Having native spatial support also makes it easy and fun to work with spatial applications like Microsoft's Virtual Earth. Spatial data support is also discussed in Chapter 11.

Dynamic Development

Without developer support, platform products such as SQL Server would have died off a long time ago. Microsoft has always made developer productivity a high priority in all of its products. SQL Server 2008 is no exception.

SQL Server 2008 contains improvements in the T-SQL language (discussed in Chapter 12), as well as new date and time datatypes (discussed in Chapter 11) to fill the void that the existing ones created. SQL Server has also integrated itself with the LINQ effort within Microsoft. LINQ provides

a higher level of data abstraction, making it really easy to code against disparate data sources. LINQ, as it's related to SQL Server, is discussed in detail in Chapter 18.

Pervasive Insight

Since the inception of Online Analytical Processing (OLAP) services in SQL Server 7.0, Microsoft has continually strived for a self-service business intelligence model. The idea is to allow the average employee to easily ask a business intelligence question and get the results, without needing to go through various layers of DBAs and report developers. Gradually, throughout the releases of SQL Server, we have seen more tools and features that promote this behavior.

At the core of business intelligence is SQL Server Analysis Services. New in Analysis Services are enhanced cube, dimension, and attribute designers. These designers, as well as core improvements related to the monitoring, analysis, and performance tuning of Analysis Services, continually push Microsoft's business intelligence engine further into the leader category of various industry analysts' charts. Analysis Services is discussed in Chapter 20.

In order for Analysis Services to effectively mine data, it needs a great Extract, Transform, and Load (ETL) tool. Investments with SQL Server Integration Services (the replacement for Data Transformation Services in SQL Server 2000) have continued, with the addition of capabilities such as caching transformations, enhanced lookup transformations, data profiling, and a set of expanded data sources. These exciting enhancements are discussed in Chapter 21.

Over the past few years, Microsoft has acquired a few companies in the reporting market. Some of these acquisitions, like Dundas and its graphical reporting controls, have shown up in Reporting Services in SQL Server 2008. The Reporting Services engine has also been upgraded to release its dependency on Internet Information Server (IIS), among other well-anticipated features. Chapter 19 covers Reporting Services in SQL Server 2008.

SQL Server 2008 Editions

At the time of this writing, the SQL Server 2008 editions are essentially the same as those that were available for SQL Server 2005. Five main SQL Server 2008 editions are available:

Enterprise Edition: This is primarily used for business-critical, large-scale online transaction processing (OLTP), large-scale reporting, data warehousing, and server consolidation requirements. Enterprise Edition comes with more than 60 features that are not found in Standard Edition. Some of these features are significant enough to entice those who have always said that Standard Edition was good enough. Features found only in Enterprise Edition are data and backup compression, audits that use extended events, and Resource Governor, to name a few. The gap of features between Standard Edition and Enterprise Edition is far greater in SQL Server 2008 than it was in SQL Server 2005.

Standard Edition: This edition is primarily used for departmental applications and small to medium-sized OLTP loads. Standard Edition comes with most of the powerful reporting capabilities found in SQL Server Reporting Services and makes a great reporting and analytics server as well.

Workgroup Edition: This edition includes the basic SQL Server relational database capabilities, as well as some of the replication technologies. This makes Workgroup Edition good for running branch office applications and performing remote synchronization with other geographically separated servers. Workgroup Edition is considerably less expensive than Standard Edition.