

**Microsoft**

Foreword by Donald Farmer  
*Principal Program Manager, US-SQL Analysis Services  
Microsoft Corporation*

# Smart Business Intelligence Solutions with Microsoft® SQL Server® 2008



**Lynn Langit**

Kevin S. Goff, Davide Mauri, Sahil Malik, and John Welch

**PUBLISHED BY**

Microsoft Press  
A Division of Microsoft Corporation  
One Microsoft Way  
Redmond, Washington 98052-6399

Copyright © 2009 by Kevin Goff and Lynn Langit

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Control Number: 2008940532

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 QWT 4 3 2 1 0 9

Distributed in Canada by H.B. Fenn and Company Ltd.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at [www.microsoft.com/mspress](http://www.microsoft.com/mspress). Send comments to [mspinput@microsoft.com](mailto:mspinput@microsoft.com).

Microsoft, Microsoft Press, Access, Active Directory, ActiveX, BizTalk, Excel, Hyper-V, IntelliSense, Microsoft Dynamics, MS, MSDN, PerformancePoint, PivotChart, PivotTable, PowerPoint, ProClarity, SharePoint, Silverlight, SQL Server, Visio, Visual Basic, Visual C#, Visual SourceSafe, Visual Studio, Win32, Windows, Windows PowerShell, Windows Server, and Windows Vista are either registered trademarks or trademarks of the Microsoft group of companies. Other product and company names mentioned herein may be the trademarks of their respective owners.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

**Acquisitions Editor:** Ken Jones

**Developmental Editor:** Sally Stickney

**Project Editor:** Maureen Zimmerman

**Editorial Production:** Publishing.com

**Technical Reviewer:** John Welch; Technical Review services provided by Content Master, a member of CM Group, Ltd.

**Cover:** Tom Draper Design

*For Mahnaz Javid and for her work  
with the Mona Foundation, which she leads*

*—Lynn Langit, author*



# Contents at a Glance

|          |  |     |
|----------|--|-----|
| Part I   | <b>Business Intelligence for Business Decision Makers and Architects</b>   |     |
| 1        | Business Intelligence Basics . . . . .   | 3   |
| 2        | Visualizing Business Intelligence Results . . . . .  | 27  |
| 3        | Building Effective Business Intelligence Processes . . . . .   | 61  |
| 4        | Physical Architecture in Business Intelligence Solutions . . . . .   | 85  |
| 5        | Logical OLAP Design Concepts for Architects . . . . .  | 115 |
| Part II  | <b>Microsoft SQL Server 2008 Analysis Services for Developers</b>  |     |
| 6        | Understanding SSAS in SSMS and SQL Server Profiler . . . . .   | 153 |
| 7        | Designing OLAP Cubes Using BIDS . . . . .  | 183 |
| 8        | Refining Cubes and Dimensions . . . . .  | 225 |
| 9        | Processing Cubes and Dimensions . . . . .  | 257 |
| 10       | Introduction to MDX . . . . .  | 293 |
| 11       | Advanced MDX . . . . .   | 329 |
| 12       | Understanding Data Mining Structures . . . . .   | 355 |
| 13       | Implementing Data Mining Structures . . . . .  | 399 |
| Part III | <b>Microsoft SQL Server 2008 Integration Services for Developers</b>   |     |
| 14       | Architectural Components of Microsoft SQL Server 2008 Integration Services . . . . .                                     | 435 |
| 15       | Creating Microsoft SQL Server 2008 Integration Services Packages with Business Intelligence Development Studio . . . . . | 463 |
| 16       | Advanced Features in Microsoft SQL Server 2008 Integration Services . . . . .  | 497 |
| 17       | Microsoft SQL Server 2008 Integration Services Packages in Business Intelligence Solutions . . . . .                     | 515 |

|                |  |            |
|----------------|--|------------|
| <b>18</b>      | <b>Deploying and Managing Solutions in Microsoft SQL Server 2008 Integration Services</b> .....      | <b>539</b> |
| <b>19</b>      | <b>Extending and Integrating SQL Server 2008 Integration Services</b> .....                          | <b>567</b> |
| <b>Part IV</b> | <b>Microsoft SQL Server Reporting Services and Other Client Interfaces for Business Intelligence</b> |            |
| <b>20</b>      | <b>Creating Reports in SQL Server 2008 Reporting Services</b> .....                                  | <b>603</b> |
| <b>21</b>      | <b>Building Reports for SQL Server 2008 Reporting Services</b> .....                                 | <b>627</b> |
| <b>22</b>      | <b>Advanced SQL Server 2008 Reporting Services</b> .....   | <b>647</b> |
| <b>23</b>      | <b>Using Microsoft Excel 2007 as an OLAP Cube Client</b> .....                                       | <b>671</b> |
| <b>24</b>      | <b>Microsoft Office 2007 as a Data Mining Client</b> .....   | <b>687</b> |
| <b>25</b>      | <b>SQL Server Business Intelligence and Microsoft Office SharePoint Server 2007</b> .....            | <b>723</b> |

# Table of Contents

|                       |        |
|-----------------------|--------|
| Foreword .....        | .xix   |
| Acknowledgments ..... | .xxi   |
| Introduction .....    | .xxiii |

## Part I **Business Intelligence for Business Decision Makers and Architects**

|  |          |
|--|----------|
| <b>1 Business Intelligence Basics .....</b>          | <b>3</b> |
| Business Intelligence and Data Modeling .....        | 3        |
| OLTP and OLAP .....                                  | 5        |
| Online Transactional Processing .....                | 5        |
| Online Analytical Processing .....                   | 8        |
| Common BI Terminology .....                          | 11       |
| Data Warehouses .....                                | 11       |
| Data Marts .....                                     | 13       |
| Cubes .....  | 13       |
| Decision Support Systems .....                       | 13       |
| Data Mining Systems .....                            | 14       |
| Extract, Transform, and Load Systems .....           | 14       |
| Report Processing Systems .....                      | 15       |
| Key Performance Indicators .....                     | 15       |
| Core Components of a Microsoft BI Solution .....     | 16       |
| SQL Server 2008 Analysis Services .....              | 16       |
| SQL Server 2008 Reporting Services .....             | 19       |
| SQL Server 2008 .....                                | 19       |
| SQL Server 2008 Integration Services .....           | 20       |
| Optional Components of a Microsoft BI Solution ..... | 21       |
| Query Languages Used in BI Solutions .....           | 23       |
| MDX .....  | 23       |
| DMX .....  | 24       |

 **What do you think of this book? We want to hear from you!**

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

[www.microsoft.com/learning/booksurvey/](http://www.microsoft.com/learning/booksurvey/)

|          |   |           |
|----------|---|-----------|
|          | XMLA .....  | 24        |
|          | RDL .....   | 24        |
|          | Summary .....   | 25        |
| <b>2</b> | <b>Visualizing Business Intelligence Results .....</b>                | <b>27</b> |
|          | Matching Business Cases to BI Solutions .....                         | 27        |
|          | Top 10 BI Scoping Questions .....                                     | 30        |
|          | Components of BI Solutions .....                                      | 31        |
|          | Understanding Business Intelligence from a User's Perspective .....   | 34        |
|          | Demonstrating the Power of BI Using Excel 2007 .....                  | 36        |
|          | Understanding Data Mining via the Excel Add-ins .....                 | 45        |
|          | Viewing Data Mining Structures Using Excel 2007 .....                 | 47        |
|          | Elements of a Complete BI Solution .....                              | 50        |
|          | Reporting—Deciding Who Will Use the Solution .....                    | 51        |
|          | ETL—Getting the Solution Implemented .....                            | 52        |
|          | Data Mining—Don't Leave It Out .....                                  | 53        |
|          | Common Business Challenges and BI Solutions .....                     | 54        |
|          | Measuring the ROI of BI Solutions .....                               | 56        |
|          | Summary .....   | 58        |
| <b>3</b> | <b>Building Effective Business Intelligence Processes .....</b>       | <b>61</b> |
|          | Software Development Life Cycle for BI Projects .....                 | 61        |
|          | Microsoft Solutions Framework .....                                   | 62        |
|          | Microsoft Solutions Framework for Agile Software Development .....    | 63        |
|          | Applying MSF to BI Projects .....                                     | 65        |
|          | Phases and Deliverables in the Microsoft Solutions Framework .....    | 65        |
|          | Skills Necessary for BI Projects .....                                | 72        |
|          | Required Skills .....   | 72        |
|          | Optional Skills .....   | 74        |
|          | Forming Your Team .....   | 76        |
|          | Roles and Responsibilities Needed When Working with MSF .....         | 76        |
|          | Summary .....   | 84        |
| <b>4</b> | <b>Physical Architecture in Business Intelligence Solutions .....</b> | <b>85</b> |
|          | Planning for Physical Infrastructure Change .....                     | 85        |
|          | Creating Accurate Baseline Surveys .....                              | 85        |
|          | Assessing Current Service Level Agreements .....                      | 87        |
|          | Determining the Optimal Number and Placement of Servers .....         | 89        |
|          | Considerations for Physical Servers .....                             | 91        |

|   |            |
|---|------------|
| Considerations for Logical Servers and Services .....                     | 92         |
| Understanding Security Requirements .....                                 | 95         |
| Security Requirements for BI Solutions .....                              | 95         |
| Backup and Restore .....  | 106        |
| Backing Up SSAS .....   | 106        |
| Backing Up SSIS .....   | 107        |
| Backing Up SSRS .....   | 108        |
| Auditing and Compliance .....   | 108        |
| Auditing Features in SQL Server 2008 .....                                | 111        |
| Source Control .....  | 111        |
| Summary .....   | 114        |
| <b>5 Logical OLAP Design Concepts for Architects .....</b>                | <b>115</b> |
| Designing Basic OLAP Cubes .....  | 115        |
| Star Schemas .....  | 116        |
| Denormalization .....   | 125        |
| Back to the Star .....  | 125        |
| Other Design Tips .....   | 132        |
| Modeling Snowflake Dimensions .....                                       | 133        |
| More About Dimensional Modeling .....                                     | 138        |
| Understanding Fact (Measure) Modeling .....                               | 146        |
| Other Considerations in BI Modeling .....                                 | 148        |
| Summary .....   | 150        |
| <b>Part II Microsoft SQL Server 2008 Analysis Services for Developers</b> |            |
| <b>6 Understanding SSAS in SSMS and SQL Server Profiler .....</b>         | <b>153</b> |
| Core Tools in SQL Server Analysis Services .....                          | 153        |
| Baseline Service Configuration .....                                      | 157        |
| SSAS in SSMS .....  | 160        |
| How Do You Query SSAS Objects? .....                                      | 170        |
| Using MDX Templates .....   | 175        |
| Using DMX Templates .....   | 179        |
| Using XMLA Templates .....  | 180        |
| Closing Thoughts on SSMS .....  | 181        |
| Summary .....   | 182        |

|          |  |            |
|----------|--|------------|
| <b>7</b> | <b>Designing OLAP Cubes Using BIDS</b>                   | <b>183</b> |
|          | Using BIDS   | 183        |
|          | Offline and Online Modes                                 | 184        |
|          | Working in Solution Explorer                             | 186        |
|          | Data Sources in Analysis Services                        | 188        |
|          | Data Source Views  | 190        |
|          | Roles in Analysis Services                               | 195        |
|          | Using Compiled Assemblies with Analysis Services Objects | 196        |
|          | Building OLAP Cubes in BIDS                              | 198        |
|          | Examining the Sample Cube in Adventure Works             | 201        |
|          | Understanding Dimensions                                 | 204        |
|          | Attribute Hierarchies                                    | 206        |
|          | Attribute Relationships                                  | 207        |
|          | Translations   | 209        |
|          | Using Dimensions   | 210        |
|          | Measure Groups   | 211        |
|          | Beyond Star Dimensions                                   | 215        |
|          | Building Your First OLAP Cube                            | 218        |
|          | Selecting Measure Groups                                 | 219        |
|          | Adding Dimensions  | 220        |
|          | Summary  | 223        |
| <b>8</b> | <b>Refining Cubes and Dimensions</b>                     | <b>225</b> |
|          | Refining Your First OLAP Cube                            | 225        |
|          | Translations and Perspectives                            | 225        |
|          | Key Performance Indicators                               | 228        |
|          | Actions  | 233        |
|          | Calculations (MDX Scripts or Calculated Members)         | 239        |
|          | Using Cube and Dimension Properties                      | 243        |
|          | Time Intelligence  | 245        |
|          | SCOPE Keyword  | 246        |
|          | Account Intelligence and Unary Operator Definitions      | 246        |
|          | Other Wizard Options                                     | 250        |
|          | Currency Conversions                                     | 251        |
|          | Advanced Cube and Dimension Properties                   | 254        |
|          | Summary  | 255        |

|           |   |            |
|-----------|---|------------|
| <b>9</b>  | <b>Processing Cubes and Dimensions</b>                    | <b>257</b> |
|           | Building, Processing, and Deploying OLAP Cubes            | 257        |
|           | Differentiating Data and Metadata                         | 258        |
|           | Working in a Disconnected Environment                     | 259        |
|           | Working in a Connected Environment                        | 261        |
|           | Understanding Aggregations                                | 261        |
|           | Partitioning  | 263        |
|           | Choosing Storage Modes: MOLAP, HOLAP, and ROLAP           | 267        |
|           | OLTP Table Partitioning                                   | 268        |
|           | Other OLAP Partition Configurations                       | 270        |
|           | Implementing Aggregations                                 | 270        |
|           | Aggregation Design Wizard                                 | 271        |
|           | Usage-Based Optimization Wizard                           | 274        |
|           | SQL Server Profiler                                       | 275        |
|           | Aggregation Designer: Advanced View                       | 277        |
|           | Implementing Advanced Storage with MOLAP, HOLAP, or ROLAP | 278        |
|           | Proactive Caching   | 279        |
|           | Notification Settings for Proactive Caching               | 282        |
|           | Fine-Tuning Proactive Caching                             | 283        |
|           | ROLAP Dimensions  | 284        |
|           | Linking   | 285        |
|           | Writeback   | 285        |
|           | Cube and Dimension Processing Options                     | 287        |
|           | Summary   | 292        |
| <b>10</b> | <b>Introduction to MDX</b>                                | <b>293</b> |
|           | The Importance of MDX                                     | 293        |
|           | Writing Your First MDX Queries                            | 295        |
|           | MDX Object Names  | 296        |
|           | Other Elements of MDX Syntax                              | 296        |
|           | MDX Core Functions  | 299        |
|           | Filtering MDX Result Sets                                 | 306        |
|           | Calculated Members and Named Sets                         | 307        |
|           | Creating Objects by Using Scripts                         | 309        |
|           | The <i>TopCount</i> Function                              | 310        |
|           | <i>Rank</i> Function and Combinations                     | 312        |
|           | <i>Head</i> and <i>Tail</i> Functions                     | 315        |
|           | Hierarchical Functions in MDX                             | 316        |

|   |            |
|---|------------|
| Date Functions .....                                  | 321        |
| Using Aggregation with Date Functions .....           | 324        |
| About Query Optimization .....                        | 326        |
| Summary .....   | 327        |
| <b>11 Advanced MDX .....</b>                          | <b>329</b> |
| Querying Dimension Properties .....                   | 329        |
| Looking at Date Dimensions and MDX Seasonality .....  | 332        |
| Creating Permanent Calculated Members .....           | 333        |
| Creating Permanent Calculated Members in BIDS .....   | 334        |
| Creating Calculated Members Using MDX Scripts .....   | 335        |
| Using <i>If</i> .....                                 | 337        |
| About Named Sets .....                                | 338        |
| About Scripts .....                                   | 341        |
| Understanding SOLVE_ORDER .....                       | 343        |
| Creating Key Performance Indicators .....             | 345        |
| Creating KPIs Programmatically .....                  | 348        |
| Additional Tips on KPIs .....                         | 348        |
| Using MDX with SSRS and PerformancePoint Server ..... | 349        |
| Using MDX with SSRS 2008 .....                        | 349        |
| Using MDX with PerformancePoint Server 2007 .....     | 352        |
| Summary .....   | 354        |
| <b>12 Understanding Data Mining Structures .....</b>  | <b>355</b> |
| Reviewing Business Scenarios .....                    | 355        |
| Categories of Data Mining Algorithms .....            | 358        |
| Working in the BIDS Data Mining Interface .....       | 360        |
| Understanding Data Types and Content Types .....      | 361        |
| Setting Advanced Data Properties .....                | 363        |
| Choosing a Data Mining Model .....                    | 365        |
| Picking the Best Mining Model Viewer .....            | 368        |
| Mining Accuracy Charts and Prediction .....           | 373        |
| Data Mining Algorithms .....                          | 376        |
| Microsoft Naïve Bayes .....                           | 376        |
| Microsoft Decision Trees Algorithm .....              | 381        |
| Microsoft Linear Regression Algorithm .....           | 383        |
| Microsoft Time Series Algorithm .....                 | 383        |
| Microsoft Clustering Algorithm .....                  | 386        |
| Microsoft Sequence Clustering .....                   | 389        |
| Microsoft Association Algorithm .....                 | 391        |

|  |     |
|--|-----|
| Microsoft Neural Network Algorithm ..... | 394 |
| Microsoft Logistic Regression .....      | 395 |
| The Art of Data Mining .....             | 396 |
| Summary .....                            | 396 |

## **13 Implementing Data Mining Structures ..... 399**

|  |     |
|--|-----|
| Implementing the CRISP-DM Life Cycle Model ..... | 399 |
| Building Data Mining Structures Using BIDS ..... | 401 |
| Adding Data Mining Models Using BIDS .....       | 404 |
| Processing Mining Models .....                   | 407 |
| Validating Mining Models .....                   | 409 |
| Lift Charts .....                                | 410 |
| Profit Charts .....                              | 413 |
| Classification Matrix .....                      | 415 |
| Cross Validation .....                           | 417 |
| Data Mining Prediction Queries .....             | 419 |
| DMX Prediction Queries .....                     | 421 |
| DMX Prediction Functions .....                   | 423 |
| Data Mining and Integration Services .....       | 426 |
| Data Mining Object Processing .....              | 429 |
| Data Mining Clients .....                        | 431 |
| Summary .....                                    | 432 |

## **Part III Microsoft SQL Server 2008 Integration Services for Developers**

### **14 Architectural Components of Microsoft SQL Server 2008 Integration Services ..... 435**

|  |     |
|--|-----|
| Overview of Integration Services Architecture .....  | 436 |
| Integration Services Packages .....  | 438 |
| Tools and Utilities for Developing, Deploying, and Executing Integration Services Packages ..... | 438 |
| The Integration Services Object Model and Components .....                                       | 442 |
| Control Flow .....   | 442 |
| Data Flow .....  | 444 |
| Variables .....  | 445 |
| Expressions .....  | 447 |
| Connection Managers .....  | 448 |
| Event Handlers and Error Handling .....  | 450 |

|  |            |
|--|------------|
| The Integration Services Runtime . . . . .   | 452        |
| The Integration Services Data Flow Engine . . . . .  | 453        |
| Data Flow Buffers. . . . .   | 454        |
| Synchronous Data Flow Outputs. . . . .   | 458        |
| Asynchronous Data Flow Outputs . . . . .   | 459        |
| Log Providers . . . . .  | 459        |
| Deploying Integration Services Packages . . . . .  | 460        |
| Package Configurations . . . . .   | 461        |
| Package Deployment Options. . . . .  | 461        |
| Summary . . . . .  | 462        |
| <b>15 Creating Microsoft SQL Server 2008 Integration Services Packages with Business Intelligence Development Studio . . . .</b> | <b>463</b> |
| Integration Services in Visual Studio 2008. . . . .  | 463        |
| Creating New SSIS Projects with the Integration Services Project Template . . . . .  | 464        |
| Viewing an SSIS Project in Solution Explorer . . . . .   | 466        |
| Using the SSIS Package Designers . . . . .   | 467        |
| Working with the SSIS Toolbox . . . . .  | 469        |
| Choosing from the SSIS Menu. . . . .   | 472        |
| Connection Managers. . . . .   | 473        |
| Standard Database Connection Managers . . . . .  | 473        |
| Other Types of Connection Managers. . . . .  | 474        |
| Control Flow . . . . .   | 474        |
| Control Flow Tasks . . . . .   | 476        |
| Control Flow Containers. . . . .   | 478        |
| Precedence Constraints . . . . .   | 480        |
| Data Flow. . . . .   | 482        |
| Data Flow Source Components. . . . .   | 483        |
| Destination Components . . . . .   | 485        |
| Transformation Components. . . . .   | 486        |
| Integration Services Data Viewers . . . . .  | 488        |
| Variables. . . . .   | 490        |
| Variables Window . . . . .   | 490        |
| Variable Properties . . . . .  | 491        |
| System Variables. . . . .  | 493        |
| Expressions . . . . .  | 493        |
| Variables and Default Values Within a Package . . . . .  | 494        |
| Summary . . . . .  | 495        |

|           |  |            |
|-----------|--|------------|
| <b>16</b> | <b>Advanced Features in Microsoft SQL Server 2008</b>                |            |
|           | <b>Integration Services</b> . . . . .                                | <b>497</b> |
|           | Error Handling in Integration Services . . . . .                     | 497        |
|           | Events, Logs, Debugging, and Transactions in SSIS . . . . .          | 499        |
|           | Logging and Events . . . . .   | 501        |
|           | Debugging Integration Services Packages . . . . .                    | 505        |
|           | Checkpoints and Transactions . . . . .                               | 506        |
|           | Configuring Package Transactions . . . . .                           | 507        |
|           | Best Practices for Designing Integration Services Packages . . . . . | 509        |
|           | Data Profiling . . . . .   | 510        |
|           | Summary . . . . .  | 514        |
| <b>17</b> | <b>Microsoft SQL Server 2008 Integration Services Packages</b>       |            |
|           | <b>in Business Intelligence Solutions.</b> . . . . .                 | <b>515</b> |
|           | ETL for Business Intelligence . . . . .                              | 515        |
|           | Loading OLAP Cubes. . . . .  | 516        |
|           | Using Integration Services to Check Data Quality . . . . .           | 516        |
|           | Transforming Source Data . . . . .                                   | 519        |
|           | Using a Staging Server . . . . .                                     | 520        |
|           | Data Lineage. . . . .  | 524        |
|           | Moving to Star Schema Loading . . . . .                              | 525        |
|           | Loading Dimension Tables . . . . .                                   | 525        |
|           | Loading Fact Tables. . . . .   | 527        |
|           | Updates . . . . .  | 530        |
|           | Fact Table Updates. . . . .  | 532        |
|           | Dimension Table Updates. . . . .                                     | 532        |
|           | ETL for Data Mining. . . . .   | 533        |
|           | Initial Loading. . . . .   | 533        |
|           | Model Training . . . . .   | 534        |
|           | Data Mining Queries . . . . .  | 535        |
|           | Summary . . . . .  | 538        |
| <b>18</b> | <b>Deploying and Managing Solutions in Microsoft SQL Server</b>      |            |
|           | <b>2008 Integration Services</b> . . . . .                           | <b>539</b> |
|           | Solution and Project Structures in Integration Services . . . . .    | 539        |
|           | Source Code Control . . . . .  | 540        |
|           | Using Visual SourceSafe . . . . .                                    | 541        |
|           | The Deployment Challenge . . . . .                                   | 546        |
|           | Package Configurations . . . . .                                     | 548        |

|   |            |
|---|------------|
| Copy File Deployment . . . . .  | 552        |
| BIDS Deployment . . . . .   | 553        |
| Deployment with the Deployment Utility . . . . .                          | 556        |
| SQL Server Agent and Integration Services . . . . .                       | 558        |
| Introduction to SSIS Package Security . . . . .                           | 559        |
| Handling Sensitive Data and Proxy Execution Accounts . . . . .            | 563        |
| Security: The Two Rules . . . . .   | 564        |
| The SSIS Service . . . . .  | 564        |
| Summary . . . . .   | 565        |
| <br>  |            |
| <b>19 Extending and Integrating SQL Server 2008</b>                       |            |
| <b>Integration Services . . . . .</b>                                     | <b>567</b> |
| Introduction to SSIS Scripting . . . . .                                  | 567        |
| Visual Studio Tools for Applications . . . . .                            | 568        |
| The Script Task . . . . .   | 568        |
| The <i>Dts</i> Object . . . . .   | 571        |
| Debugging Script Tasks . . . . .  | 572        |
| The Script Component . . . . .  | 573        |
| The <i>ComponentMetaData</i> Property . . . . .                           | 580        |
| Source, Transformation, and Destination . . . . .                         | 582        |
| Debugging Script Components . . . . .                                     | 587        |
| Overview of Custom SSIS Task and Component Development . . . . .          | 587        |
| Control Flow Tasks . . . . .  | 591        |
| Data Flow Components . . . . .  | 593        |
| Other Components . . . . .  | 594        |
| Overview of SSIS Integration in Custom Applications . . . . .             | 596        |
| Summary . . . . .   | 600        |
| <br>  |            |
| <b>Part IV Microsoft SQL Server Reporting Services and Other</b>          |            |
| <b>Client Interfaces for Business Intelligence</b>                        |            |
| <br>  |            |
| <b>20 Creating Reports in SQL Server 2008 Reporting Services. . . . .</b> | <b>603</b> |
| Understanding the Architecture of Reporting Services . . . . .            | 603        |
| Installing and Configuring Reporting Services . . . . .                   | 606        |
| HTTP Listener . . . . .   | 608        |
| Report Manager . . . . .  | 609        |
| Report Server Web Service . . . . .                                       | 609        |
| Authentication . . . . .  | 610        |
| Background Processing (Job Manager) . . . . .                             | 612        |

|   |            |
|---|------------|
| Creating Reports with BIDS .....  | 612        |
| Other Types of Reports.....   | 621        |
| Sample Reports .....  | 622        |
| Deploying Reports.....  | 623        |
| Summary .....   | 625        |
| <b>21 Building Reports for SQL Server 2008 Reporting Services . . . .</b> | <b>627</b> |
| Using the Query Designers for Analysis Services .....                     | 627        |
| MDX Query Designer .....  | 628        |
| Setting Parameters in Your Query .....                                    | 631        |
| DMX Query Designer .....  | 633        |
| Working with the Report Designer in BIDS .....                            | 635        |
| Understanding Report Items .....  | 638        |
| List and Rectangle Report Items .....                                     | 639        |
| Tablix Data Region.....   | 639        |
| Using Report Builder .....  | 643        |
| Summary .....   | 646        |
| <b>22 Advanced SQL Server 2008 Reporting Services .....</b>               | <b>647</b> |
| Adding Custom Code to SSRS Reports.....                                   | 647        |
| Viewing Reports in Word or Excel 2007.....                                | 649        |
| URL Access.....   | 651        |
| Embedding Custom ReportViewer Controls .....                              | 652        |
| About Report Parameters.....  | 656        |
| About Security Credentials.....   | 657        |
| About the SOAP API .....  | 658        |
| What Happened to Report Models?.....                                      | 660        |
| Deployment—Scalability and Security.....                                  | 662        |
| Performance and Scalability .....   | 663        |
| Advanced Memory Management.....   | 665        |
| Scaling Out .....   | 666        |
| Administrative Scripting .....  | 667        |
| Using WMI.....  | 668        |
| Summary .....   | 669        |
| <b>23 Using Microsoft Excel 2007 as an OLAP Cube Client .....</b>         | <b>671</b> |
| Using the Data Connection Wizard .....                                    | 671        |
| Working with the Import Data Dialog Box .....                             | 674        |
| Understanding the PivotTable Interface.....                               | 675        |
| Creating a Sample PivotTable.....   | 678        |

|  |            |
|--|------------|
| Offline OLAP .....   | 681        |
| Excel OLAP Functions .....   | 683        |
| Extending Excel .....  | 683        |
| Summary .....  | 685        |
| <b>24 Microsoft Office 2007 as a Data Mining Client .....</b>                                    | <b>687</b> |
| Installing Data Mining Add-ins .....   | 687        |
| Data Mining Integration with Excel 2007 .....  | 689        |
| Using the Table Analysis Tools Group .....   | 690        |
| Using the Data Mining Tab in Excel 2007 .....  | 700        |
| Data Mining Integration in Visio 2007 .....  | 714        |
| Client Visualization .....   | 718        |
| Data Mining in the Cloud .....   | 720        |
| Summary .....  | 721        |
| <b>25 SQL Server Business Intelligence and Microsoft Office<br/>SharePoint Server 2007 .....</b> | <b>723</b> |
| Excel Services .....   | 723        |
| Basic Architecture of Excel Services .....   | 724        |
| Immutability of Excel Sheets .....   | 726        |
| Introductory Sample Excel Services Worksheet .....   | 726        |
| Publishing Parameterized Excel Sheets .....  | 729        |
| Excel Services: The Web Services API .....   | 732        |
| A Real-World Excel Services Example .....  | 733        |
| SQL Server Reporting Services with Office SharePoint Server 2007 .....                           | 736        |
| Configuring SQL Server Reporting Services<br>with Office SharePoint Server 2007 .....            | 737        |
| Authoring and Deploying a Report .....   | 738        |
| Using the Report in Office SharePoint Server 2007: Native Mode .....                             | 740        |
| Using the Report in Office SharePoint Server 2007:<br>SharePoint Integrated Mode .....           | 742        |
| Using the Report Center Templates .....  | 744        |
| PerformancePoint Server .....  | 745        |
| Summary .....  | 745        |
| Index .....  | 747        |

**What do you think of this book? We want to hear from you!**

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

[www.microsoft.com/learning/booksurvey/](http://www.microsoft.com/learning/booksurvey/)

# Foreword

When Lynn Langit's name appears in my inbox or RSS feeds, I never know what to expect—only that it will be interesting! She may be inviting me to share a technical webcast, passing pithy comments about a conference speaker, or recalling the sight of swimming elephants in Zambia, where Lynn tirelessly promotes information technology as a force for improving health care. On this occasion, it was an invitation to write a foreword for this, her latest book, *Smart Business Intelligence Solutions with Microsoft SQL Server 2008*. As so often, when Lynn asks, the only possible response is, "Of course—I'd be happy to!"

When it comes to business intelligence, Lynn is a compulsive communicator. As a Developer Evangelist at Microsoft, this is part of her job, but Lynn's enthusiasm for the technologies and their implications goes way beyond that. Her commitment is clear in her presentations and webcasts, in her personal engagements with customers across continents, and in her writing. Thinking of this, I am more than pleased to see this new book, especially to see that it tackles the SQL Server business intelligence (BI) technologies in their broad scope.

Business intelligence is never about one technology solving one problem. In fact, a good BI solution can address many problems at many levels—tactical, strategic, and even operational. Part I, "Business Intelligence for Business Decision Makers and Architects," explores these business scenarios.

To solve these problems, you will find that your raw data is rarely sufficient. The BI developer must apply business logic to enrich the data with analytical insights for business users. Without this additional business logic, your system may only tell the users what they already know. Part II, "Microsoft SQL Server 2008 Analysis Services for Developers," takes a deep look at using Analysis Services to create OLAP cubes and data mining models.

By their nature, these problems often require you to integrate data from across your business. SQL Server 2008 Integration Services is the platform for this work, and in Part III, "Microsoft SQL Server 2008 Integration Services for Developers," Lynn tackles this technology. She not only covers the details of building single workloads, but also sets this work in its important architectural context, covering management and deployment of the integration solutions.

Finally, in Part IV, "Microsoft SQL Server Reporting Services and Other Client Interfaces for Business Intelligence," there is a detailed exploration of the options for designing and publishing reports. This section also covers other popular "clients"—the applications through which business users interact with your BI solution. So, even if you are a Microsoft Office Excel user, there is valuable information here.

When all of these elements—integration, analysis, and reporting—come together, you know you are implementing a "smart solution," the essence of this most helpful book.

I know from my own work at Microsoft, presenting and writing about BI, how difficult it is to find good symmetry between technology and the business case. I also know how important it is. Architects may build smart technology solutions, but enterprise decision makers put the *business* into *BI*. For these readers, Lynn makes very few assumptions. She quickly, yet quite thoroughly, takes the reader through a basic taxonomy of the moving parts of a BI solution.

However, this book is more than a basic introduction—it gets down to the details you need to build effective solutions. Even experienced users will find useful insights and information here. For example, all OLAP developers work with Analysis Services data source views. However, many of them do not even know about the useful data preview feature. In Chapter 7, “Designing OLAP Cubes Using BIDS,” Lynn not only describes the feature, but also includes a good example of its use for simple validation and profiling. It is, for me, a good measure of a book that it finds new things to say even about the most familiar features.

For scenarios that may be less familiar to you, such as data mining, Lynn carefully sets out the business cases, the practical steps to take, and the traps to avoid. Having spent many hours teaching and evangelizing about data mining myself, I really admire how Lynn navigates through the subject. In one chapter, she starts from the highest level (“Why would I use data mining?”) to the most detailed (“What is the CLUSTERING\_METHOD parameter for?”), retaining a pleasant and easy logical flow.

It is a privilege to work at Microsoft with Lynn. She clearly loves working with her customers and the community. This book captures much of her enthusiasm and knowledge in print. You will enjoy it, and I will not be surprised if you keep it close at hand on your desk whenever you work with SQL Server 2008.

*Donald Farmer*  
*Principal Program Manager, US-SQL Analysis Services*  
*Microsoft Corporation*

# Acknowledgments

Many people contributed to making this book. The authors would like to acknowledge those people and the people who support them.

## Lynn Langit

Thanks to all those who supported my efforts on this book.

First I'd like to thank my inspiration and the one who kept me going during the many months of writing this book—Mahnaz Javid—and the work of her Mona Foundation. Please prioritize caring for the world's needy children and take the time to contribute to organizations that do a good job with this important work. A portion of the proceeds of this book will be donated to the Mona Foundation. For more information, go to <http://www.monafoundation.org>.

Thanks to my colleagues at Microsoft Press: Ken Jones, Sally Stickney, Maureen Zimmerman; to my Microsoft colleagues: Woody Pewitt, Glen Gordon, Mithun Dhar, Bruno Terkaly, Joey Snow, Greg Visscher, and Scott Kerfoot; and to the SQL Team: Donald Farmer, Francois Ajenstadt, and Zack Owens.

Thanks to my co-writers and community reviewers: Davide Mauri, Sahil Malik, Kevin Goff, Kim Schmidt, Mathew Roche, Ted Malone, and Karen Henderson.

Thanks especially to my technical reviewer, John Welch. John, I wish I hadn't made you work so hard!

Thanks to my friends and family for understanding the demands that writing makes on my time and sanity: Lynn C, Teri, Chrys, Esther, Asli, Anton, and, most especially, to my mom and my daughter.

## Davide Mauri

A very big thanks to my wife Olga, who always supports me in everything I do; and to Gianluca, Andrea, and Fernando, who allowed me to realize one of my many dreams!

## Sahil Malik

Instead of an acknowledgment, I'd like to pray for peace, harmony, wisdom, and inner happiness for everyone.



# Introduction

So, why write? What is it that makes typing in a cramped airline seat on an 11-hour flight over Africa so desirable? It's probably because of a love of reading in general, and of learning in particular. It's not by chance that my professional blog at <http://blogs.msdn.com/SoCalDevGal> is titled "Contagious Curiosity." To understand why we wrote this particular book, you must start with a look at the current landscape of business intelligence (BI) using Microsoft SQL Server 2008.

Business intelligence itself really isn't new. Business intelligence—or data warehousing, as it has been traditionally called—has been used in particular industries, such as banking and retailing, for many years. What is new is the accessibility of BI solutions to a broader audience. Microsoft is leading this widening of the BI market by providing a set of world-class tools with SQL Server 2008. SQL Server 2008 includes the fourth generation of these tools in the box (for no additional fee) and their capabilities are truly impressive. As customers learn about the possibilities of BI, we see ever-greater critical mass adoption. We believe that within the next few years, it will be standard practice to implement both OLTP and (BI) OLAP/data mining solutions for nearly every installation of SQL Server 2008.

One of the most significant hindrances to previous adoption of BI technologies has not been the quality of technologies and tools available in SQL Server 2008 or its predecessors. Rather, what we have found (from our real-world experience) is that a general lack of understanding of BI capabilities is preventing wider adoption. We find that developers, in particular, lack understanding of BI core concepts such as OLAP (or dimensional) modeling and data mining algorithms. This knowledge gap also includes lack of understanding about the capabilities of the BI components and tools included with SQL Server 2008—SQL Server Analysis Services (SSAS), SQL Server Integration Services (SSIS), and SQL Server Reporting Services (SSRS).

The gap is so significant, in fact, that it was one of the primary motivators for writing this book. Far too many times, we've seen customers who lack understanding of core BI concepts struggle to create BI solutions. Ironically, the BI tools included in SQL Server 2008 are in some ways too easy to use. As with many Microsoft products, a right-click in the right place nearly always starts a convenient wizard. So customers quickly succeed in building OLAP cubes and data mining structures; unfortunately, sometimes they have no idea what they've actually created. Often these solutions do not reveal their flawed underlying design until after they've been deployed and are being run with production levels of data.

Because the SQL Server 2008 BI tools are designed to be intuitive, BI project implementation is pleasantly simple, as long as what you build properly implements standard BI concepts. If we've met our writing goals, you'll have enough of both conceptual and procedural knowledge after reading this book that you can successfully envision, design, develop, and deploy a BI project built using SQL Server 2008.

## Who This Book Is For

This book has more than one audience. The primary audience is professional developers who want to start work on a BI project using SSAS, SSIS, and SSRS. Our approach is one of inclusiveness—we have provided content targeted at both beginning and intermediate BI developers. We have also included information for business decision makers who wish to understand the capabilities of the technology and Microsoft’s associated tools. Because we believe that appropriate architecture is the underpinning of all successful projects, we’ve also included information for that audience.

We assume that our readers have production experience with a relational database. We also assume that they understand relational database queries, tables, normalization and joins, and other terms and concepts common to relational database implementations.

Although we’ve included some core information about administration of BI solutions, we consider IT pros (or BI administrators) to be a secondary audience for this book.

## What This Book Is About

This book starts by helping the reader develop an intuitive understanding of the complexity and capabilities of BI as implemented using SQL Server 2008, and then it moves to a more formal understanding of the concepts, architecture, and modeling. Next, it presents a more process-focused discussion of the implementation of BI objects, such as OLAP cubes and data mining structures, using the tools included in SQL Server 2008.

Unlike many other data warehousing books we’ve seen on the market, we’ve attempted to attain an appropriate balance between theory and practical implementation. Another difference between our book and others is that we feel that data mining is a core part of a BI solution. Because of this we’ve interwoven information about data mining throughout the book and have provided three chapters dedicated to its implementation.

## Part I, “Business Intelligence for Business Decision Makers and Architects”

The goal of this part of the book is to answer these questions:

- Why use BI?
- What can BI do?
- How do I get started?

In this first part, we address the business case for BI. We also introduce BI tools, methods, skills, and techniques. This section is written for developers, business decision makers, and

architects. Another way to look at our goal for this section is that we've tried to include all of the information you'll need to understand *before* you start developing BI solutions using SQL Server Analysis Services in the Business Intelligence Development Studio (BIDS) toolset.

**Chapter 1, “Business Intelligence Basics”** In this chapter, we provide a practical definition of exactly what BI is as implemented in SQL Server 2008. Here we define concepts such as OLAP, dimensional modeling, and more. Also, we discuss tools and terms such as BIDS, MDX, and more. Our aim is to provide you with a foundation for learning more advanced concepts.

**Chapter 2, “Visualizing Business Intelligence Results”** In this chapter, we look at BI from an end user's perspective using built-in BI client functionality in Microsoft Office Excel 2007. Here we attempt to help you visualize the results of BI projects—namely, OLAP cubes and data mining models.

**Chapter 3, “Building Effective Business Intelligence Processes”** In this chapter, we examine software development life-cycle processes that we use when envisioning, designing, developing, and deploying BI projects. Here we take a closer look at Microsoft Solutions Framework (and other software development life cycles) as applied to BI projects.

**Chapter 4, “Physical Architecture in Business Intelligence Solutions”** In this chapter, we examine best practices for establishing baselines in your intended production BI environment. We cover tools, such as SQL Server Profiler and more, that can help you prepare to begin a BI project. We also talk about physical servers—especially, number and placement. We include an introduction to security concepts. We close by discussing considerations for setting up a BI development environment.

**Chapter 5, “Logical OLAP Design Concepts for Architects”** In this chapter, we take a close look at core OLAP modeling concepts—namely, dimensional modeling. Here we take a look at star schemas, fact tables, dimensional hierarchy modeling, and more.

## Part II, “Microsoft SQL Server 2008 Analysis Services for Developers”

This part provides you with detailed information about how to use SSAS to build OLAP cubes and data mining models. Most of this section is focused on using BIDS by working on a detailed drill-down of all the features included. As we'll do with each part of the book, the initial chapters look at architecture and a simple implementation. Subsequent chapters are where we drill into intermediate and, occasionally, advanced concepts.

**Chapter 6, “Understanding SSAS in SSMS and SQL Server Profiler”** In this chapter, we look at OLAP cubes in SQL Server Management Studio and in SQL Server Profiler. We start here because we want you to understand how to script, maintain, and move objects that you've created for your BI solution. Also, SQL Server Profiler is a key tool to help you understand underlying MDX or DMX queries from client applications to SSAS structures.

**Chapter 7, “Designing OLAP Cubes Using BIDS”** In this chapter, we begin the work of developing an OLAP cube. Here we start working with BIDS, beginning with the sample SSAS database Adventure Works 2008 DW.

**Chapter 8, “Refining Cubes and Dimensions”** In this chapter, we dig deeper into the details of building OLAP cubes and dimensions using BIDS. Topics include dimensional hierarchies, key performance indicators (KPIs), MDX calculations, and cube actions. We explore both the cube and dimension designers in BIDS in great detail in this chapter.

**Chapter 9, “Processing Cubes and Dimensions”** In this chapter, we take a look at cube metadata and data storage modes. Here we discuss multidimensional OLAP (MOLAP), hybrid OLAP (HOLAP), and relational OLAP (ROLAP). We also look at the aggregation designer and discuss aggregation strategies in general. We also examine proactive caching.

**Chapter 10, “Introduction to MDX”** In this chapter, we depart from using BIDS and present a tutorial on querying by using MDX. We present core language features and teach via many code examples in this chapter.

**Chapter 11, “Advanced MDX”** In this chapter, we move beyond core language features to MDX queries to cover more advanced language features. We also take a look at how the MDX language is used throughout the BI suite in SQL Server 2008—that is, in BIDS for SSAS and SSRS.

**Chapter 12, “Understanding Data Mining Structures”** In this chapter, we take a look at the data mining algorithms that are included in SSAS. We examine each algorithm in detail, including presenting configurable properties, so that you can gain an understanding of what is possible with SQL Server 2008 data mining.

**Chapter 13, “Implementing Data Mining Structures”** In this chapter, we focus on practical implementation of data mining models using SSAS in BIDS. We work through each tab of the data mining model designer, following data mining implementation from planning to development, testing, and deployment.

## Part III, “Microsoft SQL Server 2008 Integration Services for Developers”

The goal of this part is to give you detailed information about how to use SSIS to develop extract, transform, and load (ETL) packages. You'll use these packages to load your OLAP cubes and data mining structures. Again, we'll focus on using BIDS while working on a detailed drill-down of all the features included. As with each part of the book, the initial chapters look at architecture and start with a simple implementation. Subsequent chapters are where we drill into intermediate and, occasionally, advanced concepts.

**Chapter 14, “Architectural Components of Microsoft SQL Server 2008 Integration Services”** In this chapter, we examine the architecture of SSIS. Here we take a look at the data flow pipeline and more.

**Chapter 15, “Creating Microsoft SQL Server 2008 Integration Services Packages with Business Intelligence Development Studio”** In this chapter, we explain the mechanics of package creation using BIDS. Here we present the control flow tasks and then continue by explaining data flow sources, destinations, and transformations. We continue working through the BIDS interface by covering variables, expressions, and the rest of the BIDS interface.

**Chapter 16, “Advanced Features in Microsoft SQL Server 2008 Integration Services”** In this chapter, we begin by taking a look at the error handling, logging, and auditing features in SSIS. Next we look at some common techniques for assessing data quality, including using the new Data Profiling control flow task.

**Chapter 17, “Microsoft SQL Server 2008 Integration Services Packages in Business Intelligence Solutions”** In this chapter, we take a look at extract, transform, and load processes and best practices associated with SSIS when it’s used as a tool to create packages for data warehouse loading. We look at this using both OLAP cubes and data mining models.

**Chapter 18, “Deploying and Managing Solutions in Microsoft SQL Server 2008 Integration Services”** In this chapter, we drill into the details of SSIS package deployment and management. Here we look at using Visual SourceSafe (VSS) and other source control solutions to manage distributed package deployment.

**Chapter 19, “Extending and Integrating SQL Server 2008 Integration Services”** In this chapter, we provide an explanation about the details of extending the functionality of SSIS packages using .NET-based scripts.

## **Part IV, “Microsoft SQL Server Reporting Services and Other Client Interfaces for Business Intelligence”**

The goal of this part is to give you detailed information about how to select and implement client interfaces for OLAP cubes and data mining structures. We’ll look in great detail at SSRS. In addition, we’ll examine using Excel, Visio, or Office SharePoint Server 2007 as your BI client of choice. We’ll look at SSRS architecture, then at designing reports using BIDS and other tools. Then we’ll move to a detailed look at implementing other clients, including a discussion of the process for embedding results in a custom Windows Form or Web Form application. As we do with each part of the book, our first chapters look at architecture, after which we start with simple implementation. Subsequent chapters are where we drill into intermediate and, occasionally, advanced concepts.

**Chapter 20, “Creating Reports in SQL Server 2008 Reporting Services”** In this chapter, we present the architecture of SQL Server Reporting Services. We cover the various parts and pieces that you’ll have to implement to make SSRS a part of your BI solution.

**Chapter 21, “Building Reports for SQL Server 2008 Reporting Services”** In this chapter, we drill into the detail of building reports using BIDS. We take a look at the redesigned interface and then look at the details of designing reports for OLAP cubes and data mining models.

**Chapter 22, “Advanced SQL Server 2008 Reporting Services”** In this chapter, we look at programmatically extending SSRS as well as other advanced uses of SSRS in a BI project. Here we look at using the ReportViewer control in custom SSRS clients. We also take a look at the new integration between SSRS and Excel and Word 2007.

**Chapter 23, “Using Microsoft Excel 2007 as an OLAP Cube Client”** In this chapter, we walk through the capabilities included in Excel 2007 as an OLAP cube client. We take a detailed look at the PivotTable functionality and also examine the PivotChart as a client interface for OLAP cubes.

**Chapter 24, “Microsoft Office 2007 as a Data Mining Client”** In this chapter, we look at using the SQL Server 2008 Data Mining Add-ins for Office 2007. These add-ins enable Excel 2007 to act as a client to SSAS data mining. We look at connecting to existing models on the server as well creating temporary models in the Excel session using Excel source data. We also examine the new tools that appear on the Excel 2007 Ribbon after installing the add-ins.

**Chapter 25, “SQL Server 2008 Business Intelligence and Microsoft Office SharePoint Server 2007”** In this chapter, we look at integration between SQL Server Reporting Services and SharePoint technologies. We focus on integration between SSRS and Office SharePoint Server 2007. Here we detail the integrated mode option for SSRS and Office SharePoint Server 2007. We also look at the Report Center template included in Office SharePoint Server 2007 and detail just how it integrates with SSRS. We have also included information about Excel Services.

## Prerelease Software

This book was written and tested against the release to manufacturing (RTM) 2008 version of SQL Server Enterprise software. Microsoft released the final version of Microsoft SQL Server 2008 (build number 10.0.1600.22) in August 2008. We did review and test our examples against the final release of the software. However, you might find minor differences between the production release and the examples, text, and screen shots in this book. We made every attempt to update all of the samples shown to reflect the RTM; however, minor variances in screen shots or text between the community technology preview (CTP) samples and the RTM samples might still remain.

## Hardware and Software Requirements

You'll need the following hardware and software to work with the information and examples provided in this book:

- Microsoft Windows Server 2003 Standard edition or later. Microsoft Windows Server 2008 Enterprise is preferred. The Enterprise edition of the operating system is required if you want to install the Enterprise edition of SQL Server 2008.
- Microsoft SQL Server 2008 Standard edition or later. Enterprise edition is required for using all features discussed in this book. Installed components needed are SQL Server Analysis Services, SQL Server Integration Services, and SQL Server Reporting Services.
- SQL Server 2008 Report Builder 2.0.
- Visual Studio 2008 (Team System is used to show the examples).
- Office SharePoint Server 2007 (Enterprise Edition or Windows SharePoint Services 3.0).
- Office 2007 Professional edition or better, including Excel 2007 and Visio 2007.
- SQL Server 2008 Data Mining Add-ins for Office 2007.
- 1.6 GHz Pentium III+ processor or faster.
- 1 GB of available, physical RAM.
- 10 GB of hard disk space for SQL Server and all samples
- Video (800 by 600 or higher resolution) monitor with at least 256 colors.
- CD-ROM or DVD-ROM drive.
- Microsoft mouse or compatible pointing device.

## Find Additional Content Online

As new or updated material becomes available that complements this book, it will be posted online on the Microsoft Press Online Developer Tools Web site. The type of material you might find includes updates to book content, articles, links to companion content, errata, sample chapters, and more. This Web site is located at [www.microsoft.com/learning/books/online/developer](http://www.microsoft.com/learning/books/online/developer) and is updated periodically.

Lynn Langit is recording a companion screencast series named "How Do I BI?" Find this series via her blog at <http://blogs.msdn.com/SoCalDevGal>.

## Support for This Book

Every effort has been made to ensure the accuracy of this book. As corrections or changes are collected, they will be added to a Microsoft Knowledge Base article.

Microsoft Press provides support for books at the following Web site:

*<http://www.microsoft.com/learning/support/books/>*

## Questions and Comments

If you have comments, questions, or ideas regarding the book, or questions that are not answered by visiting the site above, please send them to Microsoft Press via e-mail to

*[mspinput@microsoft.com](mailto:mspinput@microsoft.com)*

Or via postal mail to

Microsoft Press

*Attn: Smart Business Intelligence Solutions with Microsoft SQL Server 2008 Editor*

One Microsoft Way

Redmond, WA 98052-6399

Please note that Microsoft software product support is not offered through the above addresses.

## Chapter 1

# Business Intelligence Basics

Many real-world business intelligence (BI) implementations have been delayed or even derailed because key decision makers involved in the projects lacked even a general understanding of the potential of the product stack. In this chapter, we provide you with a conceptual foundation for understanding the broad potential of the BI technologies within Microsoft SQL Server 2008 so that you won't have to be in that position. We define some of the basic terminology of business intelligence, including OLTP and OLAP, and go over the components, both core and optional, of Microsoft BI solutions. We also introduce you to the development languages involved in BI projects, including MDX, DMX, XMLA, and RDL.

If you already know these basic concepts, you can skip to Chapter 2, "Visualizing Business Intelligence Results," which talks about some of the common business problems that BI addresses.

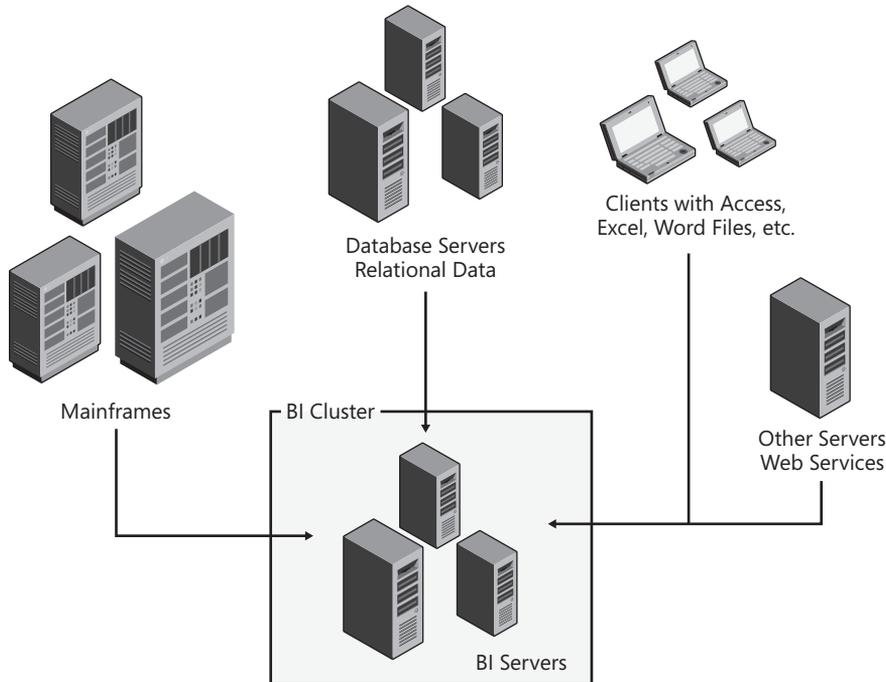
## Business Intelligence and Data Modeling

You'll see the term *business intelligence* defined in many different ways and in various contexts. Some vendors manufacture a definition that shows their tools in the best possible light. You'll sometimes hear BI summed up as "efficient reporting." With the BI tools included in SQL Server 2008, business intelligence is much more than an overhyped, supercharged reporting system. For the purposes of this book, we define business intelligence in the same way Microsoft does:

*Business intelligence solutions include effective storage and presentation of key enterprise data so that authorized users can quickly and easily access and interpret it. The BI tools in SQL Server 2008 allow enterprises to manage their business at a new level, whether to understand why a particular venture got the results it did, to decide on courses of action based on past data, or to accurately forecast future results on the basis of historical data.*

You can customize the display of BI data so that it is appropriate for each type of user. For example, analysts can drill into detailed data, executives can see timely high-level summaries, and middle managers can request data presented at the level of detail they need to make good day-to-day business decisions. Microsoft BI usually uses data structures (called *cubes* or *data mining structures*) that are optimized to provide fast, easy-to-query decision support. This BI data is presented to users via various types of reporting interfaces. These formats can include custom applications for Microsoft Windows, the Web, or mobile devices as well as Microsoft BI client tools, such as Microsoft Office Excel or SQL Server Reporting Services.

Figure 1-1 shows a conceptual view of a BI solution. In this figure, multiple types of source data are consolidated into a centralized data storage facility. For a formal implementation of a BI solution, the final destination container is most commonly called a cube. This consolidation can be physical—that is, all the source data is physically combined onto one or more servers—or logical, by using a type of a view. We consider BI conceptual modeling in more detail in Chapter 5, “Logical OLAP Design Concepts for Architects.”



**FIGURE 1-1** Business intelligence solutions present a consolidated view of enterprise data. This view can be a physical or logical consolidation, or a combination of both.

Although it's possible to place all components of a BI solution on a single physical server, it's more typical to use multiple physical servers to implement a BI solution. Microsoft Windows Server 2008 includes tremendous improvements in virtualization, so the number of physical servers involved in a BI solution can be greatly reduced if you are running this version. We talk more about physical modeling for BI solutions in Chapter 5.

Before we examine other common BI terms and components, let's review two core concepts in data modeling: OLTP and OLAP.

# OLTP and OLAP

You've probably heard the terms *OLTP* and *OLAP* in the context of data storage. When planning SQL Server 2008 BI solutions, you need to have a solid understanding of these systems as well as the implications of using them for your particular requirements.

## Online Transactional Processing

OLTP stands for *online transactional processing* and is used to describe a relational data store that is designed and optimized for transactional activities. Transactional activities are defined as inserts, updates, and deletes to rows in tables. A typical design for this type of data storage system is to create a large number of normalized tables in a single source database.

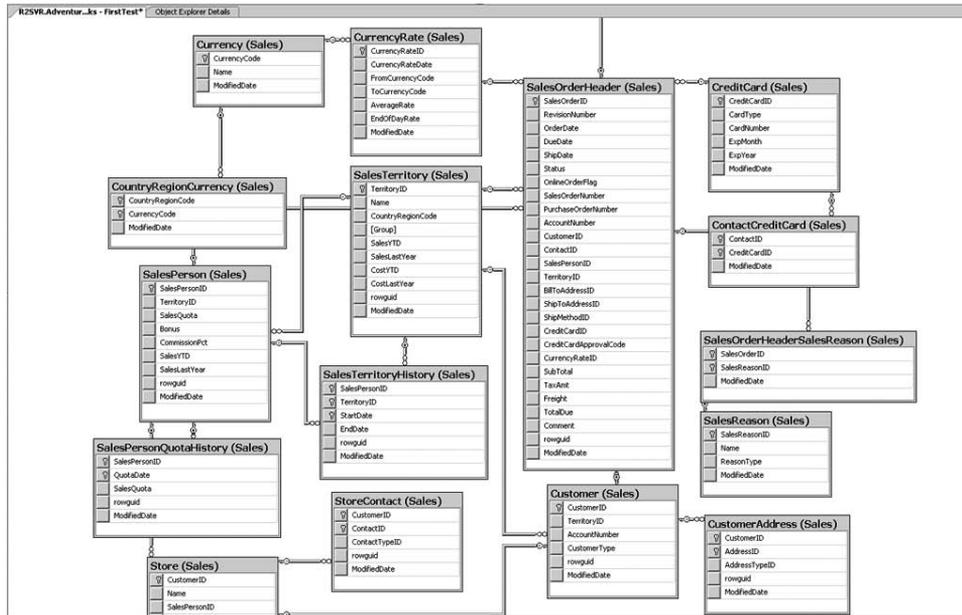
### Relational vs. Nonrelational Data

SQL Server 2008 BI solutions support both relational and nonrelational source data. Relational data usually originates from a relational database management system (RDBMS) such as SQL Server 2008 (or an earlier version of SQL Server) or an RDBMS built by a different vendor, such as Oracle or IBM. Relational databases generally consist of a collection of related tables. They can also contain other objects, such as views or stored procedures.

Nonrelational data can originate from a variety of sources, including Windows Communication Foundation (WCF) or Web services, mainframes, and file-based applications, such as Microsoft Office Word or Excel. Nonrelational data can be presented in many formats. Some of the more common formats are XML, TXT, CSV, and various binary formats.

Normalization in relational data stores is usually implemented by creating a primary-key-to-foreign-key relationship between the rows in one table (often called the *parent* table) and the rows in another table (often called the *child* table). Typically (though not always), the rows in the parent table have a one-to-many relationship with the rows in the child table. A common example of this relationship is a Customer table and one or more related [Customer] Orders tables. In the real world, examples are rarely this simple. Variations that include one-to-one or many-to-many relationships, for example, are possible. These relationships often involve many source tables.

Figure 1-2 shows the many tables that can result when data stores modeled for OLTP are normalized. The tables are related by keys.



**FIGURE 1-2** Sample from AdventureWorks OLTP database

The primary reasons to model a data store in this way (that is, normalized) are to reduce the total amount of data that needs to be stored and to improve the efficiency of performing inserts, updates, and deletes by reducing the number of times the same data needs to be added, changed, or removed. Extending the example in Figure 1-2, if you inserted a second order for an existing customer and the customer's information hadn't changed, no new information would have to be inserted into the Customer table; instead, only one or more rows would have to be inserted into the related Orders tables, using the customer identifier (usually a key value), to associate the order information with a particular customer. Although this type of modeling is efficient for these activities (that is, inserting, updating, and deleting data), the challenge occurs when you need to perform extensive reading of these types of data stores.

To retrieve meaningful information from the list of Customers and the particular Order information shown in Figure 1-2, you'd first have to select the rows meeting the report criteria from multiple tables and then sort and match (or join) those rows to create the information you need. Also, because a common business requirement is viewing aggregated information, you might want to see the total sales dollar amount purchased for each customer, for example. This requirement places additional load on the query processing engine of your OLTP data store. In addition to selecting, fetching, sorting, and matching the rows, the engine also has to aggregate the results.

If the query you make involves only a few tables (for example, the Customer table and the related SalesOrderHeader tables shown in Figure 1-2), and if these tables contain a small

number of rows, the overhead incurred probably would be minimal. (In this context, the definition of a “small” number is relative to each implementation and is best determined by baseline performance testing during the development phase of the project.) You might be able to use a highly normalized OLTP data store to support both CRUD (create, retrieve, update, delete) and read-only (decision support or reporting) activities. The processing speed depends on your hardware resources and the configuration settings of your database server. You also need to consider the number of users who need to access the information simultaneously.

These days, the OLTP data stores you are querying often contain hundreds or even thousands of source tables. The associated query processors must filter, sort, and aggregate millions of rows from the related tables. Your developers need to be fluent in the data store query language so that they can write efficient queries against such a complex structure, and they also need to know how to capture and translate every business requirement for reporting. You might need to take additional measures to improve query (and resulting report) performance, including rewriting queries to use an optimal query syntax, analyzing the query execution plan, providing hints to force certain execution paths, and adding indexes to the relational source tables. Although these strategies can be effective, implementing them requires significant skill and time on the part of your developers. Figure 1-3 shows an example of a typical reporting query—not the complexity of the statement, but the number of tables that can be involved.

```

SQLQuery1.sql -...istrator (56))* WIN-VVP8K0GA4...008 - F01xx02
SELECT
    emp.BusinessEntityID
    ,p.FirstName + ' ' + COALESCE(p.MiddleName, '') + ' ' + p.LastName AS [FullName]
    ,soh.TerritoryID
    ,emp.JobTitle
    ,sp.BusinessEntityID AS Expr1
    ,soh.SalesPersonID
    ,terr.Name
    ,terr.CountryRegionCode
    ,soh.SalesOrderID
    ,sod.ProductID
    ,sod.UnitPrice
    ,soh.OrderDate
    ,soh.DueDate
FROM
    Sales.SalesPerson sp
        INNER JOIN
            Person.Person p ON sp.BusinessEntityID = p.BusinessEntityID
        INNER JOIN
            Sales.SalesTerritory terr ON sp.TerritoryID = terr.TerritoryID
        INNER JOIN
            Sales.SalesOrderHeader soh ON sp.BusinessEntityID = soh.SalesPersonID
            AND
                terr.TerritoryID = soh.TerritoryID
        INNER JOIN
            Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID
            AND
                soh.SalesOrderID = sod.SalesOrderID
        INNER JOIN
            HumanResources.Employee emp ON sp.BusinessEntityID = emp.BusinessEntityID
            AND
                p.BusinessEntityID = emp.BusinessEntityID

```

**FIGURE 1-3** Sample reporting query against a normalized data store

SQL Server 2008 includes a powerful tool, the Database Engine Tuning Advisor, that can assist you in manual tuning efforts, though the amount of time needed to implement and maintain manual query tuning can become significant. Other costs are also involved with OLTP query optimization, the most significant of which is the need for additional storage space and maintenance tasks as indexes are added. A good way to think of the move from OLTP alone to a combined solution that includes both an OLTP store and an OLAP store is as a continuum that goes from OLTP (single RDBMS) relational to relational copy to OLAP (cube) nonrelational. In particular, if you're already making a copy of your OLTP source data to create a more efficient data structure from which to query for reporting and to reduce the load on your production OLTP servers, you're a prime candidate to move to a more formalized OLAP solution based on the dedicated BI tools included in SQL Server 2008.

## Online Analytical Processing

OLAP stands for *online analytical processing* and is used to describe a data structure that is designed and optimized for analytical activities. Analytical activities are defined as those that focus on the best use of data for the purpose of reading it rather than optimizing it so that changes can be made in the most efficient way. In fact, many OLAP data stores are implemented as read-only. Other common terms for data structures set up and optimized for OLAP are decision support systems, reporting databases, data warehouses, or cubes.

As with OLTP, definitions of an OLAP data store vary depending on who you're talking to. At a minimum, most professionals would agree that an OLAP data store is modeled in a denormalized way. When denormalizing, you use very wide relational tables (those containing many columns) with deliberately duplicated information. This approach reduces the number of tables that must be joined to provide query results and lets you add indexes. Reducing the size of surface area that is queried results in faster query execution. Data stores that are modeled for OLAP are usually denormalized using a specific type of denormalization modeling called a star schema. We cover this technique extensively in Chapter 4, "Physical Architecture in Business Intelligence Solutions."

Although using a denormalized relational store addresses some of the challenges encountered when trying to query an OLTP (or normalized) data store, a denormalized relational store is still based on relational tables, so the bottlenecks are only partially mitigated. Developers still must write complex queries for each business reporting requirement and then manually denormalize the copy of the OLTP store, add indexes, and further tune queries as performance demands. The work involved in performing these tasks can become excessive.

Figure 1-4 shows a portion of a denormalized relational data store. We are working with the AdventureWorksDW sample database, which is freely available for download and is designed to help you understand database modeling for loading OLAP cubes. Notice the numerous columns in each table.

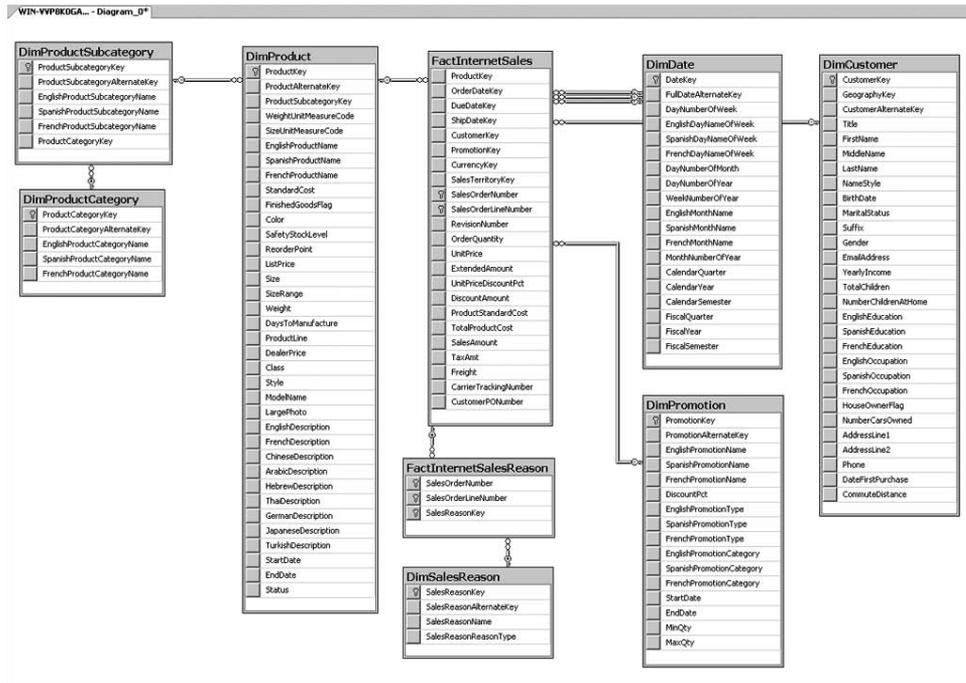


FIGURE 1-4 Portion of AdventureWorksDW

Another way to implement OLAP is to use a cube rather than a group of tables. A cube is one large store that holds all associated data in a single structure. The structure contains not only source data but also pre-aggregated values. A cube is also called a *multidimensional data store*.

## Aggregation

Aggregation is the application of some type of mathematic function to data. Although aggregation can be as simple as doing a SUM on numeric values, in BI solutions, it is often much more complex. In most relational data stores, the number of aggregate functions available is relatively small. For SQL Server 2008, for example, Transact-SQL contains 12 aggregate functions: AVG, MIN, CHECKSUM\_AGG, SUM, COUNT, STDEV, COUNT\_BIG, STDEVP, GROUPING, VAR, MAX, and VARP. Contrast this with the number of built-in functions available in the SQL Server cube store: over 150. The number and type of aggregate functions available in the cube store are more similar to those available in Excel than to those in SQL Server 2008.

So, what exactly does a cube look like? It can be tricky to visualize a cube because most of us can imagine only in three dimensions. Cubes are  $n$ -dimensional structures because they can store data in an infinite number of dimensions. A conceptual rendering of a cube is shown in

Figure 1-5. Cubes contain two major aspects: facts and dimensions. Facts are often numeric and additive, although that isn't a requirement. Facts are sometimes called measures. An example of a fact is "gross sales amount." Dimensions give meaning to facts. For example, you might need to be able to examine gross sales amount by time, product, customer, and employee. All of the "by xxx" values are dimensions. Dimensional information is accessed via a hierarchy of information along each dimensional axis. You'll also hear the term Unified Dimensional Model (UDM) to describe an OLAP cube because, in effect, such a cube "unifies" a group of dimensions. We discuss this type of modeling in much greater detail in Chapter 5.

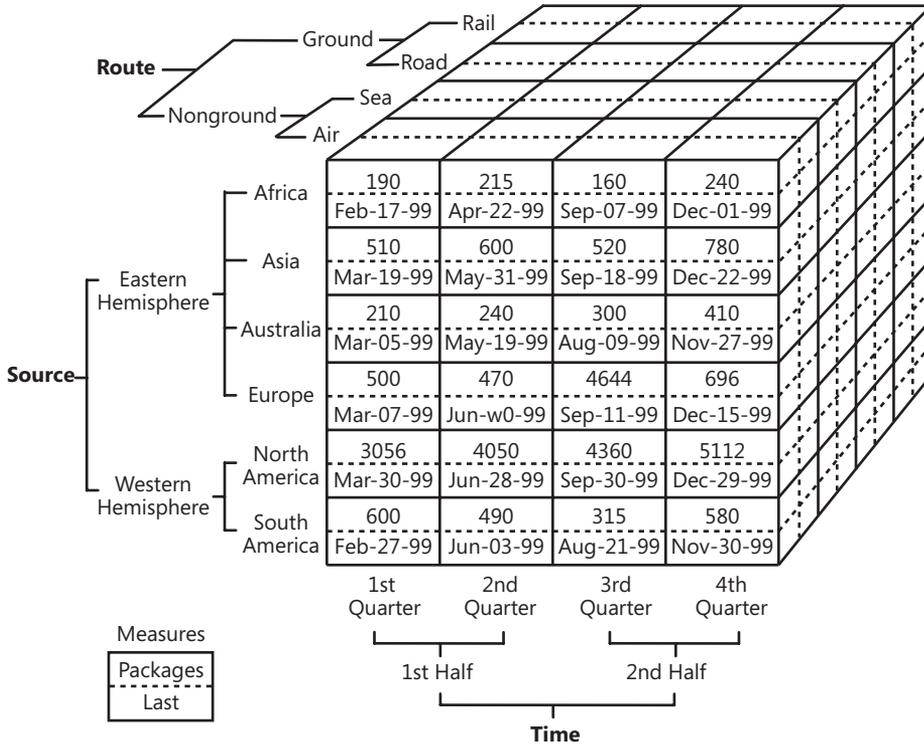


FIGURE 1-5 Sample cube structure

You might be wondering why a cube is preferable to a denormalized relational data store. The answer is efficiency—in terms of scalability, performance, and ease of use. In the case of using the business intelligence toolset in SQL Server 2008, you'll also get a query processing engine that is specifically optimized to deliver fast queries, particularly for queries that involve aggregation. We'll review the business case for using a dedicated OLAP solution in more detail in Chapter 2. Also, we'll look at real-world examples throughout the book.

The next phase of understanding an OLAP cube is translating that *n*-dimensional structure to a two-dimensional screen so that you can visualize what users will see when working with an OLAP cube. The standard viewer for a cube is a pivot table interface, a sample of which

is shown in Figure 1-6. The built-in viewer in the developer and administrative interfaces for SQL Server 2008 OLAP cubes both use a type of pivot table to allow developers and administrators to visualize the cubes they are working with. Excel 2007 PivotTables are also a common user interface for SQL Server 2008 cubes.

| Country        | Internet Total Product Cost | Internet Total Product Cost | Internet Total Product Cos |
|----------------|-----------------------------|-----------------------------|----------------------------|
| Australia      | \$746,262.98                | \$1,281,980.88              | \$1,766,196.35             |
| Canada         | \$85,947.36                 | \$371,743.45                | \$286,747.10               |
| France         | \$103,352.53                | \$303,137.86                | \$583,581.34               |
| Germany        | \$131,201.02                | \$314,916.20                | \$603,620.08               |
| United Kingdom | \$167,825.31                | \$346,553.84                | \$750,036.58               |
| United States  | \$623,893.04                | \$1,294,523.74              | \$1,567,314.25             |
| Grand Total    | \$1,858,482.24              | \$3,912,855.98              | \$5,557,495.71             |

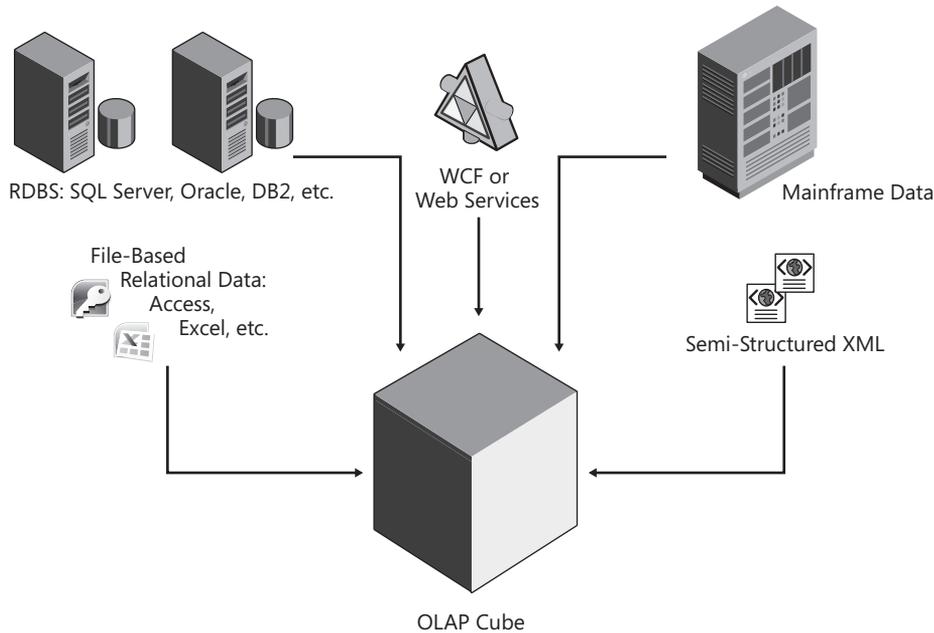
FIGURE 1-6 SQL Server 2008 cube presented in a PivotTable

## Common BI Terminology

A number of other conceptual terms are important to understand when you're planning a BI solution. In this section, we'll talk about several of these: data warehouses; data marts; cubes; decision support systems; data mining systems; extract, transform, and load systems; report processing systems; and key performance indicators.

### Data Warehouses

A data warehouse is a single structure that usually consists of one or more cubes. Figure 1-7 shows the various data sources that contribute to an OLAP cube. Data warehouses are used to hold an aggregated, or rolled-up (and most commonly) read-only, view of the majority of an organization's data. This structure includes client query tools. When planning and implementing your company's data warehouse, you need to decide which data to include and at what level of detail (or granularity). We explore this concept in more detail in "Extract, Transform, and Load Systems" later in this chapter.



**FIGURE 1-7** Conceptual OLAP cube

The terms *OLAP* and *data warehousing* are sometimes used interchangeably. However, this is a bit of an oversimplification because an OLAP store is modeled as a cube or multidimensionally, whereas a data warehouse can use either denormalized OLTP data or OLAP. OLAP and data warehousing are not new technologies. The first Microsoft OLAP tools were part of SQL Server 7. What is new in SQL Server 2008 is the inclusion of powerful tools that allow you to implement a data warehouse using an OLAP cube (or cubes). Implementing BI solutions built on OLAP is much easier because of improved tooling, performance, administration, and usability, which reduces total cost of ownership (TCO).

### Pioneers of Data Warehousing

Data warehousing has been available, usually implemented via specialized tools, since the early 1980s. Two principal thought leaders of data warehousing theory are Ralph Kimball and Bill Inmon. Both have written many articles and books and have popular Web sites talking about their extensive experience with data warehousing solutions using products from many different vendors.

To read more about Ralph Kimball's ideas on data warehouse design modeling, go to <http://www.ralphkimball.com>. I prefer the Kimball approach to modeling and have had good success implementing Kimball's methods in production BI projects. For a simple explanation of the Kimball approach, see [http://en.wikipedia.org/wiki/Ralph\\_Kimball](http://en.wikipedia.org/wiki/Ralph_Kimball).

## Data Marts

A data mart is a defined subset of enterprise data, often a single cube from a group of cubes, that is intended to be consolidated into a data warehouse. The single cube represents one business unit (for example, marketing) from a greater whole (for example, the entire company). Data marts were the basic units of organization in the OLAP tools that were included in earlier versions of SQL Server BI solutions because of restrictions in the tools themselves. The majority of these restrictions were removed in SQL Server 2005. Because of this, data warehouses built using the tools provided by SQL Server 2008 often consist of one huge cube. This is not the case with many competitive OLAP products. There are, of course, exceptions to this single-cube design. However, limits in the product stack are not what determines this type of design; rather, it is determined by OLAP modeler or developer preferences.

## Cubes

As described earlier in the chapter, a BI cube is a data structure used by classic data warehousing products (including SQL Server 2008) in place of many relational tables. Rather than containing tables with rows and columns, cubes consist of dimensions and measures (or facts). Cubes can also contain data that is pre-aggregated (usually summed) rather than included as individual items (or rows). In some cases, cubes contain a complete copy of production data; in other cases, they contain subsets of source data.

In SQL Server 2008, cubes are more scalable and perform better than in previous versions of SQL Server, so you can include data with much more detail than you could include when using previous versions of the SQL Server 2008 OLAP tool, with many fewer adverse effects on scalability and performance. As in previous versions, when you are using SQL Server 2008, you will copy the source data from any number of disparate source systems to the destination OLAP cubes via extract, transform, and load (ETL) processes. (You'll find out more about ETL shortly.)

We talk a lot about cubes in this book, from their physical and logical design in Chapters 5 and 6 to the use of the cube-building tools that come with SQL Server 2008 in Part II, "Microsoft SQL Server 2008 Analysis Services for Developers."

## Decision Support Systems

The term *decision support system* can mean anything from a read-only copy of an OLTP data store to a group of OLAP cubes, or even a mixture of both. If the data source consists of only an OLTP data store, then this type of store can be limited in its effectiveness because of the challenges discussed earlier in this chapter, such as the difficulty of efficient querying and the overhead required for indexes. Another way to think about a decision support system is some type of data structure (such as a table or a cube) that is being used as a basis for developing end-user reporting. End-user in this context means all types or categories of users. These

usually include business decision makers, middle managers, and general knowledge workers. It is critical that your solution be able to provide data summarized at a level of detail that is useful to these various end-user communities. The best BI solutions are intuitive for the various end-user communities to work with—little or no end-user training is needed.

In this book, we focus on using the more efficient OLAP data store (or cube) as a source for a decision support system.

## Data Mining Systems

Data mining can be understood as a complementary technique to OLAP. Whereas OLAP is used to provide decision support or the data to prove a particular hypothesis, data mining is used in situations in which you have no solid hypothesis about the data. For example, you could use an OLAP cube to verify that customers who purchased a certain product during a certain timeframe had certain characteristics. Specifically, you could prove that customers who purchased cars during December 2007 chose red-colored cars twice as often as they picked black-colored cars if those customers shopped at locations in postal codes 90201 to 90207. You could use a data mining store to automatically correlate purchase factors into buckets, or groups, so that decision makers could explore correlations and then form more specific hypotheses based on their investigation. For example, they could decide to group or cluster all customers segmented into “car purchasers” and “non-car-purchasers” categories. They could further examine the clusters to find that “car purchasers” had the following traits most closely correlated, in order of priority: home owners (versus non-home owners), married (versus single), and so on.

Another scenario for which data mining is frequently used is one where your business requirements include the need to predict one or more future target values in a dataset. An example of this would be the rate of sale—that is, the number of items predicted to be sold over a certain rate of time.

We explore the data mining support included in SQL Server 2008 in greater detail in Chapter 6, “Understanding SSAS in SSMS and SQL Server Profiler,” in the context of logical modeling. We also cover it in several subsequent chapters dedicated to implementing solutions that use data mining.

## Extract, Transform, and Load Systems

Commonly expressed as ETL, *extract, transform, and load* refers to a set of services that facilitate the extraction, transformation, and loading of the various types of source data (for example, relational, semi-structured, and unstructured) into OLAP cubes or data mining structures. SQL Server 2008 includes a sophisticated set of tools to accomplish the ETL processes associated with the initial loading of data into cubes as well as to process subsequent

incremental inserts of data into cubes, updates to data in cubes, and deletions of data from cubes. ETL is explored in detail in Part III, “Microsoft SQL Server 2008 Integration Services for Developers.” A common error made in BI solutions is underestimating the effort that will be involved in the ETL processes for both the initial OLAP cube and the data mining structure loads as well as the effort involved in ongoing maintenance, which mostly consists of inserting new data but can also include updating and deleting data. It is not an exaggeration to say that up to 75 percent of the project time for the initial work on a BI project can be attributed to the ETL portion of the project. The “dirtiness,” complexity, and general incomprehensibility of the data originating from various source systems are factors that are often overlooked in the planning phase. By “dirtiness,” we mean issues such as invalid data. This can include data of an incorrect type, format, length, and so on.

## Report Processing Systems

Most BI solutions use more than one type of reporting client because of the different needs of the various users who need to interact with the cube data. An important part of planning any BI solution is to carefully consider all possible reporting tools. A common production mistake is to under-represent the various user populations or to clump them together when a more thorough segmentation would reveal very different reporting needs for each population.

SQL Server 2008 includes a report processing system designed to support OLAP data sources. In Part IV, “Microsoft SQL Server Reporting Services and Other Client Interfaces for Business Intelligence,” we explore the included tools—such as SQL Server Reporting Services, Office SharePoint Server 2007, and PerformancePoint Server—as well as other products that are part of the Microsoft product suite for reporting.

## Key Performance Indicators

Key performance indicators (KPIs) are generally used to indicate a goal that consists of several values—actual, target, variance to target, and trend. Most often, KPIs are expressed and displayed graphically—for example, as different colored traffic lights (red, yellow, or green). KPIs usually include drill-down capabilities that allow interested decision makers to review the data behind the KPI. KPIs can be implemented as part of an OLAP system, and they are often part of reporting systems, which are most typically found on reporting dashboards or score-cards. It is quite common for business requirements to include the latter as part of a centralized performance management strategy. Microsoft’s BI tools include the ability to create and view KPIs. You can create KPIs from nearly any type of data source, such as OLAP cubes, Excel workbooks, or SharePoint lists.

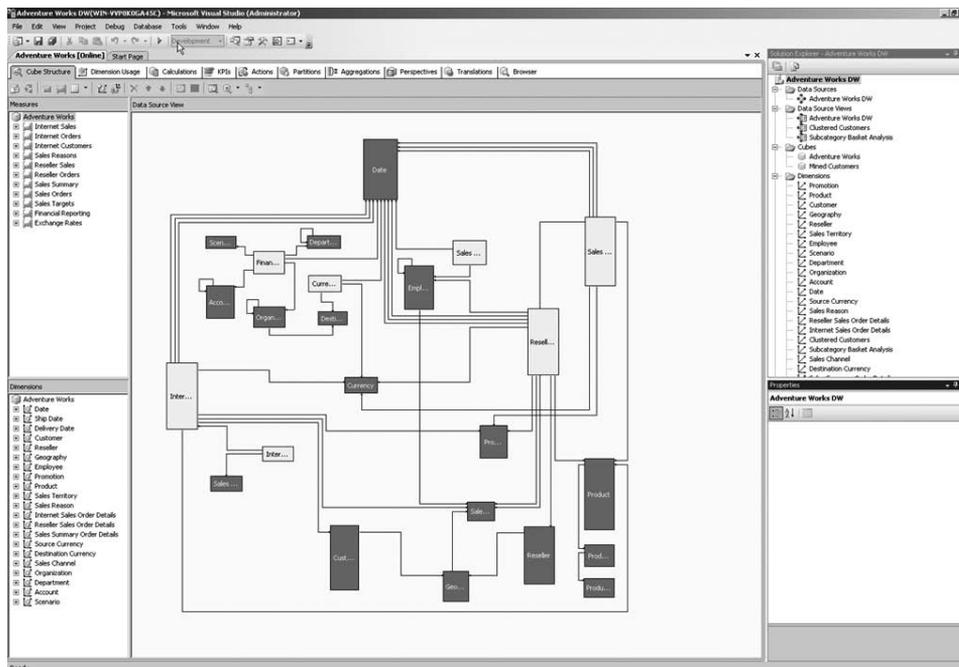
## Core Components of a Microsoft BI Solution

The core components of a Microsoft BI solution are SQL Server Analysis Services (SSAS), SQL Server Reporting Services (SSRS), and SQL Server 2008 itself. SQL Server is often used as a data source or an intermediate repository for data as it is being validated in preparation for loading into an OLAP cube. The ETL toolset, called SQL Server Integration Services (SSIS), also requires a SQL Server 2008 license. Each of these core services comes with its own set of management interface tools. The SSAS development interface is the Business Intelligence Development Studio (BIDS); SSIS and SSRS use the same development interface. The administrative interface all three use is SQL Server Management Studio (SSMS).

We typically use SQL Server Reporting Services in our solutions. Sometimes a tool other than SSRS will be used as the report engine for data stores developed using SSAS. However, you will typically use, at a minimum, the components just listed for a BI solution built using SQL Server 2008. For example, you could purchase or build a custom application that doesn't use SSRS to produce reports. In that case, the application would include data queries as well as calls to the SSAS structures via the application programming interfaces (APIs) included for it.

### SQL Server 2008 Analysis Services

SSAS is the core service in a Microsoft BI solution. It provides the storage and query mechanisms for the data used in OLAP cubes for the data warehouse. It also includes sophisticated OLAP cube developer and administrative interfaces. SSAS is usually installed on at least one dedicated physical server. You can install both SQL Server 2008 and SSAS on the same physical server, but this is done mostly in test environments. Figure 1-8 shows the primary tool you'll use to develop cubes for SSAS, the Business Intelligence Development Studio. BIDS opens in a Microsoft Visual Studio environment. You don't need a full Visual Studio installation to develop cubes for SSAS. If Visual Studio is not on your development machine, when you install SSAS, BIDS installs it as a stand-alone component. If Visual Studio is on your development machine, BIDS installs as a component (really, a set of templates) in your existing Visual Studio instance.



**FIGURE 1-8** AdventureWorksDW in the OLAP cube view within BIDS



**Note** If you're running a full version of Visual Studio 2008 on the same machine where you intend to work with SSAS, you must install Service Pack 1 (SP1) for Visual Studio 2008.

## AdventureWorksDW

AdventureWorksDW is the sample data and metadata that you can use while learning about the tools and capabilities of the SQL Server 2008 BI tools. We provide more information about how to work with this sample later in this chapter. All screen shots in this book show this sample being used. The samples include metadata and data so that you can build OLAP cubes and mining structures, SSIS packages, and SSRS reports. These samples are also available on Microsoft's public, shared-source Web site: CodePlex at <http://codeplex.com/SqlServerSamples>. Here you'll find the specific locations from which you can download these samples. Be sure to download samples that match your version (for example, 2008 or 2005) and platform (x86 or x64). When running the samples, be sure to use the sample for your edition of SQL Server.

## Data Mining with Analysis Services 2008

SSAS also includes a component that allows you to create data mining structures that include data mining models. Data mining models are objects that contain source data (either relational or multidimensional) that has been processed using one or more data mining algorithms. These algorithms either classify (group) data or classify and predict one or more column values. Although data mining has been available since SSAS 2000, its capabilities have been significantly enhanced in the SQL Server 2008 release. Performance is improved, and additional configuration capabilities are available. Figure 1-9 shows a data mining model visualizer that comes with SQL Server 2008. Data mining visualizers are included in the data mining development environment (BIDS), as well as in some client tools, such as Excel. Chapter 12, “Understanding Data Mining Structures,” and Chapter 13, “Implementing Data Mining Structures,” cover the data mining capabilities in SSAS in more detail.

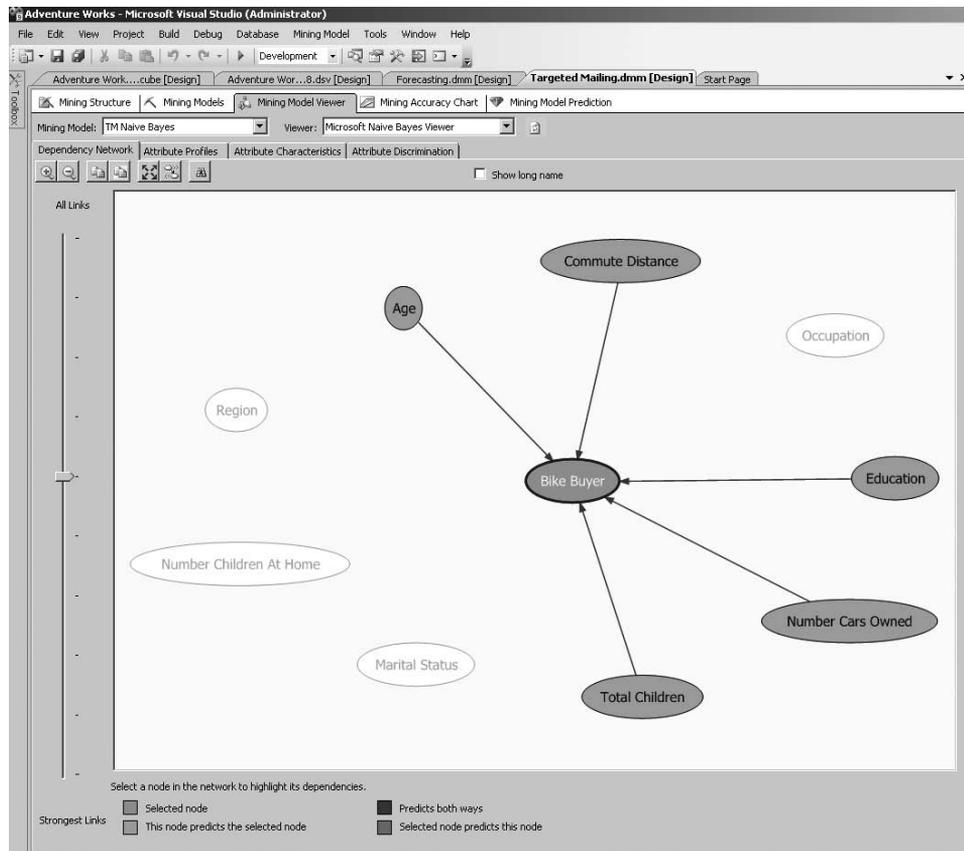


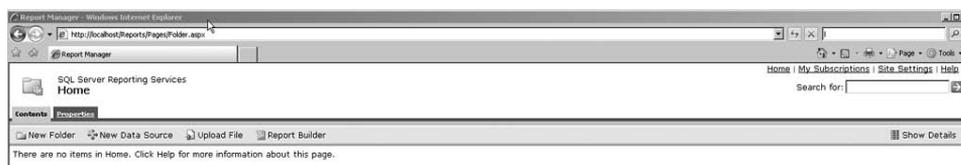
FIGURE 1-9 Business Intelligence Development Studio (BIDS) data mining visualizer

## SQL Server 2008 Reporting Services

Another key component in many BI solutions is SQL Server Reporting Services (SSRS). When working with SQL Server 2008 to perform SSRS administrative tasks, you can use a variety of included tools such as SSMS, a reporting Web site, or a command-line tool.

The enhancements made in SQL Server 2008 Reporting Services make it an attractive part of any BI solution. The SSRS report designer in BIDS includes a visual query designer for SSAS cubes, which facilitates rapid report creation by reducing the need to write manual queries against OLAP cube data. SSRS includes another report-creation component, Report Builder, which is intended to be used by analysts, rather than developers, to design reports. SSRS also includes several client tools: a Web interface (illustrated in Figure 1-10), Web Parts for Microsoft Office SharePoint Server, and client components for Windows Forms applications.

We discuss all flavors of reporting clients in Part IV.



**FIGURE 1-10** SSRS reports can be displayed by using the default Web site interface.

## SQL Server 2008

In addition to being a preferred staging source for BI data, SQL Server 2008 RDBMS data is often a portion of the source data for BI solutions. As we mentioned earlier in this chapter, data can be and often is retrieved from a variety of relational source data stores (for example, Oracle, DB2, and so forth). To be clear, data from any source for which there is a provider can be used as source data for an OLAP (SSAS) cube, which means data from all versions of SQL Server along with data from other RDBMS systems.

A SQL Server 2008 installation isn't strictly required to implement a BI solution; however, because of the integration of some key toolsets that are part of nearly all BI solutions, such as SQL Server Integration Services—which is usually used to perform the ETL processes for the OLAP cubes and data mining structures—most BI solutions should include at least one SQL Server 2008 installation. As we said earlier, although the SQL Server 2008 installation can be on the same physical server where SSAS is installed, it is more common to use a dedicated server.

You use the SQL Server Management Studio to administer OLTP databases, SSAS (OLAP) cubes and data mining models, and SSIS packages. The SSMS interface showing only the Object Explorer is shown in Figure 1-11.

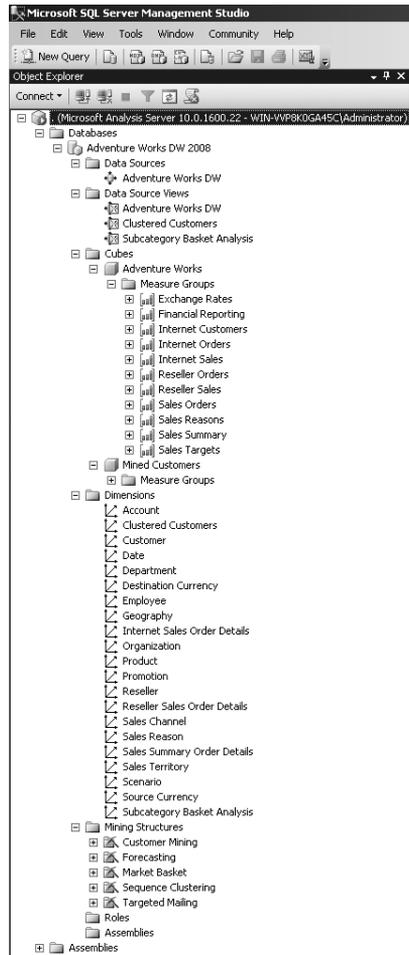


FIGURE 1-11 SSMS in SQL Server 2008

## SQL Server 2008 Integration Services

SSIS is a key component in most BI solutions. This toolset is used to import, cleanse, and validate data prior to making the data available to SSAS for OLAP cubes or data mining structures. It is typical to use data from many disparate sources (for example, relational, flat file, XML, and so on) as source data to a data warehouse. For this reason, a sophisticated toolset such as SSIS facilitates the complex data loads (ETL) that are common to BI solutions. The units of execution in SSIS are called *packages*. They are a type of XML files that you can consider to be a set of instructions that are designed using visual tools in BIDS. We discuss planning, implementation, and many other considerations for SSIS packages in Part III.

You'll use BIDS to design, develop, execute, and debug SSIS packages. The BIDS SSIS package design environment is shown in Figure 1-12.

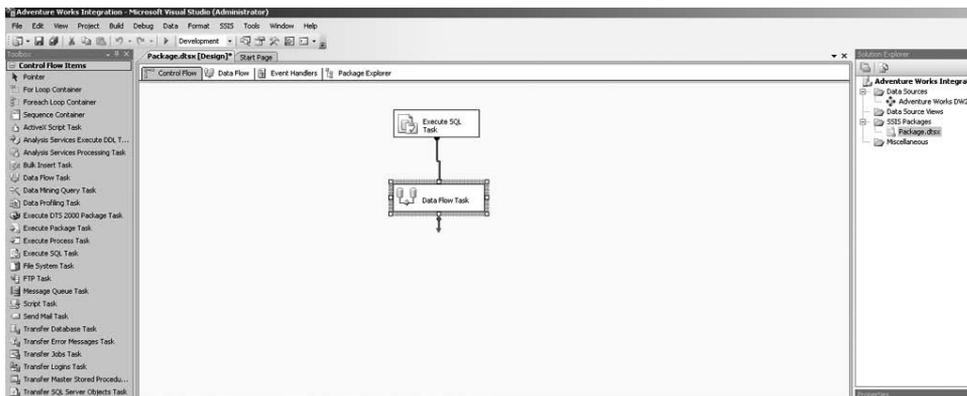


FIGURE 1-12 BIDS SSIS package designer

## Optional Components of a Microsoft BI Solution

In addition to the components that are included with SQL Server 2008, a number of other Microsoft products can be used as part of your BI solution. Most of these products allow you to deliver the reports generated from Analysis Services OLAP cubes and data mining structures in formats customized for different audiences, such as complex reports for business analysts and summary reports for executives.

Here is a partial list of Microsoft products that include integration with Analysis Services OLAP cubes and data mining models:

- **Microsoft Office Excel 2007** Many companies already own Office 2007, so using Excel as a BI client is often attractive for its low cost and relatively low training curve. In addition to being a client for SSAS OLAP cubes through the use of the PivotTable feature or the Data Mining Add-ins for SQL Server 2008, Excel can also be a client for data mining structures. (Note that connecting to an OLAP data source from Excel 2003 only does require that MS Query be installed. MS Query is listed under optional components on the Office installation DVD.)
- **Microsoft Word 2007** SSRS reports can be exported as Word documents that are compatible with Office 2003 or 2007.
- **Microsoft Visio 2007** Using the Data Mining Add-ins for SQL Server 2008, you can create customized views of data mining structures.
- **Microsoft Office SharePoint Server 2007** Office SharePoint Server 2007 contains both specialized Web site templates designed to show reports (with the most common one being the Report Center) as well as Web Parts that can be used to display individual reports on Office SharePoint Server 2007 Web pages. A Web Part is a pluggable user interface (UI) showing some bit of content. It is installed globally in the SharePoint

Portal Server Web site and can be added to an Office SharePoint Server 2007 portal Web page by any user with appropriate permissions.

- **Microsoft PerformancePoint Server** PerformancePoint Server allows you to quickly create a centralized Web site with all of your company's performance metrics. The environment is designed to allow business analysts to create sophisticated dashboards that are hosted in a SharePoint environment. These dashboards can contain SSRS reports and visualizations of data from OLAP cubes as well as other data sources. It also has a strong set of products that support business forecasting.

PerformancePoint Server includes the functionality of the Business Scorecard Manager and ProClarity Analytics Server. Its purpose is to facilitate the design and hosting of enterprise-level scorecards via rich data-visualization options such as charts and reports available in Reporting Services, Excel, and Visio. PerformancePoint Server also includes some custom visualizers, such as the Strategy Map.



**Note** We are sometimes asked what happened to ProClarity, a company that had provided a specialized client for OLAP cubes. Its target customer was the business analyst. Microsoft acquired ProClarity in 2006 and has folded features of its products into PerformancePoint Server.

Microsoft also offers other products—such as Dynamics, Project Server, and BizTalk Server—that use the Analysis Services storage mechanism and query engine. In addition to Microsoft products that are designed to integrate with the BI tools available in SQL Server 2008, you might elect to use some other developer products to improve productivity if your project's requirements call for .NET coding. Recall that the primary development tool is BIDS and that BIDS does not require a Visual Studio installation. Microsoft has found that BI developers are frequently also .NET developers, so most of them already have Visual Studio 2008. As was mentioned, in this situation, installing SSAS, SSIS, or SSRS installs the associated developer templates into the default Visual Studio installation.

Another consideration is the management of source code for large or distributed BI development teams. In this situation, you can also elect to add Visual Studio Team System (VSTS) for source control, automated testing, and architectural planning.

The data that you integrate into your BI solution might originate from relational sources. These sources can, of course, include SQL Server 2008. They can also include nearly any type of relational data—SQL Server (all versions), Oracle, DB2, Informix, and so forth. It is also common to include nonrelational data in BI solutions. Sources for this data can include Microsoft Access databases, Excel spreadsheets, and so forth. It is also common to include text data (often from mainframes). This data is sometimes made available as XML. This XML might or might not include schema and mapping information. If complex XML processing is part of your requirements, you can elect to use BizTalk Server to facilitate flexible mapping and loading of this XML data.

You might be thinking at this point, “Wow, that’s a big list! Am I required to buy (or upgrade to) all of those Microsoft products in order to implement a BI solution for my company?” The answer is no. The only service that is *required* for an OLAP BI solution is SSAS. Also, many companies provide tools that can be used as part of a Microsoft BI solution. Although we occasionally refer to some third-party products in this book, we’ll focus primarily on using Microsoft’s products and tools to build a BI solution.

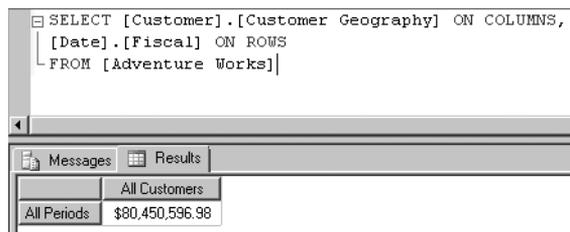
## Query Languages Used in BI Solutions

When working with BI solutions built on SSAS cubes and data mining structures, you use several query languages. The primary query language for OLAP cubes is MDX. SSAS also includes the ability to build data mining structures. To query the data in these structures, you use DMX. XMLA is a specialized administrative scripting language used with SSAS objects (OLAP cubes, SSIS packages, and data mining structures). Finally, RDL is the XML dialect behind SSRS reports. In the following sections, we briefly describe each language and provide a sample.

### MDX

MDX, which stands for *Multidimensional Expressions*, is the language used to query OLAP cubes. Although MDX is officially an open standard and some vendors outside of Microsoft have adopted parts of it in their BI products, the reality is that comparatively few working .NET developers are proficient in MDX. A mitigating factor is that the need to manually write MDX in a BI solution can be relatively small—with not nearly as much Transact-SQL as you would manually write for a typical OLTP database. However, retaining developers who have at least a basic knowledge of MDX is an important consideration in planning a BI project. We review core techniques as well as best practices for working with MDX in Chapter 10, “Introduction to MDX,” and Chapter 11, “Advanced MDX.”

The MDX query language is used to retrieve data from SSAS cubes. A simple MDX query is shown in Figure 1-13. Although MDX has an SQL-like structure, it is far more difficult to master because of the complexity of the SSAS source data structures—which are multidimensional OLAP cubes.



**FIGURE 1-13** A sample MDX query

## DMX

Data Mining Extensions (DMX) is used to query Analysis Services data mining structures. (We devote several future chapters to design and implementation of SSAS data mining structures.) Although this language is based loosely on Transact-SQL, it contains many elements that are unique to the world of data mining. As with MDX, very few working .NET developers are proficient in DMX. However, the need for DMX in BI solutions is relatively small because the SSAS data mining structure in BIDS provides tools and wizards that automatically generate DMX when you create those structures. Depending on the scope of your solution, retaining developers who have at least a basic knowledge of DMX might be an important consideration in planning a BI project that includes a large amount of data mining. A simple DMX query is shown in Figure 1-14.

| CHILDREN_CA | PARENT_UNIQ | NODE_DESCR           | NODE_RULE | MARGINAL_RU | NODE_PROBA       | MARGINAL_PR      | NODE_DISTRIBUTION | NODE_SUPPORT | MSOLAP_MOD | MSOLAP_NOD | MSOLAP_NOCE   |
|-------------|-------------|----------------------|-----------|-------------|------------------|------------------|-------------------|--------------|------------|------------|---------------|
| 10          |             | All                  |           |             | 1                | 1                | NODE_DISTRIBUTION | 19454        |            |            | Cluster Model |
| 0           | 000         | Number of Cars ...   |           |             | 0.14348831618... | 0.14348831618... | NODE_DISTRIBUTION | 2901         |            |            | Cluster 1     |
| 0           | 000         | 0 <= Number of ...   |           |             | 0.14629146580... | 0.14629146580... | NODE_DISTRIBUTION | 2578         |            |            | Cluster 2     |
| 0           | 000         | Number of Cars ...   |           |             | 0.13012643717... | 0.13012643717... | NODE_DISTRIBUTION | 2490         |            |            | Cluster 3     |
| 0           | 000         | 0 <= Number of ...   |           |             | 0.11759568012... | 0.11759568012... | NODE_DISTRIBUTION | 2154         |            |            | Cluster 4     |
| 0           | 000         | OccupationMa...      |           |             | 0.11282204136... | 0.11282204136... | NODE_DISTRIBUTION | 1985         |            |            | Cluster 5     |
| 0           | 000         | Yearly Income=...    |           |             | 0.10399101151... | 0.10399101151... | NODE_DISTRIBUTION | 1916         |            |            | Cluster 6     |
| 0           | 000         | Number of Cars ...   |           |             | 0.09682883345... | 0.09682883345... | NODE_DISTRIBUTION | 1810         |            |            | Cluster 7     |
| 0           | 000         | Yearly Income=...    |           |             | 0.06119756775... | 0.06119756775... | NODE_DISTRIBUTION | 1168         |            |            | Cluster 8     |
| 0           | 000         | Yearly Income=...    |           |             | 0.04649757628... | 0.04649757628... | NODE_DISTRIBUTION | 911          |            |            | Cluster 9     |
| 0           | 000         | 4 <= Total Childs... |           |             | 0.03120087579... | 0.03120087579... | NODE_DISTRIBUTION | 571          |            |            | Cluster 10    |

FIGURE 1-14 Sample DMX query

## XMLA

XML for Analysis (XMLA) is used to perform administrative tasks in Analysis Services. It is an XML dialect. Examples of XMLA tasks include viewing metadata, copying, backing up databases, and so on. As with MDX and DMX, this language is officially an open standard, and some vendors outside of Microsoft have chosen to adopt parts of it in their BI products. Again, the reality is that very few developers are proficient in XMLA. However, you will seldom author any XMLA from scratch; rather, you'll use the tools and wizards inside SQL Server 2008 to generate this metadata. In SSMS, when connected to SSAS, you can right-click on any SSAS object and generate XMLA scripts using the graphical user interface (GUI). XMLA is used to define SSAS OLAP cubes and data mining structures.

## RDL

RDL, or the Report Definition Language, is another XML dialect that is used to create Reporting Services reports. As with the other BI languages, RDL is officially an open standard,

and some vendors outside of Microsoft have chosen to adopt parts of it in their BI products. You rarely need to manually write RDL in a BI solution because it is generated for you automatically when you design a report using the visual tools in BIDS. We'll review core techniques as well as best practices for working with RDL in future chapters.

## Summary

In this chapter, we covered basic data warehousing terms and concepts, including BI, OLTP, OLAP, dimensions, and facts (or measures). We defined each term so that you can better understand the possibilities you should consider when planning a BI solution for your company.

We then introduced the BI tools included with SQL Server 2008. These include SSAS, SSIS, SSRS, and Data Mining Add-ins. For each of these BI tools, we defined what parts of a BI solution's functionality that particular tool could provide. Next, we discussed other Microsoft products that are designed to be integrated with BI solutions built using SSAS OLAP cubes or data mining structures. These included Excel, Word, and Office SharePoint Server 2007. We also touched on the integration of SSAS into PerformancePoint Server. We concluded our conceptual discussion with a list and description of the languages involved in BI projects.

In Chapter 2, we work with the sample database AdventureWorksDW, which is included in SQL Server 2008, so that you get a quick prototype SSAS OLAP cube and data mining structure up and running. This is a great way to begin turning the conceptual knowledge you've gained from reading this chapter into practical understanding.



## Chapter 6

# Understanding SSAS in SSMS and SQL Server Profiler

We covered quite a bit of ground in the preceding five chapters—everything from the key concepts of business intelligence to concepts, languages, processes, and modeling methodologies. By now, we’re sure you’re quite ready to roll up your sleeves and get to work in the Business Intelligence Development Studio (BIDS) interface. Before you start, though, we have one more chapter’s worth of information. In this chapter, we’ll explore the ins and outs of some tools you’ll use when working with Microsoft SQL Server Analysis Services (SSAS) objects. After reading this chapter, you’ll be an expert in not only SQL Server Profiler but also SQL Server Management Studio (SSMS) and other tools that will make your work in SSAS very productive. We aim to provide valuable information in this chapter for all SSAS developers—from those of you who are new to the tools to those who have some experience. If you’re wondering when we’re going to get around to discussing BIDS, we’ll start that in Chapter 7, “Designing OLAP Cubes Using BIDS.”

## Core Tools in SQL Server Analysis Services

We’ll begin this discussion of the tools you’ll use to design, create, populate, secure, and manage OLAP cubes by taking an inventory. That is, we’ll first list all the tools that are part of SQL Server 2008. Later in this chapter, we’ll look at useful utilities and other tools you can get for free or at low cost that are *not* included with SQL Server 2008. We mention these because we’ve found them to be useful for production work. Before we list our inventory, we’ll talk a bit about the target audience—that is, who Microsoft thinks will use these tools. We share this information so that you can choose the tools that best fit your style, background, and expectations.

SQL Server 2008 does not install SSAS by default. When you install SSAS, several tools are installed with the SSAS engine and data storage mechanisms. Also, an SSAS installation does not require that you install SQL Server Database Engine Services. You’ll probably want to install SQL Server Database Engine Services, however, because some of the tools that install with it are useful with SSAS cubes. SQL Server 2008 installation follows the minimum-installation paradigm, so you’ll probably want to verify which components you’ve installed before exploring the tools for SSAS. To come up with this inventory list, follow these steps:

1. Run Setup.exe from the installation media.

2. On the left side of the Installation Center screen, click Installation and then select New SQL Server Stand-Alone Installation Or Add Features To An Existing Installation on the resulting screen.
3. Click OK after the system checks complete, and click the Install button on the next screen.
4. Click Next on the resulting screen. Then select the Add Features To An Existing Instance Of SQL Server 2008 option, and select the appropriate instance of SQL Server from the list. After you select the particular instance you want to verify, click Next to review the features that have been installed for this instance.

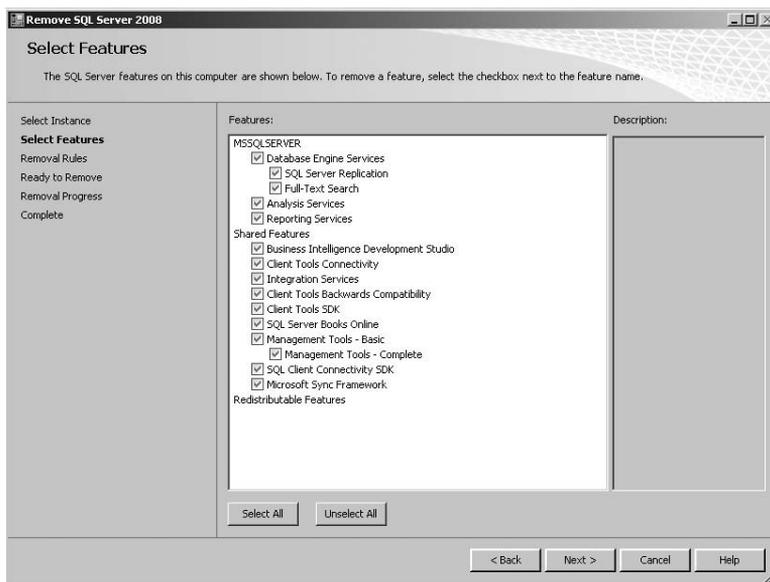
### Who Is an SSAS Developer?

Understanding the answer to this question will help you to understand the broad variety of tools available with SSAS OLAP cubes in SQL Server 2008. Unlike classic .NET developers, SSAS developers are people with a broad and varied skill set. Microsoft has provided some tools for developers who are comfortable writing code and other tools for developers who are not. In fact, the bias in SSAS development favors those who prefer to create objects by using a graphical user interface rather than by writing code. We mention this specifically because we've seen traditional application developers, who are looking for a code-first approach, become frustrated with the developer interface Microsoft has provided.

In addition to providing a rich graphical user interface in most of the tools, Microsoft has also included wizards to further expedite the most commonly performed tasks. In our experience, application developers who are familiar with a code-first environment often fail to take the time to understand and explore the development environments available for SSAS. This results in frustration on the part of the developers and lost productivity on business intelligence (BI) projects.

We'll start by presenting information at a level that assumes you've never worked with any version of Microsoft Visual Studio or Enterprise Manager (or Management Studio) before. Even if you have experience in one or both of these environments, you might still want to read this section. Our goal is to maximize your productivity by sharing our tips, best practices, and lessons learned.

Note in Figure 6-1 that some components are shared to the SQL Server 2008 instance, but others install only when a particular component is installed. As mentioned in previous chapters, SQL Server 2008 no longer ships with sample databases. If you want to install the AdventureWorks OLTP and OLAP samples, you must download them from CodePlex. For instructions on where to locate these samples and how to install them, see Chapter 1, "Business Intelligence Basics."



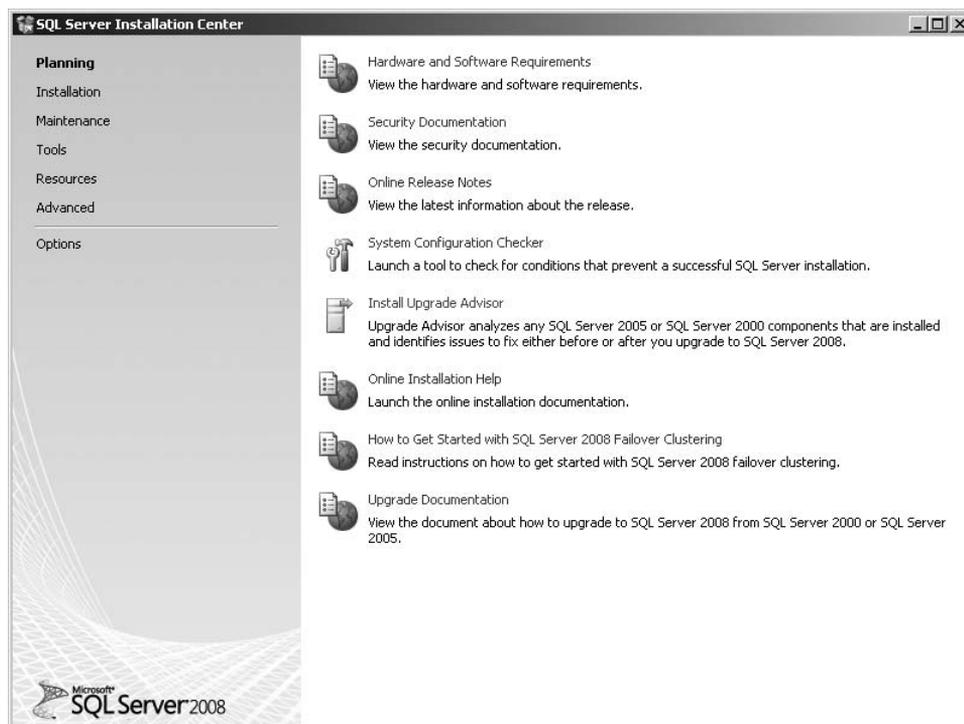
**FIGURE 6-1** Installed features are shown on the Select Features page.

After you've verified that everything you expected to be installed is actually installed in the particular instance of SSAS, you're ready to start working with the tools. A variety of tools are included; however, you'll do the majority of your design and development work in just one tool—BIDS.

For completeness, this is the list of tools installed with the various SQL Server components:

- **Import/Export Wizard** Used to import/export data and to perform simple transformations
- **Business Intelligence Development Studio** Primary SSAS, SSIS, and SSRS development environment
- **SQL Server Management Studio** Primary SQL Server (all components) administrative environment
- **SSAS Deployment Wizard** Used to deploy SSAS metadata (\*.asdatabase) from one server to another
- **SSRS Configuration Manager** Used to configure SSRS
- **SQL Server Configuration Manager** Used to configure SQL Server components, including SSAS
- **SQL Server Error and Usage Reporting** Used to configure error/usage reporting—that is, to specify whether or not to send a report to Microsoft

- **SQL Server Installation Center** New information center, as shown in Figure 6-2, which includes hardware and software requirements, baseline security, and installed features and samples
- **SQL Server Books Online** Product documentation
- **Database Engine Tuning Advisor** Used to provide performance tuning recommendations for SQL Server databases
- **SQL Server Profiler** Used to capture activity running on SQL Server components, including SSAS
- **Report Builder 2.0** Used by nondevelopers to design SSRS reports. This is available as a separate download and is not on the SQL Server installation media.



**FIGURE 6-2** The SQL Server Installation Center

A number of GUI tools are available with a complete SQL Server 2008 installation. By *complete*, we mean that all components of SQL Server 2008 are installed on the machine. A full installation is not required, or even recommended, for production environments. The best practice for production environments is to create a *reduced attack surface* by installing only the components and tools needed to satisfy the business requirements. In addition, you should secure access to powerful tools with appropriate security measures. We'll talk more about security in general later in this chapter, and we'll describe best practices for locking

down tools in production environments. For now, we'll stay in an exploratory mode by installing everything and accessing it with administrator privileges to understand the capabilities of the various tools.

Because BIDS looks like the Visual Studio interface, people often ask us if an SSAS instance installation requires a full Visual Studio install. The answer is no. If Visual Studio is *not* installed on a machine with SSAS, BIDS, which is subset of Visual Studio, installs. If Visual Studio is installed, the BIDS project templates install inside of the Visual Studio instance on that machine.

The core tools you'll use for development of OLAP cubes and data mining structures in SSAS are BIDS, SSMS, and SQL Server Profiler. Before we take a closer look at these GUI tools, we'll mention a couple of command-line tools that are available to you as well.

In addition to the GUI tools, several command-line tools are installed when you install SQL Server 2008 SSAS. You can also download additional free tools from CodePlex. One tool available on the CodePlex site is called BIDS Helper, which you can find at <http://www.codeplex.com/bidshelper>. It includes many useful features for SSAS development. You can find other useful tools on CodePlex as well. We'll list only a couple of the tools that we've used in our projects:

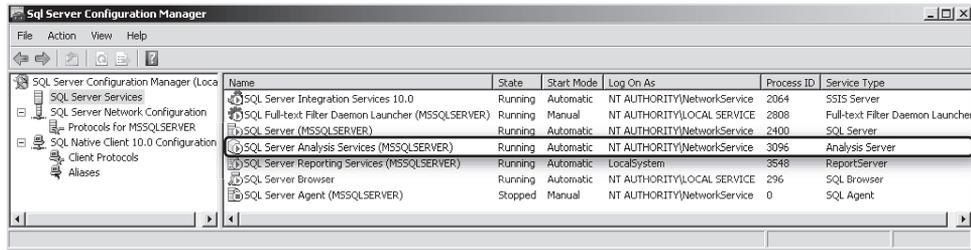
- **ascmd.exe** Allows you to run XMLA, DMX, or DMX scripts from the command prompt (available at <http://www.codeplex.com/MSFTASProdSamples>)
- **SQLPS.exe** Allows you to execute Transact-SQL via the Windows PowerShell command line—mostly used when managing SQL Server source data for BI projects.

As we mentioned, you'll also want to continue to monitor CodePlex for new community-driven tools and samples. Contributors to CodePlex include both Microsoft employees and non-Microsoft contributors.

## Baseline Service Configuration

Now that we've seen the list of tools, we'll take a look at the configuration of SSAS. The simplest way to do this is to use SQL Server Configuration Manager. In Figure 6-3, you can see that on our demo machine, the installed instance of SSAS is named MSSQLSERVER and its current state is set to Running. You can also see that the Start Mode is set to Automatic. The service log on account is set to NT AUTHORITY\Network Service. Of course, your settings may vary from our defaults.

Although you can also see this information using the Control Panel Services item, it's recommended that you view and change any of this information using the SQL Server Configuration Manager. The reason for this is that the latter tool properly changes associated registry settings when changes to the service configuration are made. This association is not necessarily observed if configuration changes are made using the Control Panel Services item.



**FIGURE 6-3** SQL Server Configuration Manager

The most important setting for the SSAS service itself is the Log On (or service) account. You have two choices for this setting. You can select one of three built-in accounts: Local System, Local Service, or Network Service. If you do not do that, you can use an account that has been created specifically for this purpose either locally or on your domain. Figure 6-4 shows the dialog box in SQL Server Configuration Manager where you set this. Which one of these choices is best and why?

Our answer depends on which environment you're working in. If you're exploring or setting up a development machine in an isolated domain, or as a stand-alone server, you can use any account. As we show in Figures 6-3 and 6-4, we usually just use a local account that has been added to the local administrator's group for this purpose. We do remind you that this circumvention of security is appropriate only for *nonproduction* environments, however.



**FIGURE 6-4** The SSAS service account uses a Log On account.

SQL Server Books Online contains lots of information about log-on accounts. You'll want to review the topics "Setting Up Windows Service Accounts" and "Choosing the Service

Account” for details on exactly which permissions and rights are needed for your particular service (user) account.

We’ll distill the SQL Server Books Online information down a bit because, in practice, we’ve seen only two configurations. Most typically, we see our clients use either a local or domain lower-privileged (similar to a regular user) account. Be aware that for SSAS-only installations, the ability to use a domain user account as the SSAS logon account is disabled. One important consideration specific to SSAS is that the service logon account information is used to encrypt SSAS connection strings and passwords. This is a further reason to use an isolated, monitored, unique, low-privileged account.

## Service Principal Names

What is a Service Principal Name (SPN)? An SPN is a particular type of Domain Name System (DNS) record. When you associate a service account with SSAS at the time of installation, an SPN record is created. If your SSAS server is part of a domain, this record is stored in your domain DNS database. It’s required for some authentication scenarios (particular client tools). If you change the service account for SSAS, you must delete the original SPN and create a new SPN record for DNS. You can do this with the setSPN.exe tool available from the Windows Server Resource Kit.

Here’s further guidance from SQL Server Books Online:

*“Service SIDs are available in SQL Server 2008 on Windows Server 2008 and Windows Vista operating systems to allow service isolation. Service isolation provides services a way to access specific objects without having to either run in a high-privilege account or weaken the object’s security protection. A SQL Server service can use this identity to restrict access to its resources by other services or applications. Use of service SIDs also removes the requirement of managing different domain accounts for various SQL Server services.*

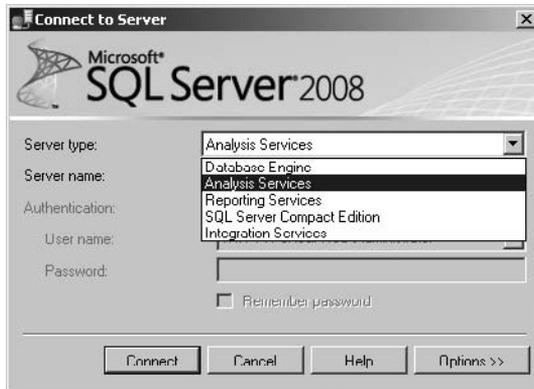
*A service isolates an object for its exclusive use by securing the resource with an access control entry that contains a service security ID (SID). This ID, referred to as a per-service SID, is derived from the service name and is unique to that service. After a SID has been assigned to a service, the service owner can modify the access control list for an object to allow access to the SID. For example, a registry key in HKEY\_LOCAL\_MACHINE\SOFTWARE would normally be accessible only to services with administrative privileges. By adding the per-service SID to the key’s ACL, the service can run in a lower-privilege account, but still have access to the key.”*

Now that you’ve verified your SSAS installation and checked to make sure the service was configured correctly and is currently running, it’s time to look at some of the tools you’ll use to work with OLAP objects. For illustration, we’ve installed the AdventureWorks DW2008 sample OLAP project found on CodePlex, because we believe it’s more meaningful to explore

the various developer surfaces with information already in them. In the next chapter, we'll build a cube from start to finish. So if you're already familiar with SSMS and SQL Server Profiler, you might want to skip directly to that chapter.

## SSAS in SSMS

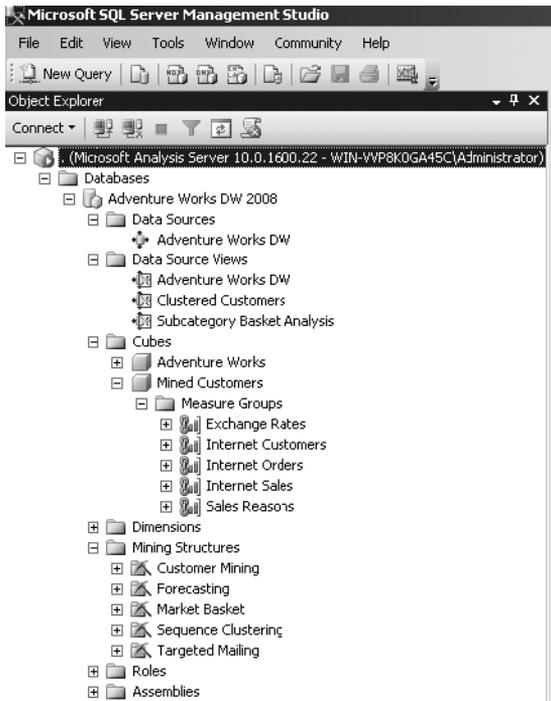
Although we don't believe that the primary audience of this book is administrators, we do choose to begin our deep dive into the GUI tools with SSMS. SQL Server Management Studio is an administrative tool for SQL Server relational databases, SSAS OLAP cubes and data mining models, SSIS packages, SSRS reports, and SQL Server Compact edition data. The reason we begin here is that we've found the line between SSAS developer and administrator to be quite blurry. Because of a general lack of knowledge about SSAS, we've seen many an SSAS developer being asked to perform administrative tasks for the OLAP cubes or data mining structures that have been developed. Figure 6-5 shows the connection dialog box for SSMS.



**FIGURE 6-5** SSMS is the unified administrative tool for all SQL Server 2008 components.

After you connect to SSAS in SSMS, you are presented with a tree-like view of all SSAS objects. The top-level object is the server, and databases are next. Figure 6-6 shows this tree view in Object Explorer. An OLAP database object is quite different than a relational database object, which is kept in SQL Server's RDBMS storage. Rather than having relational tables, views, and stored procedures, an OLAP database consists of data sources, data source views, cubes, dimensions, mining structures, roles, and assemblies. All of these core object types are represented by folders in the Object Explorer tree view. These folders can contain child objects as well, as shown in Figure 6-6 in the Measure Groups folder that appears under a cube in the Cubes folder.

So what are all of these objects? Some should be familiar to you based on our previous discussions of OLAP concepts, including cubes, dimensions, and mining structures. These are the basic storage units for SSAS data. You can think of them as somewhat analogous to relational tables and views in that respect, although structurally, OLAP objects are *not* relational but multidimensional.



**FIGURE 6-6** Object Explorer lists all SSAS objects in a tree-like structure.

Data sources represent connections to source data. We'll be exploring them in more detail in this chapter and the next one. Data source views are conceptually similar to relational views in that they represent a view of the data from one or more defined data sources in the project. Roles are security groups for SSAS objects. Assemblies are .NET types to be used in your SSAS project—that is, they have been written in a .NET language and compiled as .dlls.

The next area to explore in SSMS is the menus. Figure 6-7 shows both the menu and standard toolbar. Note that the standard toolbar displays query types for all possible components—that is, relational (Transact-SQL) components, multidimensional OLAP cubes (MDX), data mining structures (DMX), administrative metadata for OLAP objects (XMLA), and SQL Server Compact edition.



**FIGURE 6-7** The SSMS standard toolbar displays query options for all possible SQL Server 2008 components.

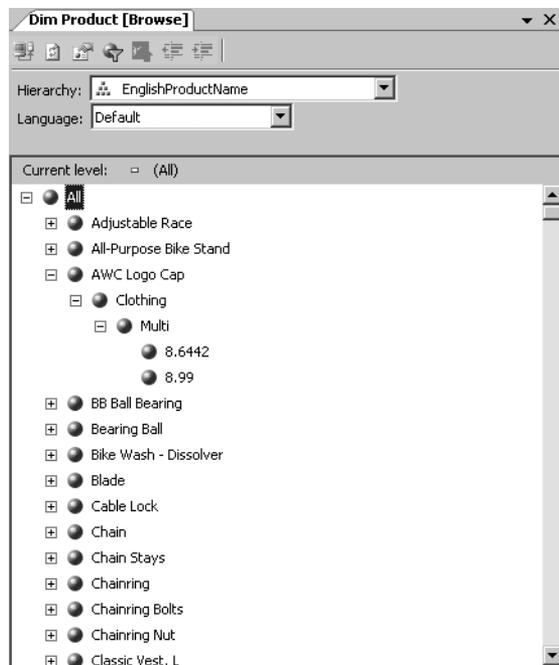
It's important that you remember the purpose of SSMS—administration. When you think about this, the fact that it's straightforward to view, query, and configure SSAS objects—but more complex to create them—is understandable. You primarily use BIDS to create OLAP objects. Because this is a GUI environment, you're also provided with guidance should you

want to examine or query objects. Another consideration is that SSMS is *not* an end-user tool. Even though the viewers are sophisticated, SSMS is designed for SSAS administrators.

## How Do I View OLAP Objects?

SSMS includes many object viewers. You'll see these same viewers built into other tools designed to work with SSAS, such as BIDS. You'll also find versions of these viewers built into client tools, such as Microsoft Office Excel 2007. The simplest and fastest way to explore cubes and mining models in SSMS is to locate the object in the tree view and then to right-click on it. For cubes, dimensions, and mining structures, the first item on the shortcut menu is Browse.

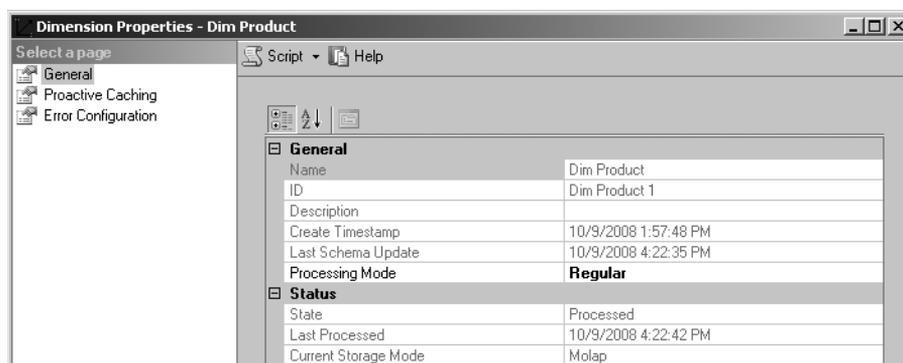
We'll begin our exploration with the Product dimension. Figure 6-8 shows the results of browsing the Product dimension. For each dimension, we have the ability to drill down to see the member names at the defined levels—in this case, at the category, subcategory, and individual item levels. In addition to being able to view the details of the particular dimensional (rollup) hierarchy, we can also select a localization (language) and member properties that might be associated with one or more levels of a dimension. In our example, we have elected to include color and list price in our view for the AWC Logo Cap clothing item. These member properties have been associated with the item (bottom) level of the product dimension.



**FIGURE 6-8** The dimension browser enables you to view the data in a dimension.

The viewing options available for dimensions in SSMS include the ability to filter and implement dimension writeback. Writeback has to be enabled on the particular dimension, and the connected user needs to have dimension writeback permission to be permitted to use this action in SSMS.

In addition to being able to view the dimension information, you can also see some of the metadata properties by clicking Properties on the shortcut menu. Be aware that you're viewing a small subset of structural properties in SSMS. As you would expect, these properties are related to administrative tasks associated with a dimension. Figure 6-9 shows the general dialog box of the Product dimension property page. Note that the only setting you can change in this view is Processing Mode. We'll examine the various processing modes for dimensions and the implications of using particular selections in Chapter 9, "Processing Cubes and Dimensions."



**FIGURE 6-9** The Dimension Properties dialog box in SSMS shows administrative properties associated with a dimension.

In fact, you can process OLAP dimensions, cubes, and mining structures in SSMS. You do this by right-clicking on the object and then choosing Process on the shortcut menu. Because this topic requires more explanation, we'll cover it in Chapter 9. Suffice it to say at this point that, from a high level, SSAS object processing is the process of copying data from source locations into destination containers and performing various associated processing actions on this data as part of the loading process. As you might expect, these processes can be complex and require that you have an advanced understanding of the SSAS objects before you try to implement the objects and tune them. For this reason, we'll explore processing in Part II of this book.

If you're getting curious about the rest of the metadata associated with a dimension, you can view this information in SSMS as well. This task is accomplished by clicking on the shortcut menu option Script Dimension As, choosing Create To, and selecting New Query Editor Window. The results are produced as pure XMLA script. You'll recall from earlier in the book that XMLA is a dialect of XML.

What you're looking at is a portion of the XMLA script that is used to define the structure of the dimension. Although you can use Notepad to create SSAS objects, because they are entirely based on an XMLA script, you'll be much more productive using the graphical user interface in BIDS to generate this metadata script. The reason you can generate XMLA in SSMS is that when you need to re-create OLAP objects, you need the XMLA to do so. So XMLA is used to copy, move, and back up SSAS objects. In fact, you can execute the XMLA query you've generated using SSMS. We take a closer look at querying later in this chapter.

Now that you've seen how to work with objects, we'll simply repeat the pattern for OLAP cubes and data mining structures. That is, we'll first view the cube or structure using the Browse option, review configurable administrative properties, and then take a look at the XMLA that is generated. We won't neglect querying either. After we examine browsing, properties, and scripting for cubes and models, we'll look at querying the objects using the appropriate language—MDX, DMX, or XMLA.

## How Do I View OLAP Cubes?

The OLAP cube browser built into SSMS is identical to the one you'll be working with in BIDS when you're developing your cubes. It's a sophisticated pivot table–style interface. The more familiar you become with it, the more productive you'll be. Just click Browse on the shortcut menu after you've selected any cube in the Object Explorer in SSMS to get started. Doing this presents you with the starter view. This view includes the use of hint text (such as Drop Totals Of Detail Field Fields Here) in the center work area that helps you understand how best to use this browser.

On the left side of the browser, you're presented with another object browser. This is where you select the items (or aspects) of the cube you want to view. You can select measures, dimension attributes, levels, or hierarchies. Note that you can select a particular measure as a filter from the drop-down list box at the top of this object browser. Not only will this filter the measures selected, it will also filter the associated dimensions so that you're selecting from an appropriate subset as you build your view.

Measures can be viewed in the Totals work area. Dimension attributes, levels, or hierarchies can be viewed on the Rows, Columns, or Filters (also referred to as *slicers*) axis. These axes are labeled with the hint text Drop xxx Fields Here. We'll look at Filters or Slicers axes in more detail later in this chapter.

At the top of the browser, you can select a perspective. A perspective is a defined view of an OLAP cube. You can also select a language. Directly below that is the Filter area, where you can create a filter expression (which is actually an MDX expression) by dragging and dropping a dimension level or hierarchy into that area and then completing the rest of the information—that is, configuring the Hierarchy, Operator, and Filter Expression options. We'll be demonstrating this shortly. To get started, drag one or more measures and a couple of dimensions to the Rows and Columns axes. We'll do this and show you our results in Figure 6-10.

To set up our first view, we filtered our list by the Internet Sales measure group in the object browser. Next we selected Internet Sales Amount and Internet Order Quantity as our measures and dragged them to that area of the workspace. We then selected the Product Categories hierarchy of the Product dimension and dragged it to the Rows axis. We also selected the Sales Territory hierarchy from the Sales Territory dimension and dragged it to the Columns axis. We also selected the Sales Territory hierarchy from the Sales Territory dimension and dragged it to the Columns axis.

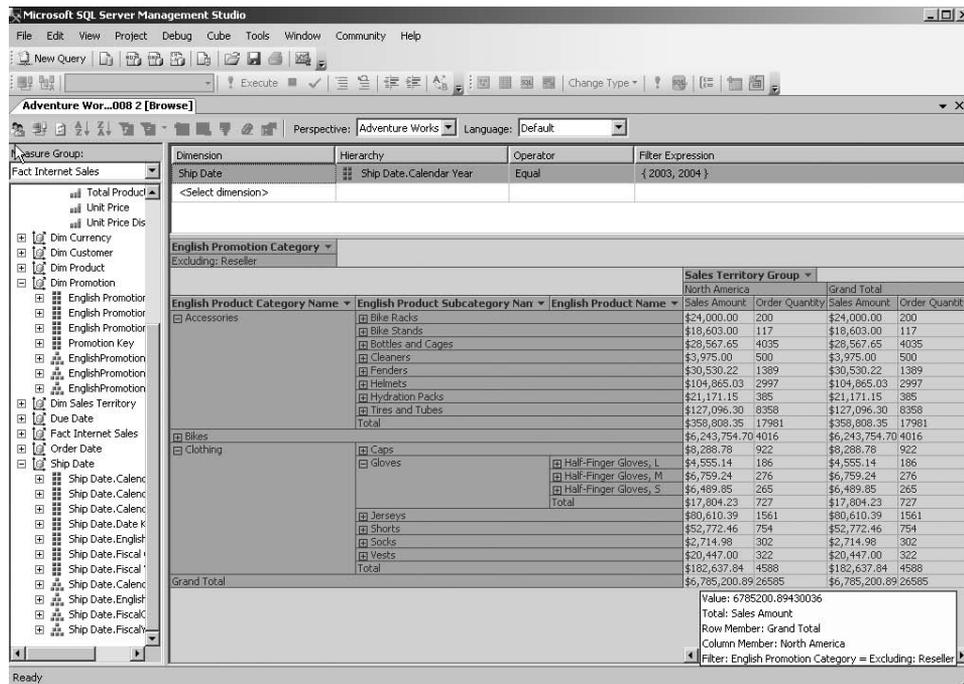
We drilled down to show detail for the Accessories product category and Gloves subcategory under the Clothing product category on the Rows axis. And finally, we filtered the Sales Territory Group information to hide the Pacific region. The small blue triangle next to the Group label indicates that a filter has been applied to this data. If you want to remove any item from the work area, just click it and drag it back to the left side (list view). Your cursor will change to an X, and the item will be removed from the view.

It's much more difficult to write the steps as we just did than to actually do them! And that is the point. OLAP cubes, when correctly designed, are quick, easy, and intuitive to query. What you're actually doing when you're visually manipulating the pivot table surface is generating MDX queries. The beauty of this interface is that end users can do this as well. Gone are the days that new query requests of report systems require developers to rewrite (and tune) database queries.

|                               |                                  |                      | Sales Territory Group |                | Grand Total    |                |
|-------------------------------|----------------------------------|----------------------|-----------------------|----------------|----------------|----------------|
| English Product Category Name | English Product Subcategory Name | English Product Name | Sales Amount          | Order Quantity | Sales Amount   | Order Quantity |
| Accessories                   | Bike Racks                       |                      | \$24,000.00           | 200            | \$24,000.00    | 200            |
|                               | Bike Stands                      |                      | \$18,603.00           | 117            | \$18,603.00    | 117            |
|                               | Bottles and Cages                |                      | \$28,567.65           | 4035           | \$28,567.65    | 4035           |
|                               | Cleaners                         |                      | \$3,975.00            | 500            | \$3,975.00     | 500            |
|                               | Fenders                          |                      | \$30,530.22           | 1389           | \$30,530.22    | 1389           |
|                               | Helmets                          |                      | \$104,865.03          | 2997           | \$104,865.03   | 2997           |
|                               | Hydration Packs                  |                      | \$21,171.15           | 385            | \$21,171.15    | 385            |
|                               | Tires and Tubes                  |                      | \$128,087.87          | 8791           | \$128,087.87   | 8791           |
|                               | Total                            |                      | \$359,799.92          | 18414          | \$359,799.92   | 18414          |
| Bikes                         |                                  |                      | \$10,821,161.9        | 5831           | \$10,821,161.9 | 5831           |
| Clothing                      | Caps                             |                      | \$9,288.78            | 922            | \$9,288.78     | 922            |
|                               | Gloves                           |                      | \$6,024.54            | 246            | \$6,024.54     | 246            |
|                               | Half-Finger Gloves, L            |                      | \$6,759.24            | 276            | \$6,759.24     | 276            |
|                               | Half-Finger Gloves, M            |                      | \$6,499.85            | 265            | \$6,499.85     | 265            |
|                               | Half-Finger Gloves, S            |                      | \$19,273.63           | 787            | \$19,273.63    | 787            |
|                               | Jerseys                          |                      | \$82,959.92           | 1608           | \$82,959.92    | 1608           |
|                               | Shorts                           |                      | \$52,772.46           | 754            | \$52,772.46    | 754            |
|                               | Socks                            |                      | \$3,930.74            | 325            | \$3,930.74     | 325            |
|                               | Vests                            |                      | \$20,447.00           | 322            | \$20,447.00    | 322            |
|                               | Total                            |                      | \$186,672.53          | 4719           | \$186,672.53   | 4719           |
| Grand Total                   |                                  |                      | \$11,367,634.3        | 28964          | \$11,367,634.3 | 28964          |

FIGURE 6-10 Building an OLAP cube view in SSMS

Let's add more sophistication to our view. To do this, we'll use the filter and slicer capabilities of the cube browser. We'll also look at the pivot capability and use the built-in common queries. To access the latter, you can simply right-click on a measure in the measures area of the designer surface and select from a shortcut menu, which presents you with common queries, such as Show Top 10 Values and other options as well. Figure 6-11 shows our results.



**FIGURE 6-11** Results of building an OLAP cube view in SSMS

Here are the steps we took to get there.

First we dragged the Promotions hierarchy from the Promotion dimension to the slicer (Filter Fields) area. We then set a filter by clearing the check boxes next to the Reseller promotion dimension members. This resulted in showing data associated only with the remaining members. Note that the label indicates this as well by displaying the text "Excluding: Reseller."

We then dragged the Ship Date.Calendar Year hierarchy from the Ship Date dimension; we set the Operator area to Equal, and in the Filter Expression area we chose the years 2003 and 2004 from the available options. Another area to explore is the nested toolbar inside of the Browser subtab. Using buttons on this tab toolbar, you can connect as a different user and sort, filter, and further manipulate the data shown in the working pivot table view. Note that there is an option to show only the top or bottom values (1, 2, 5, 10, or 25 members or a percentage). Finally, if drillthrough is enabled for this cube, you can drill through using this browser by right-clicking on a data cell and selecting that option. Drillthrough allows you

to see additional columns of information that are associated with the particular fact item (or measure) that you've selected. You should spend some time experimenting with all the toolbar buttons so that you're thoroughly familiar with the different built-in query options. Be aware that each time you select an option, you're generating an MDX query to the underlying OLAP cube.

Note also that when you select cells in the grid, additional information is shown in a tooltip. You can continue to manipulate this view for any sort of testing purposes. Possible actions also include pivoting information from the rows to the column's axis, from the slicer to the filter, and so on. Conceptually, you can think of this manipulation as somewhat similar to working with a Rubik's cube. Of course, OLAP cubes generally contain more than three dimensions, so this analogy is just a starting point.

## Viewing OLAP Cube Properties and Metadata

If you next want to view the administrative properties associated with the particular OLAP cube that you're working with (as you did for dimensions), you simply right-click that cube in the SSMS Object Browser and then click Properties. Similar to what you saw when you performed this type of action on an OLAP dimension, you'll then see a dialog box similar to the one shown in Figure 6-12 that allows you to view some properties. The only properties you can change in this view are those specifically associated with cube processing. As mentioned previously, we'll look at cube processing options in more detail in Chapter 9.

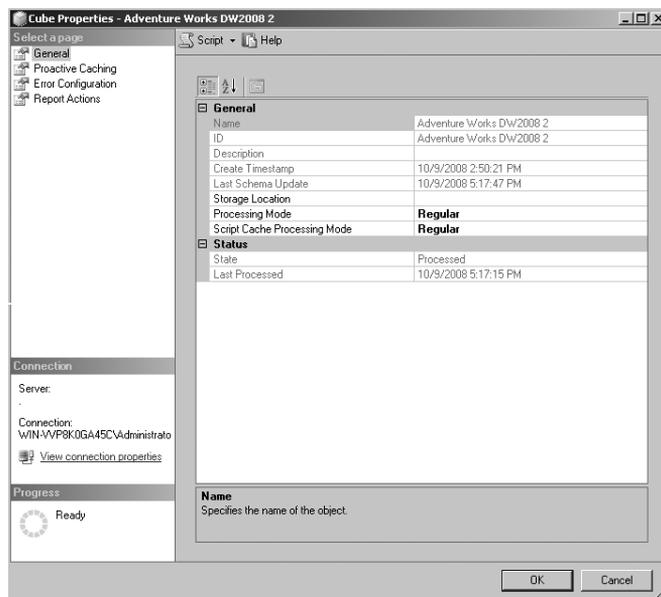


FIGURE 6-12 OLAP cube properties in SSMS

By now, you can probably guess how you'd generate an XMLA metadata script for an OLAP cube in SSMS. Just right-click the cube in the object browser and click Script Cube As on the shortcut menu, choose Create To, and select New Query Editor Window. Note also that you can generate XMLA scripts from inside any object property window. You do this by clicking the Script button shown at the top of Figure 6-12.

Now that we've looked at both OLAP dimensions and cubes in SSMS, it's time to look at a different type of object—SSAS data mining structures. Although conceptually different, data mining (DM) objects are accessed using methods identical to those we've already seen—that is, browse, properties, and script.

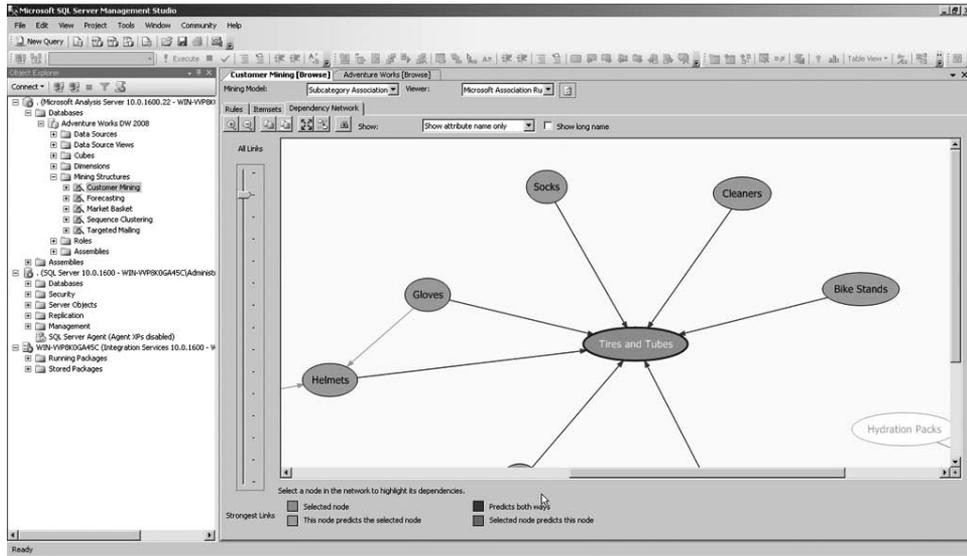
## How Do I View DM Structures?

As we begin our tour of SSAS data mining structures, we need to remember a couple of concepts that were introduced earlier in this book. Data mining structures are containers for one or more data mining models. Each data mining model uses a particular data mining algorithm. Each data mining algorithm has one or more data mining algorithm viewers associated with it. Also, each data mining model can be viewed using a viewer as well via a lift chart. New to SQL Server 2008 is the ability to perform cross validation. Because many of these viewing options require more explanation about data mining structures, at this point we're going to stick to the rhythm we've established in this chapter—that is, we'll look at a simple view, followed by the object properties, and then the XMLA. Because the viewers are more complex for data mining objects than for OLAP objects, we'll spend a bit more time exploring.

We'll start by browsing the Customer Mining data mining structure. Figure 6-13 shows the result. What you're looking at is a rendering of the Customer Clusters data mining model, which is part of the listed structure. You need to select the Cluster Profiles tab to see the same view. Note that you can make many adjustments to this browser, such as legend, number of histogram bars, and so on. At this point, some of the viewers won't make much sense to you unless you have a background using data mining. Some viewers are more intuitive than others. We'll focus on showing those in this section.

It's also important for you to remember that although these viewers are quite sophisticated, SSMS is *not* an end-user client tool. We find ourselves using the viewers in SSMS to demonstrate proof-of-concept ideas in data mining to business decision makers (BDMs), however. If these viewers look familiar to you, you've retained some important information that we presented in Chapter 2, "Visualizing Business Intelligence Results." These viewers are nearly identical to the ones that *are* intended for end users as part of the SQL Server 2008 Data Mining Add-ins for Office 2007. When you install the free add-ins, these data mining viewers become available as part of the Data Mining tab on the Excel 2007 Ribbon. Another consideration for you is this—similar to the OLAP cube pivot table viewer control in SSMS that we just finished looking at, these data mining controls are also part of BIDS.





**FIGURE 6-14** Data mining structure viewer in SSMS showing the Dependency Network view for the Microsoft Association algorithm



**Tip** You can change any of the default color schemes for the data mining viewers in SSMS by adjusting the colors via Tools, Options, Designers, Analysis Services Designers, Data Mining Viewers.

Because the processes for viewing the data mining object administrative properties and for generating an XMLA script of the object's metadata are identical to those used for OLAP objects, we won't spend any more time reviewing them here.

## How Do You Query SSAS Objects?

As with relational data, you have the ability to write and execute queries against multidimensional data in SSMS. This is, however, where the similarity ends. The reason is that when you work in an RDBMS, you need to write any query to the database using SQL. Even if you generate queries using tools, you'll usually choose to perform manual tuning of those queries. Tuning steps can include rewriting the SQL, altering the indexing on the involved tables, or both.

SSAS objects can and sometimes are queried manually. However, the extent to which you'll choose to write manual queries will be considerably less than the extent to which you'll query relational sources. What are the reasons for this? There are several:

- MDX and DMX language expertise is rare among the developer community. With less experienced developers, the time to write and optimize queries manually can be prohibitive.

- OLAP cube data is often delivered to end users via pivot table–type interfaces (that is, Excel, or some manual client that uses a pivot table control). These interfaces include the ability to generate MDX queries by dragging and dropping members of the cube on the designer surface—in other words, by visual query generation.
- SSMS and BIDS have many interfaces that also support the idea of visual query generation for both MDX and DMX. This feature is quite important to developer productivity.

What we’re saying here is that although you can create manual queries, and SSMS is the place to do this, you’ll need to do this significantly less frequently while working with SSAS objects (compared to what you have been used to with RDBMS systems). It’s very important for you to understand and embrace this difference. Visual development does *not* mean *lack* of sophistication or power in the world of SSAS.

As you move toward understanding MDX and DMX, we suggest that you first monitor the queries that SSMS generates via the graphical user interface. SQL Server Profiler is an excellent tool to use when doing this.

## What Is SQL Server Profiler?

SQL Server Profiler is an activity capture tool for the database engine and SSAS that ships with SQL Server 2008. SQL Server Profiler broadly serves two purposes. The first is to monitor activity for auditing or security purposes. To that end, SQL Server Profiler can be easily configured to capture login attempts, access specific objects, and so on. The other main use of the tool is to monitor activity for performance analysis. SQL Server Profiler is a powerful tool—when used properly, it’s one of the keys to understanding SSAS activity. We caution you, however, that SQL Server Profiler can cause significant overhead on production servers. When you’re using it, you should run it on a development server or capture only essential information.

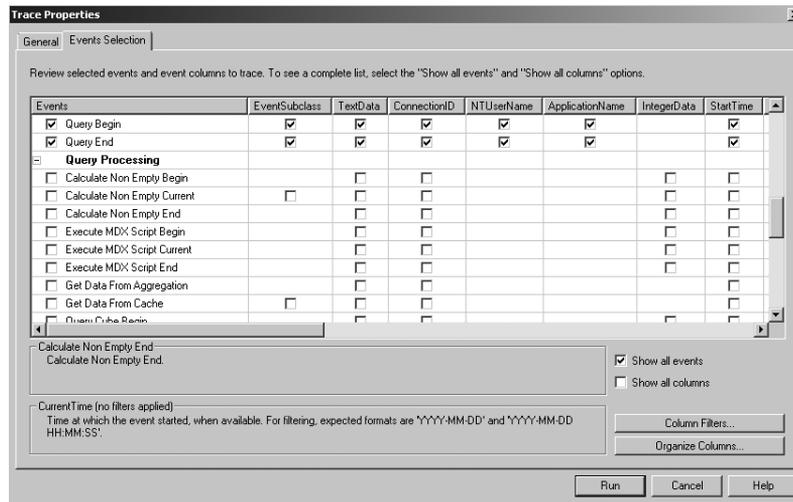
SQL Server Profiler captures are called *traces*. Appropriately capturing only events (and associated data) that you’re interested in takes a bit of practice. There are many items you can capture! The great news is that after you’ve determined the important events for your particular business scenario, you can save your defined capture for reuse as a trace template.

If you’re familiar with SQL Server Profiler from using it to monitor RDBMS data, you’ll note that when you set the connection to SSAS for a new trace, SQL Server Profiler presents you with a set of events that is specific to SSAS to select from. See the SQL Server Books Online topics “Introduction to Monitoring Analysis Services with SQL Server Profiler” and “Analysis Services Event Classes” for more detailed information. Figure 6-15 shows some of the events that you can choose to capture for SSAS objects. Note that in this view, we’ve selected Show All Events in the dialog box. This switch is off by default.

After you’ve selected which events (and what associated data) you want to capture, you can run your trace live, or you can save the results either to a file or to a relational table for you

to rerun and analyze later. The latter option is helpful if you want to capture the event on a production server and then replace the trace on a development server for analysis and testing of queries.

At this point, we're really just going to use SQL Server Profiler to view MDX queries that are generated when you manipulate the dimension and cube browsers in SSMS. The reason we're doing this is to introduce you to the MDX query language. You can also use SQL Server Profiler to capture generated DMX queries for data mining structures that you manipulate using the included browsers in SSMS.

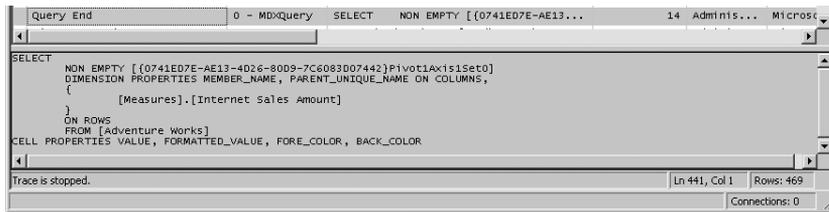


**FIGURE 6-15** SQL Server Profiler allows you to capture SSAS-specific events for OLAP cubes and data mining structures.

To see how query capture works, just start a trace in SQL Server Profiler, using all of the default capture settings, by clicking Run on the bottom right of the Trace Properties dialog box. With the trace running, switch to SSMS, right-click on the Adventure Works sample cube in the object browsers, click Browse, and then drag a measure to the pivot table design area.

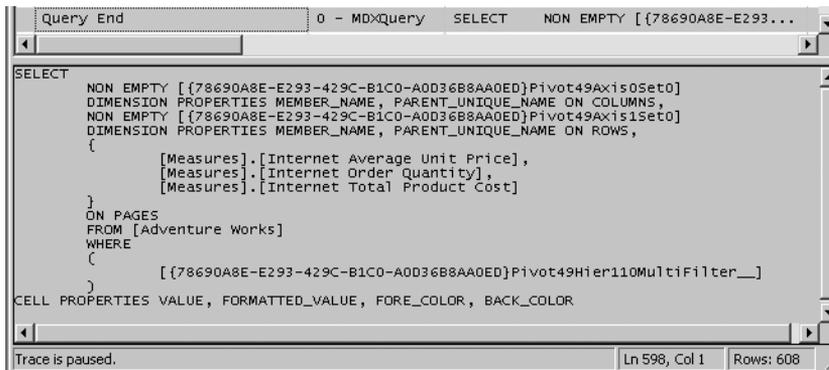
We dragged the Internet Sales Amount measure for our demo. After you've done that, switch back to SQL Server Profiler and then click on the pause trace button on the toolbar. Scroll through the trace to the end, where you should see a line with the EventClass showing *Query End* and EventSubclass showing *0 - MDXQuery*. Then click that line in the trace. Your results should look similar to Figure 6-16.

Note that you can see the MDX query that was generated by your drag action on the pivot table design interface in SSMS. This query probably doesn't seem very daunting to you, particularly if you've worked with Transact-SQL before. Don't be fooled, however; this is just the tip of the iceberg.



**FIGURE 6-16** SQL Server Profiler allows you to view MDX query text details.

Now let's get a bit more complicated. Click the Play button in SQL Server Profiler to start the trace again. After that, return to the SSMS OLAP cube pivot table browse area and then drag and drop some dimension information (hierarchies or members) to the rows, columns, slicer, and filter areas. After you have completed this, return to SQL Server Profiler and again pause your trace and then examine the MDX query that has been generated. Your results might look similar to what we show in Figure 6-17. You can see if you scroll through the trace that *each* action you performed by dragging and dropping generated at least one MDX query.



**FIGURE 6-17** Detail of a complex MDX query

We find SQL Server Profiler to be an invaluable tool in helping us to understand exactly what type of MDX query is being generated by the various tools (whether developer, administrator, or end user) that we use. Also, SQL Server Profiler does support tracing data mining activity. To test this, you can use the SSMS Object Browser to browse any data mining model while a SQL Server Profiler trace is active. In the case of data mining, however, you're *not* presented with the DMX query syntax. Rather, what you see in SQL Server Profiler is the text of the call to a data mining stored procedure. So the results in SQL Server Profiler look something like this:

```

CALL System.Microsoft.AnalysisServices.System.DataMining.AssociationRules.
GetStatistics('Subcategory Associations')

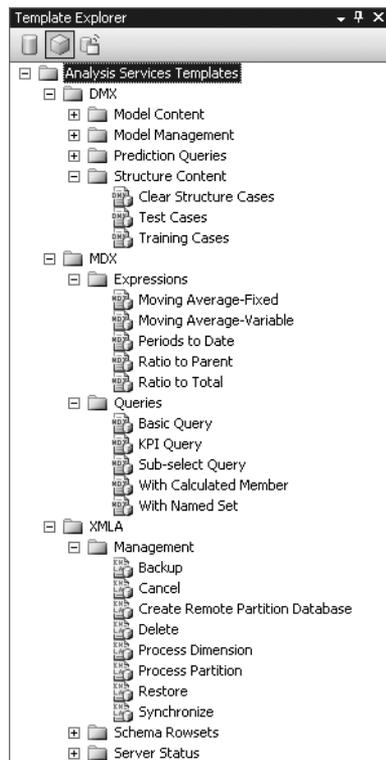
```

These results are also strangely categorized as *0 - MDXQuery* type queries in the EventSubclass column of the trace. You can also capture data mining queries using SQL Server Profiler. These queries are represented by the EventSubclass type *1 - DMXQuery* in SQL Server Profiler.

We'll return to SQL Server Profiler later in this book, when we discuss auditing and compliance. Also, we'll take another look at this tool in Chapters 10 and 11, which we devote to sharing more information about manual query and expression writing using the MDX language. Speaking of queries, before we leave our tour of SSMS, we'll review the methods you can use to generate and execute manual queries in this environment.

## Using SSAS Query Templates

Another powerful capability included in SSMS is that of being able to write and execute queries to SSAS objects. These queries can be written in three languages: MDX, DMX, and XMLA. At this point, we're not yet ready to do a deep dive into the syntax of any of these three languages; that will come later in this book. Rather, here we'd like to understand the query execution process. To that end, we'll work with the included query templates for these three languages. To do this, we need to choose Template Explorer from the View menu, and then click the Analysis Services (cube) icon to show the three folders with templated MDX, DMX, and XMLA queries. The Template Explorer is shown in Figure 6-18.



**FIGURE 6-18** SSMS includes MDX, DMX, and XMLA query templates

You can see that the queries are further categorized into functionality type in child folders under the various languages—such as Model Content and Model Management under DMX. You can also create your own folders and templates in the Template Explorer by right-clicking and then clicking New. After you do this, you're actually saving the information to this location on disk: `C:\Users\Administrator\AppData\Roaming\Microsoft\Microsoft SQL Server\100\Tools\Shell\Templates\AnalysisServices`.

## Using MDX Templates

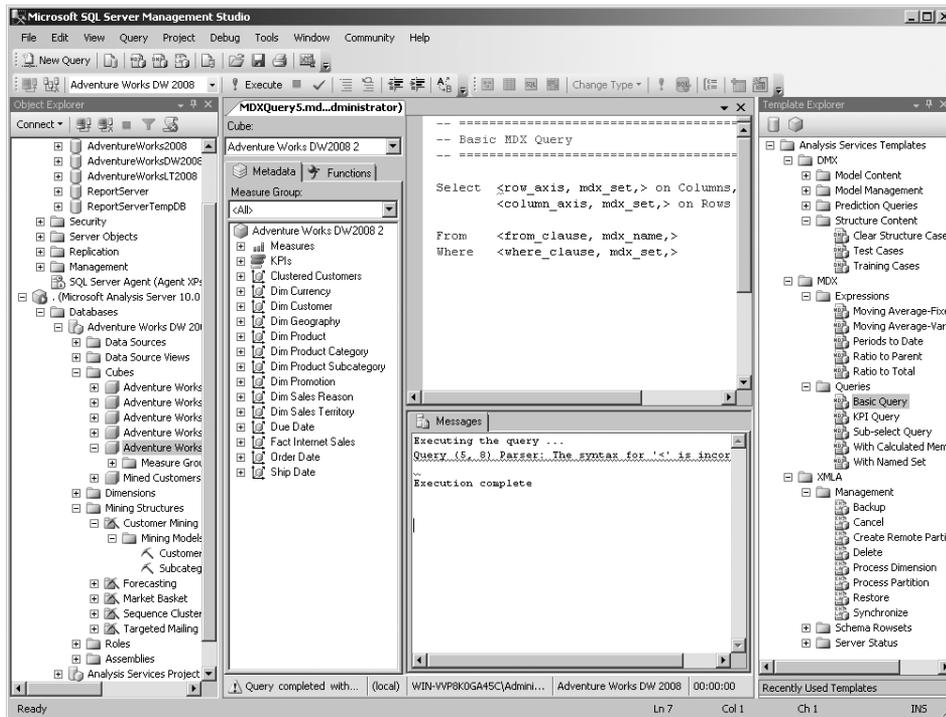
Now that you've opened the templates, you'll see that for MDX there are two types of queries: expressions and queries. Expressions use the syntax `With Member` and create a calculated member as part of a sample query. You can think of a calculated member as somewhat analogous to a calculated cell or set of cells in an Excel workbook, with the difference being that calculated members are created in  $n$ -dimensional OLAP space. We'll talk in greater depth about when, why, and how you choose to use calculated members in Chapter 9.

Queries retrieve some subset of an OLAP cube as an ADO.MD CellSet result, and they do not contain calculated members. To execute a basic MDX query, simply double-click the Basic Query template in the Template Explorer and then connect to SSAS. You can optionally write queries in a disconnected state and then, when ready, connect and execute the query. This option is available to reduce resource consumption on production servers.

You need to fill the query parameters with actual cube values before you execute the query. Notice that the query window opens yet another metadata explorer in addition to the default Object Explorer. You'll probably want to close Object Explorer when executing SSAS queries in SSMS. Figure 6-19 shows the initial cluttered, cramped screen that results if you leave all the windows open. It also shows the MDX parser error that results if you execute a query with errors. (See the bottom window, in the center of the screen, with text underlined with a squiggly line.)

Now we'll make this a bit more usable by hiding the Object Explorer and Template Explorer views. A subtle point to note is that the SSAS query metadata browser includes *two* filters: a Cube filter and, below it, a Measure Group filter. The reason for this is that SSAS OLAP cubes can contain hundreds or even thousands of measure groups.

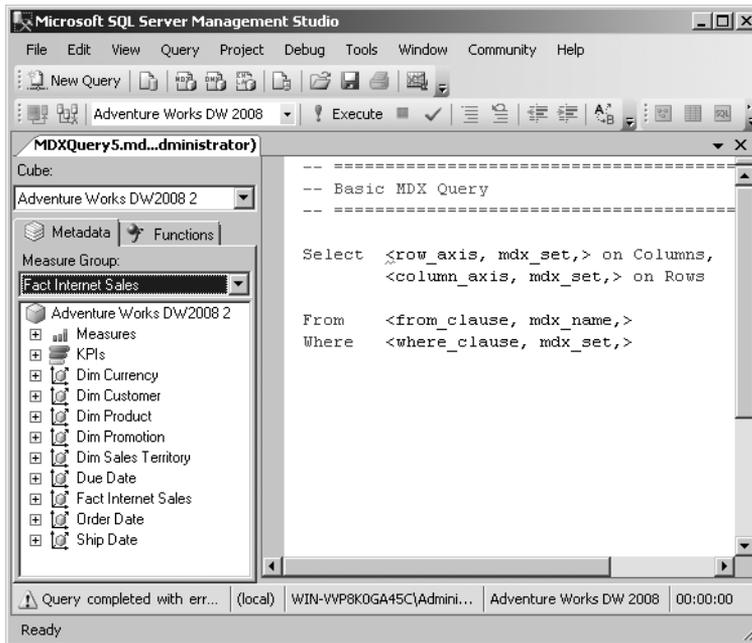
Figure 6-20 shows a cleaned-up interface. We've left the Cube filter set at the default, Adventure Works, but we've set the Measure Group filter to Internet Sales. This reduces the number of items in the viewer, as it shows only items that have a relationship to measures associated with the selected measure group. Also note that in addition to a list of metadata, this browser includes a second nested tab called Functions. As you'd expect, this tab contains an MDX function language reference list.



**FIGURE 6-19** The SSMS SSAS query screen can be quite cluttered by default.

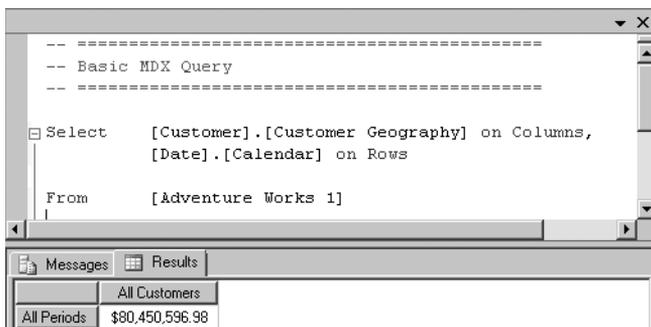
You might be wondering why you're being presented with yet another metadata interface, particularly because you're inside of a query-writing tool. Aren't you supposed to be writing the code manually here? Nope, not yet. Here's the reason why—MDX object naming is not as straightforward as it looks. For example, depending on uniqueness of member names in a dimension, you sometimes need to list the ordinal position of a member name; at other times, you need to actually list the name. Sound complex? It is. Dragging and dropping metadata onto the query surface can make you more productive if you're working with manual queries.

To run the basic query, you need to replace the items shown in the sample query between angle brackets—that is, <some value>—with actual cube metadata. Another way to understand this is to select Specify Values For Template Parameters on the Query menu. You can either type the information into the Template Parameters dialog box that appears, or you can click on any of the metadata from the tree view in the left pane and then drag it and drop it onto the designer surface template areas.



**FIGURE 6-20** The SSMS SSAS query screen with fewer items in the viewer

We'll use the latter approach to build our first query. We'll start by dragging the cube name to the From clause. Next we'll drag the Customers.Customer Geography hierarchy from the Customer dimension to the On Columns clause. We'll finish by dragging the Date.Calendar Year member from the Date hierarchy and Calendar hierarchy to the On Rows clause. We'll ignore the Where clause for now. As with Transact-SQL queries, if you want to execute only a portion of a query, just select the portion of interest and press F5. The results are shown in Figure 6-21.



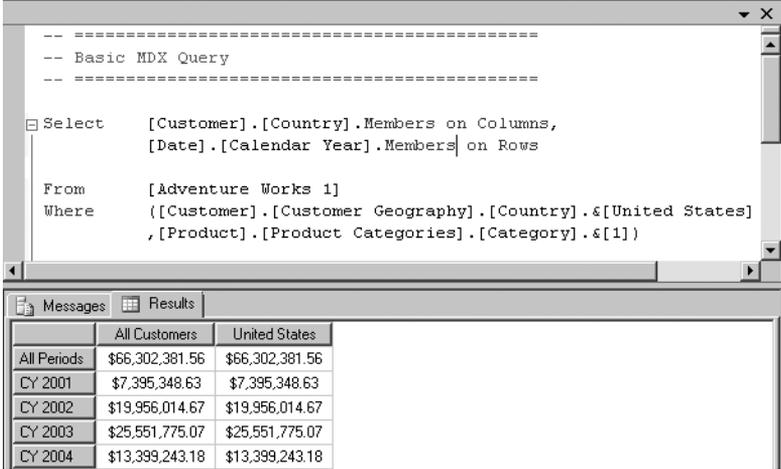
**FIGURE 6-21** SSMS SSAS query using simple query syntax

Do the results seem curious to you? Are you wondering which measure is being shown? Are you wondering why only the top-level member of each of the selected hierarchies is shown on columns and rows? As we've said, MDX is a deceptively simple language. If you've worked with Transact-SQL, which bears some structural relationship but is not very closely related at all, you'll find yourself confounded by MDX. We do plan to provide you with a thorough grounding in MDX. However, we won't be doing so until much later in this book—we'll use Chapters 10 and 11 to unravel the mysteries of this multidimensional query language.

At this point in our journey, it's our goal to give you an understanding of how to view and run prewritten MDX queries. Remember that you can also re-execute any queries that you've captured via SQL Server Profiler traces in the SSMS SSAS query environment as well.

Because we know that you're probably interested in just a bit more about MDX, we'll add a couple of items to our basic query. Notably, we'll include the MDX Members function so that we can display more than the default member of a particular hierarchy on an axis. We'll also implement the Where clause so that you can see the result of filtering. The results are shown in Figure 6-22.

We changed the dimension member information on Columns to a specific level (Country), and then we filtered in the Where clause to the United States only. The second part of the Where clause is an example of the cryptic nature of MDX. The segment `[Product].[Product Categories].[Category].&[1]` refers to the category named Bikes. We used the drag (metadata) and drop method to determine when to use names and when to use ordinals in the query. This is a time-saving technique you'll want to use as well.



```
-- =====  
-- Basic MDX Query  
-- =====  
  
Select      [Customer].[Country].Members on Columns,  
           [Date].[Calendar Year].Members on Rows  
  
From        [Adventure Works 1]  
Where       {[Customer].[Customer Geography].[Country].&[United States]  
           , [Product].[Product Categories].[Category].&[1]}
```

|             | All Customers   | United States   |
|-------------|-----------------|-----------------|
| All Periods | \$66,302,381.56 | \$66,302,381.56 |
| CY 2001     | \$7,395,348.63  | \$7,395,348.63  |
| CY 2002     | \$19,956,014.67 | \$19,956,014.67 |
| CY 2003     | \$25,551,775.07 | \$25,551,775.07 |
| CY 2004     | \$13,399,243.18 | \$13,399,243.18 |

FIGURE 6-22 MDX query showing filtering via the Where clause

## Using DMX Templates

Next we'll move to the world of DM query syntax. Again, we'll start by taking a look at the included templates in the Template Explorer. They fall into four categories: Model Content, Model Management, Prediction Queries, and Structure Content.

When you double-click on a DMX query template, you'll see that the information in the Metadata browser reflects a particular mining model. You can select different mining model metadata in the pick list at the top left of the browser. Also, the functions shown now include those specific to data mining. The Function browser includes folders for each data mining algorithm, with associated functions in the appropriate folder. Because understanding how to query data mining models requires a more complete understanding of the included algorithms, we'll simply focus on the mechanics of DMX query execution in SSMS at this point.

To do this, we'll double-click the Model Attributes sample DMX query in the Model Content folder that you access under DMX in the Template Explorer. Then we'll work with the templated query in the workspace. As with templated MDX queries, the DMX templates indicate parameters with the <value to replace> syntax. You can also click the Query menu and select Specify Values For Template Parameters as you can with MDX templates. We'll just drag the [Customer Clusters] mining model to the template replacement area. Note that you must include both the square brackets and the single quotes, as shown in Figure 6-23, for the query to execute successfully.

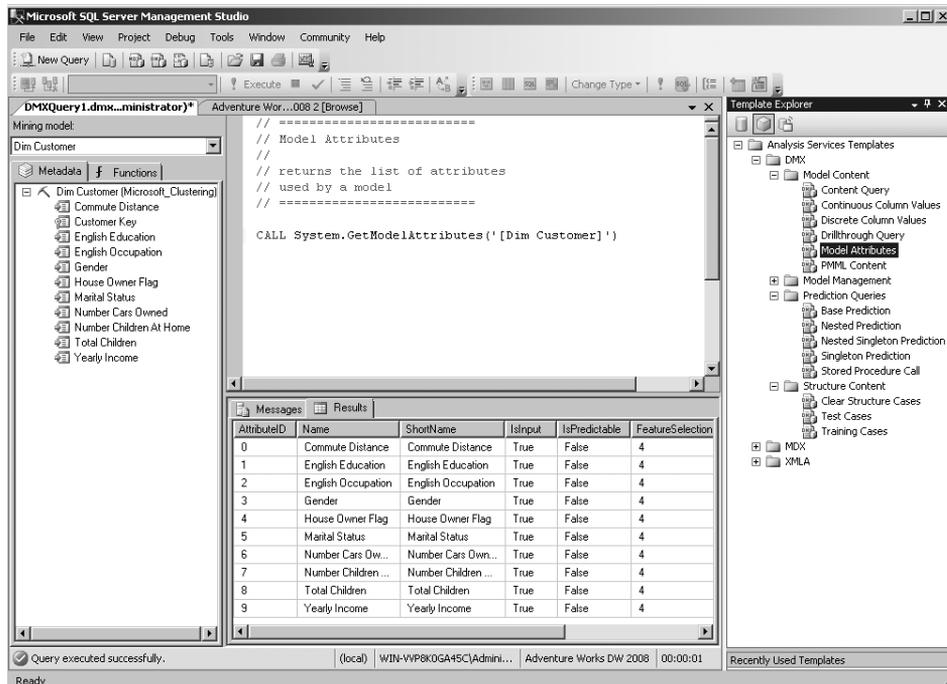


FIGURE 6-23 A DMX query showing mining model attributes

If you click on the Messages tab in the results area (at the bottom of the screen), you'll see that some DMX queries return an object of type `Microsoft.AnalysisServices.AdomdClient.AdomdDataReader`. Other DMX query types return scalar values—that is, DMX prediction queries.

For more information, see the SQL Server Books Online topic “Data Mining Extensions (DMX) Reference.”

## Using XMLA Templates

As with the previous two types of templates, SSMS is designed to be an XMLA query viewing and execution environment. The SSMS Template Explorer also includes a couple of types of XMLA sample queries. These are Management, Schema Rowsets, and Server Status. The XMLA language is an XML dialect, so structurally it looks like XML rather than a data structure query language, such as MDX or DMX (which look rather Transact-SQL-like at first glance). One important difference between MDX and XMLA is that XMLA is case-sensitive and space-sensitive, following the rules of XML in general.

Another important difference is that the Metadata and Function browsers are *not* available when you perform an XMLA query. Also, the results returned are in an XML format. In Figure 6-24, we show the results of executing the default Connections template. This shows detailed information about who is currently connected to your SSAS instance.

Be reminded that metadata for all SSAS objects—that is, OLAP dimensions, cubes, data mining models, and so on—can easily be generated in SSMS by simply right-clicking the object in the Object Browser and then clicking Script As. This is a great way to begin to understand the capabilities of XMLA. In production environments, you'll choose to automate many administrative tasks using XMLA scripting.

The templates in SSMS represent a very small subset of the XMLA commands that are available in SSAS. For a more complete reference, see the SQL Server Books Online topic “Using XMLA for Analysis in Analysis Services (XMLA).” Another technical note: certain commands used in XMLA are associated with a superset of commands in the Analysis Services Scripting Language (ASSL). The MSDN documentation points out that ASSL commands include both data definition language (DDL) commands, which define and describe instances of SSAS and the particular SSAS database, and also XMLA action commands such as Create, which are then sent to the particular object named by the ASSL. ASSL information is also referred to as binding information in SQL Server Books Online.

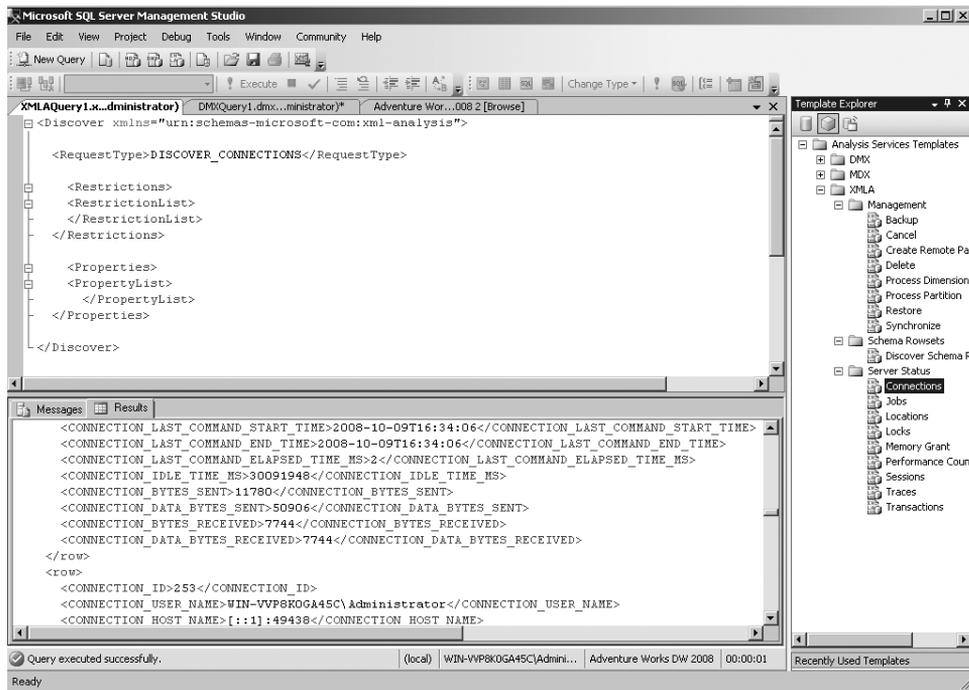


FIGURE 6-24 SSAS XMLA connections query in SSMS

## Closing Thoughts on SSMS

Although our primary audience is developers, as discussed, we've found that many SSAS developers are also tasked with performing SSAS administrative tasks. For this reason, we spent an entire chapter exploring the SSMS SSAS interface. Also, we find that using SSMS to explore built objects is a gentle way to introduce OLAP and DM concepts to many interested people. We've used SSMS to demonstrate these concepts to audiences ranging from .NET developers to business analysts. Finally, we'd like to note that we're continually amazed at the richness of the interface. Even after having spent many years with SSAS, we still frequently find little time-savers in SSMS.

## Summary

In this chapter, we took a close look at the installation of SSAS. We then discussed some tools you'll be using to work with SSAS objects. We took a particularly detailed look at SSMS because we've found ourselves using it time and time again on BI projects where we were tasked with being developers. Our real-world experience has been that SSAS developers must often also perform administrative tasks. So knowledge of SSMS can be a real time-saver. We included an introduction to SQL Server Profiler because we've found that many clients don't use this powerful tool correctly or at all because of a lack of understanding of it.

By now, we're sure you're more than ready to get started developing your first OLAP cube using BIDS. That's exactly what we'll be doing starting in the next chapter and then continuing on through several additional chapters—being sure to hit all the important nooks and crannies along the way.

# Index

## A

*AccessMode* property, 449  
account intelligence, configuring in  
    Business Intelligence Wizard,  
    243, 246–247  
*AcquireConnections* method, 581  
actions, SSAS  
    defined, 149, 233  
    drillthrough, 233, 236–238  
    regular, 233, 234–235  
    reporting, 233, 235–236  
Add SourceSafe Database Wizard,  
    541–542  
*AddRow* method, 582  
administrative scripting, SSRS,  
    667–669  
ADO.NET connection manager, 473  
ADO.NET data flow destination,  
    485, 486  
ADO.NET data flow source, 483,  
    497–498, 597  
Agent. *See* SQL Server Agent  
Aggregate data flow transformation,  
    486, 487, 488  
Aggregation Design Wizard,  
    271–273  
aggregations  
    Aggregation Design Wizard,  
    271–273  
    built-in types, 147  
    configuring, 262–263  
    creating designs manually,  
    277–278  
    defined, 9  
    and fact tables, 261  
    implementing, 270–278  
    key points, 271  
    main reason to add, 271  
    overview, 261–263  
    and query processing, 262  
    question of need for, 270–271  
    role of SQL Server Profiler,  
    275–277  
    in SQL Server cube store vs.  
    Transact-SQL, 9  
Usage-Based Optimization  
    Wizard, 274–275  
    using with date functions,  
    324–326  
    viewing and refining, 262–263

Agile Software Development.  
    *See* MSF (Microsoft Solution  
    Framework) for Agile Software  
    Development  
Algorithm Parameters dialog box,  
    367, 376–377, 378, 710  
algorithms, data mining  
    association category, 359  
    classification category, 358  
    clustering category, 359  
    configuring parameters, 367, 378  
    in data mining models, 45, 46,  
    46–47, 358  
    forecasting and regression  
    category, 359  
    Microsoft Association algorithm,  
    391–393  
    Microsoft Clustering algorithm,  
    386–389  
    Microsoft Decision Trees  
    algorithm, 381–383  
    Microsoft Linear Regression  
    algorithm, 383  
    Microsoft Logistic Regression  
    algorithm, 395–396  
    Microsoft Naïve Bayes algorithm,  
    376–381, 518  
    Microsoft Neural Network  
    algorithm, 394–395  
    Microsoft Sequence Clustering  
    algorithm, 389–390  
    Microsoft Time Series algorithm,  
    383–386  
    sequence analysis and prediction  
    category, 359  
    supervised vs. unsupervised, 376  
    viewer types for, 369–370  
ALTER MINING STRUCTURE (DMX)  
    syntax, 366  
Analysis Management Objects  
    (AMOs), 31  
Analysis Services. *See* SQL Server  
    Analysis Services (SSAS)  
Analysis Services Processing task,  
    430, 530  
analytical activities. *See* OLAP  
    (online analytical processing)  
*Ancestors* MDX function, 319–320  
*Application* class, 596, 599  
applications, custom, integrating  
    SSIS packages in, 596–600

ascmd.exe tool, 157  
ASM files, 732  
assemblies  
    compiled, using with SSAS  
    objects, 196–197  
    custom, adding to SSRS reports,  
    647–649  
    custom, creating, 197  
    default, in SSAS, 197  
association algorithms, 359  
Association Wizard, 48  
asynchronous data flow outputs,  
    459  
asynchronous transformation,  
    583–586  
attribute hierarchies, in OLAP cube  
    design, 206–207  
attribute ordering, specifying in  
    Business Intelligence Wizard,  
    244, 250  
attribute relationships, in BIDS, 139,  
    205, 207–209, 223  
Audit transformation, 524  
auditing. *See also* SQL Server  
    Profiler  
    added features in SQL Server  
    2008, 111  
    using SQL Server Profiler, 109–110  
authentication  
    credential flow in SSRS reports,  
    103  
    requesting access to Report  
    Server, 610–611  
AverageOfChildren aggregate  
    function, 147

## B

background processing, for reports  
    and subscriptions, 612  
backups and restores  
    overview, 106  
    for SQL Server Analysis Services,  
    106–107  
    for SQL Server Integration  
    Services, 107–108, 112  
    for SQL Server Reporting Services,  
    108  
Barnes and Noble, 28

- BI solutions. *See also* Business Intelligence Development Studio (BIDS)
  - case studies, 27–33
  - common challenges, 54–56
  - common terminology, 11–15
  - complete solution components, 50–54
  - customizing data display in SQL Server 2008, 3
  - defined, 3
  - development productivity tips, 70
  - in law enforcement, 29
  - localization of data, 29
  - measuring solution ROIs, 56–58
  - MSF project phases, 65–71
  - multiple servers for solutions, 4
  - and Office SharePoint Server, 723–745
  - process and people issues, 61–83
  - project implementation scope, 28
  - query language options, 23–25
  - relational and non-relational data sources, 22–23
  - reporting interfaces, 3
  - role of Microsoft Excel, 36–37, 43–50
  - sales and marketing, 29
  - schema-first vs. data-first approaches to design phase, 130
  - security requirements for solutions, 95–106
  - skills necessary for projects, 72–76
  - software life cycle, 28
  - solution core components, 16–20
  - solution optional components, 21–23
  - testing project results, 70–71
  - top 10 scoping questions, 30
  - visualizing solutions, 34–36
- BIDS. *See* Business Intelligence Development Studio (BIDS)
- BIDS Helper tool, 255, 490, 494, 510
- Biztalk Server, 22
- Boolean data type, 363
- BottomCount* MDX statement, 311
- breakpoints, inserting, 505–506
- build, defined, 259
- building phase, MSF, 68–70
- business intelligence (BI). *See also* Business Intelligence Development Studio (BIDS)
  - case studies, 27–33
  - common challenges, 54–56
  - common terminology, 11–15
  - complete solution components, 50–54
  - customizing data display in SQL Server 2008, 3
  - defined, 3
  - development productivity tips, 70
  - in law enforcement, 29
  - localization of data, 29
  - measuring solution ROIs, 56–58
  - MSF project phases, 65–71
  - multiple servers for solutions, 4
  - and Office SharePoint Server, 723–745
  - process and people issues, 61–83
  - project implementation scope, 28
  - query language options, 23–25
  - relational and non-relational data sources, 22–23
  - reporting interfaces, 3
  - role of Microsoft Excel, 36–37, 43–50
  - sales and marketing, 29
  - schema-first vs. data-first approaches to design phase, 130
  - security requirements for solutions, 95–106
  - skills necessary for projects, 72–76
  - software life cycle, 28
  - solution core components, 16–20
  - solution optional components, 21–23
  - testing project results, 70–71
  - top 10 scoping questions, 30
  - visualizing solutions, 34–36
- Business Intelligence Development Studio (BIDS). *See also* SQL Server Analysis Services (SSAS)
  - BIDS Helper tool, 255, 490, 494, 510
  - compared with Visio for creating OLAP models, 133
  - as core tool for developing OLAP cubes and data mining structures, 16, 40, 157
  - creating new SSIS project templates by using New Project dialog box, 464–465
  - creating or updating SSAS objects, 186–188
  - creating reports, 612–622
  - creating SSIS packages, 463–495
  - data mining interface, 360–375
  - defined, 155
  - Dependency Network view, 47
  - deploying reports to SSRS, 624
  - deploying SSIS packages, 553–556
  - Deployment Progress window, 41
  - development tips, 70
  - disconnected instances, 259–261
  - Error List window, 31
  - exploring dimension table data, 123
  - exploring fact table data, 120
  - MDX Query Designer, 628–631
  - New Cube Wizard, 134
  - OLAP cubes, adding capabilities, 225–255
  - OLAP cubes, using to design, 183–223
  - online vs. offline mode, 184–186
  - opening sample databases, 39–43
  - overview, 183–186
  - as primary development tool for SSIS packages, 20, 439–440, 463–495
  - processing options for cubes and dimensions, 287–291
  - relationship to Visual Studio, 16–17, 22, 41, 463
  - Report Data window, 635–638
  - resemblance to Visual Studio interface, 157
  - role designer, 195–196
  - running on x64 systems, 91
  - Solution Explorer window, 40, 46, 184, 186–188
  - source control considerations, 113
  - SSRS Toolbox, 621–622, 638
  - working with SSAS databases in connected mode, 261
  - working with two instances open, 225
- Business Intelligence Wizard
  - accessing, 243
  - Create A Custom Member Formula, 244, 251
  - Define Account Intelligence, 243, 246–247
  - Define Currency Conversion, 244, 251–254
  - Define Dimension Intelligence, 243, 250
  - Define Semiadditive Behavior, 244, 250
  - Define Time Intelligence, 243, 245
  - Specify A Unary Operator, 244, 248–250
  - Specify Attribute Ordering, 244, 250
- ByAccount aggregate function, 147

## C

- cache scopes, for queries, 326
- CacheMode* property, 364
- Calculated Columns sample package, 487
- calculated measures, 148
- calculated members
  - creating in Business Intelligence Wizard, 318, 320
  - creating in cube designer, 239–241
  - creating in query designer, 631
  - creating using WITH MEMBER statement, 307
  - defined, 175, 307
  - global vs. local, 631
  - permanent, creating using BIDS interface, 334–335
  - permanent, creating using MDX scripts, 335–336
  - pros and cons, 241
  - vs. stored measures, 298
- Calculations tab, cube designer, 201, 239–242, 334–335
- Capability Maturity Model Integration (CMMI), 65
- CAS (code access security), 648–649
- Cash Transform data flow transformation, 486
- change data capture (CDC), 524, 531
- Chart control, 638, 643
- checkpoints, in SSIS packages
  - configuring, 506
  - defined, 506
  - writing, 507
- child tables, relationship to parent table, 5
- Children* MDX function, 300, 316, 321
- Choose Toolbox Items dialog box, 591
- classification algorithms, 358
- classification matrix, 415–416
- Clean Data Wizard, 705, 706–707
- cloud-hosted data mining, 720–721
- Cluster* DMX function, 425
- Cluster Wizard, Microsoft Visio, 717–718
- ClusterDistance* DMX function, 425
- clustering algorithms, 359
- ClusterProbability* DMX function, 425
- CMMI (Capability Maturity Model Integration), 65
- code access security (CAS), 648–649
- CodePlex Web site, 37–38, 86, 157
- columns
  - in dimension tables, 121–122
  - in fact tables, 118–119, 146
  - variable-width, in data flow metadata, 456–457
- command-line tools
  - ascmd.exe tool, 157
  - DTEXEC utility, 440
  - DTEXECUI utility, 440–441
  - DTUTIL utility, 441
  - installed with SQL Server 2008, 157
  - rsconfig.exe tool, 604
  - rs.exe tool, 609
  - SQLPS.exe tool, 157
- CommandText* property, 598
- community technology preview (CTP) version, SQL Server 2008, 40
- ComponentMetaData* property, 580
- components. *See also* Script component
  - compared with tasks, 444, 567–568
  - custom, in SSIS, 587–588
  - destination, 485–486, 586–587
  - in SSIS package data flows, 444
  - transformation, 486–488
- Configuration Manager, Reporting Services, 102, 108, 155, 607, 609, 737
- Configuration Manager, SQL Server, 94, 155, 157–158
- Configuration Manager, SSRS. *See* Configuration Manager, Reporting Services
- confusion matrix. *See* classification matrix
- connection managers
  - adding to packages, 473
  - ADO.NET, 473
  - custom, 588, 594
  - defined, 468
  - Flat File, 474
  - inclusion in Visual Studio package designers, 468
  - ODBC, 473
  - OLE DB, 473
  - overview, 448–450
  - Raw File, 474
  - specifying for log providers, 502
  - types, 473–474
  - using in Script components, 580–581
  - using within Visual Studio, 473–474
- Connections* property, 571, 580
- ConnectionString* property, 581
- constraints. *See* precedence constraints
- containers
  - default error handling behavior, 499
  - generic group, 479
  - SSIS control flow, 478–479
- content types
  - Continuous, 362
  - Cyclical, 362
  - defined, 361
  - detecting in Data Mining Wizard, 402
  - Discrete, 361
  - Discretized, 362
  - Key, 362
  - Key Sequence, 362, 363
  - Key Time, 362, 363
  - Ordered, 362
  - support for data types, 363
  - Table, 362
- Continuous content type, 362
- control flow designer
  - Connection Manager window in, 468
  - Data Flow task, 476, 477–478
  - Data Profiling task, 476
  - defined, 468
  - event handling, 500–501
  - Execute Process sample, 476–478
  - Execute Process task, 476, 477
  - Execute SQL tasks, 476, 476–477, 494
  - Foreach Loop containers, 476, 478
  - For Loop containers, 478
  - Sequence containers, 478
  - Task Host containers, 478
  - task overview, 476–478
  - Toolbox window in, 469
- control flow, in SSIS packages
  - building custom tasks, 591–593
  - configuring task precedence, 480–481
  - container types, 478–479
  - Data Profiling task, 510–513
  - event handling, 450–451
  - logging events, 504
  - Lookup sample, 528
  - overview, 442–444
  - Script task, 567–568
- copying SSIS packages to deploy, 552–553
- Count aggregate function, 147
- counters. *See* performance counters
- Create A Custom Member Formula, Business Intelligence Wizard, 244, 251
- CREATE KPI statement, 348

- CreateNewOutputRows* method, 582
  - CRISP-DM life cycle model, 399–400, 409
  - cross validation, 417–418
  - CTP (community technology preview) version, SQL Server 2008, 40
  - cube browser, 41–42, 201
  - cube designer
    - accessing Business Intelligence Wizard, 243
    - Actions tab, 201, 233–239
    - Aggregations tab, 201, 262–263, 275
    - Browser tab, 41–42, 201
    - Calculations tab, 201, 239–242, 334–335
    - Cube Structure tab, 201, 201–203
    - description, 201
    - Dimension Usage tab, 126–128, 134–135, 211–212, 215
    - KPIs tab, 201, 228, 345
    - opening dimension editor, 203–204
    - Partitions tab, 201, 264, 278
    - Perspectives tab, 201, 227
    - tool for building OLAP cubes, 198–204
    - Translations tab, 201
  - cube partitions
    - defined, 263
    - defining, 265–266
    - enabling writeback, 285–286
    - overview, 263–264
    - for relational data, 268–269
    - remote, 270
    - specifying local vs. remote, 270
    - in star schema source tables, 268–269
    - storage modes, 270
    - and updates, 532
  - Cube Wizard
    - building first OLAP cube, 218–223
    - Create An Empty Cube option, 199
    - Generate Tables In The Data Source option, 199, 200
    - launching from Solution Explorer, 218
    - populating Dimension Usage tab, 128
    - Use Existing Tables option, 198–199, 200
  - CUBEKPIMEMBER OLAP function, 683
  - CUBEMEMBER OLAP function, 683
  - CUBEMEMBERPROPERTY OLAP function, 683
  - CUBERANKEDMEMBER OLAP function, 683
  - cubes, OLAP
    - adding aggregations, 263
    - assessing source data quality, 516–518
    - background, 13
    - as BI data structure, 13
    - BIDS browser, 41–42, 201
    - building in BIDS, 198–204
    - building prototypes, 50
    - building sample using Adventure Works, 37–39
    - configuring properties, 243–254
    - connecting to sample using Microsoft Excel, 43–45
    - as core of SQL Server BI projects, 115
    - core tools for development, 157
    - creating empty structures, 133
    - as data marts, 13
    - data vs. metadata, 258
    - in data warehouses, 11
    - defined, 9
    - and denormalization concept, 125
    - vs. denormalized relational data stores, 10
    - deploying, 254–255, 260–261
    - designing by using BIDS, 183–223
    - dimensions overview, 9–10, 204–210, 257–258
    - fact (measure) modeling, 146–147
    - first, building in BIDS, 218–223
    - Microsoft Excel as client, 671–684
    - modeling logical design concepts, 115–150
    - vs. OLTP data sources, 54
    - opening sample in Business Intelligence Development Studio, 39–43
    - overview of source data options, 115–116
    - partitioning data, 263–270
    - as pivot tables, 10–11
    - pivoting in BIDS browser, 42
    - presenting dimensional data, 138–142
    - processing options, 287–291
    - and ROI of BI solutions, 56–58
    - skills needed for building, 72, 74
    - star schema source data models, 116–125
    - UDM modeling, 9–10
    - updating, 530–533
    - using dimensions, 201–217
    - viewing by using SSMS Object Browser, 164–168
    - visualizing screen view, 10–11
  - CUBESET OLAP function, 683
  - CUBESETCOUNT OLAP function, 683
  - CUBEVALUE OLAP function, 683
  - currency conversions, configuring in Business Intelligence Wizard, 244, 251–254
  - CurrentMember* MDX function, 232, 313
  - custom applications, integrating SSIS packages in, 596–600
  - custom foreach enumerators, 594–595
  - custom member formulas, creating in Business Intelligence Wizard, 244, 251
  - custom SSIS objects
    - control flow tasks, 591–593
    - data flow components, 588–591, 593–594
    - deploying, 589–591
    - implementing user interfaces, 593, 594
    - overview, 587–588
    - registering assemblies in GAC, 590
    - signing assemblies, 589
  - customer relationship management (CRM) projects, skills needed for reporting, 75
  - Cyclical content type, 362
- ## D
- Data Connection Wizard, 672
  - data dictionaries, 67
  - data flow designer
    - advanced edit mode, 484
    - Calculated Columns sample package, 487
    - Connection Manager window in, 468
    - debugging Script components, 587
    - defined, 468
    - destination components, 485–486
    - error handling, 497–498
    - Execute Process sample, 482
    - overview, 482–483
    - paths, defined, 484
    - separate from control flow designer, 478
    - source components, 483–485
    - specifying Script components, 573–581
    - and SSIS data viewer capability, 488–489
    - Toolbox window in, 469
    - transformation components, 486–488

- data flow engine, in SSIS
  - asynchronous outputs, 459
  - basic tasks, 453–454
  - memory buffers, 454
  - metadata characteristics, 454–458
  - overview, 453–454
  - synchronous outputs, 458, 459
  - variable-width column issue, 456–457
- data flow, in SSIS packages. *See also* data flow engine, in SSIS
  - asynchronous component outputs, 459
  - custom components, 588–591, 593–594
  - error handling, 451–452
  - logging events, 504
  - Lookup sample, 529–530
  - overview, 444
  - Script component, 567–568
  - synchronous component outputs, 458, 459
- data flow maps, 525
- Data Flow task, 476, 477–478, 482
- data lineage, 524
- data marts, 13
- data mining
  - adding data mining models to structures using BIDS, 404–406
  - algorithms. *See* algorithms, data mining
  - ALTER MINING STRUCTURE syntax, 366
  - Attribute Characteristics view, 378, 379
  - Attribute Discrimination view, 378, 380
  - Attribute Profiles view, 378
  - background, 14
  - BIDS model visualizers, 46–47
  - BIDS visualizer for, 18
  - building objects, 407
  - building prototype model, 50
  - building structures using BIDS, 401–404
  - cloud-hosted, 720–721
  - compared with OLAP cubes, 14, 396
  - content types, 361–363
  - core tools for development, 157
  - creating structures by opening new Analysis Services project in BIDS, 401–404
  - creating structures using SSAS, 18
  - data types, 361–363
  - defined, 14
  - Dependency Network view, 370–371, 378
  - Distribution property, 363
  - DMX query language, 24, 179–180
  - end-user client applications, 431
  - feature selection, 377–381
  - future for client visualization controls, 720
  - Generic Content Tree viewer, 371–372
  - getting started, 396
  - implementing structures, 399–431
  - importance of including functionality, 53–54
  - initial loading of structures and models, 533–534
  - installing add-ins to Microsoft Office 2007, 687–688
  - Microsoft Cluster viewer, 372
  - Microsoft Excel and end-user viewer, 356
  - Microsoft Office 2007 as client, 687–721
  - model viewers in BIDS, 46
  - Modeling Flags property, 363
  - object processing, 429–431
  - OLAP cubes vs. relational source data, 401
  - prediction queries, 419–426
  - problem of too much data, 55
  - processing models/objects, 407–409
  - queries, 535–537
  - Relationship property, 363
  - and ROI of BI solutions, 57–58
  - role of Microsoft Excel add-ins, 45–47
  - sample Adventure Works cube, 357
  - sample Adventure Works structure, 46
  - skills needed for building structures, 72, 74
  - software development life cycle for BI projects, 69
  - SQL Server Analysis Services 2008 enhancements, 148–149
  - and SQL Server Integration Services, 426–428
  - tools for object processing, 429–431
  - validating models, 409–418
  - viewing sample structures in BIDS, 43
  - viewing structures by using SSMS viewers, 164, 168–170
  - viewing structures using Microsoft Excel, 47–50
- Data Mining Add-ins for Office 2007
  - downloading and installing, 47–48, 94
  - installing, 687–688
  - Table Analysis Tools group, 690–700
- Data Mining Advanced Query Editor, 704
- Data Mining Extensions. *See* DMX (Data Mining Extensions) query language
- Data Mining Model Training destination, 485, 534–535
- Data Mining Query control flow task, 535–536
- Data Mining Query data flow component, 427, 428
- Data Mining Query Task Editor, 427–428, 536
- Data Mining Query transformation component, 487, 536–537
- data mining structure designer
  - choosing data mining model, 365–368
  - handling nested tables, 364, 366
  - Mining Accuracy Chart tab, 360, 373–375, 417
  - Mining Model Prediction tab, 360, 375, 419, 424
  - Mining Model Viewer tab, 46, 360, 368–373, 408, 409
  - Mining Models tab, 360, 365–368, 404, 404–405
  - Mining Structure tab, 360, 364–365, 404
  - viewing source data, 364
- Data Mining tab, Microsoft Excel Accuracy And Validation group, 712
  - comparison with Table Tools Analyze tab, 700–701
  - Data Modeling group, 708–712
  - Data Preparation group, 705–708
  - Management group, 701–702
  - Model Usage group, 702–705
- Data Mining Wizard, 401–404, 405–406
- Data Profiling task
  - defined, 510
  - limitations, 512
  - list of available profiles, 512–513
  - new in SQL Server 2008, 478
  - profiling multiple tables, 513
  - viewing output file, 513
  - when to use, 510
- data regions, SSRS, defined, 638–655
- Tablix data region, defined, 639–642

- Data Source Properties dialog box, 614–616
- data source views (DSVs)
  - compared with relational views, 161
  - creating, 199, 201
  - defined, 190
  - examining, 190–192
  - getting started, 195
  - making changes to existing tables, 193–194
  - making changes to metadata, 192–193
  - overview, 190–191
  - as quick way to assess data quality, 518
  - required for building OLAP cubes, 199
  - in SSIS, 466, 467
- data storage containers, skills for building, 72
- data stores, OLAP. *See also* cubes, OLAP
  - denormalized, 8
  - as source for decision support systems, 13–14
- data stores, OLTP
  - query challenges, 6–7
  - reasons for normalizing, 5–6
  - relational vs. non-relational, 5
- Data Transformation Services (DTS)
  - comparison with SSIS, 446, 463–464
  - relationship to SSIS, 437–438
  - in SQL Server 2000, 546
- data types
  - Boolean, 363
  - content types supported, 363
  - date, 363
  - defined, 361
  - detecting in Data Mining Wizard, 403
  - double, 363
  - long, 363
  - text, 363
- data viewers, SSIS, 488–489, 506
- data visualization group, Microsoft Research, 34, 83, 720
- data vs. metadata, 258
- data warehouses
  - background, 12
  - compared with OLAP, 12
  - data marts as subset, 13
  - defined, 11
  - Microsoft internal, case study, 28
- database snapshots, 507
- DatabaseIntegratedSecurity* property, 668
- data-first approach to BI design, 130
- DataReader destination, 485
- DataReader* object, 598
- dataset designer, 618
- Dataset Properties dialog box, 618
- date data type, 363
- date functions, 321–326
- debugging
  - SSIS packages, 471–472, 505–506
  - SSIS script tasks, 572
  - using data viewers, 488–489, 506
- decision support systems, 13–14
- decision tables
  - fast load technique, 528
  - loading source data into, 525–526
- Decision Tree view, Microsoft Visio, 716
- Declarative Management Framework (DMF) policies, 95
- DefaultEvents* class, 596
- Define Account Intelligence, Business Intelligence Wizard, 243, 246–247
- Define Currency Conversion, Business Intelligence Wizard, 244, 251–254
- Define Dimension Intelligence, Business Intelligence Wizard, 243, 250
- Define Relationships dialog box, 212–213, 215–216, 216
- Define Semiadditive Behavior, Business Intelligence Wizard, 244, 250
- Define Time Intelligence, Business Intelligence Wizard, 243, 245
- degenerate dimension, in fact tables, 119
- denormalization
  - and OLAP cube structure, 125
  - in OLAP data stores, 8
- Dependency Network view, Microsoft Visio, 714–715
- Deploy option, BIDS Solution Explorer, 260
- deploying
  - code for custom objects, 589–591
  - reports to SSRS, 623–624
  - role and responsibility of release/operations managers, 83
  - SSIS packages, 441, 461–462, 546–558
- Deployment Progress window, 41
- Deployment Utility, 556–558
- Deployment Wizard, 155
- derived measures, 148
- Descendants* MDX function, 318–319, 321
- Description* SSIS variable property, 491
- destination components
  - data flow designer, 485–486
  - Script-type, 586–587
- developers
  - IT administrators vs. traditional developers, 81
  - keeping role separate from tester role, 81
  - manager's role and responsibility on development teams, 79–81
  - responsibility for performing administrative tasks, 160, 181
  - SSAS graphical user interface for creating objects, 154
  - types needed for BI projects, 80
- development teams
  - forming for BI projects, 76–83
  - optional project skills, 74–76
  - required project skills, 72–74
  - role and responsibility of developer manager, 79–81
  - role and responsibility of product manager, 78
  - role and responsibility of program manager, 79
  - role and responsibility of project architect, 78–79
  - role and responsibility of release/operations manager, 83
  - role and responsibility of test manager, 81–82
  - role and responsibility of user experience manager, 82–83
  - roles and responsibilities for working with MSF, 76–83
  - source control considerations, 111–113
- development tools, conducting baseline survey, 86
- deviation analysis, 360
- DimCustomer dimension table example, 122, 123
- DimCustomer snowflake schema example, 134
- dimension designer
  - accessing Business Intelligence Wizard, 243
  - Attribute Relationships tab, 141
  - Dimension Structure tab, 139
- dimension editor
  - Attribute Relationships tab, 205, 207–209, 223
  - Browser tab, 205
  - Dimension Structure tab, 205–207

- opening from cube designer, 203–204
- overview, 205
- Translations tab, 205, 209
- dimension intelligence, configuring in Business Intelligence Wizard, 243, 250
- Dimension Processing destination, 485
- dimension structures, defined, 139
- dimension tables
  - data vs. metadata, 258
  - DimCustomer example, 122, 123
  - exploring source table data, 123
  - as first design step in OLAP modeling, 131
  - generating new keys, 122, 146
  - pivot table view, 123
  - rapidly changing dimensions, 144
  - slowly changing dimensions, 142–144
  - space issue, 124
  - for star schema, 117–118, 121–125, 194
  - table view, 123
  - types of columns, 121–122
  - updating, 532–533
- Dimension Usage tab, in cube designer
  - options, 211–212
  - for snowflake schema, 135, 215
  - for star schema, 126–127, 134–135
  - using Cube Wizard to populate, 128
- dimensions
  - adding attributes, 222–223
  - adding to OLAP cube design using Cube Wizard, 220–221
  - combining with measures to build OLAP cubes, 210–214
  - configuring properties, 243–254
  - creating using New Dimension Wizard, 221
  - data vs. metadata, 258
  - enabling writeback for partitions, 285–286
  - hierarchy building, 138–139
  - non-star designs, 215–217
  - presenting in OLAP cubes, 138–142
  - processing options, 287–291
  - querying of properties, 329–332
  - rapidly changing, 144, 284
  - relationship to cubes, 257–258
  - role in simplifying OLAP cube complexity, 204–205
  - slowly changing, 142–144
  - as starting point for designing and building cubes, 204–205
  - Unified Dimensional Model, 138
  - writeback capability, 145
- disconnected BIDS instances, 259–261
- Discrete content type, 361
- Discretized content type, 362
- Distinct Count aggregate function, 147
- Distribution property, 363
- .dll files, 590
- DMF (Declarative Management Framework) policies, 95
- DMX (Data Mining Extensions) query language
  - adding query parameter to designer, 634–635
- ALTER MINING STRUCTURE syntax, 366
- background, 24
- building prediction queries, 419–421
- Cluster function, 425
- ClusterDistance function, 425
- ClusterProbability function, 425
- defined, 24
- designer, defined, 627
- designer, overview, 617
- including query execution in SSIS packages, 535–537
- Predict function, 421, 423–424, 425
- PredictHistogram function, 425
- prediction functions, 423–426
- PREDICTION JOIN syntax, 422, 427
- prediction query overview, 421–423
- PredictProbability function, 425
- PredictProbabilityStDev function, 425
- PredictProbabilityVar function, 425
- PredictStDev function, 425
- PredictSupport function, 425
- PredictTimeSeries function, 425
- PredictVariance function, 425
- queries in SSIS, 426–428
- querying Targeted Mailing structure, 633–635
- RangeMax function, 425
- RangeMid function, 425
- RangeMin function, 425
- switching from MDX designer to DMX designer, 633
- templates, 179–180
- ways to implement queries, 426

- domain controllers, conducting baseline survey, 85
- double data type, 363
- drillthrough actions, 233, 236–238, 372–373
- DROP KPI statement, 348
- .ds files, 108
- .dsv files, 108
- DSVs. *See* data source views (DSVs)
- DTEXEC utility, 440
- DTEXECUI utility, 440–441
- DTLoggedExec tool, 600
- DTS. *See* Data Transformation Services (DTS)
- Dts object, 571–572
- DtsConnection class, 597
- .dtsx files, 442, 540, 547
- DTUTIL utility, 441, 558

## E

- Enable Writeback dialog box, 286
- end users
  - decision support systems for communities, 13–14
  - reporting interface considerations, 51
  - viewing BI from their perspective, 31–50
- Enterprise Manager. *See* SQL Server Enterprise Manager
- enumerators, custom, 588, 594–595
- envisioning phase, MSF, 65–67
- Error and Usage Reporting tool, 155
- error conditions, in BIDS, 259
- error handling, in SSIS, 451–452, 497–499
- ETL (extract, transform, and load) systems
  - background, 14–15
  - as BI tool, 515–516
  - defined, 14–15
  - importance of SSIS, 52–53
  - for loading data mining models, 533–537
  - for loading OLAP cubes, 516–530
  - security considerations, 97–98
  - skills needed, 73, 76
  - SSIS as platform for, 435–462
  - for updating OLAP cubes, 530–533
- EvaluateAsExpression SSIS variable property, 491
- event handler designer
  - Connection Manager window in, 468
  - defined, 468
  - Toolbox window in, 469

event handling, in SSIS, 450–451, 499–501

events, logging, 501–505

Excel. *See also* Excel Services

- adding Data Mining tab to Ribbon, 47–48, 50
- adding Table Tools Analyze tab to Ribbon, 47–48, 419
- Associate sample workbook, 48–49
- as client for SSAS data mining structures, 101
- as client for SSAS OLAP cubes, 100–101
- configuring session-specific connection information, 101
- connecting to sample SSAS OLAP cubes, 43–45
- creating sample PivotChart, 679–680
- creating sample PivotTable, 678–679
- Data Connection Wizard, 672
- Data Mining add-ins, 73, 361, 368, 419
- data mining integration
  - functionality, 689–690
- Data Mining tab functionality, 700–712
- Dependency Network view for Microsoft Association, 48–49
- extending, 683–684
- Import Data dialog box, 674
- Offline OLAP Wizard, 681–683
  - as OLAP cube client, 671–684
  - OLAP functions, 683
  - as optional component for BI solutions, 21
- PivotTable interface, 675–677
- PivotTables as interface for OLAP cubes, 10–11
- popularity as BI client tool, 723–724
- Prediction Calculator, 419
- role in understanding data mining, 45–47
- security for SSAS objects, 100–101
- skills needed for reporting, 73, 75
- trace utility, 101
- viewing data mining structures, 47–50
  - viewing SSRS reports in, 649–650

Excel Calculation Services, 724

Excel data flow destination, 485

Excel data flow source, 483

Excel Services. *See also* Excel

- basic architecture, 724–725
- complex example, 733–736

- extending programmatically, 732–736
- immutability of Excel sheets, 726
- overview, 724
- publishing parameterized Excel sheets, 729–732
- sample worksheets, 726–729
- and Web Services API, 732–736

*ExclusionGroup* property, 579

*Execute* method, 592, 596, 597

Execute Package task, 478

Execute Process sample

- control flow tasks in, 476–478
- data flow designer, for SSIS package, 482–483
- installing, 474–475

Execute SQL tasks, 476, 476–477, 494

*ExecuteReader* method, 597, 598

Explore Data Wizard, 705–706

Expression And Configuration Highlighter, 494

Expression dialog box, 637

*Expression* SSIS variable property, 491–492

expressions

- adding to constraints, 480–481
- in dataset filters, 637
- in SSIS, 447, 493–494

Expressions List window, 494

extracting. *See* ETL (extract, transform, and load) systems

extraction history, 524

## F

fact columns, in fact tables, 118–119, 146

fact dimension (schema), 216

fact modeling, 146–147

fact rows, in fact tables, 261

fact tables

- in Adventure Works cube, 211
- data vs. metadata, 258
- degenerate dimension, 119
- exploring source table data, 120, 146–147
- FactResellerSales example, 119
- fast load technique, 527, 528
- loading initial data into, 527–530
- multiple-source, 211
- OLAP model design example, 131–132
- pivot table view, 120
- for star schema, 117–118, 118–121
- storage space concern, 121
- types of columns, 118
- updating, 532

FactResellerSales fact table example, 119

fast load technique

- for loading initial data into dimension tables, 528
- for loading initial data into fact tables, 527
- for updating fact tables, 532

File deployment, for SSIS packages, 547

file servers, conducting baseline survey, 85

files-only installation, SSRS, 607

*Filter* MDX function, 305–307, 308

filtering

- creating filters on datasets, 637
- in data mining models, 366
- source data for data mining models, 404–405

firewalls

- conducting baseline survey, 85
- security breaches inside, 97

First (Last) NonEmpty aggregate function, 147

FirstChild aggregate function, 147

Flat File connection manager, 474

Flat File data flow destination, 485

Flat File data flow source, 483

For Loop containers, SSIS, 478

foreach enumerators, custom, 594–595

ForEach Loop containers, 476, 478

*ForEachEnumerator* class, 594

forecasting algorithms, 359

FREEZE keyword, 342

functions, 299–307, 326

Fuzzy Grouping transformation, 517–518

Fuzzy Lookup transformation, 521

## G

Gauge control, 622, 638

*GetEnumerator* method, 594

global assembly cache (GAC), registering assembly files, 590

grain statements, 128–129

granularity, defined, 128

GUI (graphic user interface), SQL Server 2008

- need for developers to master, 69–70
- for SSAS developers, 154

## H

*Head* MDX function, 316  
 health care industry, business intelligence solutions, 27–28  
 hierarchical MDX functions, 316–320  
 HOLAP (hybrid OLAP), 267, 279  
 holdout test sets, 403  
 HTTP listener, 607, 607–609

## I

*IDBCommand* interface, 597  
*IDBConnection* interface, 597  
*IDBDataParameter* interface, 597  
*IIf* MDX function, 337–338  
 IIS (Internet Information Services)  
   conducting baseline survey, 86  
   not an SSRS requirement, 606, 608, 610  
   noting version differences, 86  
 Image data region, 638–655  
 Import Data dialog box, 674  
 Import/Export Wizard  
   defined, 155  
   role in developing SSIS packages, 439  
 Inmon, Bill, 12  
*Input0\_ProcessInputRow* method, 577, 578  
 Integration Services. *See* SQL Server Integration Services (SSIS)  
 IntelliSense, 633, 637  
 Internet Information Services  
   conducting baseline survey, 86  
   not an SSRS requirement, 606, 608, 610  
   noting version differences, 86  
 iteration  
   in BI projects, 62  
   in OLAP modeling, 132

## K

key columns, in fact tables, 118  
 Key content type, 362  
 key performance indicators (KPIs)  
   accessing from KPIs tab in cube designer, 201, 228, 345  
   client-based vs. server-based, 232  
   core metrics, 229  
   creating, 345–349  
   customizing, 231–232  
   defined, 15  
   defining important metrics, 55  
   metadata browser for, 229–231  
   nesting, 229  
   overview, 149, 228–233

templates for, 229–232  
   viewing in Adventure Works cube, 228, 229  
 Key Sequence content type, 362, 363  
 Key Time content type, 362, 363  
 Kimball, Ralph, 12  
 KPIs. *See* key performance indicators (KPIs)

## L

*LastChild* aggregate function, 147  
*LastChild* MDX function, 314, 324  
*LastPeriods* MDX function, 314, 324, 325  
 law enforcement, business intelligence solutions, 29  
 least-privilege security  
   accessing source data by using, 96–97, 190  
   configuring logon accounts, 98  
   when to use, 70  
 life cycle. *See* software development life cycle  
 lift charts, 410–413  
 linked objects, 285  
 list report item, 638–655  
 load balancing, 270  
*Load* method, 596  
 loading. *See* ETL (extract, transform, and load) systems  
 local processing mode, 653, 657  
 localization, 29  
 Log Events window, 503  
 log locations, 502–503  
 log providers  
   custom, 588, 594  
   overview, 459–460  
   specifying connection manager, 502  
 logging  
   for package execution, 501–505  
   question of how much, 504–505  
   SSIS log providers, 459–460, 502  
   viewing results, 503  
 logical modeling, OLAP design concepts, 115–150  
 logical servers and services  
   conducting baseline survey, 86  
   considerations, 92–94  
   service baseline considerations, 94  
 long data type, 363  
 Lookup data flow transformation, 486  
 Lookup sample, using SSIS to load dimension and fact tables, 528–530

## M

Maintenance Plan Tasks, SSIS, 471  
 many-to-many dimension (schema), 216–217  
 Matrix data region, 638–655  
 Max aggregate function, 147  
 MDX (Multidimensional Expressions)  
   query language  
     *Ancestors* function, 319–320  
     background, 23  
     *BottomCount* statement, 311  
     *Children* function, 300, 316, 321  
     core syntax, 296–305  
     creating calculated members, 307  
     creating named sets, 308, 338–340  
     creating objects by using scripts, 309  
     creating permanent calculated members, 333–336  
     in cube designer Calculations tab, 240, 241–242  
     *CurrentMember* function, 232, 313  
     date functions, 321–326  
     defined, 23  
     *Descendants* function, 318–319, 321  
     designer included in Report Builder, 644  
     *Filter* function, 305–307, 308  
     functions, 299–307, 326  
     *Head* function, 316  
     hierarchical functions, 316–320  
     *IIf* function, 337–338  
     Internet Sales example, 295, 297–299  
     and key performance indicators, 232  
     *LastChild* function, 314, 324  
     *LastPeriods* function, 314, 324, 325  
     *Members* function, 299–300, 308  
     native vs. generated, 225  
     object names, 296  
     opening query designer, 628  
     *OpeningPeriod* function, 322–323  
     operators, 297  
     *Order* function, 302–303  
     *ParallelPeriod* function, 232, 322  
     *Parent* function, 317–318  
     *PeriodsToDate* function, 333  
     query basics, 295  
     query designer, 617  
     query designer, defined, 627  
     query designer, overview, 628–631  
     query templates, 175–178  
     querying dimension properties, 329–332

- MDX, *continued*
- Rank function, 312–314
- SCOPE keyword, 246
- scripts, 341–343
- setting parameters in queries, 631–633
- Siblings function, 317
- Tail function, 315, 330–331
- TopCount function, 310
- Union function, 320
- using with PerformancePoint Server 2007, 352–354
- using with SQL Server Reporting Services (SSRS), 349–351
- warning about deceptive simplicity, 239
- working in designer manual (query) mode, 629, 630–631
- working in designer visual (design) mode, 629–631
- Ytd function, 294
- MDX IntelliSense, 633
- measure columns, in fact tables, 118–119
- Measure Group Bindings dialog box, 213–214
- Measure Group Storage Settings dialog box, 278–279
- measure groups
  - in Adventure Works cube, 211
  - creating measures, 211
  - defined, 211
  - defining relationship to dimension data, 212–213
  - enabling writeback for partitions, 286
  - how they work, 211
  - relationship to source fact tables, 146–147
  - selecting for OLAP cubes, 219
- measure modeling, 146–147
- measures, calculated compared with derived, 148
- Members MDX function, 299–300, 308
- memory management, and SSRS, 665–666
- metadata, data flow
  - characteristics, 454–458
  - how SSIS uses, 458
  - variable-width column issue, 456–457
- metadata vs. data, 258
- Microsoft Association algorithm, 391–393, 400
- Microsoft Baseline Security Analyzer, 95
- Microsoft Biztalk Server, 22
- Microsoft Clustering algorithm, 386–389, 400
- Microsoft Decision Trees algorithm
  - in classification matrix example, 415
  - defined, 400
  - in lift chart example, 412–413
  - overview, 381–383
  - in profit chart example, 414
  - for quick assessment of source data quality, 518
  - viewers for, 369–370
- Microsoft Distributed Transaction Coordinator (MS-DTC), 508
- Microsoft Dynamics, 22, 75
- Microsoft Excel. *See* Excel
- Microsoft Excel Services. *See* Excel Services
- Microsoft Linear Regression algorithm, 383, 400
- Microsoft Logistic Regression algorithm, 395–396, 400
- Microsoft Naïve Bayes algorithm, 376–381, 399, 518
- Microsoft Neural Network algorithm, 394–395, 400
- Microsoft Office 2007
  - as data mining client, 687–721
  - installing Data Mining Add-ins, 687–688
  - optional components for BI solutions, 21–22
- Microsoft Office SharePoint Server 2007. *See* Office SharePoint Server 2007
- Microsoft PerformancePoint Server (PPS)
  - integration with SQL Server Analysis Services, 745
  - as optional component for BI solutions, 22
  - skills needed for reporting, 75
  - using MDX with, 352–354
- Microsoft Project Server, 22
- Microsoft Research, 34, 83, 478, 720
- Microsoft Security Assessment Tool, 95
- Microsoft Sequence Clustering algorithm, 389–390, 400
- Microsoft Solutions Framework (MSF). *See* also MSF (Microsoft Solutions Framework) for Agile Software Development
  - Agile Software Development version, 63–65
  - alternatives to, 62
  - building phase, 68–70
  - defined, 62
  - deploying phase, 71
  - development team roles and responsibilities, 76–83
  - envisioning phase, 65–67
  - milestones, 62
  - planning phase, 67–68
  - project phases, 62, 65–71
  - role of iteration, 62
  - spiral method, 62, 64
  - stabilizing phase, 70–71
- Microsoft SQL Server 2008. *See* SQL Server 2008
- Microsoft SQL Server 2008, Enterprise edition. *See* SQL Server 2008, Enterprise edition
- Microsoft Time Series algorithm, 383–386, 400
- Microsoft Visio. *See* Visio
- Microsoft Visual Studio. *See* Visual Studio
- Microsoft Visual Studio Team System (VSTS)
  - integrating MSF Agile into, 64–65
  - reasons to consider, 22, 546
  - Team Foundation Server, 111–112
- Microsoft Visual Studio Tools for Applications (VSTA)
  - debugging scripts, 572
  - defined, 568
  - writing Script component code, 577–582
  - writing scripts, 570–572
- Microsoft Visual Studio Tools for the Microsoft Office System (VSTO), 683–684
- Microsoft Word
  - as optional component for BI solutions, 21
  - viewing SSRS reports in, 649–650
- milestones, in Microsoft Solutions Framework (MSF), 62
- Min aggregate function, 147
- mining model viewers, 46, 48–49, 356, 368, 408
- mining structure designer
  - choosing data mining model, 365–368
  - handling nested tables, 364, 366
- Mining Accuracy Chart tab, 360, 373–375, 417
- Mining Model Prediction tab, 360, 375, 419, 424
- Mining Model Viewer tab, 46, 360, 368–373, 408, 409
- Mining Models tab, 360, 365–368, 404, 404–405

- Mining Structure tab, 360, 364–365, 404
- viewing source data, 364
- Model Designer, SSRS, backing up files, 108
- model training. *See* Data Mining Model Training destination
- modeling
  - logical modeling, OLAP design concepts, 115–150
  - OLAP. *See* OLAP modeling
  - OLTP modeling, 115, 137–138
  - physical, for business intelligence solutions, 4
- Modeling Flags property, 363
- MOLAP (multidimensional OLAP), 267, 279
- MS-DTC (Microsoft Distributed Transaction Coordinator), 508
- MsDtsSrvr.ini.xml, SSIS configuration file, 112
- MSF (Microsoft Solutions Framework) for Agile Software Development
  - background, 63–65
  - built into Microsoft Visual Studio Team System, 64–65
  - defined, 63
  - development team roles and responsibilities, 77
  - project phases, 65–71
  - suitability for BI projects, 64
- MSF (Microsoft Solutions Framework) for Capability Maturity Model Integration (CMMI), 65
- .msi files, 38
- MSReportServer\_*
  - ConfigurationSetting* class, 668
  - MSReportServer\_Instance* class, 668
- multidimensional data stores. *See* cubes, OLAP
- Multidimensional Expressions. *See* MDX (Multidimensional Expressions) query language

## N

- Name* SSIS variable property, 492
- named sets, 241, 308, 338–340
- Namespace* SSIS variable property, 492
- natural language, 67
- .NET API
  - application comparison with SSIS projects in Visual Studio, 467
  - and compiled assemblies, 196–197

- developer skills, 80, 81
- skills needed for custom client reporting, 75
- for SQL Server Integration Services, 442
- using code in SSRS reports, 647–649
- using to develop custom SSIS objects, 587–588, 588
- network interface cards (NICs), conducting baseline survey, 85
- New Cube Wizard, 134
- New Table Or Matrix Wizard, 644–646
- non-relational data, defined, 5
- normalization
  - implementing in relational data stores, 5
  - reasons for using, 6–7
  - view of OLTP database example, 5

## O

- Object Explorer
  - defined, 39
  - viewing SSAS objects from SSMS, 160
  - viewing SSIS objects from SSMS, 438, 439
- object viewers, 164
- ODBC connection manager, 473
- Office 2007
  - as data mining client, 687–721
  - installing Data Mining Add-ins, 687–688
  - optional components for BI solutions, 21–22
- Office SharePoint Server 2007
  - configuration modes for working with SSRS, 737–738
  - integrated mode, installing SSRS add-in for, 742–743
  - native mode, integration of SSRS with, 740–741
  - as optional component for BI solutions, 21–22
  - Report Center, 744–745
  - skills needed for reporting, 75
  - SQL Server business intelligence and, 723–745
  - SSRS and, 604, 736–745
  - template pages, 744–745
  - Windows SharePoint Services, 94
- Offline OLAP Wizard, 681–683
- OLAP (online analytical processing)
  - characteristics, 8
  - compared with data mining, 14
  - compared with data warehousing, 12
  - defined, 8
  - Microsoft Excel functions, 683
  - modeled as denormalized, 8
  - when to use, 8
  - working offline in Microsoft Excel, 681–683
- OLAP cubes
  - adding aggregations, 263
  - assessing source data quality, 516–518
  - background, 13
  - as BI data structure, 13
  - BIDS browser, 41–42, 201
  - building in BIDS, 198–204
  - building prototypes, 50
  - building sample using Adventure Works, 37–39
  - configuring properties, 243–254
  - connecting to sample using Microsoft Excel, 43–45
  - as core of SQL Server BI projects, 115
  - core tools for development, 157
  - creating empty structures, 133
  - as data mats, 13
  - data vs. metadata, 258
  - in data warehouses, 11
  - defined, 9
  - and denormalization concept, 125
  - vs. denormalized relational data stores, 10
  - deploying, 254–255, 260–261
  - designing by using BIDS, 183–223
  - dimensions overview, 9–10, 204–210, 257–258
  - fact (measure) modeling, 146–147
  - first, building in BIDS, 218–223
  - Microsoft Excel as client, 671–684
  - modeling logical design concepts, 115–150
  - vs. OLTP data sources, 54
  - opening sample in Business Intelligence Development Studio, 39–43
  - overview of source data options, 115–116
  - partitioning data, 263–270
  - as pivot tables, 10–11
  - pivoting in BIDS browser, 42
  - presenting dimensional data, 138–142
  - processing options, 287–291
  - and ROI of BI solutions, 56–58
  - skills needed for building, 72, 74
  - star schema source data models, 116–125

OLAP cubes, *continued*  
 UDM modeling, 9–10  
 updating, 530–533  
 using dimensions, 210–217  
 viewing by using SSMS Object Browser, 164–168  
 visualizing screen view, 10–11

OLAP modeling  
 compared with OLTP modeling, 115  
 compared with views against OLTP sources, 137–138  
 as iterative process, 132  
 naming conventions, 150  
 naming objects, 132  
 need for source control, 132  
 role of grain statements, 128–129  
 tools for creating models, 130–132, 149–150  
 using Visio 2007 to create models, 130–132, 133

OLE DB connection manager, 473  
 OLE DB data flow destination, 485  
 OLE DB data flow source, 483, 484

OLTP (online transactional processing)  
 characteristics, 6  
 defined, 5  
 normalizing data stores, 5  
 querying data stores, 6–7

OLTP modeling  
 compared with OLAP modeling, 115  
 compared with OLAP views, 137–138

OLTP table partitioning, 268–269

*OnError* event, 451

*OnExecStatusChanged* event handler, 500

*OnInit* method, 648

online analytical processing.  
*See* OLAP (online analytical processing)

online transactional processing.  
*See* OLTP (online transactional processing)

*OnPostExecute* event handler, 500, 500–501

*OnProgress* event handler, 500

*OnVariableValueChanged* event handler, 500

*OpeningPeriod* MDX function, 322–323

operating environment, conducting baseline survey, 86, 87–88

optional skills, for BI projects, 74–76

Oracle, 5, 19

*Order* MDX function, 302–303

Ordered content type, 362

Outliers Wizard, 706–707

overtraining, data model, 535

## P

Package Configuration Wizard, 549–550

Package Configurations Organizer dialog box, 548–549, 551–553

package designer  
 adding connection managers to packages, 473  
 best design practices, 509–510  
 Connection Manager window in, 468  
 control flow designer, 468  
 data flow designer, 468  
 debugging packages, 471–472  
 event handler designer, 468  
 executing packages, 471–472  
 how they work, 470–472  
 navigating, 479  
 overview, 467–469  
 Toolbox window in, 469  
 viewing large packages, 479

Package Explorer, 468–469

Package Installation Wizard, 557–558

Package Store File deployment, for SSIS packages, 547

Package Store MSDB deployment, for SSIS packages, 547

packages, in SSIS  
 adding checkpoints to, 506–507  
 adding to custom applications, 596–600  
 backups and restores, 107–108  
 best practices for designing, 509–510  
 configurations, 461  
 configuring transactions in, 507–508  
 connection managers, 448–450  
 control flow, 442–444  
 control flow compared with data flow, 444–445  
 creating with BIDS, 463–495  
 data flow, 444  
 debugging, 505–506  
 default error handling behavior, 498  
 defined, 20  
 deploying and managing by using DTUTIL utility, 441, 558  
 deployment options, 461–462, 546–558  
 developing in Visual Studio, 464–472  
 documentation standards, 525  
 Encrypt Sensitive With Password encryption option, 563  
 Encrypt Sensitive With User Key encryption option, 563  
 encrypting, 554–556  
 encryption issues, 563  
 error handling, 451–452  
 event handling, 450–451, 499–501  
 executing by using DTEXEC utility, 440  
 executing by using DTEXECUI utility, 440–441  
 expressions, 447  
 external configuration file, 548–552  
 file copy deployment, 552–553  
 handling sensitive data, 563  
 keeping simple, 508, 509  
 logical components, 442–452  
 physical components, 442  
 role of SSMS in handling, 438–439  
 saving results of Import/Export Wizard as, 439  
 scheduling execution, 558–559  
 security considerations, 97–98, 559–562  
 setting breakpoints in, 505–506  
 source control considerations, 112  
 as SSIS principal unit of work, 436, 438  
 and SSIS runtime, 452  
 tool and utilities for, 438–441  
 upgrading from earlier versions of SQL Server, 440  
 variables, 445–447  
 where to store, 98, 112

PacMan (SSIS Package Manager), 600

parallel processing, in SQL Server 2008, Enterprise edition, 269

*ParallelPeriod* MDX function, 232, 322

*Parent* MDX function, 317–318

parent tables, relationship to child tables, 5

Partition Processing destination, 485

Partition Wizard, 265–266

partitions, cube  
 defined, 263  
 defining, 265–266  
 enabling writeback, 285–286  
 overview, 263–264  
 for relational data, 268–269  
 remote, 270  
 specifying local vs. remote, 270

- in star schema source tables, 268–269
  - storage modes, 270
  - and updates, 532
  - partitions, table, 268–269
  - Pasumansky, Mosha, 340, 633
  - performance counters
    - possible problems to document, 88
    - role in creating baseline assessment of operating environment, 87–88
  - Performance Visualization tool, 510
  - PerformancePoint Server
    - integration with SQL Server Analysis Services, 745
    - as optional component for BI solutions, 22
    - skills needed for reporting, 75
    - using MDX with, 352–354
  - PeriodsToDate* MDX function, 333
  - permissions, for SSRS objects, 103–104
  - perspectives
    - compared with relational views, 227
    - defined, 149
    - overview, 227–228
  - phases, in Microsoft Solutions Framework (MSF), 62
  - physical infrastructure
    - assessing servers needed for initial BI development environment, 88
    - conducting baseline survey, 85–87
    - planning for change, 85–89
  - physical modeling, for business intelligence solutions, 4
  - physical servers
    - assessing number needed for initial BI development environment, 88–89
    - for business intelligence solutions, 4
    - conducting baseline survey, 85
    - considerations, 91–92
    - consolidation, 92
    - determining optimal number and placement for initial BI development environment, 89–94
    - development server vs. test server, 91
    - installing SQL Server, 90
    - installing SSAS, 90
    - installing SSRS, 90
    - target location options for deploying SSIS packages, 547–548
    - typical initial BI installation, 90
  - pie chart, adding to Microsoft Excel sheet, 729–731
  - PivotCharts, Microsoft Excel
    - adding to workbooks, 679–680
    - creating views, 44
  - PivotTable Field List, 650–651, 652
  - PivotTables, Microsoft Excel
    - connecting to sample cubes, 43–44
    - creating, 678–679
    - creating PivotChart views, 44
    - dimensional information, 678–679
    - formatting, 679
    - as interface for cubes, 10–11
    - overview, 675–680
    - ways to pivot view of data, 44
  - planning phase, MSF, 67–68
  - PostExecute* method, 577, 578, 587
  - PostLogMessage* method, 580
  - PPS. *See* Microsoft PerformancePoint Server (PPS)
  - precedence constraints, 480–481
  - Predict* DMX function, 421, 423–424, 425, 427
  - PredictHistogram* DMX function, 425
  - prediction algorithms, 359
  - Prediction Calculator, Microsoft Excel, 419
  - prediction functions, 423–426
  - PREDICTION JOIN syntax, 422, 427
  - predictive analytics, 148–149, 355, 366, 426
  - Predictive Model Markup Language (PMML), 409
  - PredictProbability* DMX function, 425
  - PredictProbabilityStDev* DMX function, 425
  - PredictProbabilityVar* DMX function, 425
  - PredictStDev* DMX function, 425
  - PredictSupport* DMX function, 425
  - PredictTimeSeries* DMX function, 425
  - PredictVariance* DMX function, 425
  - PreExecute* method, 577, 587
  - proactive caching
    - fine tuning, 283–284
    - notification settings, 282
    - overview, 279–282
  - Process Cube dialog box, 288–289
  - Process Progress dialog box, 407, 429–430
  - processing layer, security considerations, 97–98
  - processing time, 270
  - ProcessInput* method, 583, 584–585, 585, 586
  - ProcessInputRow* method, 584, 585, 586, 587. *See also* *Input0\_ProcessInputRow* method
  - processors, multiple, 454, 523
  - ProClarity, 22
  - product managers
    - job duties, 78
    - role and responsibility on development teams, 78
  - profit charts, 413–414
  - program managers
    - job duties, 79
    - role and responsibility on development teams, 79
  - project architects, role and responsibility on development teams, 78–79
  - Project Real, 28
  - Project Server, 22
  - Propagate* variable, 501
  - prototypes, building during MSF planning phase, 68
  - proxy accounts, 563
  - proxy servers, conducting baseline survey, 85
- Q**
- queries. *See also* DMX (Data Mining Extensions) query language; MDX (Multidimensional Expressions) query language
    - cache scopes, 326
    - challenges in OLTP data stores, 6–7
    - creating in report designer, 616–618
    - creating named sets, 338–340
    - creating permanent calculated members, 333–336
    - manually writing, 54–55
    - MDX basics, 295
    - optimizing, 326
  - query browsers
    - Cube filter, 175
    - Measure Group filter, 175
  - query designer
    - multiple types, 616–618
    - for reports, 627–638
    - setting parameters in queries, 631–633

query languages, 23–25. *See also*  
 DMX (Data Mining Extensions)  
 query language; MDX  
 (Multidimensional Expressions)  
 query language; XMLA (XML  
 for Analysis) query language  
 Query Parameters dialog box, 632  
 query templates  
   for DMX (Data Mining Extensions)  
   query language, 179–180  
 execution process, 174–175  
 for MDX (Multidimensional  
 Expressions) query language,  
 175–178  
 for XMLA (XML for Analysis) query  
 language, 180  
 query tuning, 276

## R

*RaiseChangedEvent* SSIS variable  
 property, 492  
*RangeMax* DMX function, 425  
*RangeMid* DMX function, 425  
*RangeMin* DMX function, 425  
*Rank* MDX function, 312–314  
 rapidly changing dimensions, 144  
 Raw File connection manager, 474  
 Raw File data flow destination, 485,  
 507  
 Raw File data flow source, 483,  
 484–485  
 RDBMS (relational database  
 management systems)  
   conducting baseline survey of  
   servers, 85  
   defined, 5  
   SQL Server data as source data, 19  
 RDL. *See* Report Definition  
 Language (RDL)  
 .rdl files, 108, 112–113  
 .rds files, 108  
*ReadOnlyVariables* property, 577  
*ReadWriteVariables* property, 578  
 Recordset destination, 485  
 rectangle report item, 639–655  
 regression algorithms, 359  
 Re-label Wizard, 706, 707  
 relational data  
   defined, 5  
   normalizing, 5  
   partitioning, 268–269  
   tables for denormalizing, 8  
 Relationship property, 363  
*ReleaseConnections* method, 581  
 release/operations managers  
   job duties, 83  
   role and responsibility on  
   development teams, 83  
 remote partitions, 270  
 remote processing mode, 653,  
 656–657  
 Report Builder  
   creating report, 644–646  
   defined, 19, 156  
   user interface, 643  
   version issues, 604, 643  
 Report Data window, 618  
 Report Definition Language (RDL)  
   creating metadata, 635  
   defined, 24–25  
   role in report building, 621, 623,  
   624  
   version issues, 624  
 report designer  
   adding report items to designer  
   surface, 638–639  
   backing up files, 108  
   building tabular report, 619–620  
   creating queries, 616–618  
   fixing report errors, 621  
   illustrated, 614  
   opening, 614  
   previewing reports, 620  
   tabular reports, 619–620  
   types of report data containers,  
   618  
   using Tablix data region to build  
   reports, 640–642  
   version enhancements, 19, 614  
   working with MDX query results,  
   635–643  
 Report Manager Web site, 604, 609  
 report models, 660–662  
 report processing modes, 653  
 report processing systems, 15. *See*  
*also* SQL Server Reporting  
 Services (SSRS)  
 Report Project Property Pages  
 dialog box, 623–624  
 Report Properties dialog box,  
 647–648  
 Report Server Web service  
   authentication for access,  
   610–611  
   authoring and deploying reports,  
   738–740  
   defined, 603–604  
   job manager, 612  
   overview, 609–610  
 Reporting Services. *See* SQL Server  
 Reporting Services (SSRS)  
 Reportingservices.exe.config  
 file, 108  
 reporting-type actions, 233,  
 235–236  
 reports. *See also* report processing  
 systems; SQL Server Reporting  
 Services (SSRS)  
   adding custom code to, 647–651  
   authentication credential flow in  
   SSRS, 103  
   building for SSRS, 627–646  
   cleaning and validating data for,  
   55  
   client interface considerations, 51  
   considering end-user audiences,  
   51  
   creating by using New Table Or  
   Matrix Wizard, 644–646  
   creating in SSRS, 603–624  
   creating with BIDS, 612–622  
   defining data sources, 613–614  
   defining project-specific  
   connections, 627  
   deploying to SSRS, 623–624  
   query designers for, 627–638  
   samples available, 622  
   setting parameters in queries,  
   631–633  
   Toolbox for, 621–622, 638  
   using Tablix data region to build,  
   640–642  
   viewing in Microsoft Excel, 650  
   viewing in Word, 649–650, 650  
 Reports Web site, 604  
 ReportViewer  
   control features, 652–656  
   embedding custom controls,  
   652–656  
   entering parameter values,  
   656–657  
   security credentials, 657–658  
 required skills, for BI projects, 72–74  
 restores. *See* backups and restores  
 ROLAP (relational OLAP)  
   dimensional data, 284–285  
   in Measure Group Storage  
   Settings dialog box, 279  
   overview, 267–268  
 roles, in SSAS, 195–196  
 .rptproj files, 108  
 rsconfig.exe tool, 604  
 rs.exe tool, 609  
 Rsmgrpolicy.config file, 108  
 Rreportsrvr.config file, 108  
 Rssvrpolicy.config file, 108  
 runtime, SSIS, 452

## S

- Sample Data Wizard, 705, 707–708
- Save Copy Of Package dialog box, 554–556
- scalability, and SSRS, 662–664
- scaling out, in SQL Server 2008, Enterprise edition, 666–667
- schema-first approach to BI design, 130
- SCOPE keyword, MDX query language, 246, 341–342
- Scope SSIS variable property, 492
- Script component. *See also* Script Transformation Editor dialog box
  - compared with Script task, 567–568
  - connection managers in, 580–581
  - as data source, 573
  - debugging, 587
  - destination-type, 586–587
  - selecting Transformation type option, 574
  - source-type, 582
  - synchronous and asynchronous transformation, 582–586
  - type options, 573–574
  - writing code, 577–581
- Script task
  - compared with Script component, 567–568
  - defined, 478
  - using to define scripts, 568–570
- Script Task Editor dialog box, 568–570
- Script Transformation Editor dialog box
  - Connection Managers page, 580
  - Inputs And Outputs page, 576, 578–579, 579
  - Input Columns page, 574–576
  - opening, 574
- scripting
  - limitations, 587
  - Script task compared with Script component, 567–568
  - for SSRS administrative tasks, 667–669
- ScriptLanguage* property, 568
- scripts, MDX, 341–343
- security. *See also* least-privilege security
  - best practices, 70, 564
  - BIDS, for solutions, 98–99
  - BIDS, when creating SSIS packages, 98
  - custom client considerations, 104–106
  - encrypting packages when deploying, 554–556
  - handling sensitive SSIS package data, 563
  - overview of SSIS package issues, 559–562
  - passing report credentials through ReportViewer, 657–658
  - proxy execution accounts for SSIS packages, 79
- Security Assessment Tool, 95
- security requirements
  - in development environment, 70
  - overview, 95–106
- Select Script Component type dialog box, 573–574
- semiadditive behavior, configuring in Business Intelligence Wizard, 244, 250
- sequence analysis algorithms, 359
- Sequence containers, 478
- servers. *See* logical servers and services; physical servers
- service level agreements (SLAs)
  - availability strategies, 87
  - conducting baseline survey, 87
  - reasons to create in BI projects, 87
- Service Principal Name (SPN), 159
- Shared Data Source Properties dialog box, 613–614
- SharePoint Server. *See* Office SharePoint Server 2007
- Siblings* MDX function, 317
- signing assemblies, 589
- skills, for BI projects
  - optional, 74–76
  - required, 72–74
- .sln files, 98
  - backing up, 108
- Slowly Changing Dimension transformation, 533
- Slowly Changing Dimension Wizard, SSIS, 143–144
- slowly changing dimensions (SCD), 142–144
  - .smdl files, 108, 112–113
  - snapshots, database, 507
- snowflake schema
  - DimCustomer example, 134
  - on Dimension Usage tab of cube designer, 135, 215
  - overview, 134
  - when to use, 136–137
- software development life cycle, 61–71
- Solution Explorer, in BIDS, 40, 46, 184, 186–188
- Solution Explorer, Visual Studio
  - configuring SSAS object properties, 243
  - data sources and data source views, 466, 467
  - SSIS Packages folder, 466–467
  - viewing SSIS projects in, 466–467
- solutions, defined, 98, 539
- SOLVE\_ORDER keyword, 343–344
- source code control/source control systems, 540–542
- source control, 111–113, 132
- source data
  - accessing by using least-privileged accounts, 96–97
  - cleaning, validating, and consolidating, 69
  - collecting connection information, 96–97
  - loading into decision tables, 525–526
  - non-relational, 5
  - performing quality checks before loading mining structures and models, 533–534
  - querying OLTP data stores, 6–7
  - relational, 5
  - structure names, 68
  - transformation issues, 519–523
- source data systems, upgrading to SQL Server 2008, 89
- Specify A Unary Operator, Business Intelligence Wizard, 244, 248–250
- Specify Attribute Ordering, Business Intelligence Wizard, 244, 250
- spiral method, 62, 64
- SQL Server 2000, upgrading
  - production servers for SSAS, SSIS, and SSRS, 90
- SQL Server 2008
  - command-line tools installed, 157
  - complete installation, 156–157
  - as core component of Microsoft BI solutions, 16, 19
  - customizing data display, 3
  - Database Engine Tuning Advisor, 8
  - documenting sample use, 86
  - downloading and installing Data Mining Add-ins for Office 2007, 47–48
  - downloading and installing sample databases, 154
  - feature differences by edition, 37, 58

- SQL Server 2008, *continued*
  - installing sample databases, 37–41
  - installing samples in development environment, 86
  - minimum-installation paradigm, 153–154
  - new auditing features, 111
  - online transactional processing, 5–8
  - security features, 70
- SQL Server 2008, Enterprise edition
  - parallel processing, 269
  - scaling out, 666–667
- SQL Server Agent, 558–564
- SQL Server Analysis Services (SSAS). *See also* SQL Server Management Studio (SSMS); SQL Server Profiler
  - aggregation types, 147
  - background, 16–17
  - backups and restores, 106–107
  - baseline service configuration, 157–159
  - BIDS as development interface, 16
  - BIDS as tool for developing cubes, 16
  - building sample OLAP cube, 37–39
  - considering where to install, 89
  - as core component of Microsoft BI solutions, 16
  - core tools, 153–181
  - creating data mining structures, 18
  - creating roles, 195–196
  - credentials and impersonation
    - information for connecting to data source objects, 98–99
  - Cube Wizard, 128
  - data source overview, 188–190
  - data source views (DSVs), 190–195
  - database relationship between cubes and dimensions, 257–258
  - default assemblies, 197
  - defined, 16
  - deploying Adventure Works 2008 to, 41
  - Deployment Wizard, 155
  - dimension design in, 140–142
  - documenting service logon
    - account information, 94
  - exploring fact table data, 120
  - installing, 153
  - installing multiple instances, 90
  - linked objects, 285
  - logon permissions, 98
  - mastering GUI, 69–70
  - performance counters for, 87
  - providers for star schema source data, 117
  - query designers, using for
    - creating reports, 627–638
  - querying objects in SSMS, 170–175
  - reasons for installing SQL Server Database Engine Services with, 153
  - as requirement for OLAP BI solutions, 23
  - roles in, 195–196
  - scaling to multiple machines, 91
  - security considerations, 98–99
  - source control considerations, 112, 113
  - SSMS as administrative interface, 16
  - using compiled assemblies with objects, 196–197
  - using OLAP cubes vs. OLTP data sources, 54
  - viewing configuration options in SSMS, 93
  - viewing SSAS objects in Object Explorer, 160
  - viewing what is installed, 153–154
  - working on databases in BIDS in connected mode, 261
- SQL Server Books Online, defined, 156
- SQL Server Compact destination, 485
- SQL Server Configuration Manager, 94, 155, 157–158
- SQL Server Database Engine, 108
- SQL Server Database Engine Tuning Advisor, 8, 156
- SQL Server destination, 485
- SQL Server Enterprise Manager, 463–464
- SQL Server Error and Usage Reporting, defined, 155
- SQL Server Installation Center, defined, 156
- SQL Server Integration Services (SSIS)
  - architectural components, 435–462
  - architectural overview, 436–438
  - backups and restores, 107–108, 112
  - BIDS as development interface, 16
  - BIDS as tool for implementing packages, 20
  - comparison with Data Transformation Services, 446, 463–464
  - considering where to install, 89
  - creating ETL packages, 55
  - in custom applications, 596–600
  - custom task and component development, 587–595
  - data flow engine, defined, 436
  - data flow engine, overview, 453–459
  - and data mining by DMX query, 426–428
  - data mining object processing, 430
  - defined, 20
  - documenting service logon
    - account information, 94
  - error handling, 497–499
  - ETL skills needed, 76
  - event handling, 499–501
  - history, 437–438
  - as key component in Microsoft BI solutions, 20
  - log providers, 459–460
  - mastering GUI, 69–70
  - MsDtsSrvr.ini.xml configuration file, 112
  - .NET API overview, 442
  - object model and components, 442–452
  - object model, defined, 436
  - package as principal unit of work, 436, 438
  - performance counters for, 87
  - relationship to Data Transformation Services, 437
  - runtime, defined, 436
  - runtime, overview, 452
  - scaling to multiple machines, 91
  - scripting support, 567–587
  - security considerations for packages, 97–98
  - service, defined, 436
  - Slowly Changing Dimension Wizard, 143–144
  - solution and project structures, 539–540
  - source control considerations, 112
  - SSMS as administrative interface, 16
  - upgrading packages from earlier versions of SQL Server, 440
  - ways to check data quality, 516–518
- SQL Server Management Studio (SSMS)
  - backups of deployed SSAS solutions, 106–107
  - connecting to SSAS in, 160

- as core tool for developing
    - OLAP cubes and data mining structures in SSAS, 157
  - data mining object processing, 431
  - defined, 19, 155, 160
  - menus in, 161
  - object viewers, 164
  - opening query editor window, 295
  - processing OLAP objects, 163
  - querying SSAS objects, 170–175
  - role in handling SSIS packages, 438–439
  - verifying Adventure Works installation, 39
  - viewing configuration options available for SSAS, 93
  - viewing data mining structures, 164, 168–170
  - viewing dimensions, 163
  - viewing OLAP cubes, 164–168
  - viewing OLAP objects, 162–164
  - working with SSIS Service, 564–565
- SQL Server Profiler
- as core tool for developing
    - OLAP cubes and data mining structures in SSAS, 157
  - defined, 156
  - how query capture works, 172–174
  - overview, 171–172
  - role in designing aggregations, 275–277
  - using for access auditing, 109–110
- SQL Server Reporting Services (SSRS)
- adding custom code to reports, 647–649
  - architecture, 603–605
  - authentication credential flow for reports, 103
  - background, 19
  - backups and restores, 108
  - BIDS as development interface, 16
  - building reports for, 627–646
  - command-line utilities, 604
  - Configuration Manager, 102, 108, 155, 607, 609, 737
  - configuring environment for report deployment, 623–624
  - configuring with Office SharePoint Server, 737–738
  - considering where to install, 89
  - as core component of Microsoft BI solutions, 16
  - creating reports, 603–624
  - defined, 19
  - deploying reports, 623–624
  - documenting service logon
    - account information, 94
  - feature differences by edition, 606
  - installing add-in for Microsoft Office SharePoint Server, 742–743
  - installing and configuring, 606–612
  - integration with Office SharePoint Server, 604
  - mastering GUI, 69–70
  - performance counters for, 87
  - query design tools, 616–618
  - Report Builder, 19
  - and scalability, 662–664
  - scaling to multiple machines, 91
  - security decisions during installation and setup, 102–104
  - skills needed for reporting, 73, 75
  - source control considerations, 112–113
  - SSMS as administrative interface, 16
  - storing metadata, 604
  - using in SharePoint integrated mode, 742–743
  - using in SharePoint native mode, 740–741
  - using MDX with, 349–351
  - Web site interface, 19
  - Windows Management Instrumentation, 668–669
  - SQLPS.exe tool, 157
  - SSAS. *See* SQL Server Analysis Services (SSAS)
  - SSAS Deployment Wizard, 155
  - SSIS. *See* SQL Server Integration Services (SSIS)
  - SSIS Package Manager (PacMan), 600
  - SSIS Package Store, 552, 554, 564
  - SSIS Package Upgrade Wizard, 440
  - SSIS Performance Visualization tool, 510
  - SSIS Service, 564–565
  - SSMS. *See* SQL Server Management Studio (SSMS)
  - SSRS. *See* SQL Server Reporting Services (SSRS)
  - SSRS Web Services API, 658–659
  - stabilizing phase, MSF, 70–71
  - staging databases, when to use, 520–523, 524, 531
  - star schema
    - comparison with non-star designs, 215–217
    - conceptual view, 125
    - for denormalizing, 30
    - dimension tables, 117–118, 121–125, 194
    - Dimension Usage tab, in cube designer, 126–127, 134–135
    - fact tables, 117–118, 118–121, 194
    - Microsoft changes to feature, 210–211
    - moving source data to, 525–530
    - for OLAP cube modeling, 116–125
    - on-disk storage, 116–117
    - physical vs. logical structures, 116–117
    - reasons to create, 126, 126–127
    - tables vs. views, 116–117
    - visualization, 117–118
  - storage area networks, 91
  - Subreport data region, 638–655
  - Sum aggregate function, 147
  - Synchronize command, to back up and restore, 107
  - synchronous data flow outputs, 458, 459
  - synchronous transformation, 583–586
  - SynchronousInputID* property, 578–579
  - system variables, in SSIS, 445–446, 493
- ## T
- Table content type, 362
  - Table data region, 638–655
  - table partitioning, defined, 269. *See also* OLTP table partitioning
  - tables
    - parent vs. child, 5
    - relational, for denormalizing, 8
  - Tablix container, defined, 622
  - Tablix data region, defined, 639–642
  - tabular report designer, 619–620
  - Tail* MDX function, 315, 330–331
  - Task* class, 591–592
  - Task Host containers, 478
  - tasks
    - compared with components, 444, 445
    - custom, 587–588
    - default error handling behavior, 499
    - in SSIS package control flow, 442–444
  - taxonomies
    - documenting, 67–68
    - role in naming of OLAP objects, 132

Team Foundation Server, 38, 540  
 teams. *See* development teams  
 Template Explorer, 174, 422  
 test managers  
   job duties, 81  
   keeping role separate from developer role, 81  
   role and responsibility on development teams, 81–82  
 testing. *See* stabilizing phase, MSF  
 testing plans, 70–71  
 text data type, 363  
 Textbox data region, 638–655  
*This* function, 342  
 time intelligence, configuring in Business Intelligence Wizard, 243, 245  
 Toolbox, SSIS  
   adding objects to, 591  
   overview, 469  
 Toolbox, SSRS, in BIDS, 621–622, 638  
 Toolbox, Visual Studio, 652, 654  
 tools  
   ascmd.exe tool, 157  
   BIDS Helper tool, 255, 490, 494, 510  
   downloading from CodePlex Web site, 37–38, 86, 157  
   DTEXEC utility, 440  
   DTEXECUI utility, 440–441  
   DTUTIL utility, 441  
   installed with SQL Server 2008, 157  
   rsconfig.exe tool, 604  
   rs.exe tool, 609  
   SQLPS.exe tool, 157  
*TopCount* MDX function, 310  
 Trace Properties dialog box, 276  
 Tracer utility, Microsoft Excel, 101  
 transactional activities. *See* OLTP (online transactional processing)  
 transactions, package, 507–508  
 Transact-SQL  
   aggregate functions, 9  
   queries, 54–55  
 transformation components, 486–488  
 transformations, built-in, 578  
 transforming. *See* ETL (extract, transform, and load) systems  
 translations  
   for cube metadata, 225–226  
   SSAS, defined, 149

## U

UDM (Unified Dimensional Model), 9–10, 138  
 unary operators, specifying in Business Intelligence Wizard, 244, 248–250  
 Unified Dimensional Model (UDM), 9–10, 138  
*Union* MDX function, 320  
 Upgrade Package Wizard, 487  
 upgrading SSIS packages from earlier versions of SQL Server, 440  
 URLs (uniform resource locators)  
   enhanced arguments, 651  
   implementing access, 651–652  
 Usage-Based Optimization Wizard, 274–275  
 user experience managers, role and responsibility on development teams, 82, 82–83  
 user interfaces (UIs)  
   role and responsibility of user experience managers, 82–83  
   skills needed for creating, 73, 75  
 utilities  
   ascmd.exe tool, 157  
   BIDS Helper tool, 255, 490, 494, 510  
   downloading from CodePlex Web site, 37–38, 86, 157  
   DTEXEC utility, 440  
   DTEXECUI utility, 440–441  
   DTUTIL utility, 441  
   installed with SQL Server 2008, 157  
   rsconfig.exe tool, 604  
   rs.exe tool, 609  
   SQLPS.exe tool, 157

## V

*Validate* method, 592, 594  
*Value* SSIS variable property, 492–493  
*ValueType* SSIS variable property, 493  
 variables, in SSIS  
   adding to packages, 490  
   *Description* property, 491  
   differences related to SSIS platform, 490–493  
   *EvaluateAsExpression* property, 491  
   *Expression* property, 491–492  
   *Name* property, 492  
   *Namespace* property, 492

opening Variables window, 490  
 overview, 445–447  
 properties, 491–493  
*RaiseChangedEvent* property, 492  
*Scope* property, 492  
 system, 493  
*Value* property, 492–493  
*ValueType* property, 493  
 ways to use, 494–495  
 variable-width columns, in data flow metadata, 456–457  
 Virtual PC, setting up test configurations, 37  
 virtualization, 4, 91  
 Visio  
   adding Data Mining template, 47–48  
   data mining integration, 714–718  
   data mining integration functionality, 689–690  
   Data Mining toolbar, 715–718  
   as optional component for BI solutions, 21  
   using to create OLAP models, 130–132, 133  
 Vista. *See* Windows Vista, IIS version differences  
 Visual SourceSafe (VSS)  
   checking files in and out, 544–545  
   creating and configuring VSS database, 541–542  
   creating and configuring VSS user accounts, 542  
   History dialog box, 545  
   Lock-Modify-Unlock Model option, 541–542  
   overview, 540  
   storing solution files, 542–544  
   using Add SourceSafe Database Wizard, 541–542  
 Visual Studio. *See also* Solution Explorer, Visual Studio  
 Adventure Works.sln file, 38, 40  
 embedding custom ReportViewer controls, 652–656  
 as location for SSIS package development, 464–472  
 relationship to BIDS, 16–17, 22, 41, 463  
 relationship to SQL Server Integration Services, 440  
 resemblance to BIDS interface, 157  
 signing custom object assemblies, 589  
 SSIS menu, 472  
 SSIS package designers, 467–472  
 Toolbox, 652, 654

- usefulness in BI development, 86
- viewing new SSIS project template in, 465–466
- VSTO (Microsoft Visual Studio Tools for the Microsoft Office System), 683–684
- VSTS (Visual Studio Team System) integrating MSF Agile into, 64–65
- reasons to consider, 22, 546
- Team Foundation Server, 111–112

## W

- Warnings tab, in BIDS, 259
- waterfall method, 62
- Web Parts, 21–22, 740–741
- Web servers, conducting baseline survey, 85
- Web Services API, and Excel Services, 732–736
- Windows Communication Foundation (WCF), 732–733
- Windows Management Instrumentation (WMI), 668–669

- Windows Reliability and Performance Monitor tool, 87
- Windows Server 2003, IIS version differences, 86
- Windows Server 2008
  - IIS version differences, 86
  - Performance Monitor counters, 523
  - Reliability and Performance Monitor tool, 87
  - virtualization improvements, 4
- Windows SharePoint Services and Office SharePoint Server 2007, 94
- Windows Vista, IIS version differences, 86
- Windows-on-Windows 32-bit applications, 91
- Word 2007
  - as optional component for BI solutions, 21
  - viewing SSRS reports in, 649–650
- writeback
  - defined, 98
  - overview, 145

- storing changes to dimensions, 285–286

## X

- XML data flow source, 483
- XML for Analysis. *See* XMLA (XML for Analysis) query language
- XMLA (XML for Analysis) query language
  - background, 24
  - defined, 24
  - query templates, 180
  - source control considerations, 113
  - using for data mining object processing, 431
  - viewing scripts, 164

## Y

- Ytd* MDS function, 294



# About the Authors

Several authors contributed chapters to this book.

## Lynn Langit

Lynn Langit is a developer evangelist for Microsoft. She works mostly on the West Coast of the United States. Her home territory is southern California. Lynn spends most of her work hours doing one of two things: speaking to developers about the latest and greatest technology, or learning about new technologies that Microsoft is releasing. She has spoken at TechEd in the United States and Europe as well as at many other professional conferences for developers and technical architects. Lynn hosts a weekly webcast on MSDN Channel 9 called “geekSpeak.” She is a prolific blogger and social networker. Lynn’s blog can be found at <http://blogs.msdn.com/SoCalDevGal>.

Prior to joining Microsoft in 2007, Lynn was the founder and lead architect of her own company. There she architected, developed, and trained technical professionals in building business intelligence (BI) solutions and other .NET projects. A holder of numerous Microsoft certifications—including MCT, MCITP, MCDBA, MCS.D.NET, MCSE, and MSF—Lynn also has 10 years’ experience in business management. This unique background makes her particularly qualified to share her expertise in developing successful real-world BI solutions using Microsoft SQL Server 2008. This is Lynn’s second book on SQL Server business intelligence.

In her spare time, Lynn enjoys sharing her love of technology with others. She leads Microsoft’s annual DigiGirlz day and camp in southern California. DigiGirlz is a free educational program targeted at providing information about careers in technology to high-school girls. Lynn also personally volunteers with a group of technical professionals who provide support to a team of local developers building and deploying an electronic medical records system (SmartCare) in Lusaka, Zambia. For more information about this project, go to <http://www.opensmartcare.org>.

## Davide Mauri

Davide wrote Chapters 15 and 16 in the SQL Server Integration Services (SSIS) section and kindly reviewed Lynn’s writing for the remainder of the SSIS chapters.

Davide holds the following Microsoft certifications: MCP, MCAD, MCDBA, Microsoft Certified Teacher (MCT), and Microsoft Most Valued Professional (MVP) on SQL Server. He has worked with SQL Server since version 6.5, and his interests cover the whole platform, from the relational engine to Analysis Services, and from architecture definition to performance tuning.

Davide also has a strong knowledge of XML, .NET, and the object-oriented design principles, which provides him with the vision and experience to handle the development of complex business intelligence solutions.

Having worked as a Microsoft Certified Teacher for many years, Davide is able to share his knowledge with his co-workers, helping his team deliver high-quality solutions. He also works as a mentor for Solid Quality Mentors and speaks at many Italian-language and international BI events.

## Sahil Malik

Sahil Malik wrote Chapter 25 in the SQL Server Reporting Services (SSRS) section.

Sahil, the founder and principal of Winsmarts, has been a Microsoft MVP and INETA speaker for many years. He is the author of many books and numerous articles. Sahil is a consultant and trainer who delivers training and talks at conferences internationally.

## Kevin Goff

Kevin wrote Chapters 10 and 11 on MDX in the SQL Server Analysis Services (SSAS) section.

## John Welch

John Welch was responsible for the technical review of this book.

John is Chief Architect with Mariner, a consulting firm specializing in enterprise reporting and analytics, data warehousing, and performance management solutions. He has been working with business intelligence and data warehousing technologies for six years, with a focus on Microsoft products in heterogeneous environments. He is a Microsoft MVP, an award given to him for his commitment to sharing his knowledge with the IT community. John is an experienced speaker, having given presentations at Professional Association for SQL Server (PASS) conferences, the Microsoft Business Intelligence conference, Software Development West (SD West), the Software Management Conference (ASM/SM), and others.

John writes a blog on business intelligence topics at <http://agilebi.com/cs/blogs/bipartisan>. He writes another blog focused on SSIS topics at <http://agilebi.com/cs/blogs/jwelch/>. He is also active in open source projects that help make the development process easier for Microsoft BI developers, including BIDS Helper (<http://www.codeplex.com/bidshelper>), an add-in for Business Intelligence Development Studio that adds commonly needed functionality to the environment. He is also the lead developer on ssisUnit (<http://www.codeplex.com/ssisUnit>), a unit testing framework for SSIS.

