Chapter 3  Connecting to DBMSs  37

DBMS Connections  37
Connection Files in BI Query  38
Types of Connections  38
Connection Files in Use  43

Creating Connection Files  43
About Governor Settings with Connection Files  44
Specifying Governor Settings with Connection Files  45
Optimizing Connections with Row Buffering  46
Testing Connection Files  47

Managing Connection Files  48
Connection File Permissions  48
Setting the Default Connection File  49
Setting Automatic Connections  50
Connections to Multiple Databases  51
Assigning Connection Files to Design Windows  51
Setting the Location of the Connection File Folder  52
Updating, Editing, and Deleting Connection Files  53

Publishing Connection Files  53
Security for Connection Files  54
Retrieving Connection Files  55

Connecting to DBMSs  56
Guidelines for Connecting to DBMSs  57
Disconnecting from DBMSs  58
# Contents

## Chapter 4 Planning Data Models

**Assessing Users** ................................................. 63
  Types of User Requirements .................................. 64

**Designing Data Models** ........................................... 65
  Physical Data Models ........................................... 66
  Logical Data Models ............................................ 66
  Ad Hoc Data Models ............................................. 67
  Ad Hoc/Report Data Models .................................... 67
  Modular Data Models ............................................ 68
  Executive Information System Data Models .................. 69

**Optimizing the Database** ........................................ 70
  Creating Lookup Tables ........................................ 70
  Adding Indices .................................................. 71
  Denormalizing Tables ........................................... 71
  Storing Common Summarizations ............................... 72
  Reorganizing Table Structures ................................ 72
  Providing Metadata ............................................. 74

## Chapter 5 Creating Data Models

**Main Components of Data Models** ............................ 76

**Creating Data Models** ........................................... 77
  Opening New Data Models ....................................... 78

**Creating Design Windows** ....................................... 78
Deleting or Renaming a Design Window ........................................ 79

Creating Data Objects .............................................................. 79
  Loading Tables from the Database ......................................... 80
  Data Object and Attribute Names ......................................... 81
  Guidelines for Renaming Data Objects and Attributes ............ 81
  Simplifying Display Names Automatically .............................. 82
  Renaming Data Objects Manually ......................................... 83
  Renaming Attributes Manually ............................................ 83
  Organizing Attributes ....................................................... 84

Creating Relationships Between Data Objects ......................... 85
  Types of Relationships ..................................................... 85
  Relationship Icons and Connecting Lines ............................... 87
  Creating Relationships Automatically ................................... 87
  Creating Embedded Relationships ....................................... 89
  Creating Actual Relationships ............................................ 89
  Connecting Data Objects to an Actual Relationship ............... 89
  Naming Relationships ....................................................... 90

Specifying Join Conditions .................................................... 91
  Types of Joins .................................................................. 92
  Join Requirements ........................................................... 94
  Available Outer Join Syntax Types ...................................... 95
  Outer Join Check Boxes ..................................................... 96
  Creating Join Conditions .................................................... 97
  Including Prompts and Variables in Join Conditions ............. 98
  Troubleshooting Missing Outer Join Operators ..................... 99
  Editing Join Conditions ...................................................... 100

Duplicating Data Objects ....................................................... 101
  Copying and Pasting Tables ................................................ 103
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifying Correlation Names</td>
<td>103</td>
</tr>
<tr>
<td><strong>Merged Data Objects</strong></td>
<td>104</td>
</tr>
<tr>
<td>When to Merge Data Objects</td>
<td>105</td>
</tr>
<tr>
<td>Merging Data Objects</td>
<td>106</td>
</tr>
<tr>
<td>Editing Merged Objects</td>
<td>107</td>
</tr>
<tr>
<td>Unmerging Data Objects</td>
<td>108</td>
</tr>
<tr>
<td><strong>Query Data Objects (QDOs)</strong></td>
<td>108</td>
</tr>
<tr>
<td>Converting Queries into Data Objects</td>
<td>109</td>
</tr>
<tr>
<td>Editing Query Data Objects</td>
<td>110</td>
</tr>
<tr>
<td><strong>Correlated Subqueries in BI Query</strong></td>
<td>111</td>
</tr>
<tr>
<td>Creating Correlated Subqueries with QDOs</td>
<td>112</td>
</tr>
<tr>
<td><strong>Creating Tables</strong></td>
<td>113</td>
</tr>
<tr>
<td>Creating Data Objects</td>
<td>114</td>
</tr>
<tr>
<td>Creating Tables Directly in the Database</td>
<td>115</td>
</tr>
<tr>
<td>Reloading and Populating Tables</td>
<td>116</td>
</tr>
<tr>
<td>Creating Tables Using SQL Scripts</td>
<td>116</td>
</tr>
<tr>
<td><strong>Checking the Design</strong></td>
<td>117</td>
</tr>
<tr>
<td><strong>Securing Data Models</strong></td>
<td>118</td>
</tr>
<tr>
<td><strong>Chapter 6</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Creating Effective Designs for Data Models</strong></td>
<td>119</td>
</tr>
<tr>
<td><strong>Basic Design Principles</strong></td>
<td>119</td>
</tr>
<tr>
<td>Considering Audience and Content</td>
<td>120</td>
</tr>
<tr>
<td>Consistency in Design</td>
<td>120</td>
</tr>
<tr>
<td>Restraint</td>
<td>121</td>
</tr>
<tr>
<td>Contrast</td>
<td>122</td>
</tr>
<tr>
<td>Providing Unity</td>
<td>122</td>
</tr>
<tr>
<td><strong>Establishing a Layout</strong></td>
<td>123</td>
</tr>
<tr>
<td>Available BI Query Design Tools</td>
<td>123</td>
</tr>
</tbody>
</table>
Identifying Selected Objects in Design Mode ........................................ 123
Setting the Design Window Size ....................................................... 124
Changing Layout Grid Options ......................................................... 125
Moving Objects Within a Design Window ........................................... 125
Aligning Objects ................................................................. 126
Making Objects the Same Size ...................................................... 127
Spacing Objects Evenly .............................................................. 127
Copying and Pasting Objects .......................................................... 127
Deleting Objects ................................................................. 128
Repeating Objects ................................................................. 129
Kinking Relationship Lines .......................................................... 130

Adding Emphasis ................................................................. 131
Typeface ................................................................. 131
White Space ................................................................. 132
Boxes ................................................................. 133
Graphics ................................................................. 133
Color ................................................................. 133

Adding Ornaments and Other Objects ........................................... 134
Creating Ornaments .............................................................. 135
Special Text Strings in Ornaments and Buttons ............................... 136
Copying Ornaments from the Clipboard ........................................ 137
Creating Shape Ornaments ...................................................... 137
Editing Shape Ornaments ....................................................... 138
Assigning Icons to Data Objects .................................................. 139

Specifying Colors ............................................................... 140
Specifying Window Colors .......................................................... 140
Specifying Object and Design Window Colors ................................. 141
Specifying Colors for Selected Objects ........................................... 142
## Contents

Specifying Colors for Individual Design Windows ........................................ 142
Creating New Colors .................................................................................. 143
Transparent Objects .................................................................................... 143

**OLE Objects as Ornaments** ................................................................. 144

- OLE Requirements and Limitations ......................................................... 145
- Types of OLE Objects ............................................................................. 145
- Linking Objects ....................................................................................... 146
- Embedding Objects ............................................................................... 147
- Editing Imported Objects ....................................................................... 147
- Viewing and Managing Links ................................................................ 148

### Chapter 7 Adding Buttons to Data Models ...................................... 151

**Buttons in BI Query** ............................................................................ 151
- Creating Buttons .................................................................................... 152

**Specifying Button Appearance** ......................................................... 153

**Linking Buttons** .................................................................................... 155
- Link Targets for Query Buttons ............................................................. 155
- Linking Buttons to Queries .................................................................... 156
- Linking Buttons to Windows ................................................................ 157
- Buttons and OLE Automation ................................................................ 157
- Linking Buttons to Applications, Documents, or Scripts ..................... 158
- Linking Buttons to Reports or Hypercubes .......................................... 159
- Path Variables in Button Links .............................................................. 160
- Creating, Modifying, and Deleting Path Variables .............................. 161
- Hyperlinks to URLs .............................................................................. 162
- Linking Buttons to URLs ...................................................................... 165
Chapter 8 Controlling User Access

Types of User Access ................................................................. 169
Setting Data Model Passwords .................................................. 170
  Setting Data Model Passwords ................................................ 170
  Changing or Removing Passwords ............................................ 171
  Retrieving Forgotten User Passwords ...................................... 171
Controlling Access to the Data .................................................. 172
  Securing Access to Columns ................................................... 173
  Object Qualifiers ................................................................. 174
  Securing Access to Rows ........................................................ 176
  Securing Access to Data Values .............................................. 177
Setting Security for Individual Users ........................................... 178
  Setting Security with BI Server .............................................. 178
  Setting Security with Variables ............................................. 179
  Associating Data with User Names .......................................... 180
  Qualifying Data Objects with User or Group Variables ............. 182
Setting BI Query Permissions .................................................. 183
  Suggested Permission Settings for Novice Users ...................... 183
  Suggested Permission Settings for Advanced Users ................. 184
  Permissions for Editing and Saving Data Models ..................... 184
  Permissions for Modifying SQL Strings .................................. 185
  Calculated Attributes Permissions ......................................... 186
  Save Queries Permission ..................................................... 186
  Dynamic Relationship Permissions ........................................ 186
  Freehand Queries Permission ............................................... 187
  Custom Checks Permission .................................................. 187
  Update Database Permission ............................................... 187
  Submit to Table Permission ................................................ 188
Chapter 9  Providing User Help  191

Types of Help  .................................................. 191
Metadata  ......................................................... 192
  Providing Metadata in a Data Model  ......................... 192
  How BI Query Displays Metadata  ............................ 193
Adding Metadata to the Database  ............................... 193
Specifying Metadata Sources  .................................. 195
  Basic SQL for Metadata Queries  ............................ 195
  SQL for Data Object Metadata  ............................... 196
  SQL for Attribute Metadata  .................................. 196
  Creator and Database Names in Metadata Queries ...... 197
  Creating and Editing Metadata Queries .................... 198
Loading Metadata  .............................................. 199
Providing Attribute Descriptions  ............................ 200
  Adding Descriptions to the Attribute Dialog Box .... 201
  The Attribute Descriptions Table  ......................... 201
  Creating the Attribute Descriptions Table ............. 202
  Populating the Attribute Description Table ............ 203
  Creating the Attribute Description Query ............... 204
  Creating Attribute Descriptions Using Results Files 205
Annotating Existing Help  ...................................... 206
Creating Additional Help Files  .............................. 206
## Chapter 10  Saving and Distributing Data Models  209

**Saving Data Models** ................................................. 209
- Locations for Saving Data Models .......................... 210
- Saving Split Data Models ................................ 210
- Saving Combined Data Models ............................... 211

**Distributing Combined Data Models** ......................... 211

**Distributing Split Data Models** ............................... 212
- Storing Each Layer in Separate Folders ................. 212
- Establishing a Path to the Administrator-Layer File  .... 213

**Methods for Distributing Data Models** ..................... 213
- Distributing with File Servers ............................ 214
- Distributing by Diskette ................................. 215
- Distributing by E-Mail .................................. 215

**Distributing Revisions by Saving to the Database** ......... 216
- Save-to-Database Tables ................................ 217
- Attributes in the GQLMeta2 Table ....................... 218
- Attributes in Other Save-to-Database Tables .......... 219
- Granting Permissions to Save-to-Database Tables ...... 219
- Saving Initial Models to the Database .................. 219
- Saving Revised Models to the Database ................. 220

## Chapter 11  Publishing Data Models with BI Server  223

**Publishing with BI Server** .................................. 223
- Requirements for Publishing Data Models ............... 223
- Data Model Packages ..................................... 224
- Preparing to Publish Data Models ....................... 225
- Publishing Data Models .................................. 226
Contents

Setting Data Model Visibility ........................................... 227
Granting BI Web Ad Hoc Querying ................................. 228

Controlling Access with BI Server ................................. 229
Users and Groups in BI Server .................................. 230
Access Inheritance Rules ............................................. 230
The Set Security Dialog Box ........................................ 231
Setting Security on Published Items .............................. 232

Chapter 12  Balancing Network Load ............................. 233
Methods for Balancing Network Load ......................... 233
Running Efficient Queries ......................................... 234
Enforcing Connect per Query ....................................... 234
Qualifying with Data Values Files ............................... 235
Restricting Queries .................................................... 235
Submitting Queries to Database Tables ....................... 236
Modifying the SQL to Improve Query Efficiency ......... 236
Setting Query Limits with Governors ......................... 237
Estimating Database Resources ................................. 238
Ranking Data Objects for Oracle Databases .................. 238

Chapter 13  Maintaining Data Models ......................... 241
Maintaining Data Models ............................................. 241
Maintaining Split Data Models .................................... 242
Maintaining Combined Data Models ............................. 243
Creating Data Model Summaries ................................. 243
Refreshing Data Models ............................................ 244
Determining Changes to Data Objects ......................... 244
Applying Changes to the Data Model ............................................. 245
Refreshing Queries ................................................................. 247

Manually Editing Database Names in the Data Model ............. 247
Editing Database Names of Data Objects and Actual Relationships ............. 248

Maintaining Outer Joins .............................................................. 249
Syntax Consistency ................................................................. 249
Updating Outer Joins in a Design Window ................................. 250
Converting Legacy Outer Join Operators Manually ....................... 250
Converting Legacy Outer Join Operators Automatically ................. 251
Setting Classic Outer Join Syntax for Queries .............................. 252

Chapter 14 Searching Data Models ............................................. 253
Searching Data Models: The Basics ............................................. 253
Search Criteria Categories ....................................................... 254

Specific Criteria for Searching Data Models ............................. 256
Search Criteria for Data Objects ............................................... 256
Searchable Data Object Properties .......................................... 257
Search Criteria for Attributes ................................................ 258
Searchable Attribute Properties ............................................. 259
Search Criteria for Results Sets ............................................... 260
Search Criteria for Buttons ..................................................... 260
Searchable Button Properties ............................................... 262
Search Criteria for Ornaments ............................................... 262
Search Criteria for Relationships ............................................ 263
Searchable Properties for Relationships .................................... 264
Search Criteria for Variables ................................................. 265

Searching with the Find and Replace Dialog Box ....................... 265
Searching for NULL Values ..................................................... 267
Chapter 15  Updating Database Records  271

Requirements for Updating Database Records  ................. 271
Controlling Database Updates  .................................... 272
  Setting Key Fields ............................................. 273
  Assigning the Update Database Permission ..................... 274
  Denying Update Permission for Specific Attributes .......... 274
Adding, Modifying, and Deleting Records  ....................... 274
  Spreadsheet and Form View .................................. 275
  Adding, Modifying, and Deleting Records ...................... 275
Adding Records from Another Application: The Basics ......... 276
  Preparing a Text File to Add Records ......................... 277
  Importing a File to Update Records .......................... 278
Applying Updates to the Database  .............................. 279
  Correcting Updating Errors ................................. 279

Chapter 16  Accessibility and Technical Support  281

General Accessibility ............................................ 281
  Microsoft Accessibility Options ............................. 283
Technical Support ............................................... 283
  Using the Trace Utility ....................................... 284

Index  285
This section provides information on the following:

- “Introducing BI Query” on page 17
- “Basic Concepts” on page 18

**Introducing BI Query**

BI Query is a query and reporting tool that provides a comprehensive solution for accessing, analyzing, and presenting data stored in enterprise databases. BI Query lets you extract the information you need using a data model—a graphical representation of the database. By using the data model you can form queries without needing to know SQL (Structured Query Language—the language used for retrieving data from most databases).

**For Administrators**

BI Query provides the flexibility to tailor information access to the exact needs of business users. The administrator makes business-critical information available while maintaining data security, quality, and integrity.
For Business Users
BI Query provides an easy-to-use, visual way to query databases, integrate data with other applications, and generate reports.

BI Query Applications
The BI Query product line consists of three applications—Admin, User, and Update:

BI Query Admin
Lets the BI Query administrator manage the use of the program by users. The BI Query administrator can design data models, set permissions for users, set passwords, and control access to the database and the functionality of BI Query. With permission from the DBMS administrator, the BI Query administrator can also update the tables in the database.

BI Query User
Lets users run queries provided by others and, depending on their user permissions, create ad hoc queries of their own.

BI Query Update
Provides users with the same functionality as BI Query User with the additional ability—depending on their DBMS permissions—to update tables in the database.

Basic Concepts
The following sections describe the basic concepts behind BI Query’s graphical approach to extracting information from corporate databases.

Database Components
A database is a collection of related information. The basic components of a database are as follows:
Tables
In a relational database, information is held in tables. A table usually relates to something in the real world. For example, a database might store customer names and addresses in one table, products in another, stock levels in another, purchase orders in another, and so on.

Columns
Tables are made up of columns and rows. Each column represents an attribute of the entity that the table represents. For example, customers have names, addresses, fax numbers, and so on. A table for customer information would have a column for each of these attributes. (Columns are also known as attributes.)

Rows
Each row in a table is an instance of the entity—that is, each row in the customer table gives us all the information about a customer—specific name, specific address, and so on. (Rows are also known as records.)

DBMSs
The collection of programs that manage a database constitute a database management system (DBMS). A DBMS lets users examine and manipulate data in “real world” terms—customers, orders, products—without needing to know how the computer actually stores the information.

Data Model Components
A data model is a graphical representation of the data in a database.

A data model includes data objects and the relationships between them.
When you use BI Query to get information from a database, you work with a data model. Depending on the BI Query application you are using and the permissions assigned to you, you may be able to customize data models to suit your needs. Administrators, who use BI Query Admin to design corporate data models, set permissions and preferences for each data model. Users of a model can change the preference settings, but not the permissions.

Data models consist of the following components:

**Data Objects**
Data objects are rectangular or graphical icons that represent the tables stored in the database. Each data object contains one or more attributes.

**Attributes**
An attribute represents a column of data in a database table. When you double-click a data object in a data model, an attribute window opens, listing the attributes stored in the data object.

**Relationships**
A relationship connects two data objects together and indicates that the connected objects contain at least one attribute in common. For example, an Employee data object might be related to a Department data object on the basis of a common Manager attribute. In order to include the attributes from two or more data objects in a query, the data objects must be related.

Data objects can have more than one relationship, so that you can get different information using the same objects. BI Query represents relationships as connecting lines between objects. Relationships can also appear with a diamond icon and a name.

You could use the *works in* relationship to find information about an employee who works in a particular department.

You could use the *managed by* relationship to find information about an employee who manages a particular department.
Design Windows

Design windows are the workspace in which administrators design data models and users formulate queries. Design windows contain the data objects that represent tables in the database and the relationships that tie them together. Design windows can also contain buttons (for navigating between Design windows and automating tasks) and ornaments (such as graphics and text) that can provide information and improve the usability of the model.

Buttons

Buttons automate frequent activities. They let users connect automatically to the database, run multiple queries, combine the results, and generate a report—all with a single click of the mouse. Buttons can also display Design windows, save results to files, export results to other applications, open associated document files, and launch other applications such as Visual Basic and Excel.

Ornaments

Ornaments are text and graphical objects such as titles, logos, borders, backgrounds, and notes that provide additional information, act as visual organizers, and improve the appearance of Design windows.
Queries

A query is a request for information from a database. In order to retrieve information using the data objects and relationships in a data model, you create a query and run it (submit it to the database). The results returned by the database for a given query are known as a results set.

The first step in creating a query is to select attributes from at least one data object. You may also want to qualify one or more attributes to restrict the results to the particular information that interests you—such as the sales information for a particular store.

The designer of a data model typically creates queries and saves them with the data model; users of the model can then open the queries from the data model and run them. The designer can also let users edit existing queries and create their own (called ad hoc queries).

Reports

Once you have gathered the data you want by querying, you may need to present it in a report. BI Query has two report generators, BI Query Reports and BI Query Standard Reports, that let you produce your own professional-looking reports.

BI Query Reports provides a flexible reporting environment that includes tables, crosstabs, charts, and maps. You can open BI Query Reports from within BI Query or directly from the BI Query program group (under Hummingbird) in the Windows Start menu. BI Query Reports has its own online Help system, which you can access from its Help menu.

💡 For more information on BI Query Reports, see the BI Query Reports User’s Guide.

BI Query Standard Reports is an integrated component of BI Query. For more information on Standard Reports, see the BI Query Queries User’s Guide.

Types of Data Models

BI Query provides two types of data models: split and combined.
Split Data Models

A split data model lets users add their own customizations to a data model (including changes to their preferences and other default settings) and retain those customizations when a new version of the data model is distributed. This type of data model is suitable for most sites.

A split data model is stored as two layers—an administrator layer and a user layer.

Administrator Layer

The administrator layer of a split data model is the data model that the administrator creates and edits. It consists of a file with the extension .gqa, a Queries folder, a DataVals folder, and a Reports folder. The administrator-layer file includes all the components that make up a basic data model. The administrator distributes this file, plus the associated folders, to all users.

User Layer

The user layer of a split data model consists of a file with the extension .gqu, a Queries folder, a DataVals folder, and a Reports folder. Along with the administrator layer, the administrator distributes an essentially empty copy of the user-layer file to all users. The user-layer file contains a few default settings, such as preferences. It also contains the path to the administrator layer; when a user opens the user-layer file, BI Query uses the path to find the administrator layer. BI Query then combines all the elements stored in the user-layer file with the elements stored in the administrator-layer file.
The following table shows the distribution of components in a split data model:

<table>
<thead>
<tr>
<th>Layer Component</th>
<th>Administrator Layer</th>
<th>User Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data model file</td>
<td><code>datamodel.gqa:</code></td>
<td><code>datamodel.gqu:</code></td>
</tr>
<tr>
<td></td>
<td>• <strong>Design</strong> windows</td>
<td>• <strong>Design</strong> windows</td>
</tr>
<tr>
<td></td>
<td>• data objects</td>
<td>• calculated attributes</td>
</tr>
<tr>
<td></td>
<td>• relationships</td>
<td>• buttons</td>
</tr>
<tr>
<td></td>
<td>• calculated attributes</td>
<td>• ornaments</td>
</tr>
<tr>
<td></td>
<td>• buttons</td>
<td>• drawing objects</td>
</tr>
<tr>
<td></td>
<td>• ornaments</td>
<td>• preferences</td>
</tr>
<tr>
<td></td>
<td>• drawing objects</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• permissions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• default preferences</td>
<td></td>
</tr>
<tr>
<td>Queries folder</td>
<td>• query files</td>
<td>• query files</td>
</tr>
<tr>
<td></td>
<td>• subfolders</td>
<td>• subfolders</td>
</tr>
<tr>
<td></td>
<td>• prompts file</td>
<td>• prompts file</td>
</tr>
<tr>
<td></td>
<td>• variables</td>
<td>• variables</td>
</tr>
<tr>
<td></td>
<td>• standard report specifications</td>
<td>• standard report specifications</td>
</tr>
<tr>
<td>Reports folder</td>
<td>• BI Query reports</td>
<td>• BI Query reports</td>
</tr>
<tr>
<td>Data Values folder</td>
<td>• data values query files</td>
<td>• data values query files</td>
</tr>
<tr>
<td></td>
<td>• data values results files</td>
<td>• data values results files</td>
</tr>
<tr>
<td></td>
<td>• data sets</td>
<td>• data sets</td>
</tr>
</tbody>
</table>

Because it is the user-layer file that contains the path to the administrator layer, opening a split data model requires opening the user layer. BI Query combines the contents of the user-layer file with the administrator-layer file.
The first time a user opens a split data model, the user sees only the data model stored in the administrator layer. When the user layer is stored in a folder separate from the administrator layer, BI Query creates empty Queries, DataVals, and Reports folders to store user-defined queries, prompts, variables, data values files, data sets, and reports. BI Query saves any other customizations the user adds (such as extra buttons, ornaments, and Design windows) in the user-layer file.

The next time the user reopens the data model, the user sees the administrator’s data model as well as all the customizations he or she has added. The changes made by one user do not affect those made by other users. Users cannot make changes to the administrator layer.

## Combined Data Models

A combined data model consists of one data model file (with extension .gql) and the associated Queries, DataVals, and Reports folders. Each time the administrator revises a combined data model and distributes it to users, any user customizations are overwritten.

Typically, combined data models are created for organizations that do not allow users to make any changes to the data model. The following table lists the components of a combined data model:

<table>
<thead>
<tr>
<th>Layer Component</th>
<th>Combined Data Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data model file</td>
<td><code>datamodel.gql</code></td>
</tr>
<tr>
<td></td>
<td>· Design windows</td>
</tr>
<tr>
<td></td>
<td>· data objects</td>
</tr>
<tr>
<td></td>
<td>· relationships</td>
</tr>
<tr>
<td></td>
<td>· calculated attributes</td>
</tr>
<tr>
<td></td>
<td>· buttons</td>
</tr>
<tr>
<td></td>
<td>· ornaments</td>
</tr>
<tr>
<td></td>
<td>· drawing objects</td>
</tr>
<tr>
<td></td>
<td>· permissions</td>
</tr>
<tr>
<td></td>
<td>· default preferences</td>
</tr>
<tr>
<td>Layer Component</td>
<td>Combined Data Model</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>Queries folder</td>
<td>• query files</td>
</tr>
<tr>
<td></td>
<td>• subfolders</td>
</tr>
<tr>
<td></td>
<td>• prompts file</td>
</tr>
<tr>
<td></td>
<td>• variables</td>
</tr>
<tr>
<td></td>
<td>• standard report specifications</td>
</tr>
<tr>
<td>Reports folder</td>
<td>• BI Query reports</td>
</tr>
<tr>
<td>Data Values folder</td>
<td>• data values query files</td>
</tr>
<tr>
<td></td>
<td>• data values results files</td>
</tr>
<tr>
<td></td>
<td>• data sets</td>
</tr>
</tbody>
</table>

**Toolbars**

Toolbars are collections of buttons that provide access to related functions. You can display or hide each toolbar using the **Toolbar** command on the **View** menu or by right-clicking on a toolbar in the main program window.

Visible toolbars have one of two states: docked or floating. A docked toolbar occupies a fixed space on any edge of the main program window. A floating toolbar appears in its own window within the main program window or the desktop.

You can reposition a docked toolbar by clicking on the raised bar within the toolbar and dragging it to the new position. You can also turn a docked toolbar into a floating toolbar (by dragging the toolbar into the main program window) or turn a floating toolbar into a docked toolbar (by dragging the toolbar to an edge of the main window).

💡 To move a floating toolbar over an edge of the main program window—without docking the toolbar—press the CTRL key before moving the toolbar.

BI Query supplies the following toolbars:

- Standard toolbar
- Drawing toolbar
- Layout toolbar
• Super Query toolbar
• Navigation toolbar
• Query toolbar
• Query Window toolbar

💡 For more information on these toolbars, see the Help for your BI Query application.

BI Query Operation Modes

BI Query operates in one of two modes depending on the type of activities you want to perform.

Run Mode

Lets you perform activities that require access to the database (running queries, updating tables, submitting SQL scripts, and so on).

Design Mode

Lets you modify the design and layout of the data model, create buttons and ornaments, and perform other activities that do not require a connection to the database.

You can switch between these two modes by clicking the Design Mode button on the Standard toolbar. The status bar indicates the current operation mode.

The Drawing and Layout toolbars are available only in Design mode. The buttons on the Drawing toolbar let you create objects and assign properties to them (color, line thickness, and so on). The Layout toolbar provides buttons for resizing and aligning objects.

Running BI Query with BI Server

BI Server is an application server that uses a common set of services and a common repository of information to provide data for Hummingbird BI applications. The BI Server Repository is a storehouse for enterprise information produced using Hummingbird BI applications. It stores data models, queries, results sets, reports, multi-dimensional data sources, and associated metadata. If you are in a BI Server environment, you will be able to take advantage of BI Server’s publishing, retrieving, scheduling, and security features.
Specific terms are used to distinguish Server-related actions from local (computer-based) actions. Locally saved data (on your desktop computer) is saved or opened. Material stored on the BI Server Repository is published or retrieved.

If you have been assigned the appropriate system permissions, you can publish BI Query information to the Repository and you can retrieve information you have published as well as information published by others. By publishing to the Repository, you ensure that the information is accessible to other BI Query users and to BI Web users. Because you can set security on items you publish, publishing also provides a secure way to share your information with other users.

For example, if you are using BI Query Admin to create data models for distribution throughout the enterprise, publishing them to the Repository makes them easy to administer and maintain. When you publish a data model, you can restrict access to the model to a certain set of users.

For more information on using BI Query in a BI Server environment, see “Publishing with BI Server” on page 223.

**Personal Versus Shared Files**

When you create certain BI Query files, you can specify whether the file is personal or shared.

**Personal Files**

Are unique to each user of the data model. No user can access another user’s personal files. In a split data model, personal files reside in the user layer. In a combined data model in a BI Server environment, the personal files for a given user reside in a special area of the Repository for the user; in a local copy of a combined data model, all files are effectively personal. Personal files apply only to BI Query User/Update users.

**Shared Files**

Are available to all users of the data model. All users can access shared files, but only BI Query administrators can edit or delete shared files. In a split data model, shared files reside in the administrator layer. All files created by the BI Query administrator (in a combined or split data model) are effectively shared. BI Query User/Update users can share files only in a BI Server environment and only if they have the **Share queries and reports** permission. Once a user shares a file, the user cannot edit or delete it.

These distinctions apply to the following types of BI Query files:
• query files (.qry) in the Queries folder
• standard report specification files (.rpt) in the Queries folder
This section provides information on the following:

- “Starting BI Query” on page 31

Starting BI Query

When you start BI Query, the Hummingbird BI dialog box may open. This dialog box indicates that your corporate querying environment includes a central storehouse for data models, queries, results, reports, and data sources (the BI Server Repository).

You can log on to use the Repository, or you can work offline. If you log on, you can publish, secure, and retrieve queries and reports in the Repository. You can also schedule queries and reports. If you choose to work offline, you do not have access to the Repository. However, you can still connect to the database, run queries, and work with query results.

Regardless of whether you log on or work offline, the Welcome to Hummingbird BI Query dialog box appears automatically to prompt you to open a data model. (You can also open a data model after you have started the program.)

The types of data models you can open at start-up are as follows:

- Recent—the data models on your system that you have recently accessed.
Chapter 2: Getting Started

- Local—all the data models on your system; you do not need to log on to BI Server to access these models.
- Repository—the data models available in the BI Server Repository. Before you can access a Repository model, you must log on to BI Server.
- New—empty models that you can develop. Only BI Query administrators can open new data models. For more information, see “Opening New Data Models” on page 78.

Opening Data Models

A data model can be either split or combined. Combined data model files have the extension `.gql`. To open a split data model, open the user layer (extension `.gqu`).

If you are working online and you open a local copy of a published data model, you will receive a message if a version of that model has been published more recently than the date of the local copy. Depending upon the settings specified when the data model was published, you may be able to choose whether to retrieve the more recent version, or you may be required to do so.

To start BI Query and open a data model:

1. On the Start menu, navigate to the Hummingbird program group and point to BI Query; then, click your BI Query application (one of BI Query Admin, BI Query Update, or BI Query User).

2. If the Hummingbird BI dialog box appears, do one of the following:
   - To log onto the Repository, type your user name and password. In the Domain box, type your domain name. If you have access to multiple repositories, select the one you want from the BI Server list. (If you're unsure what information to provide in the dialog box, check with your BI Server administrator.) Click Log On.
   - To work without access to the Repository, click Work Offline.

3. In the Welcome to Hummingbird BI Query dialog box, do one of the following:
   - To open a data model that you have used recently, click Recent Data Model. In the Recent Data Models dialog box, select the data model you wish to open; then, click Open.
   - To open a local data model, click Local Data Model. In the Open Data Model dialog box, select the data model file you want; then, click Open.
• To retrieve a published data model from the repository, click **Repository Data Model**. In the **Retrieve Data Model** dialog box, select the data model file you want; then, click **OK**.

• To open a new data model, click **New Data Model**.

4. If the **Enter Data Model Password** dialog box opens, type your password; then, click **OK**. If a password has not been set, this dialog box does not appear.

5. If the **Enter Connection Information** dialog box opens, supply the necessary credentials (user name and password) for the connection; then, click **OK**. Contact your database administrator for more information.

6. A message box may open to indicate that a newer version of the data model exists in the database. To open the newer version, click **Yes**.

**To open a data model from within BI Query:**

1. On the **File** menu, click **Open**. In the **Open Data Model** dialog box, select the data model file you want; then, click **Open**.

2. If the **Enter Data Model Password** dialog box appears, type your password.

3. A message box may open to indicate that a newer version of the data model exists in the database. To open the newer version, click **Yes**.

**Retrieving Data Models**

When you retrieve a data model from the BI Server Repository, BI Query copies it locally. Any queries, data values files, data sets, connection files, standard reports specifications, and BI Query reports that were published with the data model are retrieved and stored in the appropriate folders. You must be logged onto BI Server to retrieve data models.

💡 For more information, see “Publishing with BI Server” on page 223.

If you have the appropriate BI Server and BI Query permissions, you can make and save changes to the copy, and then publish it to the Repository again. (The source data model remains on your local machine as originally saved unless you delete it.)

**To retrieve a data model:**

1. Do one of the following:

   • If you are starting BI Query, in the **Welcome to Hummingbird BI Query** dialog box, click **Repository Data Model**.
• If BI Query is already running, close the data model you are using; then, on the **File** menu, click **Retrieve**.

2. In the **Retrieve Data Model** dialog box, select the data model you want; then, click **OK**.

**Finding the Administrator Layer in a Split Data Model**

Each user layer of a split data model contains an internal link to the administrator layer. If the administrator-layer file has been moved or renamed, BI Query will not be able to find it when you open the model. In this case, BI Query prompts you for the new location. To reset the internal link, you must locate the file; then, save the data model.

**To find the administrator layer when it has been moved or renamed:**

1. In the alert box that opens when you try to open the data model, click **Browse**.
2. In the **Open Data Model** dialog box, find and open the administrator-layer file.
3. On the **File** menu, click **Save**.

**Working Online or Offline**

If you are in a BI Server environment, you have the option of working online (logging on to the BI Server Repository) or offline. You can log on to BI Server when you start up BI Query. You can also log on after you have started BI Query.

When you are logged on to BI Server, an icon representing the Hummingbird BI Server Desktop Session Manager appears on the Windows taskbar. The tooltip for this icon indicates the name of the BI Server to which you are currently logged in. This can be helpful if you are working with multiple instances of BI Server.

**To log on to BI Server from within BI Query:**

1. On the **File** menu, click **Work Online**.
2. In the **Hummingbird BI** dialog box, type your user name and password. In the **Domain** box, type your domain name. (If you leave this box blank, Hummingbird BI will attempt to log you in using the default domain name.) Click **Log On**.

**To work offline after you have logged on to BI Server:**

1. On the **File** menu, click **Work Offline**.
Exiting BI Query

BI Query automatically disconnects from the DBMS whenever you close a data model or exit from BI Query.

To exit BI Query:

1. On the File menu, click Exit.
2. Click Yes to save your changes, No to quit without saving them, or Cancel to remain in BI Query.
This section provides information on the following:
- “DBMS Connections” on page 37
- “Creating Connection Files” on page 43
- “Managing Connection Files” on page 48
- “Publishing Connection Files” on page 53
- “Connecting to DBMSs” on page 56
- “Connection Files and Outer Join Support” on page 58

**DBMS Connections**

BI Query connects to database management systems (DBMSs) using a variety of network connections. A network connection establishes an electronic link between BI Query on your computer and the database server.

BI Query supports network connections that use middleware—a form of connectivity software that lets application programs retrieve information from a database. Middleware may in turn use a gateway—a layer of software between the client middleware and the DBMS that translates one vendor’s DBMS to another’s.
For an up-to-date list of supported DBMSs and middleware, visit the following site:
http://www.hummingbird.com/support/dirs/bisuite/index.html
BI Query supports connections to the listed databases as well as a variety of other DBMSs using ODBC connections, dependent upon ODBC driver availability.
BI Query sets up network connections using connection files.

**Connection Files in BI Query**

One of the first tasks in setting up BI Query in the workplace is to create connection files, test them, and distribute them to users so that they can connect to the DBMS.

Connection files store the information required to connect to a DBMS; you cannot connect until this information has been set. The information in a connection file applies to a single DBMS and is site-dependent and different for each network connection.

Typically, the BI Query administrator creates the connection files for a data model. Users can also create and edit connection files, although they cannot change the DBMS to which the file applies.

In addition to connectivity information, connection files store information about the outer join capabilities of the DBMS and the middleware. For more information, see “Connection Files and Outer Join Support” on page 58.

**Connection Folder**

The connection files for a given data model usually reside in the same folder. The data model can store the location of the connection folder; in this way, each data model can use a different set of connection files. Alternatively, data models can use the connection folder specified in the *biq.ini* file on your system. All models that use the *biq.ini* file share the same set of connection files.

💡 For more information on the connection folder, see “Setting the Location of the Connection File Folder” on page 52.

**Types of Connections**

The various types of connections in BI Query are as follows:

- automatic
- default
• per-window
• metadata
• freehand query
• current
• multiple
• DSNless

Descriptions of each of these connection types are provided in the following paragraphs.

**Automatic Connection Files**

You can set up the data model so that BI Query automatically connects to the DBMS whenever you open a data model or run a query. If the data model is not set up to connect automatically, BI Query prompts you to connect when you initiate activities requiring access to the DBMS (such as creating a calculated attribute using a database function).

💡 For more information, see “Setting Automatic Connections” on page 50.

**Default Connection Files**

You can set a default connection file for a data model. If the data model is set to connect to the DBMS automatically, BI Query uses the default connection file to connect to the DBMS. If there is no default connection file, BI Query uses the connection file last used to connect to the DBMS. For data models that have a single connection file, BI Query treats the single file as the default. If the data model is not set to connect automatically, or if there is no default or last-used connection file, BI Query prompts you to select a connection file when you initiate activities requiring access to the DBMS.

The default connection file is also used for any SQL scripts sent using `gqlcheck.dll`. For more information, see “Custom Checks Permission” on page 187.
Per-Window Connection Files

If the **Connect per Window** permission has been assigned, the data model is in connect-per-window mode and you can assign a different connection file to each **Design** window. Because each window can connect to a different DBMS, this setup is useful if your organization stores data in more than one database. When you run a query built from objects in a given window, BI Query connects to the DBMS using the connection file for that window. If a **Design** window does not have an assigned connection file, the window uses the default connection for the data model, or, if there is no default, the last-used connection.

![Note]

You do not need per-window connections to connect to multiple DBMSs. You can create and distribute several connection files for the data model, each for a different DBMS. BI Query and BI Web users can select which file—and therefore, which database—they want to use for a given query or report.

For more information, see “Assigning Connection Files to Design Windows” on page 51.

To ensure that each **Design** window uses the correct connectivity, make sure you adhere to the following guidelines when creating a data model in connect-per-window mode:

- If each window in the model uses a different connectivity (middleware), assign a connection file to each window.
- If some but not all windows in the model are assigned a connection file, ensure that you have set a default connection file and that all windows without a connection file use the default connectivity.

For example, in a data model in connect-per-window mode, any **Design** windows that are not assigned a connection file use the default or last-used file. If the last-used file connects to a database “A” using SQL*Net, any windows that are not assigned connection files must be designed to work with this connection. If the active window is designed to connect to a database “B” using ODBC, any queries or reports run from the active window may fail.
Metadata Connection File

You can set a specific connection file for metadata queries. For data objects, metadata queries replace table or correlation names with display names; for attributes, metadata queries replace column names with business terms. These queries also provide short descriptions for data objects and attributes to explain what kind of information they represent. By creating a connection file for metadata queries, you can load metadata from a database that is different from the databases specified in the default or connect-per-window connection files.

💡 For more information on metadata, see “Creating and Editing Metadata Queries” on page 198.

Freehand Query Connection Files

Unlike standard queries, when you save a freehand query, the name of the connection file is saved along with the query. When you run a freehand query, if the current connection file in BI Query is different than the one specified in the freehand query, BI Query disconnects from the current connection, connects using the connection file in the freehand query, runs the freehand query, disconnects, and reconnects using the original connection file.

When you publish or retrieve a freehand query, or a data model that includes a freehand query, the connection document associated with the freehand query is automatically published or retrieved as well.

Current Connection File

The current connection file is the file that BI Query uses to connect to the DBMS at any given moment. When you open a data model, the current connection is either the default (or last-used) connection file or, in connect-per-window mode, the file assigned to the active Design window. The status bar in BI Query displays the description (or file name, if no description exists) for the current connection file.

If the data model is not in connect-per-window mode, you can change the data model connection for the current BI Query session using the Connections dialog box. The current connection remains in effect until you close and reopen the data model (in which case, the default connection applies) or change the connection.

💡 For more information on changing the connection, see “Connecting to DBMSs” on page 56.
For data models not in connect-per-window mode, BI Query uses, in order, the current connection (if assigned), the default connection file (if it exists), or the last-used connection file.

**Multiple Connection Configurations**

You can set up multiple connection configurations for a given data model. In this case, store the connection files for each configuration in a separate folder, making them easier to find, use, edit, and distribute. This arrangement is useful if a database is being moved from one machine to another. Users can continue to use the data model with the original connection files while a copy of the database is tested on another machine using another connection configuration. This arrangement is also useful if you have several databases with identical structures; in this case, you can create a configuration for each database.

**DSNless Connection Files**

In BI Query you can create ODBC connection files that do not require a data source name (DSN). Such 'DSNless connections' can be published, and they have two key advantages over connection files that explicitly specify the DSN:

- they will work from any machine with access to the DBMS server and driver without requiring the data model user to create a specifically-named DSN entry.
- they do not require that a data source with the specified name be configured on every Hummingbird BI user workstation as well as on BI Server.

Although data model users won't need to create DSN entries, they will still need to have ODBC drivers installed on their workstations.

To create such a connection file, in BI Query you select `<User Defined>` from the Data Source box of the ODBC Connection dialog box, then enter the appropriate connection string in the Connection string box. If you know both the syntax and the parameters you want to use, you can type the string directly in the Connection string box.

Otherwise, you can use the DSN To String utility to generate the connection string for you, then simply copy and paste it into the Connection string box. For more information, see the file DSN To String Utility.txt that accompanies the utility.

The DSN To String utility is not officially supported. Hummingbird does not guarantee that the connection strings generated by the utility are correct.
Creating Connection Files

Connection Files in Use

The connection file that BI Query uses depends on the activity it performs:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Connection File Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sending SQL to the DBMS, including any of the following operations:</td>
<td>The connection file assigned to the relevant Design window (for example, the Design window where the query was created).</td>
</tr>
<tr>
<td>• running a standard query</td>
<td>If a connection file is not assigned to this window, the current connection file is used.</td>
</tr>
<tr>
<td>• creating, loading, or refreshing tables</td>
<td></td>
</tr>
<tr>
<td>• retrieving data values or loading attribute comments</td>
<td></td>
</tr>
<tr>
<td>Restoring data models from or saving them to the database</td>
<td>The default connection file, or, if there is no default, the last-used connection file.</td>
</tr>
<tr>
<td>Running a freehand query</td>
<td>The connection file for the freehand query.</td>
</tr>
<tr>
<td>Running a super query</td>
<td>The connection file for each component query.</td>
</tr>
</tbody>
</table>

Creating Connection Files

The information stored in a connection file is specific to a particular DBMS and connectivity. The following procedure outlines the steps that are common to creating any type of connection file.

💡 For more information on connection files, including connection details for specific middleware, see the Help for your BI Query application.

👨‍💻 Only BI Query administrators can specify the DBMS for a connection file.

To create a connection file:

1. On the Host menu, click Connections. The Connections dialog box opens.
2. Click New. The New Connection dialog box opens.
3. From the DBMS drop-down list, select the DBMS you want to use.
4. From the **Connectivity** list, select the middleware you want to use. The connection options in the dialog box change depending on the selected DBMS and connectivity.

5. In the **Description** dialog box, type a description for the connection file. This description appears to BI Query and BI Web users of the file.

6. Specify the other parameters required for your network connection.

7. To set connection preferences, click **Preferences**. The **Connection Preferences** dialog box opens.

8. In the **Connection Preferences** dialog box, set the options you want:
   - Set row buffering to control the transfer of data across the network and between BI Query and the database. For certain DBMSs, you can also specify the width of character data that is returned from the database.
   - Set the **Rows** and **Time Governor** preferences to limit the number of rows retrieved and elapsed query time for the connection.

   ![](image)

   For more information on these connection preferences, see ““Specifying Governor Settings with Connection Files” on page 45”.

9. Click **OK**.

10. In the **New Connection** dialog box, click **Save**.

11. In the **Save Connection Document** dialog box, specify the path and name of the file; then, click **OK**. Connection files must have a `.con` extension. Connection files are typically saved in the connection folder for the data model.

12. In the **Connections** dialog box, verify that the correct information for the new file appears in the **Available Connections** list. Click **Close**.

### About Governor Settings with Connection Files

BI Query provides two connection file preferences that you can set to optimize connection performance and minimize network load.

**Rows Governor preference**

Specifies the maximum number of rows that BI Query retrieves before prompting you to retrieve more or stop the query.
Time Governor preference

Specifies the maximum time in minutes that a query can run before BI Query prompts you to continue the query or stop it.

You can set the Rows and Time Governor preferences from the Preferences dialog box, in which case they apply to all queries. You can also set these preferences for a specific connection file (using the Connection Preferences dialog box), in which case they override the global setting for queries involving that particular connection. In either case, the values you specify for these preferences apply only if they are smaller than the values for the corresponding Rows and Time Governor permissions (set by the BI Query administrator).

💡 For more information on the Governor permissions, see “Setting Query Limits with Governors” on page 237.

Example

The Rows Governor permission for a data model is set to 20,000. If the data model has multiple connections and one connection is slower than the others, you may decide to set the Rows Governor preference for that connection to 10,000. BI Query stops all queries after 20,000 rows have been returned except those queries using the slow connection (the one whose Rows Governor preference is set to 10,000), in which case BI Query prompts you to continue or stop the query after 10,000 rows have been retrieved.

Specifying Governor Settings with Connection Files

Specifying row and time limits for queries can be an effective way of managing traffic on the network, especially for heavily used networks.

To set Row and Time Governor preferences for a connection file:

1. On the Host menu, click Connections. The Connections dialog box opens.
2. In the Name column of the Available Connections list, select the connection file; then, click Edit.
3. In the Edit Connection dialog box, click Preferences. The Connection Preferences dialog box opens.
4. In the Governors area, clear Use Preference Governors; then, select both Ask Every options.
5. Type the time or row governor values you want into the corresponding boxes.
6. Click **OK**; then, save the connection file. The settings you have applied take effect the next time you use the file to connect.

7. In the **Connections** dialog box, click **Close**.

In addition to setting the Governor preferences, you can also optimize connection performance by setting the row buffering for a connection file.

### Optimizing Connections with Row Buffering

When BI Query retrieves rows of data, it holds them in a portion of memory (a buffer) on the user’s machine until they can be transferred to BI Query. Row buffering affects the time it takes to retrieve data and consequently affects the network load. You can optimize row buffering to dramatically improve data-retrieval time and reduce network load.

BI Query lets you set two types of row buffering:

#### Middleware Row Buffering

For middleware capable of performing its own row buffering (DB2 Call Level Interface, Ingres II/Net, ODBC, and SQL*Net). You can optimize the performance of middleware by increasing its row buffering value. The default is 71.

#### DBaccess Row Buffering

For DBAccess (a utility installed with BI Query that allows BI Query to connect to the DBMS). DBAccess is installed with row buffering set to 71.

While DBAccess row buffering works with any middleware, it may conflict with middleware capable of performing its own row buffering. You may be able to regain performance lost to row-buffering conflicts by changing the middleware row-buffering setting to match the value for DBAccess row buffering. When the two row-buffering settings are the same, the performance of BI Query immediately improves.

Not all DBMSs that you connect to using ODBC support row buffering. (For example, Oracle does, but Access does not.) If the DBMS supports row buffering, it is available in BI Query; if not, the DBMS sends a single row at a time.
Setting Row Buffering

When you set row buffering, you should experiment with different values to find which work best at your site. Keep in mind that as you increase the value, BI Query allocates more temporary buffer space; this can lead to problems when BI Query retrieves very long rows.

To test row-buffering settings, start by using a fairly broad range of values—for example, Row Buffering = 10; then, Row Buffering = 150. Work towards the middle to “bracket” the optimum value.

To set row buffering for a connection:

1. On the **Host** menu, click **Connections**. The **Connections** dialog box opens.
2. In the **Name** column of the **Available Connections** list, select the connection file; then, click **Edit**.
3. In the **Edit Connection** dialog box, click **Preferences**. The **Connection Preferences** dialog box opens.
4. Type the values you want into the **Row Buffering** and **DBAccess Row Buffering** boxes. (For DB2 Call Level Interface, Ingres II/Net, ODBC, or SQL*Net, type a value less than or equal to 10.)
5. Click **OK**; then, in the **Edit Connection** dialog box, click **Save**. The settings you have applied take effect the next time you use the file to connect.
6. In the **Connections** dialog box, click **Close**.

Testing Connection Files

You should test a connection file before distributing it.

To test a connection file:

1. On the **Host** menu, click **Connections**. The **Connections** dialog box opens.
2. In the **Name** column of the **Available Connections** list, select the connection file to be tested; then, click **Connect**.
3. You may have to supply your user name or password in the **Enter Connection Information** dialog box. Type the information; then, click **OK**.
BI Query informs you if the connection is not made. If this happens, check your parameter specifications. If necessary, correct them. If the connection still cannot be made, and you have other applications that use the same connection, run one of them to verify that the connection is working. If you still cannot connect, contact your BI Query or database system administrator. If problems persist, contact Technical Support.

Managing Connection Files

After you have created the connection files for a data model, you can control their use by configuring the model in the following ways:

- Set permissions to control how BI Query stores user-specific connection information.
- Set the default connection file for the model.
- Set the connection mode for the model (connect automatically, connect per window).
- Set the location of the connection folder.

Once you have configured the data model, you are ready to distribute the connection files with the model. Keep in mind the following points:

- If users edit your connection files or create their own, supply them with the appropriate connection information (such as database name and password).
- If you want to use a connection file created with an earlier version of BI Query, you may need to update the file: open the corresponding data model and connect to the DBMS using the file. BI Query automatically updates the connection file when you connect. After updating the file, distribute it to your users.

Connection File Permissions

The connection files you create for a data model can contain all the required connection information. You can also exclude certain information, such as user IDs and passwords; in this case, users must supply the missing information each time they try to connect to a DBMS. You can set the Save Passwords and the Cache Passwords permissions to control how BI Query stores user-specific connection information.
For more information on BI Query permissions, see “Setting BI Query Permissions” on page 183.

Save Passwords
If the **Save Passwords** permission is assigned, users can save passwords (in encrypted form) in connection files. If this permission is not assigned, or if a password is not provided in the connection file, users are prompted to supply a password when they connect.

**Warning:** While the **Save Passwords** permission lets users connect without supplying their password each time, it also lets any user with access to the connection file use the connection. Exercise caution when assigning this permission.

Cache Passwords
If the **Cache Passwords** permission is assigned, any additional information that users supply when they connect to a DBMS is stored in memory until they close the data model. The **Cache Passwords** permission affects all missing connection information (not just passwords) and applies to all connection files that users may use during a session. It is available only when the **Connect per Window** or **Enforce Connect per Query** permission is assigned. If the data model uses more than one connection file, additional information is stored in memory for each connection file. You can specify the default setting for this permission using the `biq.ini` file. Contact Hummingbird Technical Support for more information.

**Warning:** Because users do not need to supply a password each time they connect, the **Cache Passwords** permission is suitable only for sites with light security concerns.

For more information on the connection permissions, see “Enforcing Connect per Query” on page 234.

Setting the Default Connection File
Typically, the BI Query administrator sets the default file before distributing the data model, but anyone with access to the model can set the default. If the connection file you want to set as the default was created using an earlier version of BI Query, you must first connect with the file; then, save it.
To set a default connection file:

1. On the **Host** menu, click **Connections**. The **Connections** dialog box opens.
2. In the **Name** column of the **Available Connections** list, select the connection file that corresponds to the connection you want. If the file does not appear in the list, click **Browse** to first locate and select the correct connection folder on your system.
3. Click **Default**. A bullet is inserted in front of the file name.
4. Click **Close**.

**Setting Automatic Connections**

You can specify whether BI Query automatically connects to the DBMS.

**Connect per Session**

If the **Connect Automatically** preference is set, BI Query connects to the default DBMS when users open the data model. The connection exists for the duration of the BI Query session. Use the **Preferences** dialog box to set the **Connect Automatically** preference.

💡 For more information on preferences, see “Setting BI Query Preferences” on page 190.

**Connect per Query**

If the **Enforce Connect per Query** permission is assigned, BI Query connects to the appropriate DBMS whenever users run a query, and then disconnects once the results are returned. This minimizes the resources used on the database server. The BI Query administrator assigns the **Enforce Connect per Query** permission using the **Permissions** dialog box. If the administrator has not assigned this permission, anyone using the data model in any BI Query application can set the **Connect per Query** preference in the **Preferences** dialog box to achieve the same effect.

For both connection types, BI Query connects automatically to the DBMS when the user opens a data model and determines if a new version has been stored in the database. (If BI Query is set to connect per query, it disconnects once the data model has been opened.)
Connections to Multiple Databases

To make multiple databases available to data model users, you can do one of the following:

• Distribute a single data model with multiple connection files, each for a different database. BI Query and BI Web users can select which file—and therefore, which database—they want to use for a given query or report.

• Distribute a single data model in connect-per-window mode with multiple connection files, each for a different DBMS. In this case, the database that BI Query uses for a given query or report depends on the active Design window. Users of the data model can combine query results to widen the scope of information they retrieve.

  For information on combining query results, see the BI Query Queries User’s Guide.

To configure the data model for per-window connections, you must assign the Connect per Window permission. This permission lets you and your users assign different connection files to different Design windows.

  For more information, see “Assigning Connection Files to Design Windows” on page 51.

• Distribute multiple copies of the same data model, each with a different default connection file and separate copies of any reports. To retrieve information from a particular database, users open the data model that corresponds to that database.

Assigning Connection Files to Design Windows

Use the Design Windows dialog box to assign a connection file to a Design window. (You can also use this dialog box to create, delete, and rename Design windows.)

Before you can assign a connection file to a Design window, the Connect per Window permission must be assigned and BI Query must be in Design mode.
To assign the Connect per Window permission:
1. On the Tools menu, click Permissions.
2. In the Permissions dialog box, select Connect per Window. Assigning this permission automatically assigns the Enforce Connect per Query permission.

To assign a connection file to a Design window:
2. From the Design Windows list, select the Design window; then, click Connection. The Choose Connection dialog box opens.
3. Do one of the following:
   - Click Use Model Default to use the default connection document for the data model.
   - From the Available Connections list, select the connection file you want to use.
     If the connection file is not listed in the Available Connections list, you need to specify the location of the file by closing the Design Windows dialog box, browsing from the Connections dialog box to the connection folder you want, then returning to step 1 to assign the connection file to the Design window.
4. Click OK.
5. In the Design Windows dialog box, click Close.

Setting the Location of the Connection File Folder

You can set a connection folder location—specific to a data model—and save the location with the model. Alternatively, you can opt to use the location specified in your biq.ini file (located under the Documents and Settings folder on your computer).

To set the location of the connection file folder:
1. On the Host menu, click Connections. The Connections dialog box opens.
2. Do one of the following:
   - To use the location specified in your biq.ini file, click biq.ini. If you want to change the location specified in your biq.ini file, click the adjacent browse button, then navigate to the folder you want to use and click OK.
To use a location specific to your data model, click **Data model**. If you want to change the location specified for your data model, click the adjacent browse button, then navigate to the folder you want to use and click **OK**.

3. Click **close**.
4. Save the data model.

**Updating, Editing, and Deleting Connection Files**

If you want to use a connection file created with an earlier version of BI Query, you may need to update the file. To update an old connection file, open the corresponding data model and connect to the DBMS using the file. BI Query automatically updates the file when you connect.

⚠️ If you edit connection files created with an earlier version of BI Query (version 4.0 or earlier), you may be prompted to specify the DBMS for each connection file.

**To edit or delete a connection file:**

1. On the **Host** menu, click **Connections**. The **Connections** dialog box opens.
2. In the **Name** column of the **Available Connections** list, select the connection file.
3. Do one of the following:
   - To edit the file, click **Edit**. In the **Edit Connection** dialog box, make your changes; then, click **Save**.
   - To delete the file, click **Delete**.

⚠️ **Be sure you choose the correct file; you cannot undo a deletion.**

4. In the **Connections** dialog box, click **Close**.

**Publishing Connection Files**

If you are in a BI Server environment, you publish the connection files for a data model when you publish the model, but you must first select which connection files you want to publish.
By default, BI Query publishes the default connection file for the data model (if one has been specified), the connection file for the metadata source (if present), and any connection files assigned to Design windows (if the data model is in connect-per-window mode). Using the Connections dialog box, you can specify the other files you want to publish.

To publish connection files:

1. On the Host menu, click Connections. The Connections dialog box opens.
2. From the Name column of the Available Connections list, select a connection file that you want to publish.
3. Click To Publish. A check mark appears beside the name of the selected file. Connection files that are published by default are indicated by a grey check mark.

If you do not want to publish a file that you have selected for publishing, select the file again and click Publish With Model.

4. Repeat steps 2–3 until you have selected all the files that you want to publish. Click Close.
5. On the File menu, click Publish. The Publish dialog box opens.
6. Click Set Security. The Set Security dialog box opens. Set the security for the connection files. For more information, see “Security for Connection Files” on page 54.
7. In the Publish dialog box, click Publish.

Security for Connection Files

You can set the user and group security for connection files when you publish them. In this way, you can control what databases users can access with a published model by denying or granting access to the corresponding connection files.

If you do not explicitly set security on published connection files, they inherit the security settings applied to the data model. As with other items in the data model, the user and group access to connection files can be granted, denied, or inherited using the Set Security dialog box. The Connections folder in the Set Security dialog box lists the default connection file and any other connection files that are to be published with the data model.
The connection file for the metadata database, and connection files that are assigned to **Design** windows (in connect-per-window mode), are not listed in the Connections folder. Instead, they inherit the security settings applied to the data model.

For more information on using the Set Security dialog box, see “Controlling Access with BI Server” on page 229.

**Retrieving Connection Files**

When you retrieve a data model from the BI Server Repository, BI Query retrieves the associated connection files to a folder on your system, depending on the information stored in the data model or in the **biq.ini** file.

If BI Query cannot find the specified folder on your system, but the data model is using the **Use biq.ini Connections Folder** option, BI Query retrieves the files to a subfolder of the default downloaded file folder on your system; the location of this folder depends on your user name and operating system.

For example, if you are running Windows 2000, BI Query retrieves connection files to the following folder if it cannot find the folder specified in the **biq.ini** file:

```
C:\Documents and Settings\username\My Documents\BI-Suite\ConDocs
```

**Connection File Retrieval Scenarios**

The following table summarizes the various retrieval scenarios:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>User retrieves a data model that was published previous to version 8.5.</td>
<td>BI Query retrieves the connection files to the folder specified in the <strong>biq.ini</strong> file.</td>
</tr>
<tr>
<td></td>
<td>If <strong>biq.ini</strong> does not list a valid folder location, BI Query retrieves the files to the folder where the program was installed. BI Query writes this new location to the <strong>biq.ini</strong> file upon exit.</td>
</tr>
</tbody>
</table>
### Scenario | Result
--- | ---
User retrieves a data model that does not use the **Use biq.ini Connections Folder** option. | If the specified folder exists, BI Query retrieves the connection files to the folder location set for the data model.

If the folder does not exist, BI Query creates the folder and retrieves the connection files to the folder location set for the data model.

BI Query does not update the *biq.ini* file upon exit.

User retrieves a data model that uses the **Use biq.ini Connections Folder** option. | If the specified folder exists, BI Query retrieves the connection files to the folder location set for the data model.

If *biq.ini* does not list a folder location, BI Query retrieves the files to the folder where the program was installed. BI Query writes this new location to the *biq.ini* file upon exit.

If *biq.ini* lists a folder location, and the specified folder does not exist, BI Query retrieves the connection files to a subfolder of the default downloaded file folder on your system. BI Query writes the folder location to the *biq.ini* file upon exit.

---

**Connecting to DBMSs**

At any time, you can connect a data model to the DBMS by using the default connection file (if it has been assigned) or by specifying a different one. If there is no default file assigned and you do not specify which connection to use, BI Query connects to the DBMS using the last-used file.

💡 For more information on connecting, see ““Guidelines for Connecting to DBMSs” on page 57”.
The status bar at the bottom of the BI Query window displays the description of the current connection document if a description has been specified, otherwise it displays the path to the connection document. If the full path is too long to display, the path will be truncated. If you allow the mouse to linger over this area of the status bar, the tooltip that appears displays the full path to the current connection document.

**To connect using the default connection file for the data model:**

1. On the **Host** menu, click **Connect**.
2. If the **Enter Connection Information** dialog box opens, type in the requested information (such as database user name and password). Click **OK**.

**To change the current connection:**

1. On the **Host** menu, click **Connections**. The **Connections** dialog box opens.
2. In the **Name** column of the **Available Connections** list, select the connection file that corresponds to the connection you want. If the file does not appear in the list, click **Browse** to first locate and select the correct connection folder on your system.
3. Click **Connect** to connect the model to the database using the selected connection file.

   ![Guideline icon]

   The selected connection applies only to the current BI Query session (unless the connection is also the default for the data model).
4. If the **Enter Connection Information** dialog box opens, type in the requested information; then, click **OK**.

**Guidelines for Connecting to DBMSs**

To perform any activity that requires access to the database, BI Query must connect to the DBMS. These activities include the following:

- creating tables in the database
- loading tables from the database
- saving data models to the database
- loading data models from the database
- running queries
- sending SQL scripts to the DBMS
- updating the database
Activities in BI Query that do not require access to the database include the following:

- customizing data models (setting permissions and preferences; creating Design windows; creating and manipulating buttons and ornaments)
- publishing data models and queries to the BI Server Repository

Consider performing these activities offline so that you are not using database server resources unnecessarily. You do not need to remember when to connect. BI Query prompts you to connect whenever you initiate an activity that requires access to the database.

**Disconnecting from DBMSs**

BI Query automatically disconnects from the DBMS whenever you close a data model or exit from BI Query. You may wish to disconnect at other times. You can continue to carry out many activities in BI Query while not connected, but you cannot run queries, send an SQL script to the DBMS, or update information in the database.

To disconnect from the DBMS:

- Do one of the following:
  - On the Host menu, click Disconnect.
  - On the Standard toolbar, click the Disconnect button.

**Connection Files and Outer Join Support**

In addition to storing the information that is necessary to connect to a particular DBMS (using a particular middleware), connection files also store information about the outer join capabilities of the database. This information includes the outer join syntax types and other options supported by the database (such as nested joins and full outer joins). You need this information before you can create outer joins (in Design mode) or run queries involving outer joins.

💡 For more information on outer joins, see “Types of Joins” on page 92.
Default Versus Custom Settings

BI Query automatically updates the connection file with outer join capability information each time you use the file to connect. This information provides the default outer join settings for the connection.

You can also specify custom settings for the connection; these are stored in the connection file with the defaults. The custom settings override the defaults. If you use custom settings, you can revert to the defaults at any time. BI Query sets the defaults each time you connect to the database, even if you are using your custom settings.

Outer Join Syntax and Other Database Capabilities

BI Query supports the ODBC, Oracle, and ANSI SQL/92 types of outer join syntax. It also supports the “Classic” (ANSI SQL/89) syntax that uses the legacy join operators, *= and =*. The types of syntax that are available depend on the DBMS and the middleware used to connect to it.

The other outer join capabilities depend on the connection and the syntax. Different databases have different capabilities for each syntax. Once you connect to a database, BI Query selects the best default syntax for the connection and determines the default settings for the capabilities listed in the following table.

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left Outer Joins</td>
<td>Permits left outer joins, as in the following query:</td>
</tr>
<tr>
<td><img src="diagram1" alt="Diagram" /></td>
<td><code>SELECT * FROM A LEFT OUTER JOIN B ON A.x = B.x;</code></td>
</tr>
<tr>
<td>Right Outer Joins</td>
<td>Permits right outer joins, as in the following query:</td>
</tr>
<tr>
<td><img src="diagram2" alt="Diagram" /></td>
<td><code>SELECT * FROM A RIGHT OUTER JOIN B ON A.x = B.x;</code></td>
</tr>
<tr>
<td>Full Outer Joins</td>
<td>Permits full outer joins, as in the following query:</td>
</tr>
<tr>
<td><img src="diagram3" alt="Diagram" /></td>
<td><code>SELECT * FROM A FULL OUTER JOIN B ON A.x = B.x;</code></td>
</tr>
<tr>
<td>Nest Outer Joins</td>
<td>Permits nested outer joins, as in the following query:</td>
</tr>
<tr>
<td><img src="diagram4" alt="Diagram" /></td>
<td><code>SELECT * FROM A LEFT OUTER JOIN (B LEFT OUTER JOIN C ON B.y = C.y) ON A.x = B.x;</code></td>
</tr>
</tbody>
</table>
### Capability | Description
--- | ---
Inner and Outer Joins | Permits inner joins and outer joins in the same ODBC outer join sequence, as in the following query:

```sql
SELECT * FROM
    (A INNER JOIN B ON A.x = B.x) RIGHT OUTER JOIN C
ON C.y = B.y;
```

ODBC and SQL/92 outer joins have syntactically identical SQL except that ODBC prefaces the outer join sequence with the string “{oj}” and terminates the sequence with “}”.

Select this setting if you are using an ODBC connection to an MS Access database and you want to use both types of join in a join sequence.

Multiple Joins | Permits multiple outer joins in the same join sequence, as in the following query:

```sql
SELECT * FROM
    (A LEFT OUTER JOIN B ON A.x = B.x) LEFT OUTER JOIN C
ON C.y = B.y;
```

Select this setting if you are using an ODBC connection to an MS Access database and you want to use multiple joins in a join sequence.

Reuse Non-Preserved Side of Left or Right Outer Join in an Inner Join | Permits the use of the non-preserved side of a half outer join in a separate, inner join. This capability lets you nest inner joins inside outer joins, as in the following query:

```sql
SELECT * FROM A LEFT OUTER JOIN (B INNER JOIN C ON B.y = C.y)
ON A.x = B.x;
```

In this example, B is the non-preserved side of a half outer join with A; B also forms an inner join with C.

Order Tables Names in Join Conditions Differently Than in Outer Join Expression | Permits different ordering for tables within the `FROM` clause, as in the following:

```sql
SELECT * FROM A LEFT OUTER JOIN B ON B.x = A.x;
```

In this example, the order of A and B in the join condition (B.x = A.x) is the reverse of their order in the rest of the expression.
You can view and modify the syntax and capabilities of the current connection at any time using the **Join Syntax** dialog box.

### Missing Database Capabilities Information

Certain operations (such as editing a join or creating an outer join) require information about the capabilities of the database. If you are using a connection file from an earlier version of BI Query, this information is not available to BI Query. In this case, you must update the file by using it to connect to the database.

### Specifying Outer Join Capabilities and Syntax

Before editing (overriding) the default outer join capabilities and syntax, you should first connect to the database using a connection file. This lets BI Query determine reasonable default values.

**To set the database capabilities for a connection:**

1. On the **Host** menu, click **Connections**. The **Connections** dialog box opens.
2. In the **Name** column of the **Available Connections** list, select the connection file; then, click **Edit**.
3. In the **Edit Connection** dialog box, click **Join Syntax**. The **Join Syntax** dialog box opens.

### Capability Description Table

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
</table>
| Use Operators Other Than “=” in Outer Join Conditions | Permits the use of any join operator (such as “<“ or “>”) in an outer join, as in the following query:  
`SELECT * FROM A LEFT OUTER JOIN B ON A.x < B.x;` |
| Cross Products                      | Permits the cross-product expansion of rows in an outer join. A cross product combines the data from all attributes in all the possible ways that satisfy the join condition.  
ODBC and SQL/92 syntax denote cross products using the **CROSS JOIN** keywords in the **FROM** clause for an outer join, as in the following query:  
`SELECT * FROM A CROSS JOIN B ON A.x = B.x;` |
4. From the Join Syntax list, select the outer join syntax you want to use for the selected connection file. BI Query marks the suggested default syntax option as (Default).

You do not need to use the suggested default syntax. For example, if you use ODBC middleware that supports outer joins, but the DBMS supports another syntax that you want to use (such as ANSI SQL/92), you can select that syntax from the Join Syntax dialog box. BI Query performs any necessary syntax conversion before sending the query.

5. Do one of the following:

   • To use the default settings for the selected syntax, select Default. The dialog box automatically displays the default settings for the current connection. (The syntax you select also determines, in part, which capabilities are available for the connection.) These settings change whenever BI Query detects new defaults for the connection.

   For more information, see ““Outer Join Syntax and Other Database Capabilities” on page 59”.

   • To specify your own settings for the connection, select Custom; then, select the capabilities you want BI Query to use for the connection. These settings override the defaults.

6. Click OK. The Join Syntax dialog box closes.

7. In the Edit Connection dialog box, click Save or Save As to save your changes. Once you save the connection file with your changes, BI Query uses the new settings for all designs that use that connection file.
This section provides information on the following:

- “Assessing Users” on page 63
- “Designing Data Models” on page 65
- “Optimizing the Database” on page 70

Assessing Users

When you create a data model, you may need to deliver customized data access to a large user community with widely different needs and skills. Delivering information can entail creating one data model for the user community, and then granting or denying access to specific parts of it to individual users and groups. For users who want simplified access to information, you can provide built-in (or “canned”) queries and reports, whereas users who want the flexibility to create their own queries and reports may require ad hoc access to the database.

You have to balance users’ information needs with their computer skills, accommodating novices and “power users” alike. At the same time, you need to pay close attention to performance, ensuring that the data access environment maintains data integrity and security and minimizes network traffic problems and server overload.
Types of User Requirements

Before you can deliver information to users, you need to establish what database information they need to perform their jobs and how they’re going to use BI Query to access and present that information. You can then build the data access environment to meet the needs of your users. Consider the following points when you assess your users.

Information Needs

Users have different information needs depending on where in an organization they work. For example, users in the Sales Department need access to customer, product, and shipping data to track products shipped, warranties, and invoices. The Human Resources Department needs access to employee and payroll data to track vacation time, analyze performance, and determine how well the company is addressing employment-equity legislation. And Finance needs access to financial data to analyze sales by product, regional office, period, and so on.

Job Functions

Users’ job functions also determine how they use BI Query. Even users in the same department need different information or different ways of accessing it. For example, sales managers who need quick answers to standard questions may want to generate reports for weekly sales meetings that analyze sales by month and quarter. Sales representatives may need to build specialized queries including qualifications and calculations in order to analyze sales by product, region, and country.

Security Requirements

Different user groups have different security requirements. Human Resources, for example, needs secure access to employee and payroll records that must otherwise remain confidential. You can restrict access to specific parts of a data model as well as to certain data and features in BI Query.
Computer Skills

For users who will submit queries and use reports that you supply, you can provide a data model that provides push-button access to those activities. The data model can also contain an interface to give users access to increasingly specific data. You need to know what queries users will submit so that you can build “canned” queries and predefined reports. You also need to know how complex those queries and reports are, and how frequently they will be run, in order to balance the load on the database server.

In each user group, novices and “power users” will have varying computer skills and technical facility. Some will feel confident enough to use the advanced features of BI Query, and those advanced users may want the flexibility to specify their own SQL in queries, perform calculations, or create their own joins between tables. Novice users, on the other hand, may be content to build simple queries or use queries that you provide.

Designing Data Models

The right data model delivers the right information in the right way. It gives users access to the information they need in a way that’s easy to understand and simple to use. It strikes a balance between the number and size of tables in the database and users’ information needs; it weighs the facilities it provides against users’ computer skills and how they like to work. When a data model meets these needs, it should provide the right amount of flexibility for its users.

The facilities in a data model and the way it organizes information are essential to users’ ability to use it effectively. They’re particularly important when the data model represents a large number of tables. On the one hand, a data model that’s too complex to understand makes it difficult for users to access even the simplest information. On the other hand, a data model that’s too simple doesn’t let users access enough useful information to make the data model worthwhile.

There are a number of strategies you can use to deliver a data model that strikes the balance you need. You can choose to follow one or more of the following data model designs.
Physical Data Models

The easiest data model to build is a physical data model. A physical data model is a direct representation of the database and is modeled after the entity-relationship diagram. For example, a physical data model might store address information for customers, sales representatives, and offices in one table because it’s most efficient. The disadvantage of this design is that including all the tables in the database often makes the data model too large and complex. Also, a form of organization that might be useful for a database administrator is not necessarily going to be the most useful for business users attempting to extract information from the database. As a result, the physical data model is often not useful for users.

Logical Data Models

A logical data model presents the database in a way that makes sense from the users’ point of view. It provides only those tables and relationships that users need and reuses and combines tables for users when appropriate. A logical data model might provide address information for customers and offices in two data objects—one for customer information, the other for office information—because that’s where users would most likely look for it, even though in the database it is physically located in one table.
Ad Hoc Data Models

An ad hoc data model provides data objects and relationships that let users submit their own queries and generate their own reports and hypercubes. This type of model allows for greater flexibility. It is useful when users are constantly analyzing data and their requests for information are unique to every query. A typical ad hoc data model might include **Design** windows such as:

<table>
<thead>
<tr>
<th>Window</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>A welcome; includes the company logo.</td>
</tr>
<tr>
<td>Ad Hoc Queries (one or more)</td>
<td>Data objects and relationships.</td>
</tr>
<tr>
<td>“For Help, Contact…”</td>
<td>Information on contacting technical support.</td>
</tr>
<tr>
<td>User Queries/User Reports</td>
<td>Typically nothing; users can add their own buttons.</td>
</tr>
</tbody>
</table>

Ad Hoc/Report Data Models

An ad hoc/report data model supports a user group with varying technical skills. It provides an ad hoc interface for users whose data access requirements are unique to every query and who create their own queries, reports, and hypercubes. This data model also supplies predefined queries, reports, and hypercubes for users who don’t need to create their own.

A typical ad hoc/report data model might include **Design** windows such as:

<table>
<thead>
<tr>
<th>Window</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>A welcome; includes the company logo.</td>
</tr>
<tr>
<td>Predefined Queries/Reports</td>
<td>Buttons that automatically execute predefined queries and reports.</td>
</tr>
<tr>
<td>User Queries/User Reports</td>
<td>Typically nothing; users can add their own buttons.</td>
</tr>
<tr>
<td>Ad Hoc Queries (one or more)</td>
<td>Data objects and relationships.</td>
</tr>
</tbody>
</table>
If there are hundreds of tables in the database, you need to decide whether they can be represented in one data model. Including a large number of tables can make the data model more complex and intimidating to use—especially if some users need access to only a few tables. One method of handling a large number of tables is to build a modular data model. A modular data model displays different sets of data objects and relationships in different Design windows; each set of objects in a window is related.

If different user groups or departments use some of the same data (such as tables that store employee information), and if they’re allowed access to each other’s data, you can build a separate data access environment for each department within the same data model. For example, you can create one set of data objects and relationships for Human Resources in one Design window, one set for Manufacturing in a second window, one set for Finance in a third window, and so on.

If your company produces bicycles and carries a number of product lines, it may be beneficial to organize the data model by product line. Different Design windows can display a set of data objects and relationships for road bikes, touring bikes, racing bikes, and so on.
On the other hand, if your company produces one product but packages it in a variety of ways, you can create two groups of data objects and relationships in one Design window—one group that represents the product and another that represents the packaging.

**Executive Information System Data Models**

When your data is complex and users aren’t technically skilled, an Executive Information System (EIS) data model provides a controlled environment that’s easy to use. It can also be used in the first phase of implementation to give users time to learn BI Query and the data model. And if there’s no time for training, it gets them up and running quickly.

An EIS data model provides an EIS interface. It organizes information hierarchically and combines a number of facilities to give users a quick way of finding exactly the information they need. Each window in the series provides increasingly detailed information. Buttons also run predefined queries and reports. The result is a focused query environment in which users can quickly drill down on increasingly detailed information.

You might include the following Design windows in an EIS data model:

<table>
<thead>
<tr>
<th>Window</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>A welcome; includes the company logo.</td>
</tr>
<tr>
<td>Building Queries</td>
<td>Data objects and relationships that users can use to build queries.</td>
</tr>
<tr>
<td>Series</td>
<td>Buttons linked to, for example, areas of the world and placed over maps of those areas. Clicking these buttons runs queries by sales region, country, or domestic versus international sales, generates different types of reports, and generates reports for different aggregation levels.</td>
</tr>
<tr>
<td>“For Help, Contact…”</td>
<td>Information on contacting internal technical support.</td>
</tr>
</tbody>
</table>

**To set up an EIS:**

1. Create the Design windows you want to have in the data model.
2. Load tables into the Design windows.
3. Make ornaments such as window titles, maps, diagrams, charts, and company logos that provide visual cues to the information in each window.

4. Create the queries and reports you want associated with buttons. Use prompts in the queries so that users can insert the necessary values to retrieve exactly the information they want.

5. Make buttons that submit the queries you have saved.

6. Test the buttons to ensure they work properly.

When you save the data model, the EIS interface is saved with it.

Optimizing the Database

BI Query provides a number of facilities for efficiently modeling relational databases. For example, you can merge data objects and use tables more than once to create a more productive environment for users to access data.

However, some users may request information that’s difficult to formulate or unnecessarily inefficient to obtain, resulting in a significant load on the database server. You can make changes to the database to improve performance and help users access the data they need.

The following sections describe the ways you can optimize the database for your users.

💡 If you are unsure of the meaning of any of the terms and concepts discussed in this section, consult an SQL reference manual or your DBMS documentation. For explanations of BI Query-specific terms, see “Basic Concepts” on page 18.

Creating Lookup Tables

BI Query’s data values facilities provide a number of mechanisms for obtaining a list of values from which users can choose when qualifying queries and entering values into qualification prompts.

The default mechanism executes a `SELECT DISTINCT` query for the attribute being qualified. However, if that attribute is an unindexed column of a table with a large number of rows (called a fact table), that mechanism may result in an computationally expensive query. To alleviate that potential problem, you can create a data values query that redirects the query to a different table—typically a smaller one—called a lookup table. You can create a lookup table in the database to support that mechanism.
For more information, see “Creating Tables” on page 113.

You can use BI Query to create a lookup table by submitting a SELECT DISTINCT query and submitting the results to a named table in the database. Each time you need to update the contents of the table, submitting the results of the new query overwrites the current table. Keep in mind that you also need to develop a procedure to ensure that when the fact table changes (values are added, modified, or deleted), the values in the lookup table are updated accordingly.

Adding Indices

You can dramatically decrease the time it takes to process a query when you use indices on the tables represented in the data model. For example, indexing any attributes that are used to join two tables improves performance. Indexing the attributes being qualified or sorted also speeds up querying time. Before deciding to index an attribute, you need to weigh the benefits versus the cost of maintaining the index. To find out what the issues are for your DBMS, see your DBMS documentation.

When deciding whether to index an attribute, consider the following:

- If an attribute is often used as a qualification in a query, it’s a candidate for indexing.
- If a table is large, at least some of its attributes are candidates for indexing.
- If a table is being heavily updated (that is, if one or more authorized users are adding, updating, or deleting data), the expense of maintaining the indices may outweigh the advantage of improved performance for the users.

For more information, see “Methods for Balancing Network Load” on page 233.

When you add indices to a table, keep in mind that most DBMSs can’t use an index for qualifications that use the LIKE operator.

Denormalizing Tables

The pattern of access to a data warehouse or database used primarily for decision support is very different than the pattern of access to an operational database used to capture data. The design of databases for OnLine Transaction Processing (OLTP) usually involves reducing redundancy by creating a schema of normalized tables. However, this design creates a large number of tables, which users can find confusing.
BI Query provides facilities to simplify the data model. For example, it lets you merge data objects. But you can achieve the same simplification by denormalizing the tables in the database. Denormalizing tables has a second major benefit in that the tables are usually accessed much more efficiently than the joins between normalized tables permit.

💡 For more information, see “Merged Data Objects” on page 104.

See your DBMS documentation for further information on this procedure.

### Storing Common Summarizations

Users can easily obtain summary information from detail data using BI Query. For example, using a Sales data object, the user can create a calculated attribute called Total Sales that sums sales amounts using the Amount attribute. If many users want this summarized information, you can store it in the database and make it available as part of the data model. This pre-computing of commonly used summary information can greatly reduce the load on the database server and improve query response time for each user.

💡 For more information on calculated attributes, see the *BI Query Queries User’s Guide*.

To determine what summary information users most frequently require, you may need to undertake some consultation or analysis. For example, you can interview users to gather their requirements, or you can monitor their queries and analyze the usage patterns that appear.

### Reorganizing Table Structures

The way in which users want to present the data they retrieve in BI Query can make it desirable to reorganize the data in the database. Reorganizing tables (possibly including denormalization or summarization) can improve how BI Query information is organized in results sets and presented in reports. The ability to reorganize tables is also constrained by the manner in which the data is fed to the database in the first place and may be most appropriate in data warehouse or data mart environments.

Before deciding on the optimum benefit, be sure that any conflicting requirements are analyzed.
Examples of Table Structures

The following example shows how the organization of data in a table affects the way information is presented in a BI Query report. The report compares sales for the last two quarters, and the user wants the information to look like this:

<table>
<thead>
<tr>
<th>Company</th>
<th>2nd Last Quarter</th>
<th>Last Quarter</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC Inc.</td>
<td>23,356</td>
<td>37,893</td>
<td>62.2%</td>
</tr>
<tr>
<td>XYZ Corp.</td>
<td>762,873</td>
<td>569,581</td>
<td>-25.3%</td>
</tr>
</tbody>
</table>

It’s easy for the user to retrieve this data when the table is organized in a wide format:

<table>
<thead>
<tr>
<th>Company</th>
<th>Q1</th>
<th>Q2</th>
</tr>
</thead>
<tbody>
<tr>
<td>char(20)</td>
<td>money</td>
<td>money</td>
</tr>
</tbody>
</table>

It is not so easy for the user to retrieve this data when the table is organized in narrow format:

<table>
<thead>
<tr>
<th>Company</th>
<th>Period</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>char(20)</td>
<td>char(2); (Q1 or Q2)</td>
<td>money</td>
</tr>
</tbody>
</table>

In the case of the narrow-format table, the user has to generate two queries (one for Period = Q1 and one for Period = Q2), join the results sets, and then generate the report.
The following organization over two tables once again requires the user to submit a query containing a join (or a two-query super query) to retrieve the required information:

<table>
<thead>
<tr>
<th>Table salesq1</th>
<th>Table salesq2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Company</strong></td>
<td><strong>Company</strong></td>
</tr>
<tr>
<td>char(20)</td>
<td>char(20)</td>
</tr>
<tr>
<td><strong>Q1</strong></td>
<td><strong>Q2</strong></td>
</tr>
<tr>
<td>money</td>
<td>money</td>
</tr>
</tbody>
</table>

The choice between making “wide” tables (tables with several columns containing the same type of data) and “narrow” tables (tables with one column that indicates the type of data and one column for the data) depends on the need to compare data of each type in a report calculation. For example, it’s easier to compare (calculate with) data in the same row than data in different rows. However, it’s easier to aggregate over all the data when it’s in a single column.

### Providing Metadata

You can provide metadata for the data objects and attributes in a data model to give users a clearer understanding of their data and help them build queries. For data objects, metadata replaces database table names with display names. For attributes, it replaces database column names with business terms. And it provides short descriptions for both that explain what kind of information they represent.

💡 For more information, see “Metadata” on page 192.

You set up the necessary queries to specify the metadata source. If your database doesn’t already store metadata, you can use BI Query to create it. Loading the metadata into the data model submits the queries to the metadata source and retrieves the metadata.
Creating Data Models

This section provides information on the following:

- “Main Components of Data Models” on page 76
- “Creating Data Models” on page 77
- “Creating Design Windows” on page 78
- “Creating Data Objects” on page 79
- “Creating Relationships Between Data Objects” on page 85
- “Specifying Join Conditions” on page 91
- “Duplicating Data Objects” on page 101
- “Merged Data Objects” on page 104
- “Query Data Objects (QDOs)” on page 108
- “Correlated Subqueries in BI Query” on page 111
- “Creating Tables” on page 113
- “Checking the Design” on page 117
- “Securing Data Models” on page 118
Main Components of Data Models

A data model acts as a gateway to the data in your database. A data model consists of one or more Design windows that store data objects and the relationships between them.

Design Windows

BI Query Design windows provide a variety of tools to help you to represent the database with data objects and relationships, as well as to create ornaments that provide visual cues and buttons that automate activities.

Data Objects and Relationships

You construct a data model from database tables, typically by loading tables from the database into a Design window. These tables are represented in the data model as data objects and relationships. Once you have created the objects and the relationships between them, you specify the join conditions that govern those relationships.

You can also reverse the process, creating data objects first then making tables in the database to correspond to those objects. You can add attributes to existing as well as new data objects.

Other Features

In order to make the data model more useful to your users, you can also do the following:

- Add metadata to your model to explain its contents. For more information, see “Metadata” on page 192.
- Set data model passwords to enhance security. For more information, see “Setting Data Model Passwords” on page 170.
- Make query files and buttons. For more information on creating queries, see the BI Query Queries User’s Guide. For more information on buttons, see “Buttons in BI Query” on page 151.
- Add ornaments and apply colors. For more information, see “Adding Ornaments and Other Objects” on page 134.
- Set data model permissions and preferences. For more information, see “Setting BI Query Permissions” on page 183.
Creating Data Models

The following procedure outlines the basic steps for creating a data model. The subsequent sections of this chapter describe each step in more detail.

When you create a data model, BI Query Admin must be in Design mode.

To create a data model:

1. Open a new data model.
2. In BI Query Admin, open at least one Design window. (You begin a new data model by representing existing database tables in a Design window.)
   If you know how many Design windows you will need, you can create all of them first. If not, you can create new Design windows as you go.
3. Assign a connection file to a Design window or to the data model. You can assign different connection files to different Design windows; this makes it possible for the data model as a whole to use data in more than one DBMS.
   For more information, see “Assigning Connection Files to Design Windows” on page 51.
4. Load a database table or tables into a Design window. (The tables become data objects.)
5. Create any additional data objects you want.
   If existing database tables are not sufficient, you can create new tables in the database, and then incorporate them into the data model. You can use the following methods:
   • Create a query to retrieve existing information from the database; then, save the query results to a table.
   • Make database tables from new data objects that you have created from scratch in the data model.
6. Add any necessary attributes or calculations to data objects and relationships.
7. Create join conditions needed to define the relationships among the objects.
8. Add queries and buttons.
9. Set security, permissions, and preferences for the data model.
10. Apply colors to Design windows and place objects or ornaments in them.
11. Check the data model design.
12. Save the data model.

Opening New Data Models

The first step in creating a data model is to open a new model. The new model displays a blank **Design** window. If you want the data model to be a split data model, you must assign the **Save Split Data Model** permission.

**To open a new data model:**

1. Run BI Query Admin. In the *Welcome to Hummingbird BI Query* window, click **New Data Model**. (If you are already running BI Query Admin and you've opened a data model, close it; then, on the **File** menu, click **New**.)

2. On the **File** menu, click **Save As**, and name the untitled data model in the **Save Data Model** dialog box.

Once you have named and saved the data model, the **Design** window remains untitled. In order to give the **Design** window a title, on the **Drawing** menu, click **Windows**. The **Design Windows** dialog box opens. In the **Name** box, type a new name, click **Rename**, and then click **Close**.

Creating Design Windows

**Design** windows organize the various components of a data model: data objects, relationships, buttons, and ornaments. After you have created these components, you can copy them, or cut and paste them, from one window to another. When you are organizing data objects and creating relationships between them, keep in mind that users can create queries that specify data objects or relationships in the same **Design** window; they cannot build queries using more than one window.

If you assign the **Edit Data Model** and **Save Data Model** permissions, users can customize **Design** windows and add, copy, move, edit, and delete their own buttons and ornaments. However, in a split data model, the objects and windows that you create for the data model are stored in the administrator layer; users cannot modify or delete them.

**To create a new Design window:**

1. In Design mode, on the **Drawing** menu, click **Windows**. The **Design Windows** dialog box opens.

2. In the **Name** box, type the name for the window; then, click the right arrow button ⬤.
You can also use the **Design Windows** dialog box to rename or delete a **Design** window. For more information, see the Help for your BI Query application.

3. To specify that the window should appear when the data model opens, select the window name from the **Design Windows** list; then, click **Startup**.

4. Click **Close**.

**Deleting or Renaming a Design Window**

Use the **Design Windows** dialog box to rename or delete **Design** windows. BI Query must be in Design mode.

**To delete or rename a Design window:**

1. On the **Drawing** menu, click **Windows**.

2. In the **Design Windows** dialog box, click the name of the window you want to rename or delete.

3. Do one of the following:
   - To rename the window, type the new name into the **Name** box; then, click **Rename**.
   - To delete the window, click **Delete**.

4. Click **Close**.

**Creating Data Objects**

The data objects that appear in BI Query represent the corresponding tables in the database. You can load tables into any **Design** window in a data model, and then move the resulting data objects, or cut and paste them, from one window to another.

Data objects and the attributes they contain inherit the names of their counterparts in the database. If the database names are not meaningful to users, rename them. The names you specify in BI Query do not affect the names of the corresponding tables and columns in the database.

Limiting the number of data objects and grouping related ones in a **Design** window breaks up the information and provides a more focused querying environment that is easier to understand and faster to use. When you include data objects in multiple windows, group them appropriately, keeping in mind that users can create queries in only one window at a time.
Users can create super queries to combine results from queries created in different Design windows. For more information on super queries, see the BI Query Queries User’s Guide.

Loading Tables from the Database

Database tables are represented in the Design window by data objects. BI Query Admin must be in Design mode and must be connected to the DBMS before you can load tables into the Design window.

Provide only the tables users need. Unnecessary tables clutter Design windows and make a data model more difficult to use and maintain.

To load tables from the database:

1. On the Host menu, click Insert Data Objects.

2. If the List Tables dialog box opens (displayed for certain databases), do the following:
   a. To display all tables in the database catalog, leave the Tables box blank. Alternatively, specify a subset of tables by typing the value you want in the Tables box.
      You can use the wildcard character, %, to specify any string of characters. (For example, type S% to select all tables with names that begin with the letter S.)
   b. To display only those tables created by a specific creator or database, type the value you want in the Creator (or Database) box.
   c. Click List. The Insert Data Objects dialog box opens.

3. In the Insert Data Objects dialog box, select the tables you want to load.

4. If you have specified a metadata source, but do not wish to load display names or short descriptions, clear Metadata display names and Metadata short descriptions.

5. Select Database keys if you want to retrieve information about primary and foreign keys. You can use database key information when creating join conditions. For more information, see “Creating Relationships Between Data Objects” on page 85. If the DBMS you’re using does not support primary or foreign keys, this setting is ignored.

When you retrieve database key information, BI Query automatically sets the Key field for any attribute identified in the database as a primary key.
6. Click **Insert**; then, **Close** to close the dialog box.

You can also create tables in the database using data objects. For more information, see “Creating Tables Directly in the Database” on page 115.

**Data Object and Attribute Names**

Data objects and attributes inherit the names of their counterparts in the database. However, you should rename these objects and attributes for the following reasons:

- Database names often are not meaningful to users. You could, for example, change the database name mjr_accnts to Major Accounts.
- Attribute names become column headings in reports. Giving attributes clear, meaningful names makes it unnecessary for users to change them when they generate reports.

Unlike the names used in the database, names in a data model can be meaningful words or phrases. When users formulate queries, BI Query generates the corresponding SQL string using the actual database names of the objects and attributes. The names you apply to data objects and their attributes do not affect the corresponding tables and columns in the database.

You can rename data objects and attributes by simplifying database names automatically or by specifying new names manually. If the **Show Relationships** and **Show Relationship Names** preferences are set, you can also rename relationships. After you have renamed an object or attribute, you can still view its database name.

**Guidelines for Renaming Data Objects and Attributes**

When you are renaming items in the data model, try to adhere to the following guidelines:

- Name data objects and attributes as early in the design process as possible. Do not use placeholder names then replace them with real names later. Changing the names after you create queries may invalidate the queries.
- Give a unique name to each data object, relationship, and attribute, even if its name is not visible. Objects with the same names can cause ambiguity and errors when users attempt to open stored queries.
- Use descriptive terms to name data objects and attributes. This makes it easier for your users to understand the data model. For example, instead of retailers.cust_num, use Retailer Number.
• If table names in the database contain a creator or owner prefix, do not include this prefix in the display names for the corresponding data objects. Including a creator or owner prefix in a display name can limit your ability to replace the prefixes for table names.

  ![Tip icon] For more information, see “Replacing Owner/Creator Names for Tables” on page 269.

• If the values returned by an attribute are short, use a short attribute name so users do not need to shorten it to fit the column in a report.

**Simplifying Display Names Automatically**

BI Query can automatically simplify the display names of data objects and attributes by doing the following operations:

• replacing punctuation marks with space characters (sales.id becomes sales id)
• mixing the letter case of names (sales order becomes Sales Order)

Database names are often more readable as display names if they are modified in this way.

  ![Tip icon] You can also simplify display names by using metadata. For more information, see “Metadata” on page 192.

**To simplify attribute and object database names:**

1. In Design mode, on the **Tools** menu, click **Simplify Names**.
2. In the **Simplify Names** dialog box, select the data objects whose names you want to simplify. To list all objects in the model, select **Data objects from all windows**.
3. To replace punctuation marks with space characters, select **Remove punctuation**.
4. To mix the lettercase of names, select **Convert to mixed case**.
5. Click **Simplify**. BI Query lists the simplified names for the first object and its attributes.
6. Do one of the following:
   • Click **Skip** to view additional simplified names without actually changing them in the data model.
   • Click **Change** to accept the simplified names for the listed data object and its attributes.
• Click Change All to accept the simplified names for all the data objects and attributes you selected. Repeat as necessary. The Message window lists any changes made to the model.

Renaming Data Objects Manually

In addition to automatically simplifying data object names, you can rename data objects manually. Names can contain spaces and other punctuation characters not allowed in the SQL string for a query. All data objects must have names, and their names must be unique.

💡 If you provide the same data object more than once, you can number each instance to distinguish it from the others.

To rename a data object:

1. In Design mode, double-click the table name in the Design window.
2. In the Object Name dialog box, replace the table name with a unique object name.
3. If you want BI Query to use the display name of the object as its database name as well, select the Use as database name of data object check box. If you want to specify a database name for the object that is different from its display name, leave this check box blank. You can edit the database name using the Edit Database Name dialog box.
4. Click OK.

Renaming Attributes Manually

In addition to automatically simplifying attribute names, you can rename attributes manually. As with data object names, attribute names can contain spaces and other punctuation characters.

💡 You can also edit the database names of attributes. For more information, see “Manually Editing Database Names in the Data Model” on page 247.

To rename an attribute:

1. In Design mode, double-click the data object containing the attribute you want to name.
2. In the attribute window, click the attribute name you want to change.
3. In the **Edit Attribute** dialog box, replace the existing attribute name with a unique name. Click **OK**.

**Organizing Attributes**

Data objects display attributes in the same order as the columns in the corresponding tables. If attributes are not organized in a logical order, you can reorder them in the attribute window to make it easier for users to find the ones they need.

💡 You can create an attribute that organizes groups of attributes—for example, to set off little-used attributes from those that are used often. On the **Design** menu, click **Insert Attribute**; then, type a row of asterisks or dashes instead of a name.

You can reorder attributes in the following ways:

- sort them alphabetically or numerically
- place the most commonly used ones at the top
- group date attributes together
- group address attributes together
- group attributes that return codes at the bottom of the list

**To organize attributes:**

1. In Design mode, select a data object.
2. On the **Design** menu, click **Reorder Attributes**. The **Reorder Object Attributes** dialog box opens. The order of attributes in the **Attributes** list corresponds to their order in the attribute window.
3. In the **Reorder Object Attributes** dialog box, do one of the following:
   - To sort all attributes in the list into ascending alphanumeric order (top to bottom), click the **Sort** button.
   - To apply a different order, select an attribute from the **Attributes** list; then, use the **Up**, **Down**, **Top**, or **Bottom** buttons to move the attribute to the desired position. Repeat for all other attributes you want to order.
4. Click **OK**.
Creating Relationships Between Data Objects

Relationships between data objects are necessary for formulating queries using more than one data object. You can create relationships using any of the following methods, either alone or in combination:

- Select the data objects you want to connect, and on the Design menu click **Auto Join** to have BI Query automatically create relationships based on database keys or matching attribute names. This method can save you a lot of time especially if you have just added your data objects and have not yet created any relationships. For more information, see “Creating Relationships Automatically” on page 87.

- From the Drawing toolbar, click the **Relationship** tool, then click and drag from one data object to another. This method is useful when you only need to add one or two relationships, or when your database doesn’t support using the **Auto Join** method (such as for databases that do not have primary keys).

- Select two data objects that you want to connect, then click **Connect objects** on the Design menu. Similar to using the **Relationship** tool, this method is useful when you only need to add a few relationships.

When you create relationships using **Auto Join**, the join conditions are defined automatically. When you create relationships between data objects using the other methods, you need to specify their join conditions.

**Types of Relationships**

Data models can contain two kinds of relationships, embedded and actual. Users can also create relationships for specific queries; these are known as dynamic relationships.

Embedded and actual relationships are stored in the data model. Users can store dynamic relationships in the data model if the **Save dynamic relationships** permission is assigned. Otherwise, users can store the relationships with the queries that use them.

A connecting line and an icon between two objects indicates their relationship.
Embedded Relationships
Embedded relationships connect objects using a relationship that is formed within (embedded in) the data model (and may not exist as such in the database). An embedded relationship connects exactly two data objects. When you connect two data objects together directly (without an intermediary object), you create an embedded relationship.

Actual Relationships
Actual relationships connect two or more objects using relationships that exist in the database. For example, some tables in the database may already contain data that relates to data in other tables. You can load such a table into the data model, and then convert it into a relationship that can join other objects together. The common object is called an “actual” relationship because it represents a relationship that actually exists in the database.

Actual relationships make it possible to represent many-to-many relationships in the data model. You can convert them back to data objects.

Example
A database contains a table with information about employees, another with information about committees, and another that relates information about employees and the committees on which they serve. You can load these tables into a data model, convert the one containing the related information into an actual relationship, and then connect it to the other two tables.

The relationship between employees and committees can be one-to-many (in which one employee can serve on several committees) or many-to-many (in which one employee can serve on several committees and each committee has several employees). Once you have connected the objects using the actual relationship, users can retrieve information about employees and committees based on the committees on which the employees serve.

Dynamic Relationships
When data objects do not have a relationship, or when existing relationships do not relate the attributes the way users need them to be related, users can create dynamic relationships if you have granted them the Dynamic relationships permission and they have set the Allow dynamic relationships preference. Dynamic relationships are typically created “on the fly” for temporary use.
Creating Relationships Between Data Objects

For more information about dynamic relationships, see the *BI Query Queries User’s Guide*.

**Relationship Icons and Connecting Lines**

When you connect data objects, BI Query represents their relationship using a connecting line. If you enable the *Show relationships* preference, BI Query indicates the relationship with a diamond-shaped icon. If you enable the *Show relationship names* preference, BI Query also displays the relationship name. Actual relationships are distinguishable from embedded relationships by a shadow behind the relationship icon.

When **Design** windows contain a number of data objects, you may prefer to not display relationship icons or names in order to keep the window from appearing cluttered. In this case, users see only the data objects and the lines that connect them. When you have enough room to display relationship icons and names, they help users understand how the information in one data object relates to the information in another.

Relationship icons and their names are displayed by default. If you do not want to display them, on the **Tools** menu, click **Preferences**; then, clear the *Show relationships* and *Show relationship names* preferences. You can display relationship icons without displaying their names.

**Creating Relationships Automatically**

BI Query can create relationships for you automatically using database keys (if your database supports them and if you have enabled them in your data objects), and matching attribute names. For information on enabling database keys, see “Setting Key Fields” on page 273.

Rather than using the **Relationship** tool to create each relationship individually, you can select multiple data objects (and actual relationships), and have BI Query both discover the relationships among the selected objects and specify the join conditions automatically.

In creating the relationships, BI Query can either look for data objects in which the primary key field identified in one data object matches the foreign key field identified in another data object, or for data objects in which the primary key field identified in one data object matches an attribute name from another data object. You can specify which of these options you prefer.
The **Auto Join** command can also create relationships between data objects and actual relationships, but not between two actual relationships. You must change the data object to a relationship diamond before using **Auto Join**.

When you use the **Auto Join** command to create relationships, BI Query places the relationships and connecting lines along the shortest path between the objects. If you have selected many data objects, and they have many relationships among them, this may initially create a tangle of lines criss-crossing your **Design** window. You can try to avoid this by moving data objects that you know will have some relationship between them closer to one another before using the **Auto Join** command, or by moving data objects with many joins to the middle of the **Design** window. Otherwise, you can use the **Auto Join** command first, then click and drag the data objects until you have them where you want them.

**To specify the criteria BI Query uses to create relationships automatically:**

1. On the **Tools** menu, click **Preferences**.
2. In the **Preferences** dialog box, click **Keys**. The **Keys** dialog box appears.
3. In the **Keys** dialog box, do one of the following:
   - Click **Foreign keys** to have BI Query look for data objects in which the primary key field identified in one data object matches the foreign key field identified in another data object.
   - Click **Attribute names** to have BI Query look for data objects in which the primary key field identified in one data object matches an attribute name from another data object.
4. Click **OK**.

To automatically join on foreign keys (for all DBMS but Teradata), BI Query requires that the primary key and foreign keys match in four ways:

- sequence number
- name of table
- name of column
- name of primary key

For Teradata, only the name of the table and column in the foreign key are used.
Creating Embedded Relationships

When you connect two data objects directly, you form an embedded relationship between them. When you save the data model, the relationship is saved with it.

To create an embedded relationship:

1. In Design mode, select both data objects (SHIFT + click).
2. On the Design menu, click Connect Objects.
3. Double-click the default relationship name.
4. In the Object Name dialog box, replace the default name with the name you want; then, click OK.

You can also use the Relationship tool on the Drawing toolbar to connect two data objects.

Creating Actual Relationships

Relationships that you convert from data objects are actual relationships.

To convert a data object to an actual relationship:

1. Ensure that the Show relationships and Show relationship names preferences are set. (On the Tools menu, click Preferences, and select the appropriate preferences in the Preferences dialog box.)
2. In Design mode, click the data object.
3. On the Design menu, click Data Object <-> Relationship. The rectangle in the Design window becomes a diamond to indicate that it is a relationship.
4. Click outside the relationship to deselect it. A shadow appears behind it to indicate that it is an actual relationship (a relationship existing in the database).

You can convert actual relationships back to data objects in the same way.

Connecting Data Objects to an Actual Relationship

Once you have converted a data object to an actual relationship, you can connect other objects to it to form the relationships supported in the database.
To connect data objects to an actual relationship:

1. To connect the actual relationship to a data object, select the first data object and the actual relationship (SHIFT + click); then, on the Design menu, click Connect Objects.

2. Double-click the relationship line, and in the Join Conditions dialog box, create the join conditions you require.

   For more information on join conditions, see “Specifying Join Conditions” on page 91.

3. Repeat the process to connect the actual relationship to the other data object and to establish the join conditions.

You can also use the Relationship tool on the Drawing toolbar to connect the data objects to the actual relationship.

Naming Relationships

You must provide unique names for all relationships in a data model. Relationship names help users understand how the information in one data object relates to the information in another. As with data objects and attributes, the names you apply to relationships do not affect the names of the corresponding tables or columns in the database.

To name a relationship:

1. In Design mode, on the Tools menu, click Preferences.

2. In the Preferences dialog box, make sure Show relationships and Show relationship names are selected; then, click OK.

3. Double-click the name of a relationship.

4. In the Object Name dialog box, type the name of the relationship; then, click OK.

You should name relationships early in the design process. Do not use temporary names. Changing names after you create queries may invalidate the queries that use those relationships. (The same is true for data object names.)
Specifying Join Conditions

Once you have connected data objects by creating relationships, the next step is to create join conditions. Join conditions let users include attributes from both data objects in a query. If the data objects are related by an actual relationship, users can also include the attributes of the actual relationship.

A join condition relates two data objects by comparing their attributes. To create a join, you must select attributes from both objects (using their database names, rather than their display names) and specify the operator that compares them. When a user formulates a query using joined tables, the DBMS finds all possible combinations of data from all tables, and then eliminates the data that does not satisfy the join conditions.

💡 A join which uses the “=” operator is known as an equijoin.

A join conventionally has a “left” data object and a “right” data object. The left data object supplies the attribute that lies on the left side of the comparison operator; the right data object supplies the attribute that lies on the right side. A given join can relate more than one pair of attributes.

Example

The Employee and Committee data objects each connect to the actual relationship, ServesOn.

In order to create a query that can determine which employee serves on which committee, you must specify the join conditions for the two connections. You could join the employee_id attribute of the Employee data object with the employee_id attribute of the ServesOn relationship; then, join the committee_name attribute of ServesOn with the committee_name attribute of the Committee data object. In this case, Employee is the left member of the join with ServesOn; the join condition between them is Employee.employee_id = ServesOn.employee_id. ServesOn is the left member of the join with Committee.
Types of Joins

There are two types of joins that you can use to specify the relationship between two data objects: inner joins and outer joins.

**Inner Joins**

Inner joins—also known as regular joins—include only those rows that satisfy the join conditions. BI Query presents each inner join as a connection line with no arrow.

```
SQL Excerpt: A INNER JOIN B ON A.x = B.x
```

**Outer Joins**

Outer joins can include rows that do not match in the joined columns. Outer joins are said to be row-preserving because they can return rows that do not satisfy the join condition. There are two types of outer joins, half and full.

💡 For information on using outer joins in queries, see the *BI Query Queries User’s Guide*. 
An outer join combines the data in the joined attributes in all the possible ways that satisfy the join condition. If the joined attributes in an outer join contain duplicate values, the results set is a cross-product expansion of rows that may be larger than both of the joined objects.

**Half Outer Joins**

A half outer join includes all rows from one object and only matching rows from the other. For example, a left outer join includes all rows from the “left” data object; the “right” data object supplies only those rows that match in the joined column. A single arrow on a connection line indicates a half outer join.

![Diagram of half outer join]

**Arrows on a connection line always point away from the row-preserving object or relationship.**

BI Query sets the join position of a data object (left or right) when it generates the SQL for a query. For example, if you have created an outer join between two objects, A and B, in which only A is row-preserving, the SQL for the join could be either of the following:

- A LEFT OUTER JOIN B ON A.x = B.x
- B RIGHT OUTER JOIN A ON B.x = A.x

The join position for an object corresponds to its position in the OUTER JOIN statement.

**Full Outer Joins**

A full outer join preserves all rows from both connected objects, even if there is no match in the joined columns. A double arrow on a connection line represents a full outer join (to indicate that both sides of the relationship are row-preserving).

![Diagram of full outer join]

**SQL Excerpt: A FULL OUTER JOIN B ON A.x = B.x**
If your DBMS supports outer join syntax, you can specify an outer join in a join condition. If your DBMS or middleware does not support outer joins, you can create a combined query (super query) that runs multiple queries and performs the outer join automatically. The result is a query that retrieves the same results as if the underlying database supported the outer join. For more information on super queries, see the *BI Query Queries User’s Guide*.

**Join Requirements**

Join conditions between data objects must meet various requirements.

**Join Number**

Embedded relationships require a minimum of one set of join conditions to join the data objects they connect. If the join is an outer join, the row preservation applies to both connection lines in the relationship; BI Query mirrors the row-preservation arrows on either side of the relationship icon.

Actual relationships require a join condition for each connecting line in the relationship. If the join is an outer join, you must also set the row preservation for each connecting line.

In both types of relationship, if the row-preservation arrows point in the same direction across the connecting lines, then there is complete preservation of rows across the relationship.

**Attribute Data Type**

The attributes you select to join data objects must represent the same or similar data type. For example, integer 12252003 does not map to date 12252003 even if the numerical quantity (12252003) looks the same in both cases.

If you attempt to join attributes that have different data types, a warning appears. If the data types are similar, such as varchar and char, you can override the message.

**Outer Join Capabilities**

Before you create outer joins, you need to connect to the database. BI Query automatically determines the available outer join syntax types and other capabilities of the database when you connect. Without this information, any outer joins you create may be invalid.
For more information on outer join capabilities, see “Connection Files and Outer Join Support” on page 58.

Available Outer Join Syntax Types

You can create outer joins using the ODBC, Oracle, ANSI SQL/92 or “Classic” join syntax. The “Classic” (ANSI SQL/89) join syntax uses the legacy join operators *= and =* to specify left and right outer joins, respectively.

For a comparison of ANSI SQL/89 and SQL/92 syntax, see the Help for your BI Query application.

The syntax you can use for outer joins depends on the following:
- the DBMS that stores the tables joined in the data model
- the connectivity you use to connect to the DBMS

Not all DBMSs and connectivities support outer joins; those that do may support only a particular type of outer join syntax. BI Query determines the available syntax types for a given Design window each time you connect to a DBMS; it stores this information in the corresponding connection file. Since you can assign a separate connection file to each Design window, the available outer join syntax may change from window to window within your data model.

The connection for a Design window determines the available syntax type(s). In turn, the syntax type you select determines the particular outer join capabilities for the connection. (Different databases have different outer join capabilities depending on the outer join syntax you use.) For example, Microsoft’s SQL Server supports both the Classic syntax and the ANSI SQL/92 syntax, but has different outer join capabilities for each. For more information on outer join capabilities, see “Outer Join Syntax and Other Database Capabilities” on page 59.

Example: ANSI Syntax for Outer Joins

In the ANSI SQL/92 standard, join syntax is specified in the FROM clause, not in the WHERE clause as in most DBMS-specific syntax. The examples below show a right outer join first in ANSI SQL/89 outer join extensions syntax, and then in ANSI SQL/92 syntax.
Example: ANSI SQL/89 Syntax

```sql
SELECT *
FROM Employees E, Departments D
WHERE E.dept_id =* D.dept_id;
```

Example: ANSI SQL/92 Syntax

```sql
SELECT *
FROM Employees E RIGHT OUTER JOIN Departments D
ON (E.dept_id = D.dept_id);
```

Outer Join Check Boxes

BI Query does not use the legacy operators *= and =* to represent outer joins in the ODBC, Oracle, or ANSI SQL/92 syntax types. Instead, it uses two check boxes, Include All Left Rows and Include All Right Rows. These check boxes appear in all dialog boxes that join data objects or results sets—namely, the Join Conditions dialog box, the Choose Relationship dialog box, and the Join Columns dialog box.

The legacy outer join operators and the check boxes are not compatible. You cannot use both in a given Design window (or data model, if the entire model uses a single connection file). However, after you have created a Design window or data model using a particular outer join syntax, you can change the syntax for the purposes of building and running queries.

💡 For more information on changing the join syntax for queries, see the BI Query Queries User’s Guide.

You can also convert legacy join operators to a new syntax. Once you have converted the legacy operators in a Design window (or across the data model), you can then use the check boxes.

💡 For more information on converting legacy join operators, see “Updating Outer Joins in a Design Window” on page 250.

💡 The check boxes are available only in Design windows that use the ODBC, Oracle, or ANSI SQL/92 join syntax, and only if the Design window does not contain a legacy outer join operator.

The DBMS to which a Design window or data model connects also determines the availability of the check boxes. For example, if the DBMS does not support full outer joins, the two check boxes will not be available at the same time.
Creating Join Conditions

Use the Join Conditions dialog box to specify the join conditions for a relationship between two objects. This dialog box shows the attributes for the two objects in the Data Object 1 and Data Object 2 lists.

💡 For information on editing existing join conditions, see the Help for your BI Query application.

To create a join condition:

1. In Design mode, do one of the following:
   • For an embedded relationship, click the relationship icon or connecting line. On the Design menu, click Join Conditions.
   • For an actual relationship, click a connecting line (not the icon). On the Design menu, click Join Conditions.

2. In the Join Conditions dialog box, select an attribute from the Data Object 1 list; then, click the down arrow button. The attribute moves to the Join Conditions box on the bottom left.
   (To remove an attribute from one of the Join Conditions boxes, select it; then, click the X button.)

3. Select a related attribute from the Data Object 2 list; then, click the down arrow button. The attribute moves to the Join Conditions box on the bottom right.

4. Specify the join type using the appropriate procedure:

<table>
<thead>
<tr>
<th>Join Type</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inner join</td>
<td>Click the operator box; then, select a join operator from the pop-up menu that opens.</td>
</tr>
<tr>
<td></td>
<td>If the Design window does not use the Classic syntax, ensure that the Include All Left Rows and Include All Right Rows check boxes are cleared. If the Design window does use the Classic syntax, ensure that the join operator is not <em>= or =</em>.</td>
</tr>
</tbody>
</table>
Chapter 5: Creating Data Models

5. Click **OK**.

You cannot create a full outer join using the legacy outer join operators. However, you can create a combined query (super query) that runs multiple queries and performs the outer join automatically. For more information on super queries, see the *BI Query Queries User’s Guide*.

### Including Prompts and Variables in Join Conditions

You can include a prompt in a join condition to force users to qualify any query that specifies the related data objects. You can also include a variable in a join condition to let users qualify a query dynamically.

For tables that contain a large number of rows, prompts can improve query performance and protect users from running queries that retrieve more data than they need.

Variables can improve the flexibility of queries. For example, your database may be updated regularly to reflect current sales; including a variable for the current month into a join condition lets users obtain sales figures at any time during the month.
To create a join condition that includes a prompt or variable:

1. In the **Join Conditions** dialog box, add the attribute to which the prompt will be joined.
2. Click in the opposite column where you want to add the second attribute.
3. Click **Edit**. The **Edit Join Condition** dialog box opens.
4. Do one of the following:
   - To insert a prompt, click **Prompts**; then, use the **Prompts** dialog box to select and insert a prompt.
   - To insert a variable, click **Variables**; then, use the **Variables** dialog box to select and insert a variable.
5. Click **OK**.
6. In the **Join Conditions** dialog box, click **OK**.

**Troubleshooting Missing Outer Join Operators**

Legacy outer join operators (\*= and =*) appear in the operator pop-up menu of the **Join Conditions** dialog box if both of the following are true:

- the database and the middleware support outer joins
- the **Design** window uses the “Classic” outer join syntax

If both of these conditions are true, but the operators are not available in the **Join Conditions** dialog box, proceed as follows:

1. Ensure that the Connect Per Query preference is disabled and that you are connected to the database.
2. Verify that the database supports outer joins. (See the documentation provided with the database.)
   - BI Query checks for outer join support each time you connect; it presents the results in the **Join Syntax** dialog box.
3. Make sure the middleware you are using supports outer joins. Check the documentation provided with the middleware.
   - If the middleware does not support outer joins, you will not be able to use outer joins in BI Query. Contact your vendor to see if a newer version exists that supports outer joins.
If the database and the middleware both support outer joins and an outer join operator is still unavailable, the middleware might not be communicating correctly with BI Query. If you are not using an ODBC driver, contact Technical Support. Otherwise, proceed with the remaining steps:

4. If you are using an ODBC driver, run a trace utility, such as Dr. DeeBee Spy, to log the messages sent between BI Query and the driver. (If you do not have a trace utility, contact Technical Support.)

5. Examine the log file. (Alternatively, send the log file and a completed Technical Support form to Technical Support. A support representative will examine the log, and then contact you.)

6. Open the log file in a text editor such as Notepad. (For example, if you are using Dr. DeeBee Spy, open `drdeebee.log`.)

7. Search for the word “join”. You should find an entry in the log file. Here are two examples from Dr. DeeBee Spy log files:

<table>
<thead>
<tr>
<th>Supports Outer Joins</th>
<th>Does not Support Outer Joins</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQLGetInfo 0x01010000</td>
<td>SQLGetInfo 0x01010000</td>
</tr>
<tr>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SQL_SUCCESS</td>
<td>SQL_SUCCESS</td>
</tr>
</tbody>
</table>

One of the lines in the entry (in this case, it is the fourth line) indicates whether the ODBC driver supports outer joins.

If you see an F or an N, your ODBC driver does not support any outer joins. Contact the vendor to see if a newer version exists that supports outer joins.

If you see a T or a Y, your ODBC driver supports outer joins, but not necessarily all types. Check the **Join Conditions** dialog box again. If you still do not see the outer join operator(s) you want to use, send the log file to Technical Support.

**Editing Join Conditions**

Use the **Join Conditions** dialog box to edit existing joins.
Before you edit a join, make sure that you have used the current connection file to connect to the database. BI Query requires database capability information (stored in the connection file) for any operation involving joins.

To edit join conditions:

1. In Design mode, click the relationship or connecting line. On the Design menu, click Join Conditions. The Join Conditions dialog box opens.

2. Do any of the following:
   - To change the join operator, click in the operator box (the middle box in the bottom section of the dialog box) and select a join operator from the pop-up.
   - To replace one attribute with another, click an attribute; then, click the X button. Click a replacement attribute; then, click the down arrow button.
   - To change an attribute name or to replace an attribute with a prompt or variable, click an attribute; then, click Edit. In the Edit Join Condition dialog box, you can type over the attribute or replace the attribute with a prompt or variable by clicking Prompts or Variables, and then selecting and inserting the prompt or variable you want. Click OK.

3. In the Join Conditions dialog box, click OK.

**Duplicating Data Objects**

It may be necessary to create a copy of a data object in a Design window to let users formulate certain types of queries that would not otherwise be possible—queries in which the data object has two roles.

If you want users to be able to use both objects in a single query, you need to specify one or more correlation names. A correlation name (or alias, in SQL terminology) lets BI Query distinguish the roles of the two data objects in a query.

You must also make the display names for each object unique; in order to make the model understandable to the user, the display name of each object should reflect its content.
Example

A Design window contains a data object for an Employee table and another for a Department table. By including the Employee and Department data objects in a query, users can retrieve a list of the names of employees and the departments they work in.

Users may also need to find out information about employees who supervise a particular department. In this case, you can copy the Employee data object and create a second relationship with the Department data object that lets users retrieve that information. The join condition for the second relationship would return rows only for those employees who are supervisors (for example, Employee.title = “Supervisor”).

To reflect the new relationship with the Department data object, you could name the copy Supervisor. You also need to supply a distinct correlation name for one or both of Employee and Supervisor so that BI Query can correctly process any queries involving these objects.

By including Supervisor and Department in a query, users can retrieve a list of the names of employees and the departments they manage.

By including the Employee, Department, and Supervisor data objects in a query, users can retrieve the names of supervisors and the departments and employees they manage.
Copying and Pasting Tables

Users may find it useful to formulate queries in which a data object has more than one role. To facilitate this, you can copy existing objects and create new relationships for them with other data objects.

To duplicate data objects:

1. In Design mode, select the data object that corresponds to the table. On the Edit menu, click Copy.

2. In a Design window, click where you want the copy to appear. On the Edit menu, click Paste.

3. Give the copy a unique display name.

4. Create a relationship between it and another data object.

5. Specify a correlation name for the original object, the copy, or both.

Specifying Correlation Names

If you have created data objects from the same table, and joined them, you need to give one or both of them a correlation name. Some databases require a correlation name for both the original object and the duplicate; others require one for the duplicate. It is good practice to assign a distinct correlation name to both and do so as soon as you make the copy.

Users cannot create dynamic relationships between two identical data objects unless you have given a correlation name to at least one of them.

When you create a correlation name, BI Query preserves the database name in order to generate valid queries. You can specify any correlation name—even the same name as the data object—as long as it does not contain any spaces and is not the same as the correlation name for any other data object in the same Design window.

To specify a correlation name for a data object:

1. In Design mode, click the data object.

2. On the Design menu, click Edit Database Name.

3. In the Edit Database Name dialog box, type a correlation name into the Correlation Name box.

4. Click OK.
Merged Data Objects

You can simplify a **Design** window that contains many data objects by merging some of them. When you merge an object into another object, the attributes from the first object are added to the second; the first object is removed from the **Design** window. Merging lets you group related attributes together, even when they are stored in different database tables.

You can facilitate merging by placing objects that have the greatest number of connections to other objects on the inside of a set of objects and relationships. Place objects with fewer connections on the outside.

The following rules apply when you merge objects:

- You can merge data objects only when there is a relationship between them and the relationship has a join condition assigned to it. The join condition remains associated with the object that has been merged and is applied when a user builds a query that uses an attribute from the object.
- You cannot merge data objects joined by an actual relationship.
- Once you have merged a data object, you can merge other objects into it, but you cannot merge it into another.
- If you merge one data object into another when there is more than one relationship between them or when the first object has a relationship with a third, the relationships are lost. A useful guideline is to merge “outside objects in”: outside objects (those on the perimeter of the data model) do not generally have connections to other objects.
- You can unmerge a previously merged data object.
- The join order in a merged object matches the order in which you merge the data objects (unless you have selected either the Automatically Order Outer Joins preference or permission, in which case BI Query automatically orders joins within the merged data object).
**Example**

In the data model shown below, the Design window contains the related data objects Retailers and Retailer Phones.

![Diagram showing the relationship between Regional Offices, Retailer Phones, Products, Staff, Retailers, and Sales.](image)

When Retailer Phones is merged into Retailers, it is removed from the Design window and its attributes are added to Retailers. The join condition remains associated with Retailer Phones and is applied when a user selects an attribute from Retailers that was in Retailer Phones. Other objects can be merged into Retailers, but it cannot be merged into another.

![Updated diagram with merged Retailer Phones into Retailers.](image)

**When to Merge Data Objects**

Merging data objects is useful when the benefit—a Design window with fewer data objects—outweighs the complexity of the resulting objects. For example, if a data object is used often but contains only two attributes, and a related object contains 20 attributes, adding the two attributes of the smaller object to those of the larger one will create an object with 22 attributes—not appreciably larger or more complex than the original. If, however, you merge two objects into a third object, and the final object has 50 attributes, it may be too large, and its relationships too complex, for easy use.

When you want to merge data objects, it is good practice to identify the object that is most important from a users’ perspective, and then merge other objects into it so that users can easily find the attributes they need.
Using BI Query to update tables in the database also plays a role in determining which objects to merge. If some tables are often updated, they should not be merged. Merged tables cannot be used to update the database.

**Examples of When to Merge Objects**

Consider merging objects in the following situations:

<table>
<thead>
<tr>
<th>Situation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>A data object is not used very often or does not contain many attributes.</td>
<td>Retailer names, addresses, and phone numbers might be contained in three separate data objects, each containing a few attributes. Merging these objects creates a new object that is small enough for users to work with.</td>
</tr>
<tr>
<td>Two or more data objects contain related information, and it would be</td>
<td>Information on regional offices and regions might be contained in two separate objects. You can merge the objects because it would be logical for users to look for information about regional offices in the regions object.</td>
</tr>
<tr>
<td>logical for users to look for that information in a single object.</td>
<td></td>
</tr>
<tr>
<td>Two or more data objects contain some of the same information, and merging</td>
<td>Information on employees might be contained in two objects—one containing general information on all employees, the other information on some employees’ management experience. Merging the second object into the first results in one object that contains complete employee information.</td>
</tr>
<tr>
<td>them would create one object that contains complete information.</td>
<td></td>
</tr>
</tbody>
</table>

**Merging Data Objects**

When information is stored in more than one table in the database, but you want to represent it in the data model as one data object, you can merge data objects. The attributes from the first data object are merged into the second. The merge operation that consolidates data objects is saved with the model.
To merge one data object into another:

1. Select the data object. On the Design menu, click Merge Data Object. The Merge dialog box opens for the selected object. The title bar of the Merge dialog box indicates the selected data object.

2. Do one of the following:
   - If the selected object connects to only one other object, click OK.
   - If the selected object connects to multiple objects, select the second data object from the Merge Into list; then, click OK.

3. If the data object specified is connected to another in addition to the one you want to merge it into, a warning appears. Click OK to continue the merge or Cancel to return to the Design window.

If there is more than one relationship between the objects, or if the relationship between them is an actual relationship, you need to set new join conditions for the merged object. For more information on editing merged objects, see below.

Editing Merged Objects

When you build a query that uses attributes from a data object that has been merged, that object’s join conditions are applied. If you want to change the way BI Query restricts the results returned from the database, you can edit the object’s join conditions. The more join conditions you use, the more restricted the results.

You also need to set new join conditions if there is more than one relationship between objects in a merged object or if the relationship between them is an actual relationship.

To edit the join conditions in a merged object:

1. Click the object.
3. In the Choose Merged Table dialog box, select the merged object whose joins you want to edit. Click OK.
4. In the Join Conditions dialog box, edit the join conditions as required.

💡 For more information on editing joins, see the Help for your BI Query application.
Unmerging Data Objects

When you have merged a data object with one or more other data objects, you can unmerge it. When you unmerge a data object, you separate it from the other object it was combined with and put it back in the Design window as a distinct data object. The relationship between the unmerged data object and the data object it was previously merged with is also put back in the Design window.

- Relationships that were removed when you merged the data object are not restored. Also, join conditions that you merged using the Merge Joins command remain merged.
- The properties of the unmerged data object, such as its icon, are restored but not the original location of the object in the Design window. You may need to reposition the object.

To unmerge a data object:
1. In Design mode, click the merged data object.
2. On the Design menu, click Unmerge Data Object.
3. In the Unmerge Data Object dialog box, click the data object(s) you want to unmerge.
4. Click Unmerge.

Query Data Objects (QDOs)

You can combine information from more than one data object by creating query data objects. A query data object (QDO) is a query that has been converted into a data object. It is a specialized mechanism for querying the results of another query. Query data objects let you personalize information for users by creating a data object that contains frequently used attributes. The attributes can be from several different data objects. If your users require frequent access to specific attributes, you can make their querying tasks easier by creating a query data object that contains all of these attributes.

You cannot run default data values queries against the attributes of a query data object. As a result, if you create query data objects that others will query, it is good practice to provide data values queries or results files for them. For more information on data values files, see the BI Query Queries User's Guide.
Example

The sales team needs information from the Inventory, Sales, and Marketing data objects, but they only use one or two attributes from each data object. You can create a query data object that contains only the attributes that they use regularly. Instead of selecting two or three different data objects when they build queries, the sales team now needs to use only one data object.

Storing Query Data Objects

You can store a query data object in the data model, making it unnecessary to save the query in a separate query file. Your users can build other queries that include the query data object. And you can restore the query from the query data object, edit it, and recreate that object or create another object with it.

View Permission for Query Data Objects

A query data object has the advantages associated with creating a table in the database, but it does not require that the user have database permission to create actual tables. However, the user does need database permission to create views. When you run a query that uses a query data object, the DBMS creates “views” of the data that are substituted for actual database tables. As a result, anyone who uses the query data object must have permission to create views in the database. In order to convert a query into a data object, you must be working in BI Query Admin and have permission to create views in the database.

Converting Queries into Data Objects

A query data object is a query that has been converted into a data object in a data model.

![Note](image.png)

You can not convert a freehand query to a query data object.

You can delete, copy, rename, or move a query data object just as you would any other object in a data model. However, you cannot edit the names of query data object attributes because a query data object is a view, not a table, and there are no actual database columns represented by the attributes.
To convert a query into a data object:
1. Ensure that you have database permission to create views.
2. In BI Query Admin, create or open the query that you want to turn into a data object. Run the query and ensure that it runs to completion and returns the correct data.
3. Click in the data model where you want the data object to appear. If you opened a super query, also click in the Super Query window.
4. On the Query menu, click Convert to Data Object.
5. Specify a name for the query data object in the Query Data Object Name dialog box, and click OK.

It is good practice to remove all prompts from qualifications of a query before converting that query into a data object. Prompts may cause certain errors when you query the new data object. Once the query data object has been created, however, the queries you run against it can be qualified with prompts.

Editing Query Data Objects

You can edit a query after it has been converted by restoring it from the query data object and modifying it as you would any other query, except that you cannot edit the SQL manually. You can then reapply the modified query to the query data object or convert it into a new object. If attributes or objects are missing from the model, they are removed from the query.

To edit a query after converting it to a query data object:
1. Click the query data object to select it.
2. On the Query menu, click Restore from Data Object.
3. Edit the query using the Query menu or an attribute window. (You can't edit the SQL manually in the Query window.)
4. On the Query menu, click Convert to Data Object. In the Query Data Object Name dialog box, click OK.
5. In the Replace Query Data Object dialog box, do one of the following:
   • Click Create to convert the query into a new query data object.
   • Click Replace to convert the query and overwrite the original query data object.
Correlated Subqueries in BI Query

You can use a query data object to create a correlated subquery. Normally, when you qualify an attribute with a subquery, BI Query executes the subquery once and then uses the results to qualify the attribute. In other words, BI Query queries the database twice: once to retrieve the results for the subquery—the inner query—and once to retrieve the results of the outer query—the query that is qualified by the subquery. However, in a correlated subquery, BI Query executes the inner query for each value in the qualified attribute.

💡 For more information on building queries and subqueries, see the BI Query Queries User’s Guide.

In the following example, the Store object contains store information, such as store names and store IDs; the Sales object contains information on orders placed for each store, such as order dates and order IDs.

If you want to retrieve order information for the last order placed for each store, you need to create a correlated subquery. The inner query applies the MAXIMUM function to the order_date attribute to return the last date recorded in the Sales data object for each store:

```sql
SELECT MAX(Sales.order_date) FROM Sales
WHERE Sales.store_id = Store.store_id;
```

The outer query retrieves the order information from the Store and Sales objects, using the inner query to qualify the order_date attribute:

```sql
SELECT Store.store_name, Sales.order_id, Sales.order_date
FROM Sales, Store
WHERE Store.store_id = Sales.store_id
AND
(Sales.order_date IN (SELECT MAX(Sales.order_date) FROM Sales
WHERE Sales.store_id = Store.store_id));
```

💡 Because the inner query can return multiple values, the qualification for the order_date attribute is enclosed in parentheses and uses the IN operator.

The `WHERE` clause in the inner query ensures that the inner query returns one result (maximum order date) for each store in the outer query. If you do not include the join condition in the inner query,

```sql
SELECT MAX(Sales.order_date) FROM Sales
```

The relationship between the two objects uses the store_id attribute to join them.
then the outer query returns order information only for those stores that had orders placed on the last order date recorded in the database. In this case, the inner query does not produce a result for each store. You must include the join condition between the Store and Sales objects to retrieve the last order date for each store.

**Creating Correlated Subqueries with QDOs**

Because query data objects let you query the results of another query, you can reproduce the results of a correlated subquery using a QDO. The object you create must store the results of the inner query; you must also be able to join the created object to the existing objects in a way that reproduces the `WHERE` clause of the inner query.

**To reproduce a correlated subquery using a QDO:**

1. Create a query that returns the results of the inner query. The query must contain the following attributes:
   - the attribute that supplies the qualification values for the inner query
   - the attribute that joins the data objects that appear in the outer query
   
   For example, if you want to use the Store and Sales data objects to retrieve the order information for the last order date for each store, the query must select the following attributes:
   - `MAX(Sales.order_date)"—which retrieves the last order date
   - `Sales.store_id"—which forms the join condition between the Store and Sales data objects

2. Run the query and ensure that the results are correct (for example, they must list the last order date for each store).

3. On the **Query** menu, click **Convert to Data Object**.

4. In the **Query Data Object Name** dialog box, specify a name for the QDO (for example, Last Order Dates); then, click **OK**.

5. Join the QDO to the object from which it was created; use all the attributes in the QDO to form the join condition.
For more information on joining objects, see “Creating Relationships Between Data Objects” on page 85.

For example, to join Last Order Dates to Sales, use these conditions:

- `[Last Order Dates].store_id = Sales.store_id`
- `[Last Order Dates].MAX_order_date = Sales.order_date`

For this step to work, the Trim Relationships option must be disabled. For more information on this option, see the BI Query Queries User’s Guide.

6. Create the query to retrieve the information you want. You do not need to include in the query any of the attributes in the QDO, but you must select the relationship that joins the QDO to the other objects (to ensure that the join conditions created in step 5 apply to the query).

The WHERE clause contains the join conditions that connect the objects and the qualification that reproduces the inner query:

```sql
SELECT Store.store_name, Sales.order_id, Sales.order_date FROM [Last Order Dates], Sales, Store
WHERE Store.store_id = Sales.store_id
AND Sales.store_id = [Last Order Dates].store_id
AND Sales.order_date = [Last Order Dates].MAX_order_date;
```

7. On the Query menu, click Run.

Creating Tables

Most of the time, the tables you need already exist in the database. Occasionally, it may be necessary to create new tables that users can query.

You may find it useful to create a table that provides metadata or a list of codes and their descriptions so that users can find the meaning of a code. You might also create a table that contains descriptions for all attributes in the data model.

You can create the table in three basic steps:

1. Create a data object with the appropriate attributes. Check the data model design and fix any problems before proceeding. For more information, see “Checking the Design” on page 117.

2. Create the table in the database using one of the following methods:

   - Create the table directly using BI Query. For more information, see “Creating Tables Directly in the Database” on page 115.
   - Create the table by sending an SQL script sent to the DBMS. For more information, see “Creating Tables Using SQL Scripts” on page 116.
If you create a script, you can modify it to include additional SQL commands that you want to be executed when the tables are created.

- Run a query that retrieves existing information in the database; then, create a table from the results. For more information on this procedure, see the BI Query Queries User's Guide.

3. Populate the table with the data you want it to store.

Once you have created the tables you want, make sure that users have the appropriate database permissions to use them.

For anything more than occasional use, you should use the administrator tools that come with your DBMS for table creation.

Creating Data Objects

To create new tables in the database, you first need to create data objects and their attributes in your Design window, and then create the corresponding tables.

To create a new data object in the data model:

1. Connect to the DBMS you are using. (This establishes the data types of the attributes you create.) For more information, see “Connecting to DBMSs” on page 56.

2. If BI Query is not already in Design mode, click the Design Mode button on the Tools menu.

3. Click the Data Object button on the Drawing toolbar. Then, click in the Design window where you want the object to appear.

4. Double-click the default name ("Unnamed").

5. In the Object Name dialog box, replace the default name by typing the name you want.

6. Do one of the following:

   - If you want BI Query to use the specified display name as the database name of the object, select Use As Database Name Of Data Object; then, click OK.

   - If you want to specify a database name for the object that is different from its display name, click OK to close the Object Name dialog box. On the Design menu, click Edit Database Name. The Edit Database Name dialog box opens. In the Database Name box, type a name for the database table that will correspond to the data object. Click OK.
To add an attribute to the data object:

1. Double-click the new data object. If the data object does not contain any attributes, the Insert Attribute dialog box opens. If the object does contain one or more attributes, open the Insert Attribute dialog box by clicking Insert Attribute on the Design menu.

2. In the Insert Attribute dialog box, specify a name, data type, and database name for the attribute.

3. Click Insert; then, click Close.

Your database may require that you initially give specific names to data objects and attributes. Make sure to check your DBMS documentation before renaming data objects and attributes.

Creating Tables Directly in the Database

You can create tables directly in the database from new data objects. Users need appropriate database permissions to use the tables.

To create tables directly in the database:

1. Connect to the database.

2. In a Design window, on the Host menu, click Create DB Tables.

3. If the List Tables dialog box opens, do the following:

   a. To display all tables, leave the Tables box blank. Alternatively, to specify a subset of tables from the DBMS catalog (or system tables), type their names in the Tables box. For example, to specify all tables with names that begin with the letter C, type C%.

   b. If your database requires it, type the creator (or collection or database) name into the Creator box.

   c. Click List.

4. In the Create DB Tables dialog box, select a data object in the Data Objects list.

5. Click the right arrow button to create the selected object in the database. (It appears in the Database Tables list.) When the table is created, click Close.
Reloading and Populating Tables

When you create a table using BI Query, the DBMS may adjust the data types of attributes to whatever is most appropriate for the DBMS. To ensure that BI Query has an accurate representation of the table, it is good practice to delete the corresponding data object from the data model, and then load the table to recreate the data object.

For more information, see “Loading Tables from the Database” on page 80.

Once the table is created, you can populate it using BI Query Admin, BI Query Update, or any other tool that lets you insert records into a table.

For more information, see “Applying Updates to the Database” on page 279.

Creating Tables Using SQL Scripts

You can create tables in the database from new data objects by sending an SQL script to the DBMS. The DBMS executes the commands and creates the tables. BI Query can automatically generate the script for you.

To create a table with a SQL script:

1. On the Host menu, click Create DB Script.

2. If an alert box informs you that there are errors in the data model design, click Cancel in the alert box, correct the errors, and then repeat step 1. Otherwise, click OK.

   The generated script contains a CREATE TABLE statement for each object in the data model.

3. In the Save Script File dialog box, specify the name and location of the script; then, click Save.

4. In a text editor, open the script. Delete all CREATE TABLE statements for existing tables in the database. Make the necessary DBMS-specific modifications to the remaining statements. Add any other SQL commands as necessary. Save the script.

   For more information on adding commands, see “Additional SQL for Creating Tables” on page 117”.

5. To send the script do the following:
Checking the Design

a. On the Query menu, point to Open, then click Query.
b. In the Open Query dialog box, choose BI Query Script Files from the Files of Type list.
c. Locate and select the script file you created, then click Open. The script file opens in the Freehand Query window.
d. In the Freehand Query window, click Run.

Additional SQL for Creating Tables

When you create an SQL script, you can modify the script to include additional SQL commands. You can also modify the script that BI Query generates for creating tables in the database.

Example

To create a Retailers data object with Retailer # and Name attributes, BI Query generates the following script:

```
CREATE TABLE Retailers (Retailer # char(12), Name char(8));
```

Using any text editor, you can modify this script to include additional SQL. You can specify a segment (Sybase), tablespace (Oracle), location (Ingres II), or database (DB2) that is different from the default by appending the appropriate SQL command. To include a tablespace called NNN, you can change the SQL to the following:

```
CREATE TABLE Retailers (Retailer # char(12), Name char(8)) IN TABLESPACE NNN;
```

Similarly, you can include a prefix for the table name. Some databases require this information to identify the owner or creator name (Oracle) or database name (Teradata). If you add mydatabase. as the prefix for the table, the SQL statement becomes the following:

```
CREATE TABLE mydatabase.Retailers (Retailer # char(12), Name char(8)) IN TABLESPACE NNN;
```

Checking the Design

Before distributing a data model, be sure to check the design for consistency and completeness. You can check the design once you have completed the data model or at any stage while you are creating it.
If there are no errors in the data model design or warnings associated with it, BI Query displays a message to that effect. If there are errors or associated warnings, BI Query displays them. Correct any problems; then, check the design again to make sure you have corrected all the errors.

**To check the design of a data model:**

1. On the **Tools** menu, click **Check Design**.

2. In the **Check Design** dialog box, click **Select All** to perform all the design checks, or select the checks you want to perform.

3. Click **OK**.

4. If there are no errors or warnings, the message *No design errors found* displays in the design check window. If there are errors, they are displayed.

5. Once you have corrected errors, check the data model design again and correct any new errors that may have been introduced.

6. Close the design check window.

**Securing Data Models**

Once you have created a data model, you may want to control what information users access and which activities they carry out. By securing the data model, you can ensure that information in the database is accessible only to the users who need to see it. You can customize the level of database activity and BI Query capability to reflect the skills and needs of various users.

💡 For more information, see “Types of User Access” on page 169.
Chapter 6

Creating Effective Designs for Data Models

This section provides information on the following:

- “Basic Design Principles” on page 119
- “Establishing a Layout” on page 123
- “Adding Emphasis” on page 131
- “Adding Ornaments and Other Objects” on page 134
- “Specifying Colors” on page 140
- “OLE Objects as Ornaments” on page 144

Basic Design Principles

Effective design results from a process that organizes information to meet the needs of users. The design of a data model is thus an extension of the organization process and a way of communicating information. Although there are no hard and fast rules in design, there are some general principles you should consider in your design work.
Chapter 6: Creating Effective Designs for Data Models

Considering Audience and Content

When you design a data model, you need to choose design elements and an arrangement that work best for your organization or a particular user group. That means designing a data model that’s suitable for the audience and content—setting the right tone and incorporating graphics that support or enhance the type of information they represent. A data model for a bank, for example, will likely have a different tone and appearance than one for a bicycle company.

Appropriateness also means designing in the right proportions, where the size of any design element is appropriate for the size of the Design window, the number of elements in it, and where you want to place emphasis. For example, when you include a corporate logo, it should be large enough to establish a corporate identity but not so large that it distracts users and takes the focus away from the data objects and buttons they need to access information.

Consistency in Design

Consistency enhances clarity, readability, and reliability because it reinforces users’ expectations. The more consistent you are in your treatment of design elements, the easier it is for users to find what they need and understand what to do. When you create more than one heading in a data model, for example, use the same typeface and size for each one. When you’re naming data objects, attributes, buttons, and so on, use the same case—whether it’s lower case, upper case, title case, or sentence case. Use even spacing between objects, and make objects of the same type (for example, buttons) the same size, especially if they’re grouped together.
Be consistent in how you handle elements both within a **Design** window and from one **Design** window to another. For example, if you decide to include a corporate logo in each **Design** window, make sure it’s the same size and in the same place in each one. Introduce a change selectively—to draw attention to an element, for example.

**Restraint**

While a wide range of typefaces, graphics, and colors are available to you, keep in mind that design elements gain impact when you use them selectively and lose impact when they’re overused. In a well-designed data model, the design is invisible—users can focus on the information they need and not on the design elements used.

Keep the design simple and uncluttered. Avoid putting too many data objects in a **Design** window and creating so many relationships between them that it’s impossible to understand what information relates to what. Consider dividing data objects among multiple **Design** windows.

A “busy” data model is difficult to use  
A “clean” design is more usable
Contrast

Contrast adds “color” to a data model. It gives elements extra impact and keeps users interested. You can create contrast by adding color selectively and varying the size and shape of objects.

Providing Unity

Users need a focal point in a Design window—something that determines how they scan the window and communicates the relative importance of elements in the window. You can achieve this by organizing each Design window around one dominant visual element—a heading, the data objects and relationships in a data model, or a group of buttons. Unity is also provided by consistency in font size, font style, and headings.
Establishing a Layout

The layout or organization of your data model provides a structure for data objects, buttons, and ornaments and gives users direction for digesting the information. Different tools can help you organize design elements so that users can find the information they need quickly and easily. If you provide more than one Design window, present a consistent layout in each one.

💡 The user’s natural tendency is to read from upper left to lower right. Organize design elements to accommodate this tendency.

Available BI Query Design Tools

When designing windows, you can use menu commands or Drawing toolbar buttons to create text, icons, and picture objects for use as data objects, buttons, or ornaments. You can increase the size of a Design window to accommodate the objects you want to include.

💡 Each Design window maintains its own history of operations that you can undo or redo.

You can align objects relative to a grid, which appears on the screen in the Design window by default but does not print. A Layout toolbar of buttons for aligning objects is also available.

If you have objects that you want displayed in multiple Design windows, you can create repeating objects. Creating repeating objects not only ensures consistency of location and style, it simplifies data-model maintenance by allowing you to edit all instances of a repeating object simultaneously, and it helps reduce the overall size of the data model.

Identifying Selected Objects in Design Mode

To select an individual object, simply click the object. To select multiple objects, press SHIFT and click the additional items. You can also easily toggle all selected objects to be unselected (and vice versa) by clicking Invert Selection from the Edit menu.
How an object appears when selected depends on the type of object you select.

<table>
<thead>
<tr>
<th>This object...</th>
<th>appears this way when selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data object</td>
<td>If the data object does not have an icon associated with it, the fill color turns black. If the data object has an icon associated with it, the outline color replaces the fill color around the perimeter.</td>
</tr>
<tr>
<td>Relationship</td>
<td>The fill color turns black.</td>
</tr>
<tr>
<td>Relationship Line</td>
<td>The line thickness increases and the selection color replaces the line color.</td>
</tr>
<tr>
<td>Button</td>
<td>Eight selection handles appear around the perimeter of the object.</td>
</tr>
<tr>
<td>Ornament</td>
<td>Eight selection handles appear around the perimeter of the object.</td>
</tr>
</tbody>
</table>

The selection handles that appear for buttons and ornaments are gray if the selected object is a repeating object, otherwise they are black.

**Setting the Design Window Size**

You can change the size of the Design window. You may have to use scroll bars to view the entire window.

**To set the Design window size:**

1. In BI Query Admin, in Design mode, on the Layout menu, click Page Size.
2. In the Page Size dialog box, click Defaults to view the minimum height and width pixels for the Design window. They vary with resolution.
3. Enter the number of pixels you want (the number must be larger than the minimum) in the Height and/or Width boxes; then, click OK.
Establishing a Layout

Changing Layout Grid Options

The layout grid in BI Query helps you organize and align objects in Design windows and thus create a consistent layout from one window to the next. The grid is a useful tool for placing data objects, relationships, connecting lines, buttons, and ornaments. BI Query displays the grid by default. It appears on the screen as a series of horizontal and vertical dotted lines. In Design mode, you can turn the visibility of the grid on and off using the Snap To Grid button on the Drawing toolbar.

To align objects on the grid, you can move them manually or use the tools on the Layout toolbar. You can change the spacing between grid points, and you can place objects anywhere on the grid or have BI Query automatically place them on the closest point.

💡 Grid settings are saved with the data model.

To change layout grid options:

2. In the Grid Settings dialog box, specify the settings:
   - Select Show Grid to display the grid on the screen.
   - Select Snap to Grid to place objects on the closest grid point automatically.
3. Specify the spacing between grid points in the Width and Height boxes.
4. Click OK.

Moving Objects Within a Design Window

In order to move objects within a Design window, you must first be in design mode. If you are not in design mode, on the Tools menu, click the Design Mode button.

You can move objects within a Design window by clicking and dragging them (or using the arrow keys to move them in small increments). (To select more than one object at a time, drag the pointer over them or hold down the SHIFT key and click each one.) With the Snap To Grid option selected, objects will automatically center themselves on the nearest grid point.

For more precise adjustments, or when aligning multiple objects at the same time, use the Layout toolbar.
If you are aligning a button or ornament that you want to use in multiple Design windows, consider making the object into a repeating object. That way you need only set its location in one Design window and all other instances of the object will automatically have the same position (and other Design properties) in the Design windows in which they occur.

Adding horizontal and vertical lines can also help you organize elements in a Design window. You create lines using the Line tool.

💡 Thick lines work only if they’re set off by a lot of white space.

You can not drag an object from one Design window to another. For information on moving objects between Design windows, see “Copying and Pasting Objects” on page 127.

### Aligning Objects

You can align objects along their top, bottom, right, left, or center points.

**To align objects:**

1. Click the first object.
2. Hold down SHIFT, and then click any additional objects.
3. On the Layout menu, click Align, and then click an alignment option.

Aligning an object that has a text component (data objects, relationships, text buttons, or text ornaments) does not affect the alignment of the text within the object. The text for data objects and relationships is always centered under the object.

**To align the text inside a button or ornament:**

1. Click the button or ornament.
2. On the Drawing menu, click Edit Button or Edit Ornament.
3. In the dialog box that opens, click Edit Text. The Text Style dialog box opens.
4. In the Text Style dialog box, under Align, change the justification of the text by clicking Left, Center, or Right.
5. Click OK.
6. Click OK.
Making Objects the Same Size

You can resize buttons and ornaments according to the largest selected object. You can not resize data objects or relationships.

To make the objects the same width:
1. Select the objects you want to resize.
2. On the Layout menu, click Make Same Size, and then click Width.

To make the objects the same height:
1. Select the objects you want to resize.
2. On the Layout menu, click Make Same Size, and then click Height.

Spacing Objects Evenly

The Space Evenly Horizontally option spaces objects evenly across the page based on the location of the selected objects to the furthest left and the furthest right of the page. All other selected objects are spaced evenly between these two objects.

The Space Evenly Vertically option spaces objects evenly down the page based on the location of the selected objects closest to the top and bottom of the page. All other selected objects are spaced evenly between these two objects.

To space objects evenly across the page:
1. Select the objects you want to reposition.
2. On the Layout menu, click Space Evenly, and then click Horizontally.

To space objects evenly down the page:
1. Select the objects you want to reposition.
2. On the Layout menu, click Space Evenly, and then click Vertically.

Copying and Pasting Objects

You can copy objects (if they are not embedded or linked) within a Design window or from one Design window to another. When you paste an object, you create a new object that is an identical yet independent copy of the original object.
There are a few cases in which you might not want an exact or independent copy. For example, if you are copying a data object (with or without the accompanying relationships), you might want to copy just the physical properties of the data object but exclude other properties (such as attribute information, database names, or join conditions). You can do this using the Paste Special command.

If you are pasting a button or ornament from one Design window to another, and you do not want the copy to be independent from the original (that is, you want to keep their properties in sync through any subsequent edits made to either instance), use the Paste Repeating command.

You also use the Paste Special command when creating ornaments from OLE objects.

**To copy and paste an object:**

1. If you want to copy and paste an object in a Design window, on the Tools menu, click the Design Mode button.
2. Click the object you want to copy. (To select more than one object at a time, drag the pointer over them or hold down the SHIFT key and click each one.)
3. On the Edit menu, click Copy.
4. In the appropriate window, click the area where you want the object to be copied.
5. On the Edit menu, click one of the following:
   - To create an exact but independent copy, click Paste.
   - To create an exact copy of an ornament or button as a repeat instance of the original, click Paste Repeating.
   - To create a copy of a data object while excluding certain data associations, click Paste Special. The Paste Special dialog box appears.
6. In the Paste Special dialog box, click OK.
7. In the Paste Options dialog box, select the properties of the object that you want included in the copy; then, click OK.

**Deleting Objects**

BI Query must be in Design mode before you can delete an object. If you delete something by mistake, you can undo the deletion by clicking Undo on the Edit menu. BI Query does not require confirmation of a deletion.
To delete an object from a Design window:

1. Click the ornament, button, or data object you want to delete. To select more than one, drag the pointer over them or hold down the SHIFT key and click each one.

2. On the Edit menu, click Delete.

If the object you want to delete was created by submitting a query to the database as a table, or by creating a dynamic relationship, you can delete it using the **Delete User Objects** dialog box. (To open this dialog box, click **Delete User Object** on the Tools menu.)

**Repeating Objects**

Any time you have a button or ornament that you want to place in multiple Design windows, consider making the button or ornament into a repeating object. A repeating object is essentially a single object (button or ornament) that exists in multiple Design windows simultaneously. When you specify the properties of the object (such as name, color, size and position) in any Design window, your changes are automatically applied to all other instances of the repeating object. Creating repeating objects not only ensures consistency of location, style, and behavior, it simplifies data-model maintenance by allowing you to edit all instances of a repeating object simultaneously, and it helps reduce the overall size of the data model.

You can create a repeating object either when pasting an object from the Clipboard or when editing the object. To make it easier to locate and identify repeating objects, they appear with gray selection handles (instead of black) when selected. If your BI environment includes BI Server, you can set security for objects in the data model. Unlike the design properties of repeating objects (which must be identical for each instance of the object), you can apply different security settings for each instance of a repeating object if you want. In the **Set Security** dialog box, you can identify a particular instance of a repeating object by the unique combination of Design window and object name.

💡 If you want multiple instances of a repeating object to share the same security settings, apply the security settings to the original instance of the object before repeating it in the other Design windows.
To create a repeating object by pasting:

1. In Design mode, open the Design window in which the object you want to copy exists.
2. Select the button or ornament you want and copy it to the Clipboard.
3. Switch to the Design window into which you want to paste the object.
4. On the Edit menu, click Paste Repeating. The pasted object appears in the same location as the original object.

To create a repeating object by editing properties:

1. In Design mode, open the Design window in which the object you want to copy exists.
2. Select the button or ornament you want.
3. Do one of the following:
   - If you have selected a button, on the Drawing menu, click Edit Button. The Edit Button dialog box appears.
   - If you have selected an ornament, on the Drawing menu, click Edit Ornament. Either the Edit Ornament or the Edit Shape Ornament dialog box appears.
4. From the Locations list, select all the Design windows in which you want the object to appear (including the current one unless you intend to move the object from the current window).
   When the dialog box opens, all Design windows in which the object already exists are automatically selected.
5. Click OK.

Kinking Relationship Lines

When designing a data model, you can kink lines connecting data objects and relationships in order to avoid having lines passing through unrelated objects in a Design window containing many objects.

To kink a relationship line:

1. In Design mode, place the pointer on the relationship line where you want the kink; then, drag the line to the position you want.
2. Click outside the relationship line to deselect it.
3. Repeat as necessary.
Adding Emphasis

In order to make a data model easy to use, distinguish important elements from less important ones. For example, buttons shouldn’t be overpowered by supporting graphics. The following sections describe the design elements that let you strengthen a data model by adding emphasis where you need it.

Typeface

Typeface refers to size, shape, and spacing of letters. You can specify typeface for text buttons and ornaments. The type available to you depends on what fonts, or typefaces, you have installed on your computer. The default is Tahoma.

Avoid typefaces that draw attention to themselves. A typeface shouldn’t distract the user from the information you’re trying to communicate.

Type is divided into two categories. Serif type has “feet” attached to the ends of letters that guide the eye from letter to letter. This type is often used in body text in print to assist readability. Sans serif type has no “feet”—it has a clean, crisp, straightforward look and is often used in print for short phrases such as headings and subheadings to distinguish it from body text. Because most users find it more legible on screen than serif type, it is normally used for body text for material meant to be read from a computer screen. Arial is an example of a sans serif typeface.

Times Roman (serif) and Helvetica (sans serif) are common typefaces. You might use one typeface for labels of buttons, ornaments, or data objects, and a second one if you’re including a block of text. The key is to use only one or two typefaces and to use them consistently. Typeface influences tone (formal or informal) and adds expressiveness.

Avoid italics or script typefaces—they can be very difficult to read on screen. The same goes for reverse type (white type on a dark background). Use it only at large font sizes.

To unkink a relationship line you have just kinked:

- On the Edit menu, click Undo.

To unkink a relationship line kinked in a previous session:

- Place the pointer on the kink of the relationship line and drag until the line is straight.
Style and weight (normal, bold, italics, bold italics) influence the appearance and tone of type by adding contrast and emphasis. Size of type is also a factor. You can use large type to emphasize important information and smaller type for less important information. But don’t use large type in a small area where it’s hard to read and looks cramped. Similarly, don’t use small type in a lot of white space, or it will appear lost.

Upper case is used for emphasis in headings. Avoid using words completely in upper case except for major headings.

💡 To change all typefaces for labels in Design windows, on the Preferences menu, click Window Fonts. In the Window Fonts dialog box, select the font you want from the Design drop-down list.

Not all typefaces reproduce well on screen. You need to experiment with the typefaces installed on your computer. When you’re creating data models for distribution to other users, or when you’re designing a data model for BI Web users, you can avoid having to make adjustments from one computer to the other by using standard fonts such as Helvetica, Arial, and Times Roman.

**White Space**

White space is any blank space, whatever the background color. It allows more background to appear, making whatever it surrounds stand out. White space focuses attention and enhances readability. It opens up, or “lightens,” a Design window and avoids visual monotony. Without white space, you end up with a “run-on” effect that makes it difficult to distinguish elements in a Design window. If spacing is uneven, it can jar the eye and convey poor organization and a careless design. (However, you may occasionally want to introduce uneven spacing to create contrast.)

💡 Avoid white space in the middle of a Design window. It creates a “hole,” which gives an unfinished look.
You can create white space by limiting the number of elements you include in a Design window and by resizing and aligning them.

**Boxes**

Boxes highlight elements that are otherwise lost in their surroundings. They also organize elements. For example, you can create a box around a set of buttons to group them together and set them apart from the rest of the model. You can also add gray shading or a color to a box for more impact or to create a particular effect. Use BI Query’s **Drawing** toolbar to create boxes.

**Graphics**

Graphics can add variety to a data model and convey information. You can apply graphics to data objects and buttons to give an idea of what they do. For example, a button that displays a graphic of a report communicates that the button generates a report. You can also include graphics as a background in a Design window to establish a connection with the information that’s available. For example, a Design window can display a map of a country, and you can place buttons that run queries on specific regions in the country on the corresponding region on the map.

**Color**

Color draws attention to design elements and helps you organize and emphasize information. For example, applying one color to a group of buttons or data objects allows users to perceive them as a group. Color also evokes an emotional response and sets a tone—bright colors for excitement, subdued colors for dignity.
You can apply color to windows and objects in BI Query. For the best results, decide what you want to achieve; then, choose a main color (for the background of a Design window, for example) and complementary colors (for individual objects). Complementary colors are color combinations such as red-orange or blue-green that visually enhance one another. Apply a contrasting color to achieve a focal point in your data model or to accent smaller areas of emphasis.

Colors of text, Design windows, and objects can be specified on an individual basis. Colors of other window types can be specified on a general basis. (All windows of that type use the same colors.)

Adding Ornaments and Other Objects

Ornaments are text and graphic objects such as titles, logos, borders, and notes that provide additional information, act as visual organizers, or simply enhance the appearance of Design windows. There are several types of ornaments you can create.

Text Ornaments
Display the text you specify when you create the ornament.

Picture Ornaments
Display the image stored in a picture file. BI Query saves the image as part of the data model. You can use pictures stored in .bmp, .gif, .emf, or .wmf formats.

💡 Keep in mind that graphics can add substantially to the size of a data model.

Icon Ornaments
Display an icon. Icons are graphics with a specific dimension (32 pixels by 32 pixels). Icons are stored in .ico, .dll and .exe files. If you are using a monochrome icon, you can assign a color to it. BI Query saves the icon as part of the data model.

Shape Ornaments
Display shapes (lines, rectangles, or ovals) that you create using the tools on the Drawing toolbar.

You can also create editable ornaments using Windows’ Object Linking and Embedding (OLE).
Creating Ornaments

By default, BI Query autosizes any ornament you create to fit the text, picture, or icon it displays.

**To create an ornament that displays a text string, an icon, or a picture:**

1. In Design mode, on the **Layout** menu, click **Create Ornament**. The **Create Ornament** dialog box opens.

2. If you want to control access to the ornament using BI Server, type a name for the ornament in the **Name** box. When you publish the data model, you can set access permissions for its named components.

   Pasting text in a **Design** window in Design mode automatically creates an ornament with that text.

3. If you want to add the same ornament to multiple **Design** windows, select from the **Locations** list all of the **Design** windows in which you want the ornament to appear.

   When you open the dialog box, all **Design** windows in which the ornament appears are selected in the **Locations** list. Selecting a new entry adds the ornament to the associated **Design** window. Deselecting an entry removes the ornament from the associated **Design** window.

4. To create an ornament that displays text, select **Text**; then, click **Edit Text**. The **Text Style** dialog box opens. Use this dialog box to type the text and specify the formatting options you want. You can type the special text strings \&d and \&t to display the current date and time. For more information, see “**Special Text Strings in Ornaments and Buttons**” on page 136”.

5. To create an ornament that displays an icon, select **Icon**; then, do one of the following:

   • To use an icon on your system, click **Select Icon**. In the **Select Icon** dialog box, select the icon. If you want the icon itself to have a white background, select the **White Fill** check box. To proceed, click **OK**. (If the icon you want is not displayed, click **Browse**; then, use the **Open Icon File** dialog box to locate and open the .ico, .dll, or .exe file that contains the icon.)

   • To copy an icon from the Clipboard, click **From Clipboard**.

6. To create an ornament that displays a picture, select **Picture**; then, do one of the following:
• To use a picture file on your system, click Select Picture; then, use the Open Picture dialog box to select the file.
• To copy a picture from the Clipboard, click From Clipboard.

7. To add a border, click Border.
8. Click OK.

Often, once you create an ornament, you will want to place it under a data object or ornament. You can do this by selecting the ornament; then, clicking Send to Back on the Layout menu.

Special Text Strings in Ornaments and Buttons

BI Query provides special strings that you can use in text ornaments and buttons to automatically display the date and time. These and other strings are also available in text ornaments in standard reports.

💡 For more information on Standard Reports, see the BI Query Queries User’s Guide.

The following table describes the strings:

<table>
<thead>
<tr>
<th>String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;t</td>
<td>Displays the current time.</td>
</tr>
<tr>
<td>&amp;d</td>
<td>Displays the current date.</td>
</tr>
</tbody>
</table>
| &d/n   | Displays the current date using predefined format $n$, where $n$ is a number greater than or equal to zero (0). $n$ corresponds to a predefined format in the Date Component list in the Format Preferences dialog box. For example, &d0 displays the current date using the first predefined date format, &d1 displays the date using the second predefined format, and so on. &d is equivalent to &d0.

$n$ corresponds to a named predefined format only. The <Regional Settings>, <Custom>, and <Blank> formats do not apply in this case. |
You can enter these strings as part of the regular text within a text ornament or button. For example, a button with text “&t” displays the current time. The strings must be lowercase.

To create a text ornament or button that contains a literal version of one of the above strings, precede the string with an ampersand. For example, to make the phrase “current&timely” appear literally, type current&&timely (“current&timely” contains the reserved string &t, which automatically converts into the current time unless you precede it with an extra ampersand).

Copying Ornaments from the Clipboard

In addition to creating ornaments from files, you can create them from data stored in the Clipboard. In this way, you can use text and graphics created in other applications. The resulting ornament behaves in the same way as those you create in BI Query.

Pictures copied from the Clipboard are either Windows metafiles (the most common format for vector graphics used by applications like CorelDRAW), Windows bitmaps, or Windows device independent bitmaps.

Copying data from the Clipboard lets you use graphics in formats that BI Query does not support. However, this process may result in a loss of image quality, depending on how the source application converts graphics to metafile or bitmap format before saving them to the Clipboard.

Creating Shape Ornaments

Use the buttons on the Drawing toolbar to create and format lines, rectangles or ovals.

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>The <strong>Line</strong> button lets you draw lines.</td>
</tr>
<tr>
<td>Rectangle</td>
<td>The <strong>Rectangle</strong> button lets you draw rectangles.</td>
</tr>
<tr>
<td>Oval</td>
<td>The <strong>Oval</strong> button lets you draw ovals.</td>
</tr>
<tr>
<td>Line Width</td>
<td>The <strong>Line Width</strong> button lets you specify line thickness.</td>
</tr>
</tbody>
</table>
To create a shape ornament:

1. In Design mode, from the Drawing toolbar, click the appropriate button for the line, rectangle or oval you want to create.
2. Click and drag the mouse diagonally to draw the figure.

### Editing Shape Ornaments

There are several options available for editing shape ornaments. You can edit most of the properties of a shape ornament (such as name, location, line thickness and color) from the Edit Shape Ornament dialog box. If you only need to edit the colors, or if you are setting colors for multiple objects simultaneously, you might use the Set Object Color dialog box. You can also use the tools available from the Drawing toolbar.

If you want to set the colors for all shape ornaments in the data model, or in a particular Design window, you can also use the Design Colors dialog box. For more information, see “Specifying Object and Design Window Colors” on page 141.

### To set the thickness of a line or the outline of a rectangle or oval:

1. Select the shape ornament you want to edit.
2. Do one of the following:
   a. From the Drawing toolbar, click the Line Width button, then click the line width you want.
   b. On the Drawing menu, click Edit Ornament. In the Edit Shape Ornament dialog box, choose the line width you want from the Line Width palette.

### To set the color of a line or the outline and fill colors of a rectangle or oval:

1. Select the shape ornament you want to edit.
2. Do one of the following:
a. From the Drawing toolbar, click the Line Color button or Fill Color button, then click the color you want from the color palette.

b. On the Layout menu, click Set Object Color. In the Set Object Color dialog box, click the button for the shape property you want to edit, then click the color you want from the color palette.

c. On the Drawing menu, click Edit Ornament. In the Edit Shape Ornament dialog box, click the button for the shape property you want to edit, then click the color you want from the color palette.

To set the name of the shape ornament:
1. Select the shape ornament you want to edit.
2. On the Drawing menu, click Edit Ornament. In the Edit Shape Ornament dialog box, type the new name for the shape ornament in the Name box.

To change to a different shape:
1. Select the shape ornament you want to edit.
2. On the Drawing menu, click Edit Ornament. In the Edit Shape Ornament dialog box, click the new shape from the Shape list.

To change in which Design window a shape appears:
1. Select the shape ornament you want to edit.
2. On the Drawing menu, click Edit Ornament. In the Edit Shape Ornament dialog box, under Locations, select the Design windows in which you want the shape ornament to appear.

Assigning Icons to Data Objects

By default, when a database table is loaded into a Design window, a rectangular data object is displayed. You can choose instead to display the data object as an icon. Icons provide visual cues to the information they represent in the database and enhance the appearance of a data model.

Icons are graphics with a specific dimension (32 pixels by 32 pixels). An icon file usually has the extension .ico. If you are using a monochrome icon, you can assign a color to it. You can also apply colors to data objects.
To assign an icon to a data object:

1. In Design mode, select the data object.
2. On the Layout menu, click Select Icon. The Select Icon dialog box opens.
3. If the icon you want to use is not displayed in the Select Icon dialog box, click Browse; then, use the Open Icon File dialog box to locate and select an icon file (or a file with the extension .dll or .exe).
4. In the Select Icon dialog box, click the icon you want. (You can also replace an icon with the default rectangle by clicking Use Default.)
5. Select the White Fill check box if you want the icon to have its own white fill. For data objects, the white fill area is slightly smaller than the icon, so you will still see a strip of the data object’s fill color between the white fill of the icon and the outline of the data object.
6. Click OK.

Actual Relationships cannot display icons.

Specifying Colors

You can specify the color of windows and objects in a data model to make them more attractive and to provide visual organization. Using color can direct users’ attention to important elements of the data model.

You can specify colors for the following window and object components:

- all windows in the data model, except the Super Query or Freehand Query windows
- data objects and relationships
- buttons and ornaments

Specifying Window Colors

You can specify colors for most windows using the Window Colors dialog box. For information on setting the color for the Design window, see “Specifying Object and Design Window Colors” on page 141.

To specify colors for windows:

1. On the Tools menu, click Preferences.
2. In the Preferences dialog box, click Window Colors.
3. In the **Window Colors** dialog box, specify the colors you want.
4. Click **OK**.
5. In the **Preferences** dialog box, click **OK**.

### Specifying Object and Design Window Colors

Use the **Design Colors** dialog box to specify default colors for new objects and **Design** windows. When setting new default colors for the data model, you can also update the colors for existing design items, either throughout the entire data model or just in the current **Design** window. You can apply these changes for all design colors or just for the design colors you changed since opening the dialog box (leaving the other objects and properties alone).

**To specify default colors for objects and Design windows:**

1. On the **Tools** menu, click **Preferences**.
2. In the **Preferences** dialog box, click **Design Colors**. The **Design Colors** dialog box opens.
3. In the **Design Colors** dialog box, specify the colors you want.
4. To apply the colors to existing objects in the active **Design** window, under **Apply To**, click **Current design window only**.
5. To apply the colors to existing objects in all **Design** windows, under **Apply To**, click **All design windows**.

   ![](image)

   If you only want to set new default colors without changing any existing objects or windows, clear both check boxes.

6. Under **Use**, do one of the following:
   - Click **All design colors** if you want to apply all of the colors displayed in the dialog box.
   - Click **Changed onles only** if you only want to apply the colors you have changed since opening the dialog box.

7. Click **OK**.
8. In the **Preferences** dialog box, click **OK**.
Specifying Colors for Selected Objects

You can use the **Design Colors** dialog box to set default colors, and to update the colors for all existing objects of a given type within the current window or the entire data model. For more specific changes, use the **Set Object Color** dialog box to apply color changes to selected objects within a **Design** window.

💡 If you select multiple objects, all of the color properties of the selected objects will be updated with the settings you specify in the **Set Object Color** dialog box. Therefore, if you only want to set the color for a specific property (such as the **Fill** color) without changing the color for another property (such as the **Outline** color), use the specific tools available on the **Drawing** toolbar instead.

### To specify colors for selected objects:

1. In Design mode, click an object or, to select more than one object, SHIFT+click on the objects after the first one.
2. On the **Layout** menu, click **Set Object Color**.

   💡 To access the **Set Object Color** dialog box for relationship lines, you must right-click the relationship line whose colors you want to change.

3. In the **Set Object Color** dialog box, specify the colors you want.

   If you select multiple objects, the dialog box displays all of the possible color properties for any design item, regardless of whether the property is applicable to the specific objects you have selected. If the property exists within the selected objects, it will be updated; otherwise the property will be ignored.

4. Click **OK**.

   To see the color, click outside the object.

Specifying Colors for Individual Design Windows

You can use the **Design Colors** dialog box to set a default color for all new **Design** windows, and to update the colors for existing **Design** windows (for the current window or even the entire data model). If you only want to change the color of the current **Design** window without changing the default color for all new **Design** windows, you can use the **Set Window Color** dialog box.
To specify the color for an individual Design window:

1. In Design mode, open the Design window for which you want to change the color.
2. On the Layout menu, click Set Window Color.
3. In the Set Window Color dialog box, specify the color you want.
4. Click OK.

Creating New Colors

The palette you use to assign colors to windows and objects displays a basic set of colors. If you do not want to use these colors, you can create your own. You can experiment with a broad range of additional colors before choosing the one you want. BI Query saves a color only if you apply it to an object. If you create one or more colors that you would like to use in the future, make a note of the values in the Color dialog box so that you can readily re-create the color.

To specify a new color:

1. In the color palette, which appears when you click a color button to assign a color, click Custom Colors.
2. In the Color dialog box, click inside the large color box and use the brightness bar on the right side of the dialog box until the color that appears in the Color|Solid box is the shade you want (or type values into the available boxes).
3. To add the color to the Custom Colors palette, click Add to Custom Colors; then, click OK.

Transparent Objects

Some objects have properties that can display either a solid color, or no color. When you specify that a property has no color, it becomes transparent. Within the region that would otherwise have displayed a solid color you will instead see through to whatever colors are immediately behind the object. Having a transparent region within an object can be useful when:

- the area behind the object has detail that you do not want obscured (such as a graduated fill, or a watermark)
- you do not need or want to draw attention to the property
You can apply a transparent color to the following object properties:

<table>
<thead>
<tr>
<th>Object</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Object</td>
<td>Outline</td>
</tr>
<tr>
<td>Data Object</td>
<td>Fill</td>
</tr>
<tr>
<td>Data Object</td>
<td>Text Fill</td>
</tr>
<tr>
<td>Ornament</td>
<td>Fill</td>
</tr>
<tr>
<td>Relationship</td>
<td>Text Fill</td>
</tr>
<tr>
<td>Button</td>
<td>Fill</td>
</tr>
</tbody>
</table>

To specify an object has no color:
- In the color palette associated with one of the object properties listed above, click **None**.

## OLE Objects as Ornaments

Object Linking and Embedding (OLE) is Microsoft's object-based technology for sharing information and services across process and machine boundaries. OLE is a compound document protocol that allows one document to be embedded within or linked to another.

BI Query is an OLE container, which means that you can import OLE objects from OLE server applications into a data model without leaving BI Query. If those applications support in-place editing, you can also edit the objects from within BI Query.

Objects you can link or embed from other applications include text, picture, spreadsheet, sound, and video objects. OLE objects appear in BI Query as ornaments, which you can move and resize. You can display OLE objects as icons, rather than displaying their contents. OLE does not support multiple-page objects. If you import text into BI Query, import only one page at a time.

💡 For more information on OLE, see the user documentation for any Microsoft application, such as Word. To find out what OLE features an application supports (such as in-place editing and drag-and-drop linking), see its user documentation.
OLE Requirements and Limitations

Before you can import OLE objects into a Design window, the following must be true:

- Both BI Query and the OLE server application must be installed on your computer.
- Both BI Query and the OLE server application must be running.
- Your computer must have enough memory to accommodate the size of the object.
- The source document must have been saved.

- BI Query is not an OLE server: it doesn't create OLE objects that you can link or embed in other applications.
- You cannot create OLE ornaments in Standard Reports.
- Linked objects are not supported in BI Web.

Types of OLE Objects

OLE uses both linked and embedded objects to transfer and share information among applications.

Linked Objects

Reference a file (document, text, graphic, video, or sound) stored outside the container document. When you create a linked object in a data model, BI Query stores the link rather than the actual document. You can edit the document in the original application and the changes appear in the object in BI Query, either immediately or after you have exited and reopened BI Query, depending upon the application. You can also edit the object directly in BI Query. Some applications allow you to link portions of documents as well as whole documents. You can display linked objects as icons.
Embedded Objects

Are inserted into the container document. The object becomes part of the data model; you can use it as you would use any other ornament in BI Query. When you double-click an embedded object, the source application opens the object for editing within BI Query. Unlike linked objects, any changes you make to an embedded object in BI Query are not made to the original object. Embedded objects keep the data model self-contained so it is easy to copy and distribute. However, they also increase the file size of the data model.

Linking Objects

Use the **Insert Object** dialog box to link an object to a **Design** window. You can also use drag-and-drop linking if the OLE server application supports it. If you are using BI Query Admin, you can also copy the object to the Clipboard, and then paste it in the **Design** window as a linked object.

💡 For more information on linking objects, see the Help for your BI Query application.

To link an object:

1. In BI Query, position the insertion point where you want the object to appear; then, on the **Edit** menu, click **Insert Object**. The **Insert Object** dialog box opens.
2. In the **Insert Object** dialog box, select **Create from File**. Type the path and the name of the source file in the **File** box or click **Browse** to locate and select the file.
3. If you want to display the link as an icon, select **Display as Icon**.
4. Select **Link**; then, click **OK**.

To use drag-and-drop linking:

1. In BI Query, position the insertion point where you want the object to appear.
2. In the source application, select the object (or portion thereof) that you want to link.
3. While holding down the CTRL and SHIFT keys, drag the selection from the source application to the **Design** window in BI Query.

💡 The method you use to drag a selected object depends on the source application. Before you can drag an object from Microsoft Excel, you must move the mouse over an edge of the selection until the pointer becomes an arrow with a plus sign.
Embedding Objects

Use the **Insert Object** dialog box to embed an object in a **Design** window. You can also use drag-and-drop embedding if the OLE server application supports it. If you are using BI Query Admin, you can also copy the object to the Clipboard, and then paste it in the **Design** window as an embedded object.

💡 For more information on embedding objects, see the Help for your BI Query application.

**To embed an object in a Design window:**

1. In Design mode, position the insertion point where you want the new object to appear.
2. On the **Edit** menu, click **Insert Object**.
3. In the **Insert Object** dialog box, do one of the following:
   - To insert an existing object, click **Create from File**. In the **File** box, type the path and name of the source file. Alternatively, click **Browse** to locate and select the file.
   - To create a new object and insert it, click **Create New**; then, select an object type from the list.
4. If you want to display the object as an icon, select **Display as Icon**.
5. Click **OK**.
6. If you created a new object, edit in the **Design** window; then, click outside the object.

**To use drag-and-drop embedding:**

1. In the source application, select the object (or portion thereof) that you want to embed.
2. Drag the selection into the **Design** window.

Editing Imported Objects

You can use the **Object** command on the **Edit** menu to edit a selected OLE object. The name of the command changes depending on the type of object you have selected. For example, if you have selected a Microsoft Excel object, the menu command is **Worksheet Object**.
For information about viewing and managing linked objects, see the Help for your BI Query application.

You can either edit an OLE object within BI Query or within the original source application. If you choose the former, BI Query incorporates the menus and toolbars from the source application into its own interface; however, the File and Window menus belong exclusively to BI Query.

If you are editing a linked object, any changes you make to the source apply to the object in BI Query.

To edit a linked object:
1. In Design mode, select the object you want to edit.
2. Do one of the following:
   • To edit the object in BI Query, point to the appropriate Object command on the Edit menu (for example, Worksheet Object); then, click Edit.
   You can also double-click the object to edit it within BI Query.
   • To edit the object in the source application, point to the appropriate Object command on the Edit menu; then, click Open.
3. Edit the object; then, do one of the following:
   • If you are editing an embedded object, click outside the object to exit the application. You do not need to save your changes.
   • If you are editing a linked object, save it. If you are editing the object within BI Query, click outside the object to exit the application; otherwise, close the source application manually.

Viewing and Managing Links

Use the Links dialog box to view information about linked objects in the data model (such as the names of their source files) and to modify or update the links.

To view and manage links:
1. On the Edit menu, click Links. In the Links dialog box, click the link you want to change.
2. Make the desired changes:
a. Click **Update Now** to update it.
b. Select **Manual** if the link is to be updated only on request, or **Automatic** if it is to be updated automatically whenever a change is made to the source document. (Links are updated automatically by default.)
c. Click **Open Source** to open the source document of the object.
d. Click **Change Source** to change a link.
e. Click **Break Link** to remove the connection with the source document (which causes the linked object to become an embedded object).
Chapter 7

Adding Buttons to Data Models

This section provides information on the following:

- “Buttons in BI Query” on page 151
- “Specifying Button Appearance” on page 153
- “Linking Buttons” on page 155

Buttons in BI Query

A button in BI Query links to a particular resource and performs a particular operation based on that resource. The following table describes the resources you can link to a button and the corresponding operations that the button performs when clicked:

<table>
<thead>
<tr>
<th>Linked Resource</th>
<th>Button Operation When Clicked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stored or current query</td>
<td>Runs the query.</td>
</tr>
<tr>
<td><strong>Design</strong> window</td>
<td>Displays the window.</td>
</tr>
<tr>
<td>Application script created using OLE Automation</td>
<td>Opens the application and runs the script.</td>
</tr>
</tbody>
</table>
Chapter 7: Adding Buttons to Data Models

Creating Buttons

If you have the Edit data model and Save data model permissions (or if you are a BI Query administrator), you can create buttons in Design windows and save them with the data model.

To create a button:

1. Create the query (or other automated process) that you want the button to execute or the window that you want it to open.
2. If BI Query is not in Design mode, click the Design Mode button on the Standard toolbar.
3. On the Drawing toolbar, click the Button button ; then, click and drag the mouse over the area where you want the button to appear.
4. Double-click the button that appears. The Edit Button dialog box opens.

You can also create a button by clicking Create Button on the Drawing menu. In this case, the Create Button dialog box opens, which is identical in all respects save name to the Edit Button dialog box. You can open the Edit Button dialog box at any time by double-clicking a button in Design mode.

5. In the dialog box, type a name for the button in the Name box. (This name is the internal name for the button, not the label that will appear on the button in the user interface.)

<table>
<thead>
<tr>
<th>Linked Resource</th>
<th>Button Operation When Clicked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Runs the application.</td>
</tr>
<tr>
<td>Document</td>
<td>Opens the document in the appropriate application.</td>
</tr>
<tr>
<td>Report created in BI Query Reports</td>
<td>Opens the report within BI Query Reports.</td>
</tr>
<tr>
<td>Hypercube created in BI Cube Creator</td>
<td>Opens the hypercube within BI Cube Creator.</td>
</tr>
<tr>
<td>Web resource (URL)</td>
<td>Opens the specified URL using the default web browser on your system.</td>
</tr>
</tbody>
</table>
6. If you want to add the same button to multiple Design windows, select from the Locations list all of the Design windows in which you want the button to appear.

When you open the dialog box, all Design windows in which the button appears are selected in the Locations list. Selecting a new entry adds the button to the associated Design window. Deselecting an entry removes the button from the associated Design window.

7. Specify the appearance for the button. For more information, see “Specifying Button Appearance” on page 153.

8. Select Border if you want to apply a border to the button.

9. Select Autosize to have BI Query automatically size the button for you based on the dimensions of the text, icon, or picture that you have selected.

You can resize the button by clicking and dragging on any of the sizing handles that appear when the button is selected in Design mode.

10. If you want to provide a tip for the button, type its text into the Screen Tip box. This text opens in a pop-up box whenever the user moves the mouse pointer over the button. Typically, a screen tip describes the function of the button. For example, if the button acts as a hyperlink, you could provide the target URL as the screen tip.

11. Specify the link for the button. You can link a button to one of the following resources:
   • a query
   • a Design window
   • an application, document, or application script
   • a report or hypercube
   • a URL

   If the button links to a query, specify the output for the results as well.

12. Click OK.

Specifying Button Appearance

Use the Edit Button or Create Button dialog box to set the appearance for a button. A button can display a text string, an icon, or a picture.
To display text on a button:

1. In the Appearance area in the Edit Button or Create Button dialog box, select Text.
2. Do one of the following:
   - Click Edit Text.
   - Click From Clipboard to load text you have stored on the Clipboard; then, click Edit Text.
3. In the Text Style dialog box, type the button text (or modify the existing text). You can type the special text strings &d and &t to display the current date and time. For more information, see “Special Text Strings in Ornaments and Buttons” on page 136.
4. Assign font, size, style, color, alignment, and effects. Click OK.
5. In the Edit Button or Create Button dialog box, click OK.

To display an icon on a button:

1. In the Appearance area in the Edit Button or Create Button dialog box, select Icon.
2. Do one of the following:
   - Click Select Icon. In the Select Icon dialog box, select the icon. If you want the icon itself to have a white background, select the White Fill check box. To proceed, click OK. (If the icon you want is not displayed, click Browse; then, use the Open Icon File dialog box to locate and open the .ico, .dll, or .exe file that contains the icon.)
   - Click From Clipboard to load an icon you have stored on the Clipboard.
3. Click OK.

To display a picture on a button:

1. In the Appearance area in the Edit Button or Create Button dialog box, select Picture.
2. Do one of the following:
   - Click Select Picture. In the Open Picture dialog box, double-click the .bmp, .gif, or .wmf file you want.
   - Click From Clipboard to load a graphic you have stored on the Clipboard.
3. Click OK.
Linking Buttons

You can specify the link for a button (the operation it performs when clicked). A button can link to a query, a window, an application, a document, a script, a report, a hypercube, or a web resource.

For buttons linked to queries, you must specify the target for the results.

Link Targets for Query Buttons

When you link a button to a query, clicking the button runs the query and sends the results to a specified target. You can specify one of the following targets for a given button:

**Results Window**
Sends the query results to a results window. This is the default.

When you click a button linked to a query that sends results to a results window, BI Query runs the query then creates a standard report using the associated report specification (if one exists for the query).

**Application**
Sends the query results to an application such as Microsoft Word or Excel. Depending on the export mechanism you choose, BI Query calls menu commands in the receiving application or sends Microsoft Dynamic Data Exchange (DDE) commands to the application. Clicking the button copies the results to the Clipboard, checks whether the receiving application is already running, and, if it is not, starts it up.

DDE lets you pre-program commands in the target application’s programming language (such as Excel’s macro facility) and further manipulate the results.

**Results File**
Sends the query results to a text file. You can use this option to share results with someone who does not use BI Query.
Database Table
Sends the query results to a new table in the database and creates the corresponding data object in the active Design window. You can then use the new table to formulate future queries. If the Dynamic relationships permission is assigned and you have set the Allow dynamic relationships preference, you can create a dynamic relationship to link the new data object to the other data objects and relationships in the window. You must also have the appropriate set of permissions in order to create tables in the database. For more information, see the BI Query Queries User’s Guide.

Linking Buttons to Queries
In order to link buttons to existing queries, the queries must be stored in the Queries folder of the data model (or any of its subfolders). You can also link buttons to ad hoc queries, freehand queries, and super queries.

💡 For more information on query types, see the BI Query Queries User’s Guide.

If a query contains a prompt, you can supply a different prompt value each time you click the button so that the same query returns different results.

💡 You cannot link a button to an external query (one that is not located in the Query folder or its subfolders)

To link a button to a query:

1. Build and save the query (if necessary).
2. In the Link To area of the Edit Button or Create Button dialog box, select Query from the list on the left.
3. Do one of the following:
   • To run any ad hoc query without creating a button for each, select Current Query.
   • To link the button to a specific query, click the ellipsis button .. In the Choose Query dialog box, navigate to and select the query; then, click Open.
4. In the Output to area, specify the target for the query results:
   • To send the results to a results window, select Results Window.
   • To send the results to an application, select Application. Click Export Options; then, use the Export Options dialog box to specify the options you want. Click OK.
To send the results to a text file, select **Results File**; then, type the desired name of the file in the **File name** box. Click **Results Options**; then, use the **Results Options** dialog box to specify the options you want. Click **OK**.

💡 **For more information on export and results options, see the *BI Query Queries User’s Guide***.

- To use the results to create a new table in the database, select **Database Table**; then, type a name for the new data object in the **Database Table** box. You can also leave the box blank; in this case, you are prompted for a name each time you click the button.

5. In the **Edit Button** or **Create Button** dialog box, click **OK**.

### Linking Buttons to Windows

You can link a **Design** window to a button so that clicking the button opens the window.

**To link a button to a window:**

1. In the **Link to** area of the **Edit Button** or **Create Button** dialog box, select **Design Window** from the list on the left.

2. Select a window from the list on the right.

3. Click **OK**.

### Buttons and OLE Automation

You can link a button to an application script to perform a variety of tasks within BI Query and integrate other applications with it. An application script is a set of commands that controls the actions of one or more applications. For example, a script can run a query, send the results to an Excel worksheet, and run a macro that calculates a 12-month running total. BI Query uses OLE Automation to run application scripts.
**OLE Automation**

Automation is one of the features of Object Linking and Embedding, (OLE) Microsoft’s object-based technology for sharing information and services across applications). OLE Automation lets an application take advantage of services (functionality) provided by other OLE-enabled applications. These services can vary from application to application, but they are provided through a standard object interface.

OLE Automation requires two components:

**Automation Objects**

Programmable components of an application, such as queries and results sets in BI Query. For more information on the objects available in BI Query, click **OLE Help** from the **Help** menu of your BI Query application.

**Automation Controllers**

Development tools or applications that let users and third-party developers write scripts and create applications that drive automation objects. Microsoft Visual Basic and Visual C++ are examples of automation controllers. Using an automation controller like Visual Basic, you can control automation objects in BI Query in order to open a data model or **Design** window and run a query.

**Linking Buttons to Applications, Documents, or Scripts**

In addition to an application script, you can link a button to an application or a document (in which case BI Query launches the associated application and opens the document). You can link to any application that you can open using the **Run** dialog box from the **Start** menu.

To link a button to an application, document, or script:

1. Create the script that you want the button to execute or the document you want it to open (if necessary).

2. In the **Link to** area of the **Edit Button** or **Create Button** dialog box, select **Script** from the list on the left.

3. In the box on the right, do one of the following:
   - To run an application, type the name (or path and file name) of the application followed by any necessary parameters.
• To open a document, type the path and file name of the document. (The extension of the file must be mapped on your system to a particular application.)

• To run a script, type the name (or path and file name) of the OLE Automation controller you want to use, followed by the path and file name of the script.

4. Click **OK**.

**BI Query allows only one OLE Automation controller at a time; you cannot have a controller click a button to launch another controller.**

### Example Links for Applications, Documents, and Scripts

The following table provides some examples of possible button links:

<table>
<thead>
<tr>
<th>Specified Link</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>excel c:\sales\sales2</td>
<td>Runs Excel and opens the <em>sales2</em> worksheet in <em>c:\sales</em>.</td>
</tr>
<tr>
<td>c:\sales\sales2.xls</td>
<td>Runs Excel and opens the <em>sales2</em> worksheet (if the <em>.xls</em> extension maps to the Excel application on your system).</td>
</tr>
<tr>
<td>notepad</td>
<td>Runs Notepad.</td>
</tr>
<tr>
<td>c:\scripts\sortsales.vbp</td>
<td>Runs the specified script using Visual Basic as the OLE Automation controller.</td>
</tr>
</tbody>
</table>

### Linking Buttons to Reports or Hypercubes

If you create the same reports frequently, you can use buttons to automate the process. You can link buttons to reports in BI Query Reports and to hypercubes in BI Cube Creator. Clicking the button runs the query and creates the report or hypercube.

**To link a button to a report or hypercube:**

1. In the **Link to** area of the **Edit Button** or **Create Button** dialog box, select **BI Query Reports**, or **BI Cube Creator** from the list on the left.

2. Do one of the following:
• Click **Browse** to display a dialog box where you can locate and select the resource (report or hypercube).

• Type the path and file name of the resource into the box on the right.

3. If you want to specify the path of the resource relative to a path variable, select **Make path relative to**. In the corresponding list box, select the variable. For more information about path variables, see below.

4. If the button links to a BI Cube Creator hypercube, select **Refresh** if you want the hypercube or report to refresh when you click the button.

5. Click **OK**.

**Path Variables in Button Links**

When you create a button that links to a resource such as a report or hypercube, you can specify the path of that resource relative to a variable. If at some later time the resource moves to a different location, you can change the value of the variable; you do not need to update every button that links to that resource.

You can define your own path variables. BI Query also supplies the pre-defined variables **UserPath** and **AdminPath** that you can use.

**UserPath**

Stores the location of the data model or, if the model is split, the user layer of the model.

**AdminPath**

Stores the location of the administrator layer in a split data model.

You cannot edit **UserPath** or **AdminPath**. BI Query updates them automatically if the location of the data model changes.

Path variables are useful if you are creating data models that will be deployed at different locations for different users.
Path variables are not supported in a BI Server environment. You can publish to the BI Server Repository only those reports stored in the Reports folder for the data model. As a result, BI Web users cannot run reports from buttons that use path variables, because the resolved paths do not exist in the Repository.

However, you can link buttons to any reports in the Repository (even reports from other data models) by specifying the report as a URL using the Hummingbird Repository Retrieval Protocol. For more information on linking buttons to URLs, see “Hyperlinks to URLs” on page 162.

Path Variable Scenario

A BI Query administrator stores the reports for a particular data model in the following directory:

C:\ResourceFiles\New Reports

When the administrator creates buttons in the model that link to reports in this directory, he or she specifies the paths of those reports relative to a variable, report_location, which initially has the value

C:\ResourceFiles\New Reports

In other words, these buttons link to reports that reside in a subdirectory called NewReports, and NewReports is relative to the path specified in the report_location variable.

At some later time, the administrator moves the reports for the same data model to a new directory:

C:\CurrentFiles\New Reports

The administrator also changes the value of report_location to

C:\CurrentFiles\New Reports

Since the buttons link to resources in a NewReports subdirectory relative to report_location, they will correctly point to the new location of the reports, namely C:\CurrentFiles\New Reports. The administrator has to change only the value of the variable; he or she does not have to update the link for each button in the data model.

Creating, Modifying, and Deleting Path Variables

Before you can use a user-defined path variable in a button, you must create it first using the Path Variables dialog box.
For more information on using path variables, see the Help for your BI Query application.

**To create a path variable:**

1. On the **Edit** menu, click **Path Variables**.
2. In the **Path Variables** dialog box, click **New**. In the **Name** box, type the name of the variable.
3. In the **Path** box, type the name of the path, or click **Browse** and browse the system for the desired path.
4. In the **Description** box, type a brief summary of the purpose of the variable.
5. Click **Close** to create the variable and close the dialog box.

**To modify a path variable:**

1. On the **Edit** menu, click **Path Variables**.
2. In the **Path Variables** dialog box, select the variable from the **Variables** list.
3. Apply the necessary changes to the **Name**, **Path**, and **Description** boxes.
4. Click **Close**.

**To delete a path variable:**

1. On the **Edit** menu, click **Path Variables**.
2. In the **Path Variables** dialog box, select the variable from the **Variables** list. Click **Delete**.
3. Click **Close**.

**Hyperlinks to URLs**

You can link a button to an online help system or other resource on the Web. In this respect, the button acts as a hyperlink—a reference in one document (your data model) to another (the Web resource). When you click such a button, BI Query runs the default browser on your system and opens the specified resource. You can link to any resource that can be specified using a Uniform Resource Locator (URL).

BI Query supports the standard complement of URL protocols (such as HTTP and FTP).
Hummingbird Repository Retrieval Protocol (HRRP)

In addition to the standard URL protocols, BI Query also supports a special protocol for specifying resources in a BI Server Repository: the Hummingbird Repository Retrieval Protocol (HRRP). Using this protocol, you can open data models and reports stored in the Repository. (You can also run the queries associated with each report you open.)

When you access the URL for a data model, the data model opens in a new instance of BI Query. When you access the URL for a report, the report opens in a new instance of BI Query Reports.

URLs that use HRRP take the following form:

```
hrrp://rep_id/pkg_name/doc_type/rep_segment
```

where:

- `rep_id` is the ID for the Repository. The default `rep_id` for any BI Server Repository is `r1`.
- `pkg_name` is the package name of the data model that stores the document you want to open. The package name for a data model and its objects is the same as the name of the data model (without the file extension).

💡 For more information, see “Data Model Packages” on page 224.
- `doc_type` is the type of document you want to open.
- `rep_segment` is the particular document you want to open (either a report or an entire data model).

The following table describes the possible values for `doc_type`:

<table>
<thead>
<tr>
<th><code>doc_type</code></th>
<th>Associated Resource</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>datamodel</td>
<td>Data model file</td>
<td>Do not supply a file extension (<code>.gql</code> or <code>.gqu</code>) when specifying a data model.</td>
</tr>
<tr>
<td>report</td>
<td>BI Query report</td>
<td>Supply the usual file extension (<code>.rep</code>) to specify reports created in BI Query Reports.</td>
</tr>
<tr>
<td>xreport</td>
<td>Standard report</td>
<td>Supply a file extension of <code>.xml</code> to specify a standard report.</td>
</tr>
</tbody>
</table>
Only BI Web users of the data model can access a standard report using an HRRP hyperlink button. You can create a hyperlink to a standard report in BI Query, but you cannot access the report.

**Supported URL Protocols**

BI Query supports the following protocols for URLs:

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
</table>
| HTTP     | The Hypertext Transfer Protocol (HTTP) defines the format and transmission method of documents over the Web. For example, the following URL,  
http://www.hummingbird.com  
specifies the address of the Hummingbird web site.  
When you click the button in this case, the default web browser opens the specified URL. |
| HTTPS    | HTTPS is a secure form of HTTP using the Secure Sockets Layer (SSL) protocol. In the following example,  
https://secureSite  
secureSite is the address (domain name or IP address) of a site protected by the SSL protocol.  
When you click the button in this case, the default web browser opens the specified URL. |
| HRRP     | The Hummingbird Repository Retrieval Protocol (HRRP) lets you access files in a BI Server Repository through a browser. In the following example,  
hrrp://r1/MyModel/report/SalesUK.rep  
SalesUK.rep is a report in the MyModel data model stored in the Repository.  
When you click the button in this case, the specified resource opens in a new instance of BI Query (for a data model) or BI Query Reports (for a report). |
Linking Buttons to URLs

When the user moves the mouse pointer over a button that links to a URL, the pointer changes to a hand \(\text{(GUI)}\) to indicate that the button is a hyperlink. If you have provided a screen tip for the button, it will also appear in a pop-up box when the users moves the pointer over the button. (Typically, the tip describes the target location or lists the actual URL.)
To link a button to a URL:

1. In the **Link to** area of the **Edit Button** or **Create Button** dialog box, select **Hyperlink** from the list on the left.

2. Click **Edit**. The **Edit Hyperlink** dialog box opens.

3. From the **Address** list, select the protocol you want to use for the hyperlink.

4. In the box to the right of the list, specify the URL for the resource using the appropriate procedure:

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP/HTTPS</td>
<td>Type the URL in the box. Alternatively, click <strong>Browse</strong> and use the default web browser to locate the resource. Copy the URL from the browser and paste it into the box.</td>
</tr>
<tr>
<td>HRRP</td>
<td>Type the URL in the box. Alternatively, click <strong>Browse</strong> and use the <strong>Retrieve</strong> dialog box to select the resource.</td>
</tr>
<tr>
<td>file</td>
<td>Type the path and file name of the resource in the box, or click <strong>Browse</strong> and use the <strong>Open</strong> dialog box to select the file.</td>
</tr>
<tr>
<td>mailto</td>
<td>Type the e-mail address(es) for the principal recipient(s) in the box. Separate multiple addresses using a semicolon (;). To carbon-copy other recipients, type their addresses in the <strong>Cc</strong> box. To blind carbon-copy other recipients, type their addresses in the <strong>Bcc</strong> box. To specify a subject header for the message, type it in the <strong>Subject</strong> box.</td>
</tr>
</tbody>
</table>
5. Click **OK**.

6. In the **Appearance** area, type a description of the URL into the **Screen Tip** box. This step is optional.

7. Click **OK**.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Procedure</th>
</tr>
</thead>
</table>
| FTP      | Type the URL in the box. To access the resource on a restricted server, use the following syntax:  
  
  \[ user:pswd@server:port/path/resource \]  
  where  
  - *user* is your user name on the server  
  - *pswd* is your password on the server  
  - *server* is the server IP address or name (for example, ftp.hummingbird.com)  
  - *port* is the port number for the FTP service  
  - *path* is the local path on the server where the resource resides  
  - *resource* is the file name of the resource  

  BI Query automatically parses the string you type and populates the appropriate boxes underneath with the information. Alternatively, you could type the information in the boxes and BI Query will generate the equivalent string.
Chapter 8

Controlling User Access

This section provides information on the following:

- “Types of User Access” on page 169
- “Setting Data Model Passwords” on page 170
- “Controlling Access to the Data” on page 172
- “Setting Security for Individual Users” on page 178
- “Setting BI Query Permissions” on page 183
- “Setting BI Query Preferences” on page 190

Types of User Access

BI Query lets you control what information users can access and the activities they can carry out. You can control access to the data model by setting user passwords. You can also control what data a given user or group can access in terms of rows, columns, or data values. In addition to securing the data model and the data, you can set permissions to govern the operations that users can perform on the data and the data model.
An extra level of security is possible if you are working in a BI Server environment. In this case, you can secure the data model and its associated components by publishing them to the Repository and then specifying which users and groups have access to them.

💡 For more information on security and BI Server, see “Controlling Access with BI Server” on page 229.

### Setting Data Model Passwords

Data models can be password-protected to prevent unauthorized people from using them or making changes to them. When you open a data model with BI Query Admin, it checks to see whether an administrator password has been set. Similarly, BI Query User and BI Query Update check to see whether a user password has been set. If either has been set, the user must supply it in the Enter Data Model Password dialog box.

Administrator and user passwords are independent of each other. User passwords can be set either by users or by the administrator. Administrators can also retrieve forgotten user passwords.

In a split data model, the administrator password is stored in the administrator layer; the user password is stored in the user layer. In both cases, passwords are stored in encrypted form.

### Setting Data Model Passwords

It is recommended that administrators set user passwords only for combined data models; if you save the user layer of a split data model with BI Query Admin, all the user’s customizations become part of the administrator layer and cannot be changed by the user.

**To set an administrator password:**

1. In BI Query Admin, on the **Tools** menu, click **Permissions**.
2. In the **Permissions** dialog box, click **Admin Password**.
3. In the **Set Administrator Password** dialog box, type a password into the **New password** box; then, type it again into the **Verify password** box.
4. Click **OK**. In the **Permissions** dialog box, click **OK**.
5. Save the data model.
To set a user password for a combined data model:

1. On the **Tools** menu, click **Preferences**.
2. In the **Preferences** dialog box, click **User Password**.
3. In the **Set User Password** dialog box, type a password into the **New password** box, and then type it again into the **Verify password** box.
4. Click **OK**; then, click **OK** again in the **Preferences** dialog box.
5. Save the data model.

**Changing or Removing Passwords**

Depending on whether you want to change or remove an administrator or user password, use the **Set Administrator Password** or **Set User Password** dialog box.

**To change or remove a password:**

1. In the **Old password** box, type the existing password. It appears as a series of asterisks.
2. Do one of the following:
   - To remove the password click **OK**.
   - To alter the password, type the new password into the **New password** and **Verify password** boxes; then, click **OK**.
3. In the **Permissions** or **Preferences** dialog box, click **OK**.
4. Save the data model.

**Retrieving Forgotten User Passwords**

BI Query administrators can retrieve forgotten user passwords, provided they have access to the user’s user layer. BI Query administrators cannot retrieve their own data model passwords.

💡 To retrieve a forgotten administrator password, contact Hummingbird Technical Support.

**To retrieve a forgotten user password:**

1. Open the data model. If you are using a split data model, open the user’s user-layer file.
2. On the **Tools** menu, click **Permissions**.
3. In the **Permissions** dialog box, click **User Password**.
Do not save the data model. If you open a user’s user layer within BI Query Admin and save it, the user’s customizations become part of the administrator layer.

Controlling Access to the Data

Depending on the database and the security system your organization is using, you can set up different levels of restricted access. For example, access to tables can be restricted by the database administrator at the database level. If a user doesn’t have access and attempts to include the table in a query, the database disallows it. You can also restrict access to tables within the data model, using one or more of the following methods.

Access to Entire Tables

The most basic way to restrict access is to limit the set of tables that users can see in the data model. You should determine this set of tables before creating the model. Depending on the number of user groups and their specific information needs, you may need to create more than one data model.

When different users or user groups share certain tables in the database, you can load all the tables, and then add or remove tables as needed to create the different models. You can add tables by loading them from the database. You can remove tables by deleting the corresponding data objects or actual relationships from the data model.

For more information, see “Loading Tables from the Database” on page 80.

Access to Columns

In some cases, you can restrict access at the table level, allowing users access to some of the information in a table or none of it. You can secure access to columns by hiding or deleting the corresponding attributes in the data model.

Access to Rows

In addition to securing columns, you may be able to secure table rows by applying an object qualifier to a data object or actual relationship. Any query involving the object returns only those rows that satisfy its qualifier. Object qualifiers also let you limit access to rows on an individual user or group basis.
Access to Data Values

In other cases, you may be able to restrict access to specific values in a table, either by disabling the default data values query for an attribute or by creating a data values file for that attribute.

Securing Access to Columns

Certain information in a table may be useful to some users but not others. You can prevent users from seeing columns they don’t need by hiding the corresponding attributes in BI Query. You hide attributes by turning off the corresponding Visible flag in the attribute window in Design mode. Hidden attributes do not appear in BI Query User or BI Query Update; they are visible in BI Query Admin in Design mode (at the bottom of the attribute list).

Typically, you display only the attributes users need so that they don’t have to spend time searching for them. For example, you should hide system-generated primary and foreign keys. It is also possible to remove attributes from a data model.

Hiding attributes—rather than deleting them—makes the data model easier to maintain. If you remove an attribute from a data model and subsequently refresh the model, BI Query recognizes that the attribute exists in the database and prompts you to add it. If you hide the attribute instead, BI Query does not prompt you each time you refresh the model. If you later need to give users access to the hidden attribute, you can make it visible.

💡 For more information on refreshing the data model, see “Refreshing Data Models” on page 244.

To secure access to columns in a table:

1. In Design mode, double-click a data object or actual relationship.
2. In the Attribute window, do one of the following:
   • To remove an attribute, click the attribute line (the thick line connecting the boxes of the attribute). On the Design menu, click Delete Attribute.
   • To hide an attribute, turn off its Visible flag by clearing the corresponding check box in the Visible column.
3. Save the data model.
Object Qualifiers

Once you’ve loaded the tables you want, you can place restrictions on queries on an individual table by using an object qualifier. An object qualifier associates a specified expression with a data object. Each time a user runs a query involving the data object, BI Query adds the expression to the WHERE clause for the query.

In addition to securing data, adding a qualifier to an object lets you improve query performance and protect users from running queries that retrieve more data than they need.

For example, if you include the expression `Retailers.country = 'UK'` in the object qualifier for the Retailers table, any query involving this table returns only those rows that satisfy the expression. Object qualifiers can also include prompts, variables, and subqueries. If you include a subquery, it must return a single column of results. You can apply multiple object qualifiers to a given data object.

Basic Expression Syntax

An object qualifier typically follows the form of expressions in a WHERE clause. The most basic object qualifier expression uses the following syntax:

```
object.attribute operator testvalue
```

where:

- `object` is the data object that is being qualified (or a data object that is joined to the qualified object).
- `attribute` is an attribute in `object`.
- `operator` is one of the standard qualification operators for queries.
- `testvalue` is one or more values that qualify the attribute. `testvalue` can be an explicit data value, a prompt, a variable, another attribute, a subquery, or a combination thereof.

To specify more than one data value in `testvalue`, separate successive values with a comma and enclose the entire sequence in parentheses.

For more information on qualification operators, see the *BI Query Queries User’s Guide*. 
## Object Qualifier Examples

The following table describes some sample object qualifier expressions:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Retailers.country IN ('UK', 'Canada', 'Brazil')</strong></td>
<td>This expression tests the <em>country</em> attribute of the Retailers object against the listed values. All rows for the specified countries are returned.</td>
</tr>
<tr>
<td><strong>Orders.country = «Country»</strong></td>
<td>This expression tests the <em>country</em> attribute of the Orders object against the value supplied by the user in response to the specified prompt. All rows for the country specified by the user are returned. If the user specifies multiple countries, BI Query automatically converts the expression operator from = to IN and surrounds the prompt values in parentheses.</td>
</tr>
<tr>
<td><strong>Orders.country IN («Country»)</strong></td>
<td>This expression is identical to the previous one except that it uses the IN operator to test the <em>country</em> attribute against multiple values supplied by the prompt. The parentheses around the prompt are necessary because the prompt may return multiple values.</td>
</tr>
<tr>
<td><strong>Sales.year &lt; «Year»</strong></td>
<td>This expression tests the year attribute in the Sales object against the value of the Year variable. All rows for years previous to the current year are returned.</td>
</tr>
</tbody>
</table>
Securing Access to Rows

You can limit the access to rows in a table by using an object qualifier. Be sure to test object qualifiers before distributing them with the data model, because any object qualifiers you create are combined with users’ qualifications when they formulate queries.

For information on using object qualifiers to limit access to certain users or groups, see “Setting Security for Individual Users” on page 178.

Users can’t modify the object qualifiers you create unless you assign the Edit SQL string permission.

To secure access to rows in a table:

1. In Design mode, click the data object or actual relationship you want to qualify. On the Design menu, click Object Qualifiers.

   If the selected object is a merged object, the Choose Merged Table dialog box opens. Select the distinct object you want to qualify; then, click OK.

   For more information on merged objects, see “Merged Data Objects” on page 104.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales.year &lt; «Year» and Sales.orderID = Product.orderID</td>
<td>This expression is identical to the previous one except that it also specifies a join condition between the Sales and Product objects. You could use this expression to qualify either object.</td>
</tr>
<tr>
<td>Products.id NOT IN (SELECT Sales.product_id FROM Sales WHERE (Sales.order_amt &gt; 1000))</td>
<td>This expression tests the id attribute of the Products object against values returned by the specified subquery. All rows for product IDs not in the query results are returned. The parentheses around the subquery expression are necessary.</td>
</tr>
</tbody>
</table>
2. In the **Object Qualifiers** dialog box, click **New**. The **New Object Qualifier** dialog box opens.

3. In the **Qualifier ID** box, type a name for the qualifier. This name distinguishes the qualifier from other components in the data model when you’re assigning access to the data model. For more information on securing individual components of a data model, see “Setting Security on Published Items” on page 232.

4. In the **Expression** box, type the qualifier expression.

   ![For more information on valid expression syntax, see “Object Qualifiers” on page 174.](image)

   You can also build the expression using the available buttons, as follows:

   - To specify the database name of the attribute to which you want to restrict access, click **Attributes**; then, use the **Insert Attribute Name** dialog box to select and insert the name.

   - To insert a variable into the qualifier, click **Variables**; then, use the **Variables** dialog box to select the variable.

   - To insert a prompt into the qualifier, click **Prompts**; then, use the **Prompts** dialog box to select and insert the prompt.

   - To insert a subquery into the qualifier, click **SubQuery**; then, use the **Insert Subquery** dialog box to select the query.

   ![To include more than one prompt or variable in an object qualifier expression, separate each term with a plus sign (for example, Retailers.country = «promptA» + «variableB»).](image)

5. Click **OK**.

**Securing Access to Data Values**

Data values are all the database values for an attribute. When users formulate queries, they can qualify an attribute with one or more data values. They can select these values from a list of the database values for the qualified attribute. BI Query uses a default `SELECT DISTINCT` query to produce such a list. However, there are two ways you can limit access to this list:

- Create a data values file for the attribute. If this file is present, BI Query lists its contents instead of the default data values.
For more information on data values files, see the *BI Query Queries User’s Guide*.

- Disable access to the default query. You should disable the default query if loading all the values in the database uses network resources inefficiently—for example, when an attribute has a large number of values or when those values are numbers or dates.

  Data values in a data values file are still available to users even if you disable the default query.

**To disable the default query for an attribute:**

1. In Design mode, double-click the data object or actual relationship that stores the attribute.
2. In the Attribute window, click the box in the Data Values column to remove the check mark (the default), thereby disabling data values for that attribute.
3. Close the Attribute window.
4. Save the data model.

### Setting Security for Individual Users

To set row-level security for users, you can either use the user and group security structure provided by BI Server or use BI Query’s predefined variables.

### Setting Security with BI Server

You can apply multiple object qualifiers to a given data object. If you are working in a BI Server environment, you can individually grant or deny access to each object qualifier when you publish the data model. If you deny a particular user or group access to a qualifier, the qualifier will not apply to any query created by that user or member of that group. In this way, you can selectively apply object qualifiers to users and groups by granting them access to the qualifiers. In the Set Security dialog box, object qualifiers appear in the list for the corresponding data object in the Views folder.

The Set Security dialog box is part of another BI program module and has its own Help system.

For more information on setting security using BI Server, see “Controlling Access with BI Server” on page 229.
Setting Security with Variables

Three predefined variables (DBUserID, BIUserID, and BIGroupID) are available in BI Query Admin.

**DBUserID**
Supplies the database user name for the user currently logged on to the database.

**BIUserID**
Supplies the BI Server user name for the user currently logged on to BI Server.

**BIGroupID**
Supplies the list of groups to which the user currently logged on to BI Server belongs. You can use this variable only if the BI Server administrator has created BI Server groups and associated users with those groups.

You can use these variables to create row-level security for individual users without creating separate object qualifiers or queries for each user. To use them, you must have permission to create and secure data models. You can't edit or delete these variables.

To make use of these variables, there must be a table in the database that associates the data with user name or group name. If you include this table in a data model, you can qualify data objects and queries using one or more of these variables to ensure that only the associated data is returned to that user.

**To set up variables to supply user or group names:**
1. Associate the data in a database table with a database user name, BI Server user name, or BI Server group name.
2. Add the table to the data model.
3. Qualify a data object with the appropriate variable.

In addition to qualifying objects, you can also use the DBUserID, BIUserID, and BIGroupID variables to qualify queries directly. For more information, see the *BI Query Queries User's Guide*.

If you use the BIGroupID variable in an object qualifier, the qualifier expression you create should accept multiple values. For example, the following qualifier tests the group attribute of the SalesData table against any number of group values: `SalesData.group IN ("BIGroupID")`. The parentheses around the variable are necessary, since it may return multiple values.
Associating Data with User Names

There are two ways to associate the data in your database with a database user name or BI Server user or group name:

- Use an existing database table.
- Create a new table.

Using an Existing Database Table

In many cases, you can use an existing database table. For example, suppose your database already contains a table, Employees, that lists unique employee information.

<table>
<thead>
<tr>
<th>employee#</th>
<th>lastname</th>
<th>salary</th>
<th>bonus</th>
</tr>
</thead>
<tbody>
<tr>
<td>ay173</td>
<td>Allen</td>
<td>100,000</td>
<td>50,000</td>
</tr>
<tr>
<td>br276</td>
<td>Barbieri</td>
<td>90,000</td>
<td>55,000</td>
</tr>
<tr>
<td>ch56</td>
<td>Chen</td>
<td>125,000</td>
<td>45,000</td>
</tr>
<tr>
<td>km476</td>
<td>Kwame</td>
<td>125,000</td>
<td>45,000</td>
</tr>
</tbody>
</table>

You can add a “username” column to the table to associate database user names with existing user data (employee number, last name, salary, and bonus), and then populate the new column with the appropriate values.

<table>
<thead>
<tr>
<th>username</th>
<th>employee#</th>
<th>lastname</th>
<th>salary</th>
<th>bonus</th>
</tr>
</thead>
<tbody>
<tr>
<td>jallen</td>
<td>ay173</td>
<td>Allen</td>
<td>100,000</td>
<td>50,000</td>
</tr>
<tr>
<td>gbarbieri</td>
<td>br276</td>
<td>Barbieri</td>
<td>90,000</td>
<td>55,000</td>
</tr>
<tr>
<td>lchen</td>
<td>ch56</td>
<td>Chen</td>
<td>125,000</td>
<td>45,000</td>
</tr>
<tr>
<td>nkwame</td>
<td>km476</td>
<td>Kwame</td>
<td>125,000</td>
<td>45,000</td>
</tr>
</tbody>
</table>

To restrict rows in the SalesData table on the basis of database user name, you could create an object qualifier for SalesData with the following expression:

```
Employees.username = «DBUserID»
AND SalesData.employeeID = Employees.employee#
```
This expression specifies the join condition between the two tables. When a user runs a query involving the SalesData table, only those rows for his or her employee ID are returned.

To add a column to a table, contact your database administrator. Once the database table has been modified to include the new column, do the following:

1. Refresh the data model to add the column to the corresponding data object. For more information, see “Refreshing Data Models” on page 244.

2. Update the table with the required values. For more information, see “Adding, Modifying, and Deleting Records” on page 274.

Creating a New Table

If existing values in your database aren’t unique for individual users, you can create a separate “security table” that correlates user names with values. For instance, suppose your database includes a table for sales territories. Each of your salespeople is assigned to one or more territories. Some territories are also assigned to more than one salesperson. You can create a security table in the database to associate the salesperson’s user name with the sales territory information you want him or her to see.

<table>
<thead>
<tr>
<th>BIServerName</th>
<th>Territory</th>
</tr>
</thead>
<tbody>
<tr>
<td>mjimenez</td>
<td>4</td>
</tr>
<tr>
<td>mjimenez</td>
<td>5</td>
</tr>
<tr>
<td>mjimenez</td>
<td>6</td>
</tr>
<tr>
<td>ibeckworth</td>
<td>4</td>
</tr>
<tr>
<td>ibeckworth</td>
<td>7</td>
</tr>
</tbody>
</table>

💡 For more information, see “Creating Tables” on page 113.

To restrict rows in the SalesData table on the basis of BI Server user name, you could create an object qualifier for SalesData with the following expression:

```
SecurityTable.BIServerName = «BIUserID»
AND SalesData.region = SecurityTable.Territory
```

When a user runs a query involving the SalesData table, only those rows for the territories associated with the user are returned.
Qualifying Data Objects with User or Group Variables

To provide row-level security using one of the DBUserID, BIUserID, or BIGroupID variables, you insert the variable in an object qualifier as you would any other variable.

To qualify a data object using a user or group name variable:

1. In Design mode, click the data object or actual relationship you want to qualify. On the **Design** menu, click **Object Qualifiers**.
   
   If the selected object is a merged object, the **Choose Merged Table** dialog box opens. Select the distinct object you want to qualify; then, click **OK**.
   
2. In the **Object Qualifiers** dialog box, click **New**. The **New Object Qualifier** dialog box opens.
   
3. In the **Qualifier ID** box, type a name for the qualifier. This name distinguishes the qualifier from other components in the data model when you’re assigning access to the data model.
   
4. In the **Expression** box, specify the qualifier expression. Typically, the expression tests one of the predefined variables against a constant value (for example, «BIUserID» = 'ibeckworth') or an attribute (for example, `SalesData.user = «BIUserID»`).
   
   For more information on valid expression syntax, see “Object Qualifiers” on page 174.
   
   To insert a variable into the qualifier, click **Variables**. From the **Variables** list, select the variable you want to use; then, click **Insert**.
   
   If you include the BIGroupID variable in the expression, remember to use the IN operator (or NOT IN) and enclose the variable in parentheses. (BIGroupID may return multiple values.)
   
5. In the **New Object Qualifier** dialog box, click **OK**.
Setting BI Query Permissions

You can customize the level of database activity and BI Query capability to reflect the skills and needs of various users. Novice users with limited knowledge of the database may need simple, predefined access. Advanced users who know how to create sophisticated queries and understand complex data access strategies may need the freedom to specify their own SQL in a query or create their own joins between tables.

You can assign permissions for a data model to let users carry out certain activities in BI Query User and BI Query Update while preventing them from carrying out others. Users can’t change the permissions they’re assigned. BI Query permissions don’t override DBMS permissions. (For certain activities, you need to provide other setup options, and users need certain database permissions.)

To assign permissions:

1. In BI Query Admin, on the Tools menu, click Permissions. The Permissions dialog box opens.
2. In the Permissions dialog box, click the appropriate check boxes to grant (or deny) permissions for the current data model.
3. Click the appropriate buttons to set an administrator password, to save model information to the database, and to show a user password. If you are in a BI Server environment, you can also set security for the data models you publish to the BI Server Repository.
4. Set the Rows and Time Governor permissions to specify the maximum number of rows BI Query should retrieve or the maximum period of time that should elapse during a query.
5. Click OK.

If your data access environment includes BI Server, BI Server system permissions determine who can make use of the BI Server Repository and to what extent. Information on system permissions is available in the BI Server Administrator’s Guide.

Suggested Permission Settings for Novice Users

Suggested permissions for novice users consist of the following:
- Edit Data Model (split data model only)
- Save Data Model (split data model only)
• Save Queries
• Connect per Window
• Create Calculated Attributes
• Rows Governor: 10000
• Time Governor: no limit

**Suggested Permission Settings for Advanced Users**

Suggested permissions for advanced users consist of the following:

• Edit Data Model (split data model only)
• Save Data Model (split data model only)
• Edit SQL String
• Save Queries
• Save Passwords
• Dynamic Relationships
• Create Calculated Attributes
• Enforce Connect per Query
• Rows Governor: no limit
• Time Governor: no limit

**Permissions for Editing and Saving Data Models**

You assign the **Edit data model** permission to let users add their own **Design** windows, buttons, and ornaments to the user layer and change their preferences and other default settings. To let users save their changes, you must also assign the **Save data model** permission.

⚠️ **A combined data model, which consists of one file, is designed for sites that don’t allow users to make changes to a data model. With the permissions mentioned above, users can make changes to a combined data model, but when you distribute a new version of the data model, the changes you’ve made will overwrite users’ customizations.**
A split data model lets the administrator make changes to the administrator layer without affecting changes made by the user in the user layer. To let users save a data model in two layers (an administrator layer and a user layer), assign the **Save split data model** permission.

### Permissions for Modifying SQL Strings

Users who understand SQL and the database can modify the SQL string generated when they formulate a query. To let them do this, set the **Show SQL string** preference to display the SQL in the **Query** window; then, assign the **Edit SQL string** permission to let users modify the SQL. Users can then modify anything after the **SELECT** statement. This permission doesn’t compromise any restrictions placed on users by the DBMS.

When the **Allow non-select SQL** permission is assigned, authorized users can also perform special functions in the database, such as updating and deleting data. When the **Show SQL string** preference is set, they can type anything into the SQL string. When the **Allow non-select SQL** permission isn’t assigned, users can’t run a query for which they’ve modified the SQL string to add, delete, or update data, even when the **Edit SQL string** permission is assigned.

💡 For more information on non-SELECT SQL, see “Sending Non-SELECT SQL to DBMSs from the Query Window” on page 215.

The following table summarizes these settings:

<table>
<thead>
<tr>
<th>Settings</th>
<th>Allowed Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show SQL string preference, Edit SQL string permission</td>
<td>Modify anything after <strong>SELECT</strong> statement</td>
</tr>
<tr>
<td>Show SQL string preference, Edit SQL string permission, Allow non-select SQL permission</td>
<td>Modify anything in the SQL string</td>
</tr>
</tbody>
</table>

🔍 BI Query Update commands for inserting, updating, and deleting aren’t restricted by these settings.
Chapter 8: Controlling User Access

Calculated Attributes Permissions
When you assign the **Create calculated attributes** permission, users who need to perform calculations on their data can add attributes to an attribute window that request the DBMS to perform user-defined calculations. For example, a data model might include Quota and Sales attributes that retrieve a salesperson’s sales quotas and total sales, respectively. By creating a new attribute that calculates Quota minus Sales, they can determine the shortfall in sales quotas.

When you assign the **Save calculated attribute in model** permission, users are also able to save in the data model any calculated attributes they have created.

💡 For more information on calculated attributes, see the *BI Query Queries User’s Guide*.

Save Queries Permission
When you assign the **Save queries** permission, users can save queries in order to reuse them in BI Query or in other applications. This permission isn’t necessary for data models that provide only canned queries and reports instead of ad hoc querying capability.

Dynamic Relationship Permissions
There are many possibilities for creating relationships between tables in most databases. But creating every possible relationship would make it difficult for users to understand and use the data model. To minimize confusion and maximize flexibility, you can create the most important relationships between data objects yourself, and then assign both the **Dynamic relationships** permission and the **Allow dynamic relationships** preference. These settings let advanced users create their own relationships between data objects.

💡 For more information on dynamic relationships, see “Types of Relationships” on page 85.

For example, to satisfy the needs of most users, you might provide a relationship between the Retailers and Sales data objects, specifying an equal join that lets users retrieve information about only those retailers who have placed sales. You can then add flexibility for advanced users by allowing them to create their own join conditions—for example, to specify an outer join that retrieves retailers even when they haven’t placed an order.
By default, when users create a dynamic relationship in a query, BI Query lets the users save the relationship with the query. You can prevent users from saving dynamic relationships by denying the **Save dynamic relationships** permission.

**Freehand Queries Permission**

The **Edit Freehand Queries** permission lets advanced users create (and run) freehand queries, essentially allowing them to send any SQL commands to the DBMS to be executed. This permission doesn’t compromise any restrictions placed on users by the DBMS.

**Custom Checks Permission**

With the help of Technical Support, you can create a Dynamic Link Library (DLL) file called `gqlcheck.dll`. If the **Custom checks** permission is assigned, BI Query calls a routine in the resource file before submitting any SQL string to the DBMS. This routine accepts the SQL string that BI Query is about to send and returns a potentially modified string, which BI Query runs in its place. This feature can be used to check security, optimize performance, and so on.

To use `gqlcheck.dll` with BI Query V8.0 and later, recompile the DLL with Microsoft Visual C++ 6.0. When compiling the DLL, make sure you adhere to the `_declspec(dllexport)` calling convention.

**Update Database Permission**

Because organizations need to maintain the quality and security of data in the corporate database, only authorized users can use BI Query to make changes to corporate data. With the appropriate authorization, users can update data and create new tables in the database.

BI Query Admin and BI Query Update include a facility for adding, modifying, and deleting data in the corporate database. This facility is primarily designed for users who are responsible for maintaining database tables.

Users can update the corporate database only if the database administrator grants the appropriate database permissions and you set up the data model to perform updates by doing the following:

- assigning key fields to the tables that users are allowed to update
- assigning the **Update database** permission
Submit to Table Permission

When users formulate a query, they can send the results to a new table in the database. They can then use the table to formulate additional queries. Users can create their own tables in the database if the database administrator grants permission to create tables and if you assign the **Submit to table** permission.

It can be useful to let users create their own tables. If they’re extracting information from a number of tables, their queries take time because they use multiple joins. Summarizing the information in one table eliminates the joins and reduces the number of rows that the DBMS has to process. This gives users faster access to the information and reduces the load on the database server. Here are two examples:

- Instead of querying the Retailers and Sales tables each time they need to retrieve information about a retailer’s monthly sales, users can query a table that contains retailer sales summarized by month. This method works particularly well when users need to analyze information that isn’t changing regularly.
- By creating a table that summarizes sales for a particular year, users can query the table and perform calculations on the results without the values changing and becoming out of date.

Submit to Named Table Permission

Users can name new tables and the corresponding data objects when you assign the **Submit to named table** permission. This is useful when users send query results to a new table in the database using buttons; each time they click the button, BI Query drops and re-creates the table with the current results. When this permission isn’t assigned, BI Query creates a unique table each time. The database administrator must grant users permissions to create and drop tables.

⚠️ **Keep in mind that this permission allows users to drop the named table. Because they can enter arbitrary table names, it gives them the facility to drop any table from the database.**

Permissions for Controlling DBMS Connections

BI Query provides several permissions to let you control how users connect to the DBMS.
Connect per Window

Lets you assign a connection file to each window in a data model—rather than to the data model as a whole—so that a data model can return information from more than one database. When this permission is assigned, the **Enforce connect per query** permission is automatically set, and the **Connection** button is enabled in the **Design Windows** dialog box.

Enforce Connect per Query

This permission ensures that BI Query connects to the DBMS only when users run a query; BI Query disconnects from the DBMS once the results are returned. Connecting to the DBMS only for the duration of a query minimizes the resources being used on the server at one time.

💡 For more information, see “Enforcing Connect per Query” on page 234.

Save Passwords

This permission lets users save passwords (in encrypted form) in connection files. When set, this permission lets users connect to a DBMS without specifying their password each time.

💡 For more information on the **Save passwords** and **Cache passwords** permissions, see “Connection File Permissions” on page 48.

Cache Passwords

This permission saves missing connection information for users in memory. The **Cache passwords** permission affects all missing connection information (not just passwords) and applies to all connection files that users may use during a session.

Rows Governor and Time Governor

These permissions let you limit the connection time taken up by potentially expensive queries. You can limit the number of rows that a query returns (**Rows Governor** permission) or the maximum time that a query can run before it’s automatically stopped (**Time Governor** permission).

💡 For more information on the governor permissions, see “Setting Query Limits with Governors” on page 237.
Permissions for Sharing Queries and Reports

To let BI Web and BI Query users of the published data model share their ad hoc queries and standard report specifications with other users, set the **Share queries and reports** permission.

If this permission is assigned, users have the option of publishing queries and report specifications either as “personal,” in which case the resource is not visible to other users, or “shared,” in which case the resource is visible to all users of the data model. Once a query or report specification is shared, only the BI Query administrator can edit or delete it.

💡 For more information on personal and shared files, see “Personal Versus Shared Files” on page 28.

You can also set the **Data Model is visible in portfolio** permission for BI Web users when you publish the data model. For more information, see “Setting Data Model Visibility” on page 227.

Setting BI Query Preferences

If you don’t assign certain permissions, you can set the corresponding preferences using the **Preferences** dialog box. Keep in mind that users can’t change their permissions, but they can change their preferences.

**To set preferences:**

1. On the **Tools** menu, click **Preferences**. The **Preferences** dialog box opens.
2. In the **Preferences** dialog box, set the preferences you want to be defaults for users.
3. Click **OK**.
This section provides information on the following:

- “Types of Help” on page 191
- “Metadata” on page 192
- “Adding Metadata to the Database” on page 193
- “Specifying Metadata Sources” on page 195
- “Loading Metadata” on page 199
- “Providing Attribute Descriptions” on page 200
- “Annotating Existing Help” on page 206
- “Creating Additional Help Files” on page 206

**Types of Help**

By providing additional information for your users, you can make your data model a more useful representation of corporate information. Typical tasks you can perform to add value to the data model include the following:

- providing metadata for the data objects and attributes in the data model
- creating attribute descriptions
• customizing BI Query’s online Help
• adding your own Help files to deliver important data model-specific information to your users

Metadata

Metadata is data about data. It labels, summarizes, and suggests the questions that can be answered by the data in the corporate database. It does so in familiar terminology rather than database terminology. Like other data, metadata is stored in the database in tables. It can provide information about the database as a whole, about a particular table, or about a column in a table.

You can provide metadata for the data objects and attributes in a data model. For data objects, metadata replaces table or correlation names with display names; for attributes, it replaces column names with business terms. It also provides short descriptions for both that explain what kind of information they represent. Including metadata in a data model gives users a clearer understanding of their data and helps them build queries. Because the metadata is stored in the data model, it is always available.

Providing Metadata in a Data Model

To provide metadata, ensure that your database stores metadata (contains a metadata source). (If it does not store metadata, you can use BI Query to create it.) You can then set up the necessary queries to specify the metadata source. Loading the metadata into the data model runs the queries to the metadata source and retrieves the metadata.

To help you set up the queries you need to specify the metadata source, you can use one of a number of editable template queries. In addition to BI Query’s own, you can use template queries for metadata supplied by IBM’s DataGuide, Informatica’s PowerMart MX, and Prism’s Directory Manager (using the Prism MetaLink option). Once these queries have been edited to specify the source of the metadata for a particular data model, you or a user can load the metadata into that model.

Providing metadata in a data model involves three basic steps:
1. Add metadata to the database (if not already present) by creating tables in the database and populating them with metadata. For more information, see “Adding Metadata to the Database” on page 193.

2. Specify the metadata source by editing BI Query’s metadata query templates (or by writing your own query). For more information, see “Specifying Metadata Sources” on page 195.

3. Load the metadata into the data model. For more information, see “Loading Metadata” on page 199.

How BI Query Displays Metadata

Metadata provides display names for data objects and business terms for attributes as well as short descriptions for both. When the metadata is loaded, the display names replace the data object names, and the short descriptions are available as tool tips.

Display Names

Display names for tables provide clear, natural-language labels that users understand. For example, a table with the database name sales_na might be loaded into a Design window as the North American Sales data object. Similarly, display names for columns replace a data object’s attribute names with common business terms. For example, a column with the database name quartsale might become the Sales by Quarter attribute in the North American Sales data object.

Tool Tips

Short descriptions displayed as tool tips provide additional information about a data object or attribute. A short description in the metadata for a data object, for example, might include the number of rows returned or the frequency with which the underlying table is updated. A short description for an attribute that returns a number might include the unit of measure represented and where the data originated.

Adding Metadata to the Database

If your database does not contain metadata, you can create metadata tables for the data models you use and then populate the tables.
Creating Metadata Tables

You create metadata tables in the same way as any other table. Create one table to supply metadata for data objects and one table to supply metadata for attributes.

To create metadata tables:

1. Create a data object for table metadata. Give it a meaningful name (for example, tablemeta). Add the following attributes to the object:
   - an attribute for table names as they appear in the database (for example, tab_name)
   - an attribute for the display names of tables (for example, disp_name)
   - an attribute for the short descriptions of tables (for example, short_desc)
2. In the attribute window for the data object, select the Key check box for the table name attribute. This sets the table name attribute as a key field.
3. Create a data object for column (attribute) metadata. Give it a meaningful name (for example, columnmeta). Add the following attributes to the object:
   - an attribute for table names as they appear in the database (for example, tab_name)
   - an attribute for column names as they appear in the database (for example, col_name)
   - an attribute for the display names of columns (for example, disp_name)
   - an attribute for the short descriptions of columns (for example, short_desc)
4. In the attribute window for the object, set the table name and column name attributes as key fields.
5. Create the corresponding tables in the database for both objects. For more information, see “Creating Tables” on page 113.

💡 For more information, see “Loading Tables from the Database” on page 80.

Once you create metadata tables in the database, you can check that they have been created properly by loading them into the data model. After you load the tables, reselect the key fields. This information is retained in the data object, not in the database, and is lost when you load the tables.
Populating Metadata Tables

Populate the metadata tables using BI Query Admin, BI Query Update, or any other tool that allows you to insert data into a table. For more information, see “Requirements for Updating Database Records” on page 271.

Specifying Metadata Sources

To provide metadata for the data objects and attributes in a data model, you need to specify the source for the metadata in the database—which tables and columns store the metadata. You specify the source by using the template queries supplied with BI Query, editing them, or writing your own. In addition to a standard query template, BI Query provides editable query templates for metadata from IBM DataGuide, Informatica PowerMart MX, and Prism Directory Manager.

When you or your users load metadata, BI Query runs these queries and retrieves the corresponding metadata.

Basic SQL for Metadata Queries

The SQL for any metadata query must contain a WHERE clause in the following form:

`WHERE metatable.tab_name='«table_name»'`

where:

- `metatable` is the name of the metadata table (either for data object or attribute metadata). For DB2 and Oracle databases, it may be necessary to include an additional creator name. For Teradata databases, it may be necessary to include an additional database name.
- `tab_name` is the name of the column in `metatable` that stores data object names (either correlation or database table names).
- `«table_name»` is a special BI Query variable that supplies the correlation name of a data object in the data model; if the object does not have a correlation name, the variable supplies the database name of the object.

When you load metadata, BI Query replaces `«table_name»` successively with the names of the data objects for which there is metadata. The query associates the corresponding display name and short description with each object and attribute on the basis of the supplied names. The quotes around `«table_name»` are necessary because the variable returns a string value.
The full SQL for a metadata query depends on the type of metadata (data object or attribute) and whether the display names and short descriptions contain more than one row of data each. If the display names and short descriptions contain multiple rows of data, the SQL orders by sequence; otherwise, no ordering is required.

For metadata tables containing more than one row of data per column, BI Query ignores display names after the first row for each attribute or data object but appends the rows of descriptions sequentially.

**SQL for Data Object Metadata**

For data object metadata, the SQL must select the display name and then the short description, in that order, from the appropriate metadata table.

When display names and short descriptions each occupy only one row in the metadata table, the basic SQL for a data object metadata query takes the following form:

```sql
SELECT disp_name, short_desc FROM tablemeta
WHERE tablemeta.tab_name = '«table_name»'
```

Otherwise, the SQL takes the following form:

```sql
SELECT disp_name, short_desc SEQUENCE FROM tablemeta
WHERE tablemeta.tab_name = '«table_name»' ORDER BY SEQUENCE
```

where:

- `tablemeta` is the name of the table that contains the data object metadata.
- `disp_name` is the name of the column in `tablemeta` that contains the display names for data objects.
- `short_desc` is the name of the column in `tablemeta` that contains the short descriptions for data objects.
- `tab_name` is the name of the column in `tablemeta` that stores data object names (either correlation or database table names).

The SQL shown in this example matches BI Query’s standard template for data object metadata queries.

**SQL for Attribute Metadata**

For attribute metadata, the SQL must select the attribute database name first, and then the display name and short description.
When display names and short descriptions each occupy only one row in the metadata table, the basic SQL for an attribute metadata query takes the following form:

```
SELECT col_name, disp_name, short_desc FROM columnmeta
WHERE columnmeta.tab_name = '«table_name»'
```

Otherwise, the SQL takes the following form:

```
SELECT col_name, disp_name, short_desc SEQUENCE FROM columnmeta
WHERE columnmeta.tab_name = '«table_name»' ORDER BY SEQUENCE
```

where:

- `columnmeta` is the name of the table that contains the attribute metadata.
- `col_name` is the name of the column in `columnmeta` that contains the database names for attributes or the display name for calculated attributes.
- `disp_name` is the name of the column in `columnmeta` that contains the display names for attributes. This is not used for calculated attributes.
- `short_desc` is the name of the column in `columnmeta` that contains the short descriptions for attributes.
- `tab_name` is the name of the column in `columnmeta` that stores data object names (either correlation or database table names).

💡 The SQL shown in this example matches BI Query’s standard template for attribute metadata queries.

**Creator and Database Names in Metadata Queries**

For DB2, Oracle, or Teradata databases that require it, a creator or database name must appear in front of the metadata table name in both the `FROM` and the `WHERE` clauses, as in this example for single-row attribute metadata:

```
SELECT col_name, disp_name, short_desc FROM dbaname.columnmeta
WHERE dbaname.columnmeta.tab_name = '«table_name»'
```

IBM DataGuide queries always target a DB2 database, and DB2 always requires a creator name. For that reason, BI Query’s IBM DataGuide template supplies a default creator name in the appropriate places in those queries. Change that name to reflect the actual creator name you are using.

If you are editing a Prism or PowerMart MX query that targets a database requiring a creator or database name, you must manually insert that name into the `FROM` and `WHERE` clauses.
Creating and Editing Metadata Queries

Use the Metadata Source dialog box to create or edit a metadata query. Make sure you create a query only for the type of metadata you want. For example, if you want metadata only for columns, provide a query only in the Columns Query box; delete the query in the Tables Query box.

You can specify a connection file for the query. This lets you load metadata from a database that is different from the database for a given Design window.

You can edit BI Query’s standard template queries or queries that support other vendors’ metadata tables (IBM’s DataGuide, Informatica’s PowerMart Mx, and Prism’s Directory Manager).

To create a metadata query or edit a template query:

2. From the Vendor drop-down list, select the template query you want to use or edit. (Select Other to use or edit BI Query’s standard template).
3. Do one of the following:
   • Edit the queries in the Columns Query and Tables Query boxes.
   • Type your own queries.

   The text of metadata queries is case sensitive and must match the attributes stored in the metadata table.

4. To submit the query to a DB2, Oracle, or Teradata database requiring an Owner or Creator name, add the name in front of columnmeta in both the FROM and the WHERE clauses.
5. To insert the table variable, «table_name», into the SQL string, click Insert Table Variable.
6. If the Creator name must be included as part of the substitution for the table variable when you load metadata or refresh data objects, click Use Creator Name in Table Variable.

   BI Query uses the DB User ID value (specified in the metadata connection file) for the creator name. Make sure the letter case of the creator name in the connection file matches the case of the creator name in the metadata table.
7. Click Connection to assign a connection file to be used when the metadata source queries are run.
8. Click OK.
Loading Metadata

Once you specify the metadata source for a data model, you can load the metadata in a number of ways. Loading metadata associates the display names and descriptions with the data objects and attributes in the data model.

Both you and your users can load metadata into a data model. As a result, when the metadata changes, you can simply inform users that they need to load the metadata; you do not need to load it into the model and then distribute an updated version.

To load metadata into an existing data model:

1. On the Host menu, click Load Metadata. The Load Metadata dialog box opens.
2. To display all the tables in the data model, click Data Objects From All Windows. Display Names and Short Descriptions are selected by default. If you do not want to load one or the other, click to deselect it.
3. Select the tables you want; then, click Open.

To load metadata into a new data model:

Load tables as data objects into the data model. Metadata is loaded with the tables.

To load metadata into an existing data model and refresh the data objects at the same time:

Refresh data objects in the data model. Metadata is included with the data objects.

To refresh the metadata on startup:

1. On the Tools menu, click Preferences.
2. In the Preferences dialog box, select Refresh Metadata on Startup.
Providing Attribute Descriptions

As an alternative to providing metadata, you can provide descriptions for attributes that can help users decide when and how to use the attributes in a query. The description for a Title attribute can clarify that the attribute represents business titles (Director, Manager, Specialist), not personal titles (Mr., Mrs., Ms). The description for an attribute that is updated only at certain intervals can help users decide when to schedule their queries.

To provide attribute descriptions in BI Query you can use either the Attribute dialog box or the database table.

**Attribute Dialog Box**

You can create attribute descriptions directly in BI Query by typing them into the Attribute dialog box. Descriptions can contain up to about 400 characters, depending on the size and style of the text. Users can see the descriptions you provide by clicking the question mark button beside an attribute in the attribute window. The Attribute dialog box opens, displaying the description.

**Database Table**

In many organizations, the database already contains a catalog table or a data dictionary that stores information about tables and attributes. If you have such a table, you can create a query that retrieves the attribute information from the table, or you can store the information in a data values results file. If such a table does not exist, you can create one. When you create a query or results file that provides a description for the attribute, the Load Comment button is enabled in the Attribute dialog box. When users click the button, BI Query retrieves and displays the description.

Using data values results files for attribute descriptions reduces the load on the database server because BI Query does not need to query the database each time the user clicks the Load Comment button. Instead, BI Query reads the description for a given attribute directly from the results file. However, using a query to generate attribute descriptions provides users with up-to-date information and lets you create one file—the query file—that retrieves information for all attributes for a data object. If you use data values results files, you have to create a file for each attribute you want to describe.

💡 For more information on data values results files, see the BI Query Queries User’s Guide.
Providing Attribute Descriptions

Adding Descriptions to the Attribute Dialog Box

When attribute descriptions are not available in a table, you can create them in BI Query using the **Attribute** dialog box. When users click the question mark for an attribute in an attribute window, the description is displayed.

In a split data model, attribute descriptions you create are stored in the administrator layer. Any attribute descriptions users create are saved with their user layer and take precedence over descriptions stored in the administrator layer. Users do not normally create attribute descriptions from queries to a database table, but they can type them directly in BI Query.

You can also add an attribute at the top of an attribute window to provide a description of the data object. Insert the attribute (for example, “Data Object Description”); then, create its description. The description is saved with the data model.

To create an attribute description in BI Query:

1. Make sure BI Query is not in Design mode.
2. In an attribute window, click the question mark beside an attribute.
3. In the **Attribute** dialog box, type a description into the **Comment** box.
4. Click **OK**.

The Attribute Descriptions Table

If the database contains a catalog table or a data dictionary table from which you want to supply attribute descriptions, ensure that the table contains the following:

- one column of the database names of the attributes
- one column of the database names of the tables that store those attributes
- one column of attribute descriptions

If the database does not contain a table with attribute information, you can create one using BI Query or another tool. Be sure to populate the table with the database names of the attributes and their corresponding tables, not their BI Query names.

BI Query uses the data object you create or load to retrieve the attribute descriptions. Users can also use the data object itself to retrieve attribute descriptions. If the table is large and you want to avoid tying up the database server, you can prevent users from accessing the data object by hiding all of its attributes. (In Design mode, turn off the **Visible** flag for each attribute.)
The basic procedure to supply attribute descriptions from a table is as follows:

1. If the database contains a catalog table or a data dictionary, ensure that it contains a table with attribute information. If the database does not contain an attribute descriptions table, create one using BI Query or another tool. For more information, see “Creating the Attribute Descriptions Table” on page 202.

2. If you are using an existing table or have created your own using a third-party tool, load the table into the data model. For more information, see “Loading Tables from the Database” on page 80.

3. Populate the table with the database names of the attributes and their corresponding tables, not their BI Query names. You can use the update facility in BI Query to populate tables. For more information, see “Populating the Attribute Description Table” on page 203.

4. Create the appropriate query or data values results files. For more information, see “Creating the Attribute Description Query” on page 204 and “Creating Attribute Descriptions Using Results Files” on page 205.

Creating the Attribute Descriptions Table

The table you create needs three columns—one for the database names of the attributes, another for their corresponding table names, and a third for the attribute descriptions.

If you do not need to change the attribute and table names in the table or in BI Query later, you can assign key fields to those attributes. (Both must be key fields if more than one table has attributes with the same name; this lets queries distinguish between the attributes.) If you need to make changes to attribute and table names using BI Query, create another attribute to serve as your key field. Its sole purpose is to let you populate and update the table.

**Comment** box in the **Attribute** dialog box can display descriptions of up to about 400 characters, but your database may not support character fields longer than 256 characters. If you need more than 256 characters for a description, you can type the description across two or more rows. In this case, you do not need to add an additional column to the table to sequence the rows, even though the key fields are no longer unique.
To create the attribute descriptions table:

1. In a Design window, create the data object you want and name it (for example, “Descriptions”).
   
   For more information on creating data objects and adding attributes to them, see “Creating Data Objects” on page 114.

2. Add attributes to the data object, specifying a text or character data type for each attribute:
   - Attribute 1 must be the database name of the attribute (for example, “AttributeName”).
   - Attribute 2 must be the database name of the table (for example, “TableName”).
   - Attribute 3 must be the description of the attribute (for example, “Description”).
     
     If you need to modify attribute or table names later, create a fourth attribute as well (to serve as the key field).

3. Assign one or more key fields. For more information, see “Setting Key Fields” on page 273.

4. Connect to the DBMS; then, on the Host menu, click Create DB Tables.

5. In the Create DB Tables dialog box, select the new data object in the Data Objects list.

6. Click the right arrow button to create the selected object in the database.
   
   (It appears in the Database Tables list.)

7. When the new object is created, click Close.

Populating the Attribute Description Table

Once you create the table in the database, you can populate it using BI Query Admin, BI Query Update, or any other tool that lets you insert data into a table. You need to know the database names of the attributes (and the corresponding data objects) that you want to describe.

For more information on populating tables, see “Adding, Modifying, and Deleting Records” on page 274.

To determine the database name of an attribute:

- Do one of the following:
In the attribute window, click the question mark beside the attribute. The database name appears after the period in the DB Name box (such as store_id in Store.store_id).

In Design mode, click the attribute in the attribute window.

To determine the database name of a data object:
- In Design mode, click the data object; then, on the Design menu, click Edit Database Name.

Creating the Attribute Description Query

Once the attributes description table is populated with data, you can create a query that retrieves the description for each attribute.

For more information on building queries, see the BI Query Queries User’s Guide.

To create the query:

1. Double-click the data object that corresponds to the attribute descriptions table.
2. In the attribute window, select the attribute that represents description information.
3. Qualify the attribute that represents attribute names with the value ^0. Qualify the attribute that represents table names with the value ^1.
4. On the Query menu, point to Save, and then click Data Values Query. The Save Data Values Query dialog box opens.
5. Type query(info) into the File Name box. (Use this exact name, and use lower case.)
6. Click Save.

When users click the Load Comment button in the Attribute dialog box for a particular attribute, BI Query runs the attribute description query, replacing the qualifications ^0 and ^1 with the names of the attribute and the corresponding table, respectively. The query returns the description for the attribute, which BI Query displays in the Comment box.
Creating Attribute Descriptions Using Results Files

Instead of creating a query to retrieve attribute descriptions from the database, you can create a data values results file for each attribute you want to describe. BI Query uses the description in the results file even if you have also created the attribute description query.

To create a data values results file, you create a query that returns the description of an attribute based on the database names of the attribute and the data object that contains it.

💡 For more information on data values results files, see the *BI Query Queries User’s Guide*.

**To create a data values results file for an attribute description:**

1. Double-click the data object that corresponds to the attribute descriptions table.
2. In the attribute window, select the attribute that represents description information.
3. Qualify the attribute that represents table names with the database name of the data object.
4. Qualify the attribute that represents attribute names with the database name of the attribute.
5. On the *Query* menu, click *Run*. The results set displays the description for the attribute.
6. On the *Results* menu, click *Save*. The *Save Results* dialog box opens.
7. In the *File Name* box, type the name of the results file using the following format:
   
   ```
   data_object.attribute(info)
   ```
   
   where *attribute* is the display name of the attribute and *data_object* is the display name of the data object that contains the attribute.
   
   For example, if you are providing a description for the *Order Id* attribute in the *Sales* data object, type *Sales.Order Id(info)*.
8. Click *Save*.
9. Repeat steps 1–8 as necessary to provide descriptions for other attributes.
Chapter 9: Providing User Help

When users click the **Load Comment** button in the **Attribute** dialog box for the attribute, BI Query retrieves the description from the results file and displays the description in the **Comment** box.

You can also use a text editor to create the data values results file for an attribute description. In this case, the results file contains a single row of data (the attribute description). For more information on creating results files with a text editor, see the *BI Query Queries User’s Guide*.

### Annotating Existing Help

You can add information directly to a Help topic as an annotation. When you add an annotation, and then close the annotation window, a paper clip icon 👤 appears in the topic you annotated. Users can view the annotation by clicking the paper clip. The annotation is saved with the data model.

**To add information to an online Help topic:**

1. On the **Help** menu, click **Contents**.
2. In BI Query **Help** window, display the topic you want to annotate.
3. On **Edit** menu of the **Help** window, click **Annotate**.
4. In the **Annotate** dialog box, type the information you want.
5. Click **Save**; then, close the **Help** window.

### Creating Additional Help Files

When you want to provide more information than is appropriate for annotations in the online Help, you can create your own Help files. To do so, you need the following tools:

- A Help compiler. See your software vendor for commercially available applications.
- A text editor that can save `.rtf` (Rich Text Format) files, such as Microsoft Word.

The Help compiler translates the `.rtf` files into the required format. The files you create must follow the format for Help files. (For instructions on using the Help compiler, see the documentation for that product.)
After you have created your Help files, place them in the folder that contains either the data model (for a combined data model) or the administrator layer (for a split data model).

**Data Model Help**

You can provide additional Help on the data model you are using. If you name the Help file after the data model, when users run BI Query, it finds this Help file and adds a command to the `Help` menu. (For example, if a data model is called `mymodel.gql`, call the Help file `mymodel.hlp`. Running BI Query adds `mymodel Help` to the `Help` menu.) When users select this command, BI Query displays the online Help you have provided.

**Site-Specific Help**

You can also provide additional Help about your organization. If you name the Help file `Site.hlp`, BI Query adds the corresponding command to the `Help` menu. Users can access Site Help from the `Help` menu.
Chapter 10

Saving and Distributing Data Models

This section provides information on the following:

- “Saving Data Models” on page 209
- “Distributing Combined Data Models” on page 211
- “Distributing Split Data Models” on page 212
- “Methods for Distributing Data Models” on page 213
- “Distributing Revisions by Saving to the Database” on page 216

Saving Data Models

You can save a data model as a split model or a combined model:

- A **split data model** is recommended for organizations that let users make changes to the data model.
- A **combined data model** is designed for sites where users are not allowed to make changes. (You can prevent users from changing the data model by not assigning related permissions.)

To change a combined data model into a split one, assign the **Save Split Data Model** permission. To change a split data model into a combined one, remove the same permission.
Before you save a data model, make sure it is no longer in Design mode; otherwise, users have to switch to Run mode to build their queries.

💡 For more information on combined and split data models, see “Types of Data Models” on page 22.

### Locations for Saving Data Models

You can save combined or split data models to the following locations:

- A folder on your computer or on a networked corporate server.
- The database, if you have the **Save to Database** permission. After you make your first revisions to a model, saving a data model in the database is a means of distributing the revisions. When anyone using that model connects to the database, a message indicates that there is a new version of the data model available.

💡 For more information on saving models to the database, see “Distributing Revisions by Saving to the Database” on page 216.

You can also store in the database any reports that you created using BI Query Reports and saved with the data model in the Reports folder.

### Saving Split Data Models

Use the **Save Data Model** dialog box to save split data models. When you set the file type to **BI Query Data Model**, only the user layer file (with a `.gqu` extension) appears in the dialog box. This is because the data model can be opened only from the user layer. You can verify that the `.gqa` file has been saved by selecting **All Files** in the **Save as Type** drop-down list box.

**To save a split data model:**

1. In BI Query Admin, on the **Tools** menu, click **Permissions**.
2. In the **Permissions** dialog box, select the **Save Split Data Model** permission; then, click **OK**.
3. On the **File** menu, click **Save** and save the data model in its own folder.

💡 Be sure to keep a copy of the empty user-layer file; you need this file to make revisions to the data model.
The first time you save a split data model, BI Query Admin saves the data model in an administrator-layer file and an empty user-layer file. It also creates the Queries, DataVals, and Reports folders. (If you saved queries, data values files, or reports before saving the data model, BI Query already created the folders.)

Before you distribute a split data model, you should store its administrator and user layers in separate folders. For more information, see “Storing Each Layer in Separate Folders” on page 212.

**Saving Combined Data Models**

When you do not want users to make changes to the data model, save it as a combined data model. Saving a combined data model saves it as a single file with the extension `.gql` and creates the Queries, DataVals, and Reports folders.

**To save a combined data model:**

1. In BI Query Admin, on the **Tools** menu, click **Permissions**.
2. In the **Permissions** dialog box, ensure that the **Save Split Data Model** check box is not selected; then, click **OK**.
3. On the **File** menu, click **Save**. BI Query saves the data model as one file with the extension `.gql`.

**Distributing Combined Data Models**

When you distribute a combined data model to users, you distribute the data model file (.gql) and its Queries, DataVals, and Reports folders. When you have more than one combined data model, store each one in its own folder.

When users open data model files with BI Query User or BI Query Update, they see the core BI Query application. Typically, users are not given permission to add their own customizations to a combined data model because the changes are overwritten each time you distribute a new version of the data model.

💡 For more information on combined data models, see “Combined Data Models” on page 25.
Distributing Split Data Models

When you distribute a split data model to users, distribute the administrator-layer file (\*.gqa), its Queries, DataVals, and Reports folders, and the empty user-layer file (\*.gqu) according to the setup at your site:

- If you are storing the data model on a file server, store the administrator-layer file on the server, then distribute the empty user-layer file to each user.
- If you are storing the data model on users’ computers, distribute a copy of the administrator-layer file and the empty user-layer file to each user.

Before you distribute a split data model, keep a copy of the empty user-layer file. You need this file to revise the data model.

When a user opens the user-layer file with BI Query User or BI Query Update, the path it contains to the administrator-layer file lets BI Query find the administrator-layer file. BI Query then combines all the elements stored in both files. When the user-layer file is stored in a different folder than the administrator layer, BI Query creates empty Queries, DataVals, and Reports folders the first time the user-layer file is opened.

For more information on split data models, see “Types of Data Models” on page 22.

Storing Each Layer in Separate Folders

It is good practice to store user and administrator layers in separate folders regardless of the method used to distribute a split data model. This strategy prevents user files from being inadvertently overwritten by administrator files when you distribute new versions of the administrator layer. It also prevents administrator files from being overwritten by user files when users create and modify their own files.

If you are distributing only the .gqa file (no associated files such as queries or reports), you can store the administrator and user layers of a split data model in the same folder.

When you have more than one split data model, store each one in its own folder with their corresponding administrator and user layers stored in separate folders.
Establishing a Path to the Administrator-Layer File

If you store the administrator and user layers of a split data model in separate folders, users must reestablish the path to the administrator layer when they open the user-layer file.

You can eliminate this step by establishing the path for users before you distribute the user-layer file. Make sure you have write access to the folders containing the administrator and user layers. When users open the user-layer file, BI Query finds the administrator layer and combines the contents of the two files.

To establish the path to the administrator-layer file:

1. Do one of the following:
   - If you are distributing using diskettes or e-mail, store the administrator- and user-layer folders on your own computer in the same location as they would be stored on each user’s computer (for example, in C:\BIQuery\Admin and C:\BIQuery\User).
   - If you are distributing using the server, make sure you have the write access you need. Copy the empty user-layer file to a write-protected folder on the server and copy the administrator layer and associated folders to a separate write-protected folder.

2. Open the user-layer file and reestablish the path to the administrator-layer file. For more information, see “Finding the Administrator Layer in a Split Data Model” on page 34.

3. Save the data model to save the new path with it.

Methods for Distributing Data Models

Depending on how your site is set up, you can distribute the initial version of a data model as well as subsequent revisions using any of the following methods:

- Store it on a file server. For more information, see “Distributing with File Servers” on page 214.
- Distribute it on diskettes. For more information, see “Distributing by Diskette” on page 215.
- Distribute it by e-mail. For more information, see “Distributing by E-Mail” on page 215.
Chapter 10: Saving and Distributing Data Models

- Publish it to the BI Server Repository (if your querying environment includes BI Server and you have appropriate permissions). For more information, see “Publishing Data Models with BI Server” on page 223.

You can also distribute revisions using the database itself. For more information, see “Distributing Revisions by Saving to the Database” on page 216.

Once you distribute the data model, users can open data models from their own computers, from a file server, or from the BI Server Repository (if they have appropriate permissions).

Distributing with File Servers

If users have access to a common file server, you can use it to distribute a data model. When you distribute revisions to a split data model, the path it contains to the administrator-layer file lets BI Query open the revised version.

To distribute a combined data model from a file server:

- Copy the model to a write-protected folder on the server. Include the Queries, DataVals, and Reports folders.

To distribute a split data model from a file server:

1. Copy the empty user-layer file to a write-protected folder on the server and the administrator layer to a separate write-protected folder. Include the Queries, DataVals, and Reports folders.

2. E-mail the user layer to users or distribute it by diskette. Users can also copy the user layer to their own computers from the server.

3. Instruct users that after they have reestablished the link to the administrator-layer file, they should save the user-layer file on their computers to save the link. When users open the user-layer file, BI Query will find the administrator layer on the server and combine the contents of the two files.

You can also establish the new link for your users before you copy the data model. For more information, see “Establishing a Path to the Administrator-Layer File” on page 213.

When you revise the model, copy the revised model (or, if the model is split, the administrator layer) and associated folders to the file server. Alternatively, save the revisions to the database.

For more information, see “Saving Revised Models to the Database” on page 220.
Methods for Distributing Data Models

Distributing by Diskette

If you have a small number of workstations, it may be practical to distribute a data model using diskettes.

To distribute a combined data model using diskettes:

1. Copy the model along with the Queries, DataVals, and Reports folders onto the diskettes.
2. Copy the contents of the diskettes to the user computer.

To distribute a split data model using diskettes:

1. Copy the empty user-layer file to one folder on the diskette and the administrator layer, including the Queries, DataVals, and Reports folders, to a separate folder on the diskette.
2. Copy the contents of the diskette to each computer. Make sure the two layers remain in separate folders.
3. Open the user-layer file on the user computer. When prompted to reestablish the link, locate the administrator-layer file on the user computer and reestablish the link. Then save the data model to save the new path with it.

You can also establish the new link for your users before you copy the data model. For more information, see “Establishing a Path to the Administrator-Layer File” on page 213.

When you revise the data model, copy the revised model (or, if the model is split, the administrator layer) and associated folders to a diskette; then, copy the contents of the diskette to the appropriate folders on the user computer. Alternatively, save the revisions to the database.

For more information, see “Saving Revised Models to the Database” on page 220.

Distributing by E-Mail

If you have hundreds of users with access to different file servers and you want to save the work of distributing a data model to each server, you can distribute the models by e-mail instead. You can send e-mail directly from BI Query using any e-mail system that supports the Microsoft® Messaging API (MAPI), such as Microsoft® Mail.
To distribute the data model by e-mail:

1. On the **File** menu, click **Send**.

2. In the mail editor that opens, type in the recipient, subject, and message information as necessary.

3. If the data model is a split model, instruct your users to do the following:
   a. Store the administrator layer and associated folders in one folder and the user-layer file in another folder.
   b. Open the user-layer file. When prompted to reestablish the link, locate the administrator-layer file on the user computer and reestablish the link. Then, save the data model to save the new path with it.

   You can also establish the new link for your users before you send the data model. For more information, see “Establishing a Path to the Administrator-Layer File” on page 213.

4. Attach the data model files to the message; then, send the message.

When you revise the model, mail the revised model (or, if the model is split, the administrator layer) and associated folders to your users and have them copy the new material to their computer. Alternatively, save the revisions to the database.

   For more information, see “Saving Revised Models to the Database” on page 220.

**Distributing Revisions by Saving to the Database**

If your site has hundreds of users, or if you are running BI Query on a distributed network with offices located in different cities but sharing a common database, you need an efficient way of distributing revisions to the data model. If your site is set up to provide access to the BI Server Repository, this is the simplest and most efficient way to distribute the data model and any subsequent revisions.

If you do not have access to the BI Server Repository, you can distribute revised data models by saving them to the database. This mechanism is strictly for distributing revised versions of the data model; it cannot be used for initial distribution. However, even if you are not planning to revise the data model right away, it is good practice to store it in the database before you initially distribute it. This creates a model ID in the data model, which is required when you distribute revisions to the data model using the database.
If you store BI Query reports in a Reports folder with the data model, you can store the reports in the database too.

To save a revised version of the data model in the database, you simply save the data model in the database. For split data models, this operation saves the administrator layer (with the new ID and version number) and user layer on your computer, and then stores the administrator layer in the database. (If you specify that BI Query should also save queries, data values, and BI Query reports, they are also stored in the database.)

Combined data models and the administrator layer of split models store the model ID of the data model, its version number, and a flag that indicates whether the model has been saved in the database.

When users run BI Query User or BI Query Update and open the data model, the administrator-layer file (or the combined data model) checks for a new version in the database. If one is stored there, users have the option of loading it. If they do load it, BI Query saves the earlier version with the extension .bak.

**Save-to-Database Tables**

The first time you store the data model in the database, up to five tables are created in the order listed below:

<table>
<thead>
<tr>
<th>Table</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>GQLMnnnn</td>
<td>Information about the administrator-layer file of a split data model (or a combined data model).</td>
</tr>
<tr>
<td>GQLVnnnn</td>
<td>Contents of the data values files (optional).</td>
</tr>
<tr>
<td>GQLQnnnn</td>
<td>Contents of the query files (optional).</td>
</tr>
<tr>
<td>GQLRnnnn</td>
<td>Contents of the BI Query report files (optional).</td>
</tr>
<tr>
<td>GQLMETA2 (a metatable)</td>
<td>Model ID of the data model, its version number, the names of tables storing data for the data model, and other information. (To view the model ID in BI Query, on the Tools menu, click Permissions; then, click Model Info.)</td>
</tr>
</tbody>
</table>
Chapter 10: Saving and Distributing Data Models

The \textit{nnnn} string in the table names represents digits. All of the table names for a data model use the same digits. Each table stores the ID of the model plus other pertinent information.

The GQLMETA2 table is created only if it does not already exist in the database. All data models that are saved in the database use the same GQLMETA2 table.

The data in these tables should not be modified in any way other than by saving revised versions of a data model in the database using BI Query Admin.

💡 For more information on these tables, see BI Query Admin Help.

**Attributes in the GQLMeta2 Table**

The GQLMETA2 table contains the following attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GQLMODEL</td>
<td>The data model ID.</td>
</tr>
<tr>
<td>GQLVERS</td>
<td>The data model version number.</td>
</tr>
<tr>
<td>GQLMAND</td>
<td>Controls whether BI Query displays a message to indicate the downloading of a new version of the data model is optional or mandatory. (The message is for users’ information only; they can still choose not to download.)</td>
</tr>
<tr>
<td>GQLMTAB</td>
<td>The name of the table that stores the data model.</td>
</tr>
<tr>
<td>GQLDVTAB</td>
<td>The name of the table that stores the data values.</td>
</tr>
<tr>
<td>GQLQRYTAB</td>
<td>The name of the table that stores queries.</td>
</tr>
<tr>
<td>GQLREPTAB</td>
<td>The name of the table that stores BI Query Reports.</td>
</tr>
<tr>
<td>GQLREASON</td>
<td>The description of the changes contained in the new version of the data model.</td>
</tr>
</tbody>
</table>

Each time you store a new version of the data model in the database, the GQLVERS column in the GQLMETA2 table is updated with the new version number. The GQLMeta2nnnn table is dropped and replaced with another table (with new digits). If the Queries, DataVals, and Reports folders are also stored in the database, the old GQLQnnnn, GQLVnnnn, and GQLRnnnn tables are dropped, and new tables are created.
Attributes in Other Save-to-Database Tables

With the exception of the GQLMETA2 table, the save-to-database tables that BI Query creates all have the following attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GQLMODL</td>
<td>The data model ID.</td>
</tr>
<tr>
<td>GQLFID</td>
<td>A number identifying the query or data values file.</td>
</tr>
<tr>
<td>GQLSEQ</td>
<td>A number identifying each record in the query or data values file.</td>
</tr>
<tr>
<td>GQLDLEN</td>
<td>The length of the data in the record.</td>
</tr>
<tr>
<td>GQLDATA</td>
<td>The actual data in the record.</td>
</tr>
</tbody>
</table>

Before creating the save-to-database tables, the data for the GQLDATA columns is converted from binary format to ASCII format to be stored as a varchar. The data in the other columns does not need to be converted.

Granting Permissions to Save-to-Database Tables

To enable users to use a data model stored in the database, you must ensure that they have read access to, or SELECT permission on, the save-to-database tables BI Query creates.

If users are granted access on a table-by-table basis rather than automatic access to all new tables, you need to determine the names of the new tables. You can do that by creating a new data model, loading the GQLMETA2 table, and then using the corresponding data object to query the table and retrieve the table names.

Once you know the table names, you (or the database administrator) can grant users access to the tables by sending an SQL script to the DBMS using BI Query; alternatively, use a front-end DBMS tool.

Saving Initial Models to the Database

When you save the initial version of the data model to the database, you can minimize the size of the model by creating an empty data model with a model ID. Then, include the content in the data model and save the revised version in the database.
To save the initial version of a data model to the database:

1. Run BI Query Admin.
2. In the **Welcome to Hummingbird BI Query** dialog box, click **New Data Model**.
3. In an empty **Design** window, connect to the DBMS.
4. On the **File** menu, click **Save**.
   
  💡 Before you can save a model as a split data model, the **Save Split Data Model** permission must be assigned.

5. On the **File** menu, click **Save To Database**. The **Save to Database** dialog box opens.
6. In the **ID** box, type a model ID for the data model.
7. Click **Save**.
8. Distribute the resulting data model file.
   
  💡 For more information, see “**Methods for Distributing Data Models**” on page 213.

9. Revise the data model to include the actual content, and then save the revised version in the database.

**Saving Revised Models to the Database**

Saving a revised data model to the database creates the BI Query database tables. Each time you save a revised version of the data model in the database, BI Query deletes any older version that exists there along with the tables it created.

💡 For more information on these tables, see “**Save-to-Database Tables**” on page 217.

To save a revised data model to the database:

1. Save the data model. For more information, see “**Saving Data Models**” on page 209.

2. Make sure that you have the appropriate database permissions and that users are granted permissions to the save-to-database tables that BI Query creates. For more information, see “**Granting Permissions to Save-to-Database Tables**” on page 219.

3. If the data model is not in Connect per Query Mode, connect to the database. Ensure that the model is not in Design mode.
4. On the File menu, click **Save To Database**. In the **Save To Database** dialog box, specify the options you want. You can view the current values for these options by clicking **Model Info** in the Permissions dialog box.

    For more information on the Save to Database options, see BI Query Admin Help.

5. To specify additional information, click **Options**; then, do the following:
   a. In the **Save To Database Options** dialog box, specify the options you want.

    For more information on these options, see BI Query Admin Help.

   b. Click **OK**.

6. In the **Save To Database** dialog box, click **Save**.

7. If the **Save Data Model** dialog box opens, specify a name and location for the local version of the data model; then, click **Save**.

8. Test the data model to be sure the new version has been saved in the database. (In BI Query User or BI Query Update, open your empty user-layer file and connect to the database. Click **Yes** when you are asked whether you want to load the new version.)
Chapter 11

Publishing Data Models with BI Server

This section provides information on the following:

• “Publishing with BI Server” on page 223
• “Controlling Access with BI Server” on page 229

Publishing with BI Server

If your querying environment includes BI Server, and you have the appropriate system permissions, you can publish a data model to the BI Server Repository.

Requirements for Publishing Data Models

In order to publish a data model, you must first save it locally. You must also log onto BI Server and have the appropriate BI Server and BI Query permissions.

Required BI Server Permissions

The BI Server administrator assigns the system permissions that allow you to publish. To publish a data model, you need at least one of the following BI Server permissions:
Chapter 11: Publishing Data Models with BI Server

- **Publish Data Model User Layer.** If you have this permission and you are using BI Query User or BI Query Update, you can publish your user layers to the Repository.

- **Publish Data Model Admin Layer.** If you have this permission and you are using BI Query Admin, you can publish either or both layers of a split data model to the Repository; you can also publish a combined data model.

  For more information, see the *BI Server Administrator’s Guide*.

**Required BI Query Permissions**

To publish a data model, you need the following BI Query permissions:

- **Edit Data Model**
- **Save Data Model**

  If you are an administrator using BI Query Admin to publish a split data model, ensure the following:

  - You have assigned the required BI Query permissions in the data model.
  - The BI Server administrator has assigned the **Publish Data Model User Layer** permission to your users.

  For more information on these permissions, see “Setting BI Query Permissions” on page 183.

Once you save a data model locally and publish it, queries based on that data model can also be published. For more information on publishing and retrieving queries and data values files, see the *BI Query Queries User’s Guide*.

**Data Model Packages**

Publishing a data model creates a package in the BI Server Repository. A package contains the data model and its associated queries, BI Query reports, standard report specifications, data values, data sets, and connection files. (Results files are not published with a data model.) The package has the same name as the data model.

BI Query creates and then increments a version number each time you publish the data model, so that you do not overwrite earlier versions. Each time you publish, BI Query also saves the data model locally with the incremented version number.
When you publish a data model, BI Query automatically publishes the queries, connection files, and data sets, and provides you with the option of also publishing the reports. In order for these files to be published with the data model, they must be stored in the appropriate folders within the local data model folder:

- Queries and standard report specifications must be saved in the Queries folder and its subfolders.
- Connection files must be saved in the connection folder for the data model. Use the Connections dialog box to indicate which connection files you want to publish with the model.
- Data sets must be saved in the DataVals folder.
- BI Query reports must be saved in the Reports folder. (You can save reports in any location. However, to use the BI Server Repository to distribute reports with a data model, the reports must reside in the Reports folder.)

Check the PublishLogger.txt file to verify that all reports were published. (The file is located in the Windows temporary folder as defined in your system environment variables.) Check the file as soon as possible after publishing the data model because it is available only until BI Query Reports is next opened.

**Publishing Roadmap as XML**

If you want, in addition to the information normally published, you can have BI Query also publish the data model roadmap as XML. The XML lists all the data objects in each Design window, and all the attributes within each data object. You can then examine the dependencies within the data model using Impact Analysis in Genio.

If you enable this option, the time required to publish data models will increase significantly.

**To enable publishing the data model roadmap:**

1. Open the biq.ini file (located in the \BI\Query folder).
2. Edit the PublishRoadmapXML setting to Yes.
3. Save and close the biq.ini file.

**Preparing to Publish Data Models**

Before publishing a data model, make sure you do the following:
• If the queries and reports in the data model will be scheduled to run automatically, save the data model with a default connection file. Use the Connections dialog box to indicate which other connection files you want to publish with the model. Without published connection files, BI Query users who run queries from the data model must supply their own connection files; BI Web users, who can’t create their own connection files, will not be able to run queries from the model at all.

For more information, see “Publishing Connection Files” on page 53.

• Save the data model with a name that others will recognize when they retrieve it from the Repository. The name you give the data model is the name that others will see whether they open the data model in BI Query or in BI Web.

• Verify that the associated data model files are stored in the appropriate folders. For more information, see “Data Model Packages” on page 224.

• Set the user and group access for the model and its components. (On the Tools menu, click Set Security.) For more information, see “Setting Security on Published Items” on page 232.

If you want to secure your data model using the Repository, make sure the data model isn’t also available from a corporate file server. Your security settings will apply only to users who access the data model using the Repository.

• If you also want to publish the data model roadmap as XML, verify that PublishRoadmapXML=Yes in the biq.ini file.

Publishing Data Models

You can publish the user layer of a split data model from any BI Query application. If you're using BI Query Admin and you have the appropriate permissions, you also can publish the administrator layer of a split data model or a combined data model. BI Server lets you grant permission to the users you want to access the model.

For information on retrieving data models, see “Retrieving Data Models” on page 33.
To publish a data model to the Repository:

1. Log onto BI Server. For more information, see “Working Online or Offline” on page 34.

2. Do one of the following:
   - Open a saved data model.
   - Create or edit a data model; then, save it.


4. In the Description box, type information that will help others identify the data model.

5. In the Also Publish area, specify whether you want to also publish the reports with the data model.

6. To require others to retrieve the latest version of the model published, select Required in the Download area.

7. To publish the data model to a folder other than the one shown in the Folder box, do the following:
   a. Click the Browse button. The Select Folder dialog box opens.
   b. To create a new folder, select the folder under which you want the new folder to be created. Click the New Folder button and enter a name for the new folder; then, click OK.
   c. Select the folder you want.
   d. Click OK.

8. To set security for the data model and its components, click Set Security; then, use the Set Security dialog box to grant or deny access to the model. For more information, see “Setting Security on Published Items” on page 232.

9. In the Publish dialog box, click Publish.

Setting Data Model Visibility

When you publish a data model, you can grant or deny access to it to certain users and groups. You can also set the “visibility” of the model for BI Web users. If you make a data model “invisible” but grant access to it, the data model itself won’t appear in the BI Web Portfolio. As a result, BI Web users won’t be able to open the data model. However, if they have the Refresh permission for any BI Query reports stored in the data model package, they will still be able to run and refresh those reports.
When you use BI Query Reports to publish reports for a published data model, you can set the Refresh permission for selected users and groups. For more information, see the BI Query Reports User’s Guide.

**To set the visibility for a data model:**

1. On the File menu, click Publish.
2. In the Publish dialog box, click Set Security. The Set Security dialog box opens.
3. From the Data Model list, select the data model; then, grant or deny access to the data model to the appropriate BI Web users and groups.
4. From the Data Model list, expand the Permissions folder. Select Data Model Is Visible in Portfolio.
5. To make the data model available to certain BI Web users in their Portfolios, select the appropriate users or groups from the Users and Groups list; then, click Grant.
6. To make the data model invisible to certain BI Web users, select the users or groups from the Users and Groups list; then, click Deny.
   If these users have access to the data model and Refresh permission for a report, they will be able to refresh the report even though the data model does not appear in their Portfolio.
7. Click OK.
8. In the Publish dialog box, click Publish.

**Granting BI Web Ad Hoc Querying**

You can grant or deny to BI Web users the ability to create ad hoc queries. You can set this permission in the Set Security dialog box when you publish a data model. If you do not grant this permission, BI Web users of the model cannot do any of the following:

- open data objects or published queries
- create their own queries or run published queries
- edit standard reports or BI Web reports
- save BI Web reports

Users who do not have this permission can still run queries that are linked to buttons in the data model.
To control BI Web ad hoc querying:

1. On the File menu, click Publish.
2. In the Publish dialog box, click Set Security. The Set Security dialog box opens.
3. From the Data Model list, select the data model; then, grant or deny access to the data model to the appropriate BI Web users and groups.
4. From the Data Model list, expand the Permissions folder. Select Allow Web Ad Hoc Queries.
5. To grant ad hoc querying to certain BI Web users, select the appropriate users or groups from the Users and Groups list; then, click Grant.
6. To deny ad hoc querying to certain BI Web users, select the users or groups from the Users and Groups list; then, click Deny.
7. Click OK.
8. In the Publish dialog box, click Publish.

Controlling Access with BI Server

For any item you publish to the BI Server Repository, there may be people in your organization who need access to the information contained in it, people who need customized views of it, and people who should not see it at all. You must secure the data so that it can be viewed or changed only by the people with the authority to do so.

The Set Security dialog box lets you control individual as well as group access to the information you publish; you can grant or deny general access and provide individual exceptions as necessary. You exercise this control based on a structure of users and groups that the BI Server administrator creates.

Depending upon your needs, the administrator may make it possible for you to grant system permissions to other users, such as the ability to schedule queries and reports. For information on these permissions and on how they are assigned, see the BI Server Administrator’s Guide.
Users and Groups in BI Server

Your BI Server logon user name and password identify you as a user in a structure of users and groups created by your BI Server administrator. The administrator assigns system permissions that determine the extent to which you can use the features of BI Server, such as the ability to publish and retrieve items and schedule the items you publish.

Using the user and group structure provided by your BI Server administrator, you can grant or deny access to the items you publish. If you find that you cannot use features or perform activities appropriate to your work, ask your administrator to review the system permissions that have been assigned to you.

Access Inheritance Rules

In the BI Server security structure, users inherit group access according to the following rules:

<table>
<thead>
<tr>
<th>Inheritance Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group members inherit their group’s access.</td>
<td>When you grant or deny access to a group, all members of the group inherit that setting (if a group has access, each member has access and if a group is denied access, each member is denied access).</td>
</tr>
<tr>
<td>Members of more than one group inherit from the groups that grant access.</td>
<td>If a user belongs to one group that is granted access and to another group that is denied access, the user has access.</td>
</tr>
<tr>
<td>Member security settings override inherited group settings.</td>
<td>You can give a group member a setting different from that of the member’s group. When you do, the overriding setting always applies to that member, even when you extend the membership to other groups. You can remove an overriding setting to let a member inherit from the group again.</td>
</tr>
</tbody>
</table>
The Set Security Dialog Box

You can set security for data models, connection files, queries, reports, and other data model components using the **Set Security** dialog box. This dialog box is available from any BI Query dialog box that lets you save or publish items. In BI Query Admin, you can also open the dialog box by clicking **Set Security** on the **Tools** menu.

The **Set Security** dialog box organizes the components of the data model into the following folders:

**Connections**
Lists the connection files that have been selected for publishing and the default connection file (if it exists).

**DataVals**
Lists the data values files associated with the data model.

**Permissions**
Lists the data model permissions (such as Save Data Model and Edit Data Model).

**Queries**
Lists the queries associated with the data model.

**Views**
Lists the data model **Design** windows and the data objects, object qualifiers, attributes, relationships, buttons, and ornaments that the windows contain.

You can set security for the individual items within the folders, although you cannot directly grant or deny access to the folders themselves; they inherit the access granted or denied to the data model. You also cannot deny yourself access.

The symbols that indicate access settings in the **Set Security** dialog box are as follows:

- Green light: access specifically granted.
- Red “no entry” symbol: access specifically denied.
Chapter 11: Publishing Data Models with BI Server

Setting Security on Published Items

You can set security using one of the following methods:

**By Item**

In this case, you select the item, and then grant or deny access to that item for particular users and groups.

**By User and Group**

In this case, you select a user or a group first, and then select the items and grant or deny access to those items.

**To set security:**

1. Open the Set Security dialog box. The Set By Item page opens by default.

2. Do one of the following:
   - To set security by item, select the item from the Data Model list. From the User and Groups list, select the users and groups for whom you want to grant or deny access.
   - To set security by user and group rather than by item, click the Set By User and Group tab. From the Users and Groups list, select the user or group for whom you want to grant or deny access. From the Data Model list, select the items that you want to secure.

3. Do one of the following:
   - To grant access, click Grant.
   - To deny access, click Deny.
   - To allow a user or group to inherit security from the group of which it is a member, click Inherit.

4. When you have set security for as many items and as many users and groups as you want, click OK.

5. Publish the item.

- Green check mark: a granted access that has been inherited.
- Red x: a denied access that has been inherited.
This section provides information on the following:

- “Methods for Balancing Network Load” on page 233
- “Running Efficient Queries” on page 234

**Methods for Balancing Network Load**

The load on the network database server varies with the number of users running BI Query, how often they query the database, and the size and complexity of their queries. BI Query has many features that let you adjust and redistribute the load. If the network is slow, choose those features that minimize data transfer. If the database server is overloaded, choose those that minimize database requests and process results on the user’s machine.

💡 Modifications to the DBMS can also have a considerable impact on network load. For more information, see “Optimizing the Database” on page 70.

BI Query reduces the demands on the server by storing query results on the user’s computer so that users can manipulate results locally instead of in the database. You can take additional steps to optimize how data is processed and ensure the best performance for your site:
Chapter 12: Balancing Network Load

- You can ensure that your connection to the DBMS optimizes the process of running queries and retrieving data. For more information, see “Optimizing Connections with Row Buffering” on page 46.
- You can ensure that you run efficient queries.
- You can set limits on the size of results sets returned by queries, or the length of time queries run.
- In a BI Server environment, you can monitor and manage the load on the database server by scheduling queries. For more information on scheduling queries, see the BI Query Queries User’s Guide.

Running Efficient Queries

BI Query provides a number of features that help you ensure efficient query execution.

Enforcing Connect per Query

To run queries, users need to connect to the DBMS. The larger the number of users who connect and the longer they are connected, the greater the load on network resources as the server authenticates each user, writes log or audit records, starts a new task to handle each user, and so on.

You can minimize the resources being held by assigning the Enforce Connect per Query permission. This forces BI Query to connect only for the duration of a query—connecting each time a user runs a query and disconnecting once the results are returned. If you don’t assign this permission, users can take advantage of this facility themselves by setting the Connect per Query preference.

💡 For more information on permissions, see “Setting BI Query Permissions” on page 183.

Setting up BI Query to connect in this manner is useful when query traffic is heavy or users don’t need regular access to the database. For users who require quick response, consider setting the Connect Automatically preference to connect for the duration of a session instead. Enforce Connect Per Query is automatically set when you assign the Connect per Window permission.

💡 For more information, see “Setting Automatic Connections” on page 50.
Assigning the **Enforce Connect per Query** permission is recommended for sites that have concurrent user licenses for their database (licenses that permit only a certain number of users to connect at one time) and are reaching their maximum number of users. It isn’t recommended for sites where users run queries throughout the day because connecting and disconnecting from the database with every query takes up time and server resources.

**Qualifying with Data Values Files**

You can make qualifying a query more efficient by using data values files. BI Query offers the user a choice of data values to restrict query results. By default, BI Query requests all distinct values from the database when showing the user data values. There are two kinds of data values files:

- Data values query files retrieve a subset of values for an attribute from the database.
- Data values results files display a stored set of values; they further reduce the load on the database server because they eliminate the need to query the database each time users need a list of values.

💡 For more information on data values files, see the *BI Query Queries User’s Guide*.

You can also improve query efficiency by limiting access to data values for particular attributes.

💡 For more information, see “Securing Access to Data Values” on page 177.

**Restricting Queries**

You can qualify data objects and actual relationships in order to restrict users’ queries to specific information in the corresponding tables. This helps improve query performance by preventing users from retrieving more data than they actually need.

For example, you can qualify the Customer data object with the value Canada on the Country attribute so that any query that includes that data object retrieves only information about customers in Canada. Alternatively, you can qualify a data object with a prompt that forces users to qualify any query that includes the data object—for example, specifying one or more countries.
Users can further restrict queries using dynamic relationships (providing you have assigned the **Dynamic Relationships** permission and they have set the **Allow Dynamic Relationships** preference).

You should not assign the **Allow Unconnected Data Objects** permission unless absolutely necessary. This allows users to produce a Cartesian product and generates a large number of rows. Most users will never need to do this.

### Submitting Queries to Database Tables

The greater the number of tables users include in a query, the greater the number of joins and rows the DBMS has to process. If users regularly retrieve information from multiple tables, you can improve performance by letting them summarize the information they need in their own tables in the database. The database administrator also has to grant permission to create tables.

💡 For more information on submitting queries to tables, see the *BI Query Queries User’s Guide*.

Summarizing information in one table eliminates the joins and reduces the number of rows that the DBMS has to process. It gives users faster access to the information and reduces the load on the database server.

Users formulate a query, send the results to a new table in the database, and then use the table to formulate additional queries. You can assign the **Submit to Named Table** permission to let users name the new table as well as the corresponding data object.

You can also increase efficiency by creating query data objects that create views instead of temporary tables. These are more efficient than super queries or queries that submit to table because the intermediate data stays in the database server.

💡 For more information, see “Query Data Objects (QDOs)” on page 108.

### Modifying the SQL to Improve Query Efficiency

When users formulate a query, the corresponding SQL string generated by BI Query is visible in the **Query** window. Advanced users who understand SQL and the data in the database can recognize opportunities to make queries run more efficiently by modifying the SQL string. You can allow users to modify the SQL string by assigning the **Edit SQL String** permission and setting the **Show SQL String** preference. Users can then modify the **SELECT** statement (within the limits placed on users by the DBMS).
While the length of the SQL statement depends only on available memory, when you edit it in the Query window, it accepts up to 65,535 bytes. This allows you to create queries with thousands of data values in qualifications and large numbers of calculated attributes. Note, however, that length may also be restricted by the specific DBMS you are using.

Users with the Edit SQL String permission can also add prefixes and suffixes to the SQL string to improve query efficiency. For more information, see the BI Query Queries User’s Guide.

Setting Query Limits with Governors

Depending on the data they need or the type of analysis they want to perform, certain users may need access to more data than others. You can minimize the load on the network and the database server by using the Rows and Time Governor permissions.

Rows Governor
Specifies the maximum number of rows to be returned from a query.

Time Governor
Specifies the maximum time a query can run before it is automatically stopped.

Users can also set Rows and Time Governor preferences, which let them specify maximum values for rows retrieved and elapsed query time. When the specified row or time limit is reached, BI Query prompts users to continue or stop the query. Users can also apply these preferences to a particular connection file. The Rows and Time Governor permissions override the corresponding preferences.

For more information on Governor preferences, see “Specifying Governor Settings with Connection Files” on page 45.

To set query limits using permissions:

1. On the Tools menu, click Permissions.
2. In the Permissions dialog box, in the Governors area, select the Limit option buttons and type the values you want into the corresponding boxes under Rows and Time.
3. Click OK.
Estimating Database Resources

If you’re using a DB2 or Teradata database, you can obtain information about a standard query before you send it. This information includes the length of time it will take to return the results as well as details about the database resources the query will use (for example, which tables it will use and how many times the same table will be used). Rather than returning results, the database returns information about the query, and BI Query displays it in a window.

This facility helps you plan the queries you provide to users, and it helps users determine whether or not to issue a query. For example, a long query time may indicate a Cartesian product which you can rectify by adding the missing join. This feature can’t be applied to queries that have been filtered or combined (super queries).

To estimate database resources:

1. Ensure that the EstimateResource flag in the BIQ.ini file is set to Yes. If it isn’t, do the following:
   a. Exit BI Query.
   b. Open the BIQ.ini file and set the EstimateResource flag to Yes.
   c. Save the file.
   d. Restart BI Query.
2. Create or open a query.
3. Connect to the DBMS.
4. On the Query menu, click Estimate Resources. The Estimate Resources dialog box displays the estimated time and computer resources required if you run the current query against a Teradata or DB2 database.

Ranking Data Objects for Oracle Databases

You can improve the efficiency of queries that run against an Oracle database by ranking data objects in a data model based on the number of rows of results they can potentially return. (This feature is supported for Oracle 6+ if you’re not already running the Oracle optimizer, and only if you are using Oracle syntax—not ODBC or SQL/92.)
Tables appear in the FROM clause of an SQL statement in the order in which the corresponding data objects are ranked. The optimal sequence of tables in the statement places the table that returns the fewest number of rows, based on its WHERE conditions, last in the FROM clause. That table is processed first (and is therefore called the “driving table”). The driving table isn’t necessarily the one that contains the least amount of data but rather the one that returns the fewest number of rows of results.

In the following SQL example, the smallest_num_rows table returns the fewest rows of results and is processed first. When you rank it first, any query containing that data object has that table last in the FROM clause and has the greatest potential for being optimized.

```sql
SELECT (*)
FROM
large_num_rows, small_num_rows,
smaller_num_rows, smallest_num_rows;
```

In order to achieve the best results, you need to know how the database is being used and how many rows of results any table can return. For example, a table returns fewer rows when it’s joined to another table. (You may need to consult the documentation for your DBMS.)

When a user formulates a query using data objects that have been ranked, BI Query uses the ranking order to arrange the FROM clause for optimal performance.

To rank data objects for Oracle databases:

1. On the Host menu, click Connections. The Connections dialog box opens.
2. From the DBMS list, select Oracle; then, click Close.
3. Display the Design window that contains the data objects you want to rank.
5. From the Data Objects list, select an object; then, use the Up, Down, Top, or Bottom buttons to move the object to the desired position.
   - Click the Sort button to sort the objects alpha-numerically.
6. Repeat step 5 for all other objects until you achieve the correct order. (The object that returns the fewest number of rows must be ranked first.)
7. Click OK.
Maintaining Data Models

As the database and business needs change within your organization, you’ll receive requests for new information and facilities. BI Query provides a number of features that simplify the job of maintaining data models so you can accommodate users’ needs.

💡 To help you make changes to a data model, BI Query lets you search for and replace items in the model. For more information, see “Searching Data Models: The Basics” on page 253.
Maintaining Split Data Models

Maintaining a split data model is easier than maintaining a combined data model because you can revise the administrator layer and distribute it to users without affecting any customizations they may have made. In addition, by storing the user and administrator layers in separate folders, you can prevent the files stored in the corresponding Queries, DataVals, and Reports folders from being inadvertently overwritten when new versions of the administrator layer are distributed or when users create and modify their own files.

Changes to the administrator layer are stored in the administrator-layer file. Changes to the user layer are stored in the user-layer file.

Once you’ve made the necessary changes and saved the data model, distribute the administrator-layer file. Also distribute any new or changed queries, prompts, data values files, or BI Query reports.

To make changes to the data model:

1. In BI Query Admin, open the empty copy of the user-layer file that was created when you first saved the data model.
2. Make the necessary changes; then, save the data model.

To include a user’s customizations in a data model:

1. Make a copy of the user’s user-layer file and open it in BI Query Admin.
2. Delete any customizations that you don’t want to include; then, save the data model.

If you open a copy of a user’s user-layer file with BI Query Admin and then save the file, you will incorporate all of that user’s customizations into the administrator layer.

To include specific user customizations in a data model:

1. In BI Query Admin, open the empty copy of the user-layer file.
2. In BI Query User or BI Query Update, open the user’s user-layer file. Copy the elements you want to include in the data model.
3. In BI Query Admin, paste the copied elements; then, save the data model.
Maintaining Combined Data Models

Combined data models are easy to maintain, but be careful when distributing revised versions because they will overwrite any customizations made by your users. Users can save their changes to the data model only if they have the **Edit data model** permission. To prevent users from customizing their copies of a data model, do not assign this permission.

**To make changes to a combined data model:**

1. Run BI Query Admin.
2. Open the data model you want to change.
3. Make the necessary changes; then, save the data model.

Creating Data Model Summaries

To help you maintain a data model, you can create a report that summarizes information about the model and its associated files. The report is a tab-delimited text file (.txt) that you can open in a text editor or spreadsheet program for easy formatting, sorting, and printing.

The report provides a detailed listing of the properties of all data model components, including **Design** windows, data objects (including database keys), attributes, connection files, queries, ornaments, and prompts. You can use the report to rebuild the model from scratch, if necessary.

💡 For more information on the components you can list in the report, see the Help for your BI Query application.

**To create a data model summary:**

1. On the **File** menu, click **Data Model Summary**. The **Data Model Summary** dialog box opens.
2. Select the check boxes that correspond to the types of information you want in the summary.

⚠️ If the data model is large, the summary may be large in turn; certain text editors and spreadsheet programs may be unable to open the summary. In this case, you can create several smaller summaries and format each separately. Use the check boxes to specify which particular information appears in a given summary.
3. Click **OK**.

4. In the **Export Report To File** dialog box, specify the path and name of the summary file; then, click **Save**.

**Refreshing Data Models**

Using BI Query Admin, you can compare a data model to the current structure of the database and refresh the model to include the following changes:

- new and deleted attributes
- attributes with new data types—for example, char(12) changed to char(20) or smallint to largeint
- attributes with new Nulls flag settings
- tables that have been deleted or renamed (renamed tables are detected as deleted)
- database keys (primary and foreign keys)

After other data objects have been refreshed, query data objects (QDOs) in the data model are also refreshed. Additions or changes to metadata can be displayed at the same time. In addition to refreshing data objects, you can load any associated metadata.

BI Query displays a list of changes by table, and you select what changes to apply. BI Query provides a log of the changes that have been applied.

Once a data model has been refreshed, both you and your users can determine which queries have been affected and refresh them accordingly.

💡 For more information on refreshing queries, see the *BI Query Queries User’s Guide*.

**Determining Changes to Data Objects**

BI Query determines what changes have been made by comparing the data objects in the data model with their counterparts in the database and displaying the differences.
To determine what changes have been made:

1. Open a data model and connect to the database.

2. In a Design window, in Design mode, on the Host menu, click Refresh Data Objects.

3. In the Compare Data Objects dialog box, click Data objects from all windows to display all the tables in the data model.

4. If you want BI Query to perform case-sensitive matches when it compares the names of objects and attributes in the data model to table and column names in the database, select Case-sensitive.

5. From the Data Objects list, select the data object(s) you want to refresh.

6. If you’re loading metadata for the data objects and you don’t want to display names and short descriptions, clear Metadata display names and Metadata short descriptions.

   For more information on metadata display names and descriptions, see “How BI Query Displays Metadata” on page 193.

7. Select Database keys if you want to retrieve information about primary and foreign keys. If the DBMS you’re using does not support primary or foreign keys, this setting is ignored.

   When you retrieve database key information, and there has been a change to the primary key of a data object, BI Query automatically sets the Key field for the associated attribute in that data object. Such changes can include simply retrieving the database keys for data objects where this information was not previously available (as with earlier versions of the data model).

8. Click Compare. The Refresh Data Objects dialog box opens. BI Query displays the list of changes by table.

Applying Changes to the Data Model

Once you’ve determined the differences between the data model and the database, you can apply the changes to the data model. BI Query updates the data model and displays the changes. It also identifies the calculated attributes, object qualifiers, and join conditions that are affected by the changes; you need to update those manually.
When you apply changes from a table, BI Query automatically adds any new attributes to the data model—even if you don’t select them. Attributes that you don’t select appear in the corresponding attribute window in BI Query Admin in Design mode with the Visible flag turned off. They aren’t visible in BI Query User or BI Query Update. (You can delete an attribute from the attribute window using the Delete command on the Edit menu.)

BI Query hides these attributes instead of removing them to make it easier to maintain the data model. If BI Query removed them, it would recognize that the attributes still exist in the database and would prompt you to add them the next time you refreshed the data model. Including and hiding the attributes avoids this. Also, if you later need to give users access to these attributes, you can make them visible by turning the Visible flag on rather than refreshing the data model.

To apply a change to the data model:

1. In Design mode, on the Host menu, click Refresh Data Objects.
2. In the Compare Data Objects dialog box, select a data object, and click Compare.
3. In the Refresh Data Objects dialog box, select the change(s) you want to apply, and then click Refresh.

Interpreting Message Window Messages

Once you’ve refreshed the data model, you need to make any necessary changes to calculated attributes, object qualifiers, and join conditions that use attributes that have changed in the database. Warnings in the Message window identify changes you need to make yourself, such as the following:

***Warning: Credit Limit occurs in calculated attribute (Retailers.Credit Limit +1) of object Retailers

Read each warning carefully: it may not directly apply to the listed data object. For example, if both the Retailers and Sales data objects contain the Credit Limit attribute and Credit Limit is used in a calculated attribute in Retailers only, BI Query produces a warning for both data objects. For information on viewing and clearing messages, see “Viewing and Clearing Messages” on page 116.
Refreshing Queries

After a data model has been refreshed, some of the saved queries may not work if they include attributes or data objects that no longer exist in the database. When a user opens a data model, BI Query compares the dates on which the data model and the queries were last refreshed. If the data model was refreshed more recently than the queries, BI Query prompts the user to refresh them.

💡 For more information on refreshing queries, see the *BI Query Queries User’s Guide*.

BI Query compares each query with the data model and identifies where the query uses attributes or data objects that no longer exist—for example, in qualifications or join conditions. When you refresh queries, BI Query detects only when attributes, data objects, and join conditions have been deleted, not when they’ve simply changed. For example, it doesn’t detect changes to data type.

Only you can refresh a data model that you created, but both you and your users can refresh queries. Data values queries can’t be refreshed.

Manually Editing Database Names in the Data Model

When you refresh a data model, BI Query treats renamed tables in the database as deleted tables and cannot refresh the corresponding objects in the data model. However, you can manually rename the database names of data objects, actual relationships, and attributes within the data model before you refresh the model.

⚠️ Changing the database name of an attribute, data object, or actual relationship does not actually alter the database. You should modify the model only to make it match changes that have already occurred in the database. Improper use of this feature could damage the data model.

Your database may require that you initially give specific names to data objects, actual relationships, and attributes. Make sure to check your DBMS documentation before renaming any of these parts of the data model.
To edit the database name of an attribute:

1. In Design mode in BI Query Admin, double-click the data object or actual relationship containing the attribute you want to edit.
2. In the attribute window, click on the thick line that joins the properties of the attribute together.
3. On the Design menu, click Edit Database Name.
4. In the Edit Database Name dialog box, view the attribute's data model name and corresponding database name.
5. In the Database name box, make the necessary changes.
6. If you do not want to rename the attribute in expressions or join conditions, clear the appropriate check box(es).
7. Click OK.
8. Close the attribute window.

Editing Database Names of Data Objects and Actual Relationships

In addition to changing the display name of a data object or relationship, you can change its database name.

To edit the database name of a data object or actual relationship:

1. In Design mode in BI Query Admin, click the object or relationship whose name you want to edit.
2. On the Design menu, click Edit Database Name.
3. In the Edit Database Name dialog box, view the name of the object in the data model and the name of the corresponding table in the database.
4. In the Database name box, make the necessary changes.
5. If needed, provide a correlation name. For more information, see “Specifying Correlation Names” on page 103.
6. If you do not want to rename the object in attributes, expressions, or join conditions, clear the appropriate check box(es).
7. Click OK.
Maintaining Outer Joins

BI Query supports the syntax for ODBC, Oracle, and ANSI SQL/92 outer joins. You can still use data models that include the legacy outer join operators (\*= and \=*) without modifying the joins. You can also add joins using the legacy operators in Design windows that already contain them. However, the legacy outer join operators and the outer join check boxes are not compatible. As a result, you must modify the data model if you want to convert joins from one syntax to another.

💡 For more information on outer joins, see “Types of Joins” on page 92.

Syntax Consistency

BI Query automatically preserves outer join syntax consistency in various ways.

Split Data Model

If the user layer of a split data model contains Classic-syntax outer joins but the administrator layer has been converted to a new syntax, BI Query automatically updates the user-layer joins to the new syntax when the user next opens the model.

Queries

Queries involving Classic-syntax outer joins still work in data models that use a different syntax. BI Query automatically regenerates the SQL for the query and converts the legacy operators to the syntax in use in the model.

💡 If you do not want to update Classic-syntax queries, select the Preserve legacy query outer joins permission. For more information, see “Setting Classic Outer Join Syntax for Queries” on page 252.

Connection Files

Existing connection files that store legacy outer join information do not require modification to work in Design windows or data models that use a different join syntax. However, BI Query does not use the outer join information that the connection files store. To update the outer join information, you must either edit the connection file or use it to connect to the DBMS. For more information, see “Specifying Outer Join Capabilities and Syntax” on page 61.
Chapter 13: Maintaining Data Models

Updating Outer Joins in a Design Window

If a Design window contains “Classic” outer joins (using *= or =*), you cannot create outer joins using the outer join check boxes. Similarly, you cannot use the legacy outer join operators in a Design window that uses ODBC, Oracle, or ANSI SQL/92 syntax.

💡 BI Query stores the syntax information for a Design window or data model in the corresponding connection file. Since each Design window can have a separate connection file, the available outer join syntax may change from window to window within a data model.

To use the check boxes in a Classic-syntax Design window or data model, you must convert the existing legacy operators to a different syntax. You can convert legacy joins either manually or automatically. For more information on either procedure, see below.

Join operator conversion is necessary only if you want to use the outer join check boxes in a Design window. You do not need to convert legacy join operators to use a different syntax in queries. You can apply a given syntax to existing joins at any time (as long as the connection supports the syntax). BI Query formulates queries using the syntax you have selected (which may not be the same as the original syntax for the joins).

Converting Legacy Outer Join Operators Manually

You can convert legacy outer join operators by editing the joins directly. Before you do so, make sure that you have used the current connection file to connect to the database. BI Query requires database capability information (stored in the connection file) for any operation involving joins.

To manually convert legacy outer join operators:

1. Open a Design window that contains the legacy operators. Ensure that the current connection for the window uses the ODBC, Oracle, or ANSI SQL/92 syntax for outer joins. For more information, see “Specifying Outer Join Capabilities and Syntax” on page 61.

2. In Design mode, click the connecting line for a join that uses the legacy operator; then, on the Design menu, click Join Conditions. The Join Conditions dialog box opens.
3. For every join condition that uses a legacy operator, click the operator box (the middle box in the bottom section of the dialog box); then, select a join operator other than *= or =* from the pop-up.

4. Click OK.

5. Repeat steps 2–4 for every join that uses a legacy operator.

6. Repeat steps 1–5 for every Design window that you want to convert.

Once you have replaced all legacy outer join operators, you can reset the original outer joins using the outer join check boxes. For more information, see “Creating Join Conditions” on page 97.

**Converting Legacy Outer Join Operators Automatically**

BI Query can automatically convert legacy outer join operators for you. Automatic conversion applies only to those Design windows that contain legacy operators and support a new outer join syntax. You can convert the current Design window or all applicable Design windows in the data model.

**To automatically convert legacy outer join operators:**

1. Open a Design window that contains legacy operators. Ensure that the current connection for the window uses the ODBC, Oracle, or ANSI SQL/92 syntax for outer joins.

   ![Tip](image)

   For more information, see “Specifying Outer Join Capabilities and Syntax” on page 61.


3. In the Convert Joins dialog box, do one of the following:
   
   • To convert legacy operators in the current Design window only, select Current window only.
   
   • To convert legacy operators in all Design windows in the data model, clear Current window only. (This is the default.)

4. Click Convert.
Setting Classic Outer Join Syntax for Queries

When users run a query that uses legacy outer join operators in a data model that has been converted to a new outer join syntax, BI Query automatically converts the query to the new syntax. If this conversion does not produce the same results as the query using the old operators, you can force BI Query to treat all outer joins in the data model as if they used the Classic (ANSI SQL/89) syntax.

To use Classic syntax for legacy joins in a query:

1. On the Tools menu, click Permissions.

2. In the Permissions dialog box, select Preserve legacy query outer joins. Click OK. The Preserve legacy query outer joins permission overrides the Preserve legacy query outer joins preference.

3. Save the data model and distribute it to your users.
Searching Data Models

This section provides information on the following:
• “Searching Data Models: The Basics” on page 253
• “Specific Criteria for Searching Data Models” on page 256
• “Searching with the Find and Replace Dialog Box” on page 265

Searching Data Models: The Basics

In order to maintain a data model and to facilitate creating queries when the data model, tables, or results sets are large, you can use the Find And Replace dialog box to search for items in the model and, in some cases, replace their text. Typical applications of this dialog box include the following:
• searching for a particular data object in a cluttered Design window
• searching for a particular attribute in a very large table
• searching for a particular result in a results set window

You can also set the owner or creator name for all tables in the model by doing a search-and-replace on table name prefixes.
In BI Query User and BI Query Update, the **Find and Replace** dialog box is called the **Find** dialog box. You can use it to search for items in the data model, but you cannot use it to replace any text.

**Search Criteria Categories**

When you search for items in the data model, you must specify the following criteria (or use the defaults):

- **text**
- **type**
- **text field**
- **property**
- **search area**

**Text**

The value you want to find. You must specify a search value (or search for all values). You can force case-sensitive and whole-word searches.

**Type**

The general class of the item you want to target (for example, data objects). The default is all searchable items.

**Text Field**

The text field in the selected item that stores the search value (for example, table name). The default is all searchable text fields within the item.

**Property**

A particular quality of the selected item. BI Query searches items of the specified type only if they have the specified property. For example, you could limit a search of attributes to those attributes that are visible (that is, those attributes that have the Visible flag enabled). The default is all items of the specified type, regardless of their properties.
Search Area

The range of the search (data objects, results sets, Design windows). The default is the current active window in the data model. For example, if you open the Find and Replace dialog box when the active window is a Design window, the default search area is that Design window. If you open the dialog box when an attribute window is active, the default search area is the corresponding data object. For windows other than attribute, design, or results windows, the default search area is the entire data model.

BI Query returns a list of text fields in the items that match the specified search conditions. (Since a given item can have multiple text fields, it may appear in the list more than once.) You can use this list to examine the matching items directly and, if permitted, replace their text. You can also save the specified search conditions and the results of your search to a text file for further reference.

Examples

Using the available criteria, you could search for the following items:

- all merged data objects with a table name of “Product_ID” across the entire data model
- all calculated attributes with the Integer data type in a particular Design window (“MySales”)
- all cells that contain the value “Ontario” in one or more results sets windows

When setting up the conditions for a search, you can force case-sensitive matches, in which the returned items match the content and case of the search value:

<table>
<thead>
<tr>
<th>Search Value</th>
<th>Data Model Value</th>
<th>Case-Sensitive Match?</th>
</tr>
</thead>
<tbody>
<tr>
<td>button</td>
<td>button</td>
<td>Yes</td>
</tr>
<tr>
<td>button</td>
<td>Button</td>
<td>No</td>
</tr>
</tbody>
</table>
You can also force whole-word matches, in which the entire content of each returned item matches the search value exactly. If you do not select this option, BI Query can return items that either match the search value exactly or contain it:

<table>
<thead>
<tr>
<th>Search Value</th>
<th>Data Model Value</th>
<th>Whole-Word Match?</th>
</tr>
</thead>
<tbody>
<tr>
<td>button</td>
<td>Button</td>
<td>Yes</td>
</tr>
<tr>
<td>button</td>
<td>button2</td>
<td>No</td>
</tr>
</tbody>
</table>

**Specific Criteria for Searching Data Models**

The criteria available for a given search depend on the type of item you want to locate.

**Search Criteria for Data Objects**

You can limit your search on data objects to specific text fields and to specific types of data objects (on the basis of a particular property).

You can search on one or more of the following data object text fields:

<table>
<thead>
<tr>
<th>Text Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display name</td>
<td>The name of the object as it appears in a <strong>Design</strong> window. The display name can be edited within the data model or can be set using metadata. For more information on metadata, see “How BI Query Displays Metadata” on page 193.</td>
</tr>
<tr>
<td>Table name</td>
<td>The name of the table in the database. For some databases, the table name includes an owner or creator name or a volume or subvolume ID. The table name appears in the SQL for a query.</td>
</tr>
<tr>
<td>Query data object SQL</td>
<td>The SQL string associated with the object if it is a query data object. For more information on query data objects, see “Query Data Objects (QDOs)” on page 108.</td>
</tr>
</tbody>
</table>
Specific Criteria for Searching Data Models

Searchable Data Object Properties

In addition to all data objects, you can selectively search those that have one of the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query data object</td>
<td>A query that has been converted into a data object. For more information, see “Query Data Objects (QDOs)” on page 108.</td>
</tr>
<tr>
<td>User object</td>
<td>The object created by users when they send query results to the database as a table. For more information on user-created objects, see the BI Query Queries User’s Guide.</td>
</tr>
<tr>
<td>Merged object</td>
<td>A data object that has been merged into another object. Objects that have been merged in this way do not appear as discrete entities in a Design window. As a result, when you view a merged object that matches your search criteria, BI Query highlights the object that contains the merged object. For more information on merged objects, see “Merging Data Objects” on page 106.</td>
</tr>
</tbody>
</table>
Search Criteria for Attributes

You can limit your search on attributes to specific attribute text fields and to specific types of attributes (on the basis of a particular property).

You can search on one or more of the following attribute text fields:

<table>
<thead>
<tr>
<th>Text Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display name</td>
<td>The name of the attribute as it appears in the attribute window. The display name can be edited within the data model or can be set using metadata. For more information on metadata, see “How BI Query Displays Metadata” on page 193.</td>
</tr>
<tr>
<td>Attribute name</td>
<td>The database name of the attribute (or column). In the SQL for a query, the attribute name is prefixed with the name of the corresponding table, followed by a period (for example, Sales.productID).</td>
</tr>
<tr>
<td>Table name</td>
<td>The database name of the table that stores the attribute (or the correlation name of the table, if it has one).</td>
</tr>
<tr>
<td>Data type</td>
<td>The name of the data type for the attribute (for example, “smallint”). For more information on available data types for attributes, see the Help for your BI Query application.</td>
</tr>
<tr>
<td>Expression</td>
<td>The calculation expression for a calculated attribute. For more information on calculated attributes, see the BI Query Queries User’s Guide.</td>
</tr>
</tbody>
</table>
Specific Criteria for Searching Data Models

Searchable Attribute Properties

In addition to all attributes, you can selectively search those that have one of the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visible</td>
<td>An attribute that has the Visible flag enabled; attributes that do not have this flag enabled do not appear in the attribute window in Run mode. Attributes that are part of a join condition are often set invisible by the BI Query administrator. You can set the Visible flag in the attribute window in Design mode. For more information, see “Securing Access to Columns” on page 173.</td>
</tr>
<tr>
<td>Null</td>
<td>An attribute that permits NULL values in updates or inserts. You can set this property by enabling the Null flag in the attribute window in Design mode.</td>
</tr>
<tr>
<td>Calculated</td>
<td>An attribute that performs a predefined calculation in a query. For more information on calculated attributes, see the BI Query Queries User’s Guide.</td>
</tr>
</tbody>
</table>
Search Criteria for Results Sets

You can search on one or both of the following text fields for results sets:

- **Cell contents.** When you search a results set, its cell contents are treated as text values, regardless of their actual data type.
- **Column titles.**

To search for NULL values in a results set, leave the search value blank.

💡 Click on the title of the text column in the items found in the Find and Replace dialog box. This will put all NULL values at the top of the list.

Search Criteria for Buttons

You can limit your search to one or more specific text fields within buttons. You can also search all buttons in a specified area of the data model or selectively search those that have a specific property.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary key</td>
<td>An attribute that has unique values in the database and serves as a primary key (key field). You cannot update a table unless it has at least one primary key. You can set an attribute as a primary key in the attribute window in Design mode, or BI Query can retrieve this information from the database when uploading or refreshing tables. For more information on primary keys, see “Setting Key Fields” on page 273.</td>
</tr>
<tr>
<td>Update</td>
<td>An attribute that you can modify in the database using BI Query Admin or BI Query Update. You can set this property by enabling the Update flag in the attribute window in Design mode. For more information, see “Denying Update Permission for Specific Attributes” on page 274.</td>
</tr>
<tr>
<td>Data values</td>
<td>An attribute for which BI Query uses the default SELECT DISTINCT query to generate data values. BI Query does not use this query if the attribute has an associated data values file. For more information on data values files, see “Securing Access to Data Values” on page 177.</td>
</tr>
<tr>
<td>Foreign key</td>
<td>An attribute which the database specifies is a foreign key.</td>
</tr>
</tbody>
</table>
For more information on buttons, see “Adding Buttons to Data Models” on page 151.

You can search on one or more of the following text fields:

<table>
<thead>
<tr>
<th>Text Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The internal name of the button. This name distinguishes the button from other components of the data model.</td>
</tr>
<tr>
<td>Text</td>
<td>The display text for the button, if it displays text.</td>
</tr>
<tr>
<td>Query name</td>
<td>The name of the linked query file, if the button links to a query.</td>
</tr>
<tr>
<td>Screen tip</td>
<td>The text that appears as a screen tip when the user moves the mouse pointer over the button.</td>
</tr>
<tr>
<td>Destination</td>
<td>The name of the destination for query results, if the button links to a query. Valid Destination fields include the following:</td>
</tr>
<tr>
<td></td>
<td>• the name of the destination table, if the button sends query results to a database table</td>
</tr>
<tr>
<td></td>
<td>• the name of the destination file, if the button sends query results to a file</td>
</tr>
<tr>
<td></td>
<td>• the name of the linked script (or application or document), if the button links to a script (or application or document)</td>
</tr>
<tr>
<td></td>
<td>• the text of the linked URL, if the button is a hyperlink</td>
</tr>
</tbody>
</table>
Searchable Button Properties

In addition to all buttons, you can selectively search buttons on the basis of one of the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>The type of resource to which the button links. You can target buttons that link to one of the following resources:</td>
</tr>
<tr>
<td></td>
<td>• a query</td>
</tr>
<tr>
<td></td>
<td>• a <strong>Design</strong> window</td>
</tr>
<tr>
<td></td>
<td>• a script (or application or document)</td>
</tr>
<tr>
<td></td>
<td>• a report created in BI Query Reports</td>
</tr>
<tr>
<td></td>
<td>• a BI Cube Creator hypercube</td>
</tr>
<tr>
<td></td>
<td>• a URL (hyperlink)</td>
</tr>
<tr>
<td>Appearance</td>
<td>The display type for the button. You can search buttons that display one of the following:</td>
</tr>
<tr>
<td></td>
<td>• a text string</td>
</tr>
<tr>
<td></td>
<td>• an icon</td>
</tr>
<tr>
<td></td>
<td>• a picture</td>
</tr>
<tr>
<td>Repeating</td>
<td>Whether the button occurs in more than one <strong>Design</strong> window.</td>
</tr>
<tr>
<td>Results destination</td>
<td>The destination for query results, if the button links to a query.</td>
</tr>
<tr>
<td>Results</td>
<td>You can selectively target buttons that send the query results to one of the following destinations:</td>
</tr>
<tr>
<td>destination</td>
<td>• a results window</td>
</tr>
<tr>
<td></td>
<td>• an application</td>
</tr>
<tr>
<td></td>
<td>• a file</td>
</tr>
<tr>
<td></td>
<td>• a database table</td>
</tr>
</tbody>
</table>

Search Criteria for Ornaments

You can limit your search on ornaments to one or more text fields. You can also search all ornaments or search one particular type of ornament (on the basis of a particular property).
For more information on ornaments, see “Adding Ornaments and Other Objects” on page 134.

The following tables summarize the available search criteria for ornaments:

<table>
<thead>
<tr>
<th>Text Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The internal name of the ornament.</td>
</tr>
<tr>
<td>Text</td>
<td>The display text of the ornament, if it is a text ornament. This field is empty if the ornament does not display text.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ornament Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>An ornament that displays text.</td>
</tr>
<tr>
<td>Icon</td>
<td>An ornament that displays an icon.</td>
</tr>
<tr>
<td>Picture</td>
<td>An ornament that displays a picture.</td>
</tr>
<tr>
<td>Shape</td>
<td>An ornament that was created using the Line, Rectangle, or Oval drawing buttons on the Drawing toolbar.</td>
</tr>
<tr>
<td>Repeating</td>
<td>Whether the ornament occurs in more than one Design window.</td>
</tr>
<tr>
<td>OLE object</td>
<td>An ornament that stores an OLE object.</td>
</tr>
</tbody>
</table>

**Search Criteria for Relationships**

You can limit your search on relationships to specific text fields within relationships and to specific types of relationships (on the basis of a particular property).

For more information on relationships, see “Creating Relationships Between Data Objects” on page 85. For more information on join conditions, see “Specifying Join Conditions” on page 91.
You can search on one or more of the following text fields:

<table>
<thead>
<tr>
<th>Text Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display name</td>
<td>The name of the relationship in a Design window. The display name can be edited within the data model. If the relationship is an actual relationship, the display name can also be set using metadata. For more information on metadata, see “How BI Query Displays Metadata” on page 193.</td>
</tr>
<tr>
<td>Operator</td>
<td>The text string that represents the join operator.</td>
</tr>
<tr>
<td>Left side</td>
<td>The text string that represents the left side of the join condition.</td>
</tr>
<tr>
<td>Right side</td>
<td>The text string that represents the right side of the join condition.</td>
</tr>
</tbody>
</table>

A relationship can have more than one join condition. When you search on a particular type of join component (such as a join operator), BI Query searches all components of that type in the relationship.

**Searchable Properties for Relationships**

In addition to all relationships, you can selectively search those that have the characteristic property of one of the following relationships:

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic relationship</td>
<td>A relationship created by users when they join unconnected data objects or specify a new join condition for objects that are already joined. For more information, see “Types of Relationships” on page 85.</td>
</tr>
<tr>
<td>Actual relationship</td>
<td>A special data object that joins other data objects and represents an actual table in the database. You can also search for actual relationships as objects. In this case, you can search the text fields associated with the object (such as correlation name). For more information, see “Search Criteria for Data Objects” on page 256.</td>
</tr>
</tbody>
</table>
Searching with the Find and Replace Dialog Box

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Merge join</td>
<td>The relationship between objects in a merged object. Objects that have been merged do not appear as discrete entities in a Design window. As a result, when you view a merge join that matches your search criteria, BI Query highlights the object that contains the merged objects. For more information on merged objects, see “Merging Data Objects” on page 106.</td>
</tr>
</tbody>
</table>

Search Criteria for Variables

You can limit your search on variables to the following text fields:

<table>
<thead>
<tr>
<th>Text Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the variable. Do not include the chevron characters (» and «) as part of the search value.</td>
</tr>
<tr>
<td>Description</td>
<td>The description of the variable. This text appears in the Description box of the Edit Variable dialog box. This field is empty if the creator of the variable has not provided a description.</td>
</tr>
<tr>
<td>Expression</td>
<td>The expression that defines the variable. When you run a query that includes a variable, BI Query evaluates the corresponding expression and adds it to the query.</td>
</tr>
</tbody>
</table>

💡 For more information on variables, see the BI Query Queries User’s Guide.

Searching with the Find and Replace Dialog Box

Use the Find and Replace dialog box to search the data model and view matching items or replace their text.

To find an item:

1. On the Edit menu, click Find and Replace. The Find and Replace dialog box opens.
2. In the Find Text list box, do one of the following:
   - To match a particular value, type the value (or select a previously-entered value).
• To match any value, leave the box blank.
3. If you want returned items to match the case of the specified value exactly, select Match case. If you want to force whole-word matches, select Find whole word only.

4. From the Target type list box, select the type of item you want to find.
5. From the In text field(s) list, select the particular field(s) of the item you want to search.

6. To narrow your search on the basis of a specific property, use the With property list. Do one of the following:
   • To search only those instances of the selected item that have a certain property, click the property once. A check mark box appears to the left of the property.
   • To search only those instances of the selected item that do not have a certain property, click the property twice. A red X box appears to the left of the property.
   • To search all instances of the selected item, regardless of their properties, click (Any Properties). (This is the default.)

7. If you want to restrict the search to a certain area of the data model—to Design windows, results sets, or data objects—click Search Area. The Search Area dialog box opens.

   In the Search Area list box, select the area you want to search. From the list below the Search Area list box, select one or more instances of the selected area. Click OK.

   You can search only those areas in the data model that are relevant to your search. For example, if you select Results Set from the Target Type list box, you can search for the specified value only in results windows.

8. In the Find And Replace dialog box, click Find. If a given search is successful, the Items Found box lists the matching items; the number of matching items appears at the bottom of the dialog box.

Once you have run a search, you can save the search conditions and results to a text file. Click To File; then, specify the path and name of the file. The file also contains the name of the data model and the date of the search.
Searching for NULL Values

You can use the Find and Replace dialog box to find NULL values in a results set. You can also search for blank text in certain text fields (for example, tables that do not have correlation names).

You cannot use an blank value to search specifically for objects that do not have an object qualifier.

To search for NULL or blank values:
1. On the Edit menu, click Find and Replace. The Find And Replace dialog box opens. Leave the Find text box blank.
2. From the Target type list box, select the type of item you want to find.
3. From the In text field(s) list, select the particular field(s) of the item you want to search.
4. To narrow your search on the basis of a specific property, use the With property list.
5. To restrict the search to a certain area of the data model, click Search Area; then, use the Search Area dialog box to select the area you want to search. Click OK.
6. In the Find And Replace dialog box, click Find.
7. In the Items Found list, click the title for the Text column. This sorts all returned items alphabetically; any NULL or blank values appear at the top of the Items Found list.

Viewing and Replacing Matched Items

You can use the Items Found list in the Find And Replace dialog box to view items that match the search value. You can also use the list to replace the text of matched items. Each row in the list presents the following information:
- the matched item (for example, a particular cell in a results set)
- the value of the matched text in that item
- the type of item (for example, a results set)
- the text field that stores the matched text (for example, cell contents)
- the window that stores the item (for example, a query results window)
- any other relevant details about the item
An item may appear in the list more than once if it matched the search value in two or more fields.

Before you replace any text in the data model, ensure that your changes will not make the model unusable. For example, the replace operation should preserve the following properties of a working data model:

- Expressions for object qualifiers and calculated attributes contain valid SQL.
- Table and attribute names in the model match the corresponding names in the database.
- Join conditions do not produce runaway queries.

**To view an item in a search result:**

1. Open the **Find And Replace** dialog box and search for a value.
2. From the **Items Found** list, select the item (in the **Item** column) that you want to view.
3. Click **Goto**. (The **Find And Replace** dialog box is minimized.) BI Query opens the appropriate design/attribute/results window and highlights the selected item.

   You can go to the next item in the list by clicking **Find Next** on the **Edit** menu.

**To replace a text value in a matched item:**

1. From the **Items Found** list, select the item (in the **Item** column) whose value you want to replace.
2. In the **Replace with** drop-down list, type the new value for the matched text of the selected item (or select a previously entered value); then, click **Replace**.

   An alert box warns you that changes to the data model may make it unusable. To proceed with the changes, click **OK**.

**Do not replace the display name and table name of a data object (or an attribute) at the same time.** If the database name of a table has changed, and you run a query involving that table, BI Query locates the table using the display name of the corresponding data object. If you change both the display name and table name for a data object or attribute, BI Query cannot open any query involving the corresponding table.
Replacing Owner/Creator Names for Tables

A common maintenance problem for data model administrators is globally replacing table names in the model to match changes in table ownership in the database. (For certain DBMSs, such as Oracle, table names often contain a prefix to indicate the owner or creator.) Such a change may occur if the data model is first created for a development environment and is later moved to a production environment. In this case, the BI Query administrator must change the table names wherever they occur in the data model: in data objects, in queries, in joins, and so on.

Using the Find And Replace dialog box, you can easily locate all text values in the model that have the specific prefix, and then change those values to include the new prefix.

To replace owner/creator names for tables:

1. Create a copy of the data model. If the search-and-replace operation makes the model unusable, you can open the saved copy and rename it to the original.
2. On the Edit menu, click Find and Replace. The Find And Replace dialog box opens.
3. In the Find Text box, type the prefix that represents the table owner or creator.
4. To match the case of the specified value exactly, select Match case.
5. From the Target type drop-down list, select (All).
6. Click Find.
7. From the Items Found list, select the items (in the Item column) you want to modify.

   If the display name and table name for a data object both contain the old prefix, and you replace the prefix in both, the next time you open a query that uses this object you will get an error message (“Object x not found in the data model. BI Query will use y instead”). If this happens, re-save the query. The next time you open it, the warning will not appear.

8. In the Replace with box, type the new prefix value; then, click Replace.

   An alert box opens to warn you that changes to the data model may make it unusable. To proceed with the changes, click OK.
9. Verify the replace operation: check the names of tables as they appear in data objects, attributes, merged objects, queries, and so on. If the operation is successful, re-save all queries; then, save the data model.
Requirements for Updating Database Records

If you are using BI Query Update or BI Query Admin and the BI Query administrator has set the required permissions, you can add, modify, and delete records in the database. BI Query Update is designed primarily for maintaining tables, not for large-scale data entry or updating more than one table at a time. However, when you have a large number of records stored in another application, you can use BI Query Update to add them to the database.

続けて、データベース記録を更新するための基本的な要件を掲載します。

You cannot update database records using BI Query User.

BI Query Update is designed to update records in the following situations:

• You need to correct or replace missing data.
You need to update a single table. (Only one person can update a table at a time.)

You need to populate an empty table.

You need to add records to the database that are stored in another application such as Microsoft Excel.

In order to update records, you must satisfy the following conditions:

- You must have the **Update Database Permission** assigned.
- You must have certain database permissions (depending on the DBMS) granted by the database administrator.
- You must only update records from one data object at a time.
- The data object you want to update must have at least one key field.
- The update you want to apply to an object must include all key fields in that object.

As there is no way to specify key fields in freehand queries, you cannot use freehand query results to update database records. You can use `Insert` or `Update` statements in the freehand query itself to update database records.

### Controlling Database Updates

The BI Query administrator can control whether or not users can use either BI Query Update or BI Query Admin to update records in the database. In order for updating to take place, the BI Query administrator must do the following:

- Set at least one key field in each data object to be updated; for more information, see “Setting Key Fields” on page 273.
- Assign the **Update Database** permission, which lets users change data; for more information, see “Assigning the Update Database Permission” on page 274.

The BI Query administrator can also prevent users from updating specific attributes; for more information, see “Denying Update Permission for Specific Attributes” on page 274.
Setting Key Fields

The BI Query administrator must designate at least one of the attributes in a data object as a key field to allow the corresponding table to be updated. Key fields must contain unique data, such as ID numbers. BI Query may be able to automatically determine which attributes have been designated as primary and foreign keys in your database. If your DBMS supports database keys, and you selected the Database keys check box when inserting or refreshing the data object, BI Query will automatically select key fields for you based on the primary keys. In Design mode, attributes that have been designated as primary and foreign keys in the database appear with a key icon to the left of the Key column (the icon for a primary key has both a key and a 1). You can also create a data model summary to view detailed information about the database keys in your data model. If your DBMS does not support database keys, or if you want to make specific changes yourself, you can also manually specify key fields.

If you clear the Key field setting for an attribute identified in the database as a primary key (one that appears with the primary key icon), the primary key icon will remain but the associated attribute will no longer behave as a primary key in BI Query.

You must use BI Query Admin to set an attribute as a key field.

In Run mode, if you have set the Show primary key preference, key fields are identified in the attribute window by a key icon. If you have not set the Show primary key preference, you can click the question mark beside an attribute in the attribute window to open the Attribute dialog box. If the attribute is a key field, a key icon appears in the upper left corner of the Attribute dialog box.

To set a key field automatically:

* In Design mode, do one of the following:
  * To set a key field in a new data object, on the Host menu click Insert Data Objects. In the Insert Data Objects dialog box, select the table for which you want to create a data object, select the Database keys check box, then click Insert.
  * To set a key field in an existing data object, on the Host menu click Refresh Data Objects. In the Compare Data Objects dialog box, select the data object for which you want to set a key field, select the Database keys check box, then click Compare. In the Refresh Data Objects dialog box, click Refresh.
To set a key field manually:
1. In Design mode, double-click the data object or actual relationship that stores the attribute that you want to set as a key field.
2. In the attribute window, select the Key check box beside the attribute.

Assigning the Update Database Permission
To authorize users of BI Query Update or BI Query Admin to make changes to the data, you need to assign them the Update Database permission.

To assign the Update Database permission:
1. In BI Query Admin, on the Tools menu, click Permissions.
2. In the Permissions dialog box, select Update Database.
3. Click OK.

Denying Update Permission for Specific Attributes
BI Query assigns update permission by attribute. The permission is automatically granted for all attributes in the data model, but the BI Query administrator can deny the permission for a specific attribute to prevent users from updating the attribute. (In this case, users can double-click this attribute in an update window, but they can’t modify it. Users can, however, update this attribute when it occurs in new rows.)

You must use BI Query Admin to set the update permission for an attribute.

To deny update permission for an attribute:
1. In Design mode, double-click the data object or actual relationship that stores the attribute.
2. In the attribute window, clear the Update check box beside the attribute.

Adding, Modifying, and Deleting Records
You must use BI Query Update or BI Query Admin to access these features.
In order to update data, in most cases you either open a results file or run a query, and then select the record(s) you want to update and send them to the Update window. You can also update records without retrieving results.
Spreadsheet and Form View

When you update records, the rows you selected to update are displayed in the update window in Spreadsheet view. Spreadsheet view displays multiple records in a format similar to a results window. It is useful when you are updating the same data for more than one record or when you need to compare records when making changes.

💡 In Spreadsheet view, a key appears above a key field to indicate that the field is locked and cannot be modified.

Form view displays one row at a time and provides a way to navigate through the rows.

💡 In Form view, a key appears beside a key field to indicate that the field is locked and cannot be modified.

You can change from Spreadsheet view to Form view of an individual record by double-clicking its row number. To change back to Spreadsheet view, click the close button in the upper right corner of the Form view window. The view in which you work depends on the nature of the data and the types of changes you want to make.

Any values you change are displayed in italics, and update indicators identify whether you want to add, modify, or delete a record. The changes you make are stored in memory on your computer until you apply them to the database.

⚠️ BI Query Update does not lock rows that are being updated. This means that more than one user can make changes to the same record at one time. To avoid overwriting another user’s changes, make sure that only one person is updating a table at a time.

Adding, Modifying, and Deleting Records

You must use BI Query Update or BI Query Admin to add, modify, or delete records.
To add, modify, or delete records:

1. Do one of the following:
   - To modify or delete existing records, retrieve them. (Use only one data object and include all key fields.) Select the record(s) you want to update; then, on the Results menu, click Update.
   - To update records without retrieving results, click the data object you want to update. On the Results menu, click Open for Insert.
2. To work in Form view instead of the default Spreadsheet view, double-click the row number of a record in the Spreadsheet view.
3. To add a record, on the Results menu, click Insert Row, click in a cell; then, type the information you want.
4. To modify a record, click in a cell, make the change; then, click outside the cell.
5. To delete a record, select it; then, on the Results menu, click Delete Row.
6. Once you have completed your changes, click Do Update on the Results menu.
   For more information, see “Applying Updates to the Database” on page 279.

If BI Query displays an error message that you cannot update more than one table, you may have selected a data object that has been merged. In this case, retrieve results for attributes in one table at a time.

Adding Records from Another Application: The Basics

You must have BI Query Update or BI Query Admin to access this feature.

When records are stored in an application that does not allow you to update the database, you can import the records into BI Query, and then add them to the database. For example, you may have created a “batch” of new records that you want to add to an existing table. You may also want to populate a new table that you have created in the database.

You can import data as a text file and paste data from the Clipboard. Be sure you meet the conditions required for updating.

The basic procedure for adding records from another application is as follows:
1. Prepare a text file to add records. For more information, see “Preparing a Text File to Add Records” on page 277.

2. Specify results options before updating records. You must ensure that BI Query’s default results options match the field, record, and end-of-file separators used in the text file. Because you are importing a text file, the separators typically match BI Query’s default options:
   - The <TAB> character separates fields.
   - The <CR> character separates records.
   - No character (<None>) indicates the end of file.

   If the file uses any other separators, you must specify them. The only options that should be specified are the separators.

   For more information on specifying default results options and separators for the data model, see the BI Query Queries User’s Guide.

3. Import the file to update the records. For more information, see “Importing a File to Update Records” on page 278.

Preparing a Text File to Add Records

To add records from another application, you first prepare the text file.

To prepare the text file:

1. Store the data in a flat text file (usually tab-delimited). Do not include column headings.

2. Make sure that the columns are in the same order as those in the update window. (To view the order, double-click the data object that corresponds to the table you are updating, select all the attributes, and run a query.)

3. Include the key field and make sure that its values are unique. (For example, if the key field is Retailer #, its new values must not duplicate existing values in the table you are updating.)

4. Include all columns flagged as NOT NULL for the table you are updating.

5. To specify NULL values, use placeholder values:
   - For non-character fields such as integers, money, and dates, replace NULLs with “dummy” values. (You can clean them up after you have inserted the file.)
Chapter 15: Updating Database Records

- For character fields, replace NULLs with periods to reduce the chance of data becoming misaligned.

You can use a spreadsheet application such as Microsoft Excel to correctly insert placeholder values. Once the data is in BI Query, you can open an update window, delete the contents of the cells, and then update the database. The deleted cell contents become NULLs. BI Query does not allow you to insert blank character values (‘ ’).

Instead of using placeholders, you can also include NULL values in the text file. Be sure to include the exact number of tab characters.

6. Make sure there are no blank lines or spaces at the end of the file.

   If you have trouble importing a Microsoft Excel file into BI Query (Excel does not maintain carriage returns), open the file in Microsoft Word and save it as a DOS text file (.txt).

You can create a Microsoft Word file that contains the update data you want to import into BI Query. In the Word file, type each value, followed by a <TAB> character. At the end of each record, press ENTER. do not press ENTER after the last record. Save the file as a DOS text file (.txt).

**Importing a File to Update Records**

You must have BI Query Update or BI Query Admin to access this feature.

When you import data from a file, the data is inserted into an update window. An update indicator appears beside each row to mark it to be added to the database.

**To import a file into BI Query:**

1. In BI Query Admin or BI Query Update, click the data object whose table you are updating.
2. On the Results menu click Insert from File; then, use the Insert from File dialog box to specify a text file.

You can also import data using the Clipboard.

**To import data using the Clipboard:**

1. In the source application, copy the data to the Clipboard.
2. In BI Query, click the data object whose table you are updating. On the Results menu, click Open for Insert.
3. Paste the contents of the Clipboard into the window.
Applying Updates to the Database

You must have BI Query Update or BI Query Admin to access this feature.

The changes you make to records in BI Query are stored in memory on your computer. You can apply them to the database as you make them or all at once at the end of a BI Query session.

After all the updates are applied, the update indicators become unavailable. If you try to close the update window before applying the updates, BI Query prompts you to either complete the update, cancel closing the window, or close the window and discard all the updates.

**To apply an update to the database:**

1. If you are not already connected, connect to the DBMS.
2. In an update window, on the **Results** menu, click **Do Update**.

   ![To cancel making a change in the update window, click the update indicator for that row to remove the indicator.]

**Correcting Updating Errors**

You must have BI Query Update or BI Query Admin to access this feature.

If an error occurs while a record is being updated in the database, the update to that record and the remaining records are lost. Errors may occur, for example, when data separators or data types are incorrect in records you have imported into BI Query from another application.

**To correct an error:**

1. To correct an error in the data separators, check the BI Query results options and make sure that the separators you have specified match the separators used in the text file.

2. To correct an error in data types, identify the fields with mismatched data types by highlighting the first cell in the update window, and then use the arrow keys to move from cell to cell. If the data type in a cell is invalid, BI Query prompts you to enclose it within brace brackets ({}). Do so, or correct the data type. (Keep in mind that while BI Query may accept this data type, the DBMS may not.)

3. When you have finished correcting the error, apply the update again.
Accessibility and Technical Support

This section provides information on the following:

- “General Accessibility” on page 281
- “Technical Support” on page 283

General Accessibility

Hummingbird products are accessible to all users. Wherever possible, our software adheres to Microsoft Windows interface standards and contains a comprehensive set of accessibility features.

Access Keys

All menus have associated access keys (mnemonics) that let you use the keyboard, rather than a mouse, to navigate the user interface (UI). These access keys appear as underlined letters in the names of most UI items. (If this is not the case, press ALT to reveal them.) To open any menu, press ALT and then press the key that corresponds with the underlined letter in the menu name. For example, to access the File menu in any Hummingbird application, press ALT+F.
Once you have opened a menu, you can access an item on the menu by pressing the underlined letter in the menu item name, or you can use the arrow keys to navigate the menu list.

**Keyboard Shortcuts**

Some often-used menu options also have shortcut (accelerator) keys. The shortcut key for an item is listed beside it on the menu.

**Directional Arrows**

Use the directional arrows on the keyboard to navigate through menu items or to scroll vertically and horizontally. You can also use the directional arrows to navigate through multiple options. For example, if you have a series of radio buttons, you can use the arrow keys to navigate the possible selections.

**Tab Key Sequence**

To navigate through a dialog box, press the TAB key. Selected items appear with a dotted border. You can also press SHIFT+TAB to go back to a previous selection within the dialog box.

**SPACEBAR**

Press the SPACEBAR to select or clear check boxes, or to select buttons in a dialog box.

**ESC**

Press the ESC key to close a dialog box without applying new settings.

**Enter**

Press the ENTER key to select the highlighted item or to close a dialog box and apply the new settings. You can also press the ENTER key to close all About boxes.

**ToolTips**

ToolTips appear for all functional icons. This feature lets users use Screen Reviewers to make interface information available through synthesized speech or through a refreshable Braille display.
Microsoft Accessibility Options

Microsoft Windows environments contain accessibility options that let you change how you interact with the software. These options can add sound, increase the magnification, and create sticky keys.

To enable/disable Accessibility options:

1. In Control Panel, double-click **Accessibility Options**.
2. In the **Accessibility Options** dialog box, select or clear the option check boxes on the various tabs as required, and click **Apply**.
3. Click **OK**.

If you installed the Microsoft Accessibility components for your Windows system, you can find additional Accessibility tools in the Accessibility program group on the **Start** menu.

Technical Support

Administrators can contact Hummingbird Technical Support to report problems or suggest enhancements. We require product and company information before we can investigate any problems. For your convenience, the **Hummingbird BI Configuration Manager** utility can quickly assemble most of the required information and automatically add it to an e-mail message. Even the address is automatically filled in, so all you need to do is add a description of your problem to the body of the message and click **Send**. For more information on using the **Hummingbird BI Configuration Manager** utility, consult the utility’s online help.

To start the **Hummingbird BI Configuration Manager** utility:

- On the **Start** menu, navigate to the program group folder for your BI application (BI Server or BI Query), then click **BI Configuration Manager**.

For Technical Support services, please use the contact information for your area, or visit the Technical Support web site at:
Using the Trace Utility

Hummingbird provides a trace utility with the software to help troubleshoot problems you are having. The trace utility simplifies problem-solving by monitoring the activity of your products. If you are having problems with the software, Technical Support may ask you to run the trace utility, reproduce the problem, save the trace information, and send us the resulting trace file.

To run the trace utility, double-click `trace.exe` from one of the following locations:

- Program Files\Hummingbird\BI\Query
- Program Files\Hummingbird\BI\Server

For information on configuring the trace utility, see Trace Help.
Symbols

&& text string 137
&d text string 136
&dn text string 136
&t text string 136
*= operator 95
.bak files 217
.bmp files 134
.emf files 134
.gif files 134
.gqa files 23
    distributing 212
    saving 210
.gql files 25
    distributing 211
    saving 211
.gqu files 23
    distributing 212
    saving 210
.ico files 134, 139
.rtf files 206

.wmf files 134
=* operator 95
? in attribute windows 201
«table_name» variable 195

A

access to data, limiting 172
accessibility, general 281
activities requiring DBMS connections 57
actual relationships
    overview 86
    connecting to data objects 90
    creating 89
    editing database names 248
    from conversion of data objects 89
ad hoc
    data models 67
    queries 22
    report data models 67
adding indices, database preparation 71
adding value to data models, overview 191
administrator in BI Query 18
administrator-layer files 34
  location 34
  path 213
  saving 210
  split data model 23
advanced users 187
  suggested permission settings for 184
aligning
  objects 126
  text 126
aligning objects 125
  spacing evenly 127
Allow Dynamic Relationships preference 186
Allow Non-Select SQL permission 185
Allow Unconnected Data Objects permission 235
Allow Web Ad Hoc Queries permission 229
annotating existing Help 206
ANSI SQL/89 outer join syntax 95
ANSI SQL/92 outer join syntax 95
application scripts 157
  linking to buttons 158
applications
  adding records from 276
  assigning permissions 183
  associating data with user names 180
attribute descriptions
  creating in BI Query 201
  creating tables for 202
  in split data models 200
  populating tables 204
  retrieving by query 204
  retrieving from results files 205
attributes
  comments for 201
  definition 20
  editing database names 248
  hidden 245
  hiding in tables 173
  inserting 114
  names of, simplifying automatically 82
  removing from tables 173
  renaming 81, 83
  reordering 84
attributes, calculated
  updating 245
attributes, in BI Query tables
  GQLDATA 219
  GQLDLEN 219
  GQLFID 219
  GQLMODL 219
  GQLSEQ 219
attributes, in save-to-database tables 218
  GQLDVTAB 218
  GQLMAND 218
  GQLMODEL 218
  GQLMTAB 218
  GQLQRYTAB 218
  GQLREASON 218
  GQLREPTAB 218
  GQLVERS 218
audience design considerations 120
<table>
<thead>
<tr>
<th>automatic connections</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>automation</td>
<td></td>
</tr>
<tr>
<td>controllers</td>
<td>158</td>
</tr>
<tr>
<td>objects</td>
<td>158</td>
</tr>
<tr>
<td>simplifying display names</td>
<td>82</td>
</tr>
<tr>
<td>updating of links</td>
<td>148</td>
</tr>
<tr>
<td>autosizing buttons</td>
<td>153</td>
</tr>
<tr>
<td>Publish Data Model Admin Layer</td>
<td>224</td>
</tr>
<tr>
<td>Publish Data Model User Layer</td>
<td>224</td>
</tr>
<tr>
<td>BI Server security</td>
<td>229</td>
</tr>
<tr>
<td>access inheritance rules</td>
<td>230</td>
</tr>
<tr>
<td>access symbols</td>
<td>231</td>
</tr>
<tr>
<td>granting ad hoc querying</td>
<td>228</td>
</tr>
<tr>
<td>setting by item</td>
<td>232</td>
</tr>
<tr>
<td>setting by user or group</td>
<td>232</td>
</tr>
<tr>
<td>setting data model visibility</td>
<td>227</td>
</tr>
<tr>
<td>users and groups</td>
<td>230</td>
</tr>
<tr>
<td>BIGroupID variable</td>
<td>179</td>
</tr>
<tr>
<td>using in an object qualifier</td>
<td>182</td>
</tr>
<tr>
<td>bitmaps</td>
<td>134</td>
</tr>
<tr>
<td>BIUserID variable</td>
<td>179</td>
</tr>
<tr>
<td>setting up</td>
<td>180</td>
</tr>
<tr>
<td>using in an object qualifier</td>
<td>182</td>
</tr>
<tr>
<td>borders on buttons</td>
<td>153</td>
</tr>
<tr>
<td>boxes, design principles</td>
<td>133</td>
</tr>
<tr>
<td>buttons</td>
<td>21, 151</td>
</tr>
<tr>
<td>autosizing</td>
<td>153</td>
</tr>
<tr>
<td>creating</td>
<td>152</td>
</tr>
<tr>
<td>Dynamic Data Exchange (DDE)</td>
<td>155</td>
</tr>
<tr>
<td>executive</td>
<td>151</td>
</tr>
<tr>
<td>for creating tables in the database</td>
<td>156</td>
</tr>
<tr>
<td>link targets</td>
<td>155</td>
</tr>
<tr>
<td>buttons, appearance</td>
<td>153</td>
</tr>
<tr>
<td>borders</td>
<td>153</td>
</tr>
<tr>
<td>icons</td>
<td>154</td>
</tr>
<tr>
<td>pictures</td>
<td>154</td>
</tr>
<tr>
<td>resizing</td>
<td>153</td>
</tr>
<tr>
<td>text</td>
<td>154</td>
</tr>
<tr>
<td>buttons, linking</td>
<td>155</td>
</tr>
</tbody>
</table>

**B**

balancing network load | 233|

BI Query

overview | 17|

administrator | 18|

exiting | 35|

OLE object container | 144|

opening a data model | 32|

permissions, assigning | 183|

Reports | 22|

save-to-database tables | 217|

starting | 32|

BI Query Admin | 18|

administrator | 18|

update control | 272|

BI Query Reports | 22|

BI Query Update | 18|

BI Query User | 18|

BI Server | 27|

environment | 31|

logging on | 34|

Repository | 27|

system permissions | 28|

terminology | 28|

BI Server permissions |    |
to application scripts 158
  to hypercubes 159
  to queries 156
  to reports 159
  to URLs 166
  to windows 157
buttons, output
  default 155
  to applications 155
  to files 155
  to results windows 155

C
  Cache Passwords permission 49
  caching connection information 49
  calculated attributes
    permitting users to create 186
    permitting users to save 186
    updating 245
  case-sensitive searches 255
  catalog table, attribute descriptions 201
  check boxes for outer joins 96, 250
  checklist, data model design 77
  Classic (ANSI SQL/89) outer join syntax 95
Clipboard, importing ornaments 135
colors
  creating new 143
  design principles 133
  palette for 143
  specifying for Design windows 141
  specifying for individual Design windows 142
  specifying for individual objects 142
  specifying for objects 141
  specifying for windows 140
  specifying in data models 140
  transparent 143
columns
  access by users 173
  hiding columns for security 173
  removing columns for security 173
combined data models 22, 25
  changing 243
  distributing 211
  maintaining 243
  saving 211
  user changes in 184
  combining data objects 104, 107
  examples of 106
  when to combine objects 105
commands
  Paste Special 127
  Send to Back 135
common summarizations 72
Connect Automatically preference 50
Connect per Query preference 50
Connect per Window permission 51
connecting buttons
  to application scripts 158
  to hypercubes 159
  to queries 156
  to reports 159
  to windows 157
connecting data objects 85
  to actual relationships 90
connecting lines 87
connecting to DBMSs 56, 57
connection file
  DSNless 42
  freehand query 41
connection file folder 38
connection files 38
  assigning to Design windows 52
  changing the current connection 57
  connecting to a DBMS 56
  connection folder for 38
  connect-per-window guidelines 40
creating 43
current 41
default versus custom outer join support 59
deleting 53
editing 53
for metadata queries 41
legacy outer join information in 249
multiple configurations 42
optimizing with row buffering 47
outer join information in 58
permissions for 48
publishing 53
retrieving 55
setting security on 54
testing 47
troubleshooting 47
updating 48, 53
connection files, creating 43
connection files, default 39, 43, 56
  setting 49
connections
  automatic, setting 50
  creating files for 43
  network 37
  optimizing performance of 234
  preferences 44
  to DBMSs 43
types of 38
consistency in design 120
consolidating information with query data objects 108
container applications 145
content design considerations 120
controlling access 169
  to columns by users 173
  to information 172
  to rows by users 174, 176
  to tables 172
controlling database updates 272
controlling user capability 183
converting data objects
  to actual relationships 89
  to database tables 115
converting outer join syntax 250
converting queries to data objects 108, 110
copying objects in Design windows 127
correcting update errors 279
correlated subqueries 111
  creating 112
correlation names 101, 103
Create Calculated Attributes permission 186
creating tables
from button output 156
in the database 113, 115
in the database for efficiency 236
using SQL scripts 116
creator names in metadata queries
for DB2 195
for Oracle 195
Custom Checks permission 187
Custom Colors palette 143
customizing data models 23

D
data access, limiting 172
data dictionary in database 201
data model design 77
assigning permissions 183
business view of database 66
checking the design 117
complex data, unskilled users 69
controlling user access 169, 172
converting queries into data objects 110
creating data objects 114
deleting Design windows 79
Design windows 78
direct representation of database 66
effective layout 123
embedded relationships 89
for broad range of users 67
for independent users 67
increasing window size 124
join conditions 91, 97
large numbers of tables 68
organizing attributes 84
query data objects 108, 110
relationships between data objects 85, 89
renaming data objects 81
renaming Design windows 79
saving data models, overview 209
setting data model passwords 170
setting preferences 190
Data Model Is Visible in Portfolio permission 190, 228
data models 20, 77
adding value to, overview 191
applying database changes 245
checking for design errors 117
connect-per-window mode 40
customizing 23
determining requirements 63
finding items in 253
layout 123
legacy outer join operators in 249
loading metadata 199
loading tables from the database 79, 80
main components of 76
maintaining 241
minimizing file size 219
modular design 68
new outer joins syntax in 249
opening 32, 78
packages 224
permitting user changes 184
refreshing 244
renaming attributes 81, 83
renaming data objects 83
retrieving 33
saving initial version in database 216, 219
security 118
security requirements 64
simplifying display names 82
split 34
structure, overview 63
types 22
varied information needs 64
visibility in Portfolio 227
data models, combined 22, 25
  changing 243
  distributing 211
  maintaining 243
  saving 211
  user changes 184
data models, distributing 213
  by e-mail 215
  revisions via the database 220
  through the database, overview 216
  using a file server 214
  with BI Server 223
data models, passwords 32, 170
  changing 171
  removing 171
  setting 170
data models, saved in database
  distributing 216, 220
  saving revisions 220
data models, split 22, 23
changing 242
distributing 212
maintaining 242
saving 210
user changes 184
data models, types of
  ad hoc 67
  executive information system 69
  logical 66
  modular 68
  physical 66
data objects
  actual relationships as 89, 90
  assigning icons to 139
  converting from queries 110
  converting to database tables 115
  creating 79, 80, 114
  definition 20
  determining changes 244
differences, database and data models 244
  editing database names 248
  editing join conditions for merged 107
  names of, simplifying automatically 82
  qualifying with variables 182
  refreshing 244
  renaming 81, 83
  using more than once 101, 103
  variables, user and group 182
data objects, merging 104, 105, 107
  examples 106
  join conditions for merging 107
unmerging 108
  when to merge 105, 106
data separators and types, update errors 279
data values 70
  disabling 178
files, increasing efficiency 235
redirecting to lookup tables 70
database
  overview 18
adding metadata to 193
applying updates to 279
connections to 43, 56
differences with data models 244
moving 42
multiple, connecting to 51
organization and efficient queries 73
preparation for data model creation 70
records 18
  view permission 109
database capabilities
  default 61
  missing information 61
  options 59
  setting 61
database changes, applying to data models 244, 245
database names
  determining 204
  editing 248
  for Teradata metadata 195
  viewing 204
database preparation
adding indices 71
common summary information 72
creating lookup tables 70
denormalizing tables 71
providing metadata 74
reorganizing
table structures 72
database resources, estimating with DB2 or Teradata 238
database tables
  as data objects and relationships 76
created by BI Query 217
creating 113
creating directly in the database 115
loading 80
SQL scripts 116
DB2 databases
  creator names, metadata 195
  editing metadata template queries 198
  estimating database resources 238
DBAccess, row buffering 46
DBMS connections 18, 43, 56
DBUserID variable 179
  setting up 180
  using in an object qualifier 182
decreasing query time 71
default connection file 43
default outer join capabilities 61
deleting data model passwords 171
deleting Design windows 79
deleting objects in Design windows 128
denormalizing tables 71
descriptions for attributes 200
Design mode 27, 77
design principles 119
colors in data models 133, 140
consistency 120
content 120
effective layout 123
emphasis 131
graphics 133
layout grid 125
lines 126
ornaments 134
restraint and contrast 121
tools 123
typeface 131
unity 122
using boxes 133
white space 132
Design windows 21, 78, 79
assigning connection files to 52
copying objects 127
creating 78
deleting objects 128
increasing default size 124
moving objects 125
outer join syntax in 95
renaming 79
setting size 124
simplifying 104
specifying colors for 141, 142
disabling data values 178
disconnecting from DBMSs 58
display names
  from metadata 193
  simplifying automatically 82
displaying objects as icons
  embedded objects 147
  linked objects 146
displaying relationship icons 87
displaying tool tips 193
distributing data models 213
  by diskette 215
  by e-mail 215
  combined data models 211
  split data models 212
  using the database 216
  with a file server 214
  with BI Server 223
double ampersand string 137
DSNless connection files 42
Dynamic Data Exchange (DDE) button
  output 155
dynamic relationships 86
  permitting 186
  saving 187
Dynamic Relationships permission 186, 235
E
  Edit Data Model permission 184, 243
  Edit Freehand Queries permission 187
  Edit Model permission 187
  Edit SQL String permission 185
    when to assign 236
  editing
    shape ornaments 138
Index

editing database names
  of attributes 248
  of data objects 248
  of relationships 248
editing join conditions 101
EIS data models 69
e-mail
  using to distribute data models 215
embedded objects 145, 147
embedded relationships 86, 89
Enforce Connect per Query permission 50, 234
equijoins 91
errors
  in record updates, correcting 279
  in the design, checking 117
estimating database resources 238
executive buttons 21, 151
Executive Information System (EIS) 69
exiting BI Query 35
exporting results 155

F
file server, distributing data models 214
File Transfer Protocol (FTP) 165
file types
  .bmp 134
  .emf 134
  .gif 134
  .gqa 23, 212
  .gql 25, 211
  .gqu 23, 212
  .ico 134, 139
  .rtf 206
  .wmf 134
files
  administrator-layer 34
  connection 43, 47
  importing to update records 278
  personal 28
Rich Text Format. See .rtf
shared 28
size of, minimizing data model 219
storing split data model layers 212
URL protocol 165
Fill Color tool 141
forgotten password, retrieving 171
form view
  updating records 274, 276
freehand query
  connection files 41
FTP. See File Transfer Protocol
full outer joins 93
  using legacy operators 98

G
gateways 37
governor settings
  specifying using connection files 45
  specifying using permissions 237
gqlcheck.dll 187
GQLDATA attribute 219
GQLDLEN attribute 219
GQLDVTAB attribute 218
GQLFID attribute 219
GQLMETA2 table 217
GQLMnnnn table 217
GQLMODEL attribute 218
GQLMODL attribute 219
GQLMTAB attribute 218
GQLQnnnn table 217
GQLQRYTAB attribute 218
GQLREASON attribute 218
GQLREPTAB attribute 218
GQLRnnnn table 217
GQLSEQ attribute 219
GQLVERS attribute 218
GQLVnnnn table 217
hyperlinks
   defined 162
   using buttons as 162
HyperText Transfer Protocol (HTTP) 164
I
IBM DataGuide metadata queries 195
   editing 198
icons
   assigning to data objects 139
   dimensions 139
   file formats 134
   file types 139
   on buttons 154
   ornaments 135
   paper clip 206
   relationship 87
importing data using Clipboard 278
importing files to update records 278
importing ornaments from Clipboard 135
increasing query efficiency
   by creating query data objects 236
   by creating tables in the database 236
   by modifying SQL 236
   by restricting queries 235
common summary information 72
denormalizing tables 71
permitting users to create tables 188
providing metadata 74
ranking data objects with Oracle databases 238, 239
reorganizing table structures 72
table structures 73

half outer joins 93
help
   annotations in 206
   creating additional files 206
hiding attributes in tables 173
HRRP. See Hummingbird Repository Retrieval Protocol
HTTP. See Hypertext Transfer Protocol
HTTPS. See Secure Hypertext Transfer Protocol
Hummingbird accessibility 281
Hummingbird Repository Retrieval Protocol (HRRP) 163
hypercubes, linking to buttons 159
graphics
   as design elements 133
   file formats 134
   on buttons 154
grid
   layout 125
   settings 125
H

icons
   assigning to data objects 139
   dimensions 139
   file formats 134
   file types 139
   on buttons 154
   ornaments 135
   paper clip 206
   relationship 87
importing data using Clipboard 278
importing files to update records 278
importing ornaments from Clipboard 135
increasing query efficiency
   by creating query data objects 236
   by creating tables in the database 236
   by modifying SQL 236
   by restricting queries 235
common summary information 72
denormalizing tables 71
permitting users to create tables 188
providing metadata 74
ranking data objects with Oracle databases 238, 239
reorganizing table structures 72
table structures 73
using object qualifiers 174
with data values files 235
with governor settings 45, 237
indicators, update 279
indices, adding to tables 71
Informatica
    editing PowerMart template queries 198
    metadata queries 195
information
    data model structure 65
    meeting user requirements 64
inner joins 92
intercepting SQL 187

J
join conditions 91
    attribute data types 94
    creating 97
    database requirements 94
    editing 101
    editing for merged data objects 107
equijoins 91
in merged data objects 107
including prompts or variables 98
left and right sides 91
number required 94
outer joins 92
regular joins 92
requirements 94
updating 245
joins
    equijoins 91
    outer joins 91

K
kinking relationship lines 130

L
launching applications by using buttons 158
layers
    administrator 23
    folders for storing split data model 212
    user 23
left outer joins 93
legacy outer join operators 95, 249
    converting automatically 251
    converting manually 250
limiting data access 172
Line Color tool 141
lines
    as design elements 126
    editing 138
    ornaments 138
linked objects, editing 147
linking buttons
    overview 155
    to application scripts 158
    to hypercubes 159
    to queries 156
    to reports 159
    to windows 157
linking objects 145, 146
linking, source application requirements for 146

regular joins 92
types of 92
Index

links
  managing 148
  viewing 148
Load Comment button 200, 204, 206
loading metadata into data models 193, 199
loading tables from the database 79, 80
local data model 32
logging on to BI Server 31, 34
logical data models 66
lookup tables
  creating 70
  redirecting data values queries 70
metadata queries 196
  IBM DataGuide 195
  Informatica 195
  Prism 195
metadata queries, SQL for
  columns 196
  structure 195
tables 196
metadata tables
  creating 193
  loading into the data model 193
  populating 193
SQL for 195
metadata template queries
  editing 198
  for DB2 198
  for Oracle 198
  for Teradata 198
metafiles 134
middleware 37
  row buffering for 46
modular data models 68
moving objects within Design windows 125
multiple objects
  spacing evenly 127
metadata
  overview 192
  adding to the database 193
  alternatives 200
  as display names and tool tips 193
columns, SQL 195
creating tables 193
loading into data models 199
providing 192
queries 195
names
  correlation, specifying 103
  display, simplifying automatically 82
naming relationships 90
narrow tables 73

297
network connections 37, 43
network load 46, 233
    reducing 235, 236

0
Object Linking and Embedding (OLE)
    automation 158
    overview 144
    types of objects 145
    using with buttons 157
object qualifiers 174, 235
    controlling access to rows 176
    expression syntax 174
    prompts, variables, and subqueries 174
    updating 245
    using user and group variables 182
objects
    aligning 126
    copying in Design windows 127
    deleting in Design windows 128
    embedded, editing 147
    embedding 147
    for linking and embedding 144
    individual, specifying colors for 142
    linked, editing 147
    linking 146
    making same size 127
    moving within Design windows 125
    repeating 129
    represented by icons 146, 147
    selecting 123
    spacing evenly 127
    specifying colors 141

ODBC
    syntax for outer joins 95
offline, working 31
OLE. See Object Linking and Embedding
    opening data models 32
    new data models 78
operators
    *= 95
    *= 95
Oracle
    outer join syntax 95
    ranking data objects for efficiency 238, 239
Oracle databases
    creator names, metadata 195
    editing metadata template queries 198
    organizing attributes in data objects 84
ornaments
    as design elements 134
    autosizing 135
    creating 135, 138
    file formats 134
    icons 134
    special text strings in 136
ornaments, displaying
    icons 135
    lines 138
    pictures 135
    shapes 138
    text 135
ornaments, shape
    editing 138
outer join types
full outer joins 93
half outer joins 93
left outer joins 93
right outer joins 93
outer joins
database capabilities 59
default versus custom support 59
definition 92
missing capability information 61
setting database capabilities 61
support in connection files 58
syntax check boxes 96, 250
syntax incompatibility 249
troubleshooting missing operators 99
types of 92
using legacy joins as is 252
outer joins syntax 59
check boxes 96, 250
Classic (ANSI SQL/89) 95
converting Classic to new syntax 250, 251
in split data models 249
ODBC, Oracle, and ANSI SQL/92 95
preserving syntax consistency 249
syntax incompatibility 250
output, buttons 155
ovals
   editing 138
P
paper clip icon 206
password
   BI Server 34
passwords
   administrator 170
   and combined data models 170
   and split data models 170
data model 32, 170–171
   setting data model 170
   user 170
Paste Special command 127
pasting objects in Design windows 127
path variables
   overview 160
   creating 162
   deleting 162
   example of use 161
   in a BI Server environment 161
   modifying 162
permissions 183
   Allow Non-Select SQL 185
   Allow Unconnected Data Objects 235
   Allow Web Ad Hoc Queries 229
   Cache Passwords 49
   Connect per Window 51
   Create Calculated Attributes 186
   Custom Checks 187
   Data Model Is Visible in Portfolio 190, 228
   Dynamic Relationships 186, 235
   Edit Data Model 184, 187, 243
   Edit Freehand Queries 187
   Edit SQL String 185
   Enforce Connect per Query 50
Preserve Legacy Query Outer Joins 252
Rows Governor 237
Save Calculated Attribute In Model 186
Save Data Model 184, 187
Save Dynamic Relationships 85, 187
Save Passwords 49
Save Queries 186
Save Split Data Model 78
Share Queries and Reports 190
Share queries and reports 28
Submit to Named Table 188
Submit to Table 188
Time Governor 237
Update Database 187, 274
permissions for advanced users, suggested 184
permissions for changing data models 184
permissions for creating new tables 188
permissions for modifying SQL 185
permissions for naming new tables 188
permissions for novice users, suggested 183
permissions for saving queries 186
permissions for updating attributes 274
permissions for updating the database 187
permissions, assigning 183
personal files 28
physical data models 66
pictures
  file formats 134
  on buttons 154
  ornaments 135
placing icons on data objects 139
populating tables 116
attribute description tables 204
metadata tables 193
preferences
  Allow Dynamic Relationships 186, 235
  Connect Automatically 50, 234
  Connect per Query 50
  Rows Governor 44
  Show Join Style 92
  Show Relationship Names 81, 87
  Show Relationships 81, 87
  Show SQL String 185
  Time Governor 44
preferences for connection files 44
preparing the database 70
Preserve Legacy Query Outer Joins permission 252
Prism Directory Manager
  metadata queries 195
  template queries 198
prompts
  creating join conditions with 98
protocols, URL
  file 165
  FTP 165
  HTTP 164
  HTTPS 164
  mailto 165
  SSL 164
publishing
  connection files 53
data models 226
roadmap as XML 225
publishing data models 223, 225, 226
data model packages 224
preliminary steps for 225
requirements for 223

Q
QDO. See query data objects
qualifying
to increase efficiency 235
queries 22
ad hoc 22
attribute description, creating 204
converting to data objects 108, 110
correlated subqueries 111
governor settings 45, 237
increasing speed with indices 71
linking to buttons 156
metadata 195
refreshing 247
restored from query data objects 110
restricting on individual tables 174, 176
using legacy joins as is 252
using legacy outer join operators in 249
queries, saving
permission for users 186
Query Admin
administrator 18
control of updating 272
query data objects
overview 108
converting queries into 110
creating correlated subqueries with 112
editing 110
populating tables 116
reloading tables 116
to increase efficiency 236
updating 245
view permission requirements 109
query efficiency, increasing
by modifying SQL 236
by restricting queries 235
with data values files 235
with database tables 236
with lookup tables 70
with query data objects 236
quitting BI Query 35

R
ranking data objects for Oracle databases 239
recent data model 32
records 19
adding from other applications 276, 277
updating 276
rectangles
editing 138
refreshing data models 244
refreshing data objects 244
refreshing queries 247
refreshing query data objects 244
regular joins 92
relational database management system (RDBMS) 18
relationship lines
kinking 130
relationships 85
actual, creating 89
between data objects, overview 85
dynamic 86
embedded, creating 89
embedded, overview 86
icons 87
kinking lines 130
many-to-many 86
naming 90
one-to-many 86
reloading tables 116
removing attributes from tables 173
removing data model passwords 171
renaming attributes 81, 83
renaming data objects 81, 83
renaming Design windows 79
renaming relationships 90
reorganizing attributes 84
reorganizing table structures 72
repeating objects 129
replacing items in a data model 253, 268
reports
  BI Query Reports 22
  linking to buttons 159
  Standard Reports 22
Repository 27
resizing
  objects 127
  resizes buttons 153
  resizes Design windows 124
resources, network 233
restoring queries 110
restricting access to information 172
restricting queries on individual tables 174, 176
to increase efficiency 235
results 22
results, sending
to applications 155
to text files 155
retrieving
data models 33
retrieving forgotten user passwords 171
Rich Text Format (.rtf) 206
right outer joins 93
row buffering 46
  fine tuning 47
  setting 47
row-level security, using variables 182
rows
  controlling access by users 174, 176
  in databases 18
Rows Governor
  permission 237
  preference 44
Run mode 27

S
sample SQL, metadata queries 195
sans serif typefaces 131
Save Calculated Attribute In Model permission 186
Save Data Model permission 184
Save Dynamic Relationships permission 85, 187
Save Model permission 187
Save Passwords permission 49
Save Queries permission 186
Save Split Data Model permission 78
save-to-database tables 217
GQLMETA2 217
GQLMnnnn 217
GQLQnnnn 217
GQLRnnnn 217
GQLVnnnn 217
saving data models
  overview 209
  combined data models 211
  split data models 210
saving files
  .gqa files 210
  .gql files 211
  .gqu files 210
saving queries
  permission for 186
searching a data model 253
  archiving results 266
  replacing matched items 268
  replacing table names 269
search area 255
search categories 254
search options 255
searching for NULLs or blanks 267
using the Find and Replace dialog box 265
  viewing matched items 268
searching a data model, criteria
  for attributes 258
  for buttons 260
  for data objects 256
for ornaments 262
for relationships 263
for results sets 260
for variables 265
Secure Hypertext Transfer Protocol (HTTPS) 164
Secure Sockets Layer protocol (SSL) 164
security
  access to columns 173
  access to data values 177
  access to rows 174, 176
  access to tables 172
  checking 187
  controlling user capability 183
  data model design 64
  data model passwords 170
  for data models 118
  for individual users 179, 180
  hiding columns 173
  individual attributes 274
  limiting data access, overview 172
  removing columns 173
  row-level using predefined variables 179
  types of 169
  using object qualifiers 174, 176
selecting objects 123
Send to Back command 136
sending button output
  to DB tables 156
  to results windows 155
sending results
  to applications 155
  to tables in the database 156
to text files 155
serif typefaces 131
server 31
shape ornaments
editing 138
Share Queries and Reports permission 190
Share queries and reports permission 28
shared files 28
Show Join Style preference 92
Show Relationship Names preference 81, 87
Show Relationships preference 81, 87
Show SQL String preference 185
size
objects 127
Snap to Grid 125
special text strings
for ornaments and buttons 136
split data models
overview 23
administrator layer 23, 34
changing 242
distributing, overview 212
establishing a path to the administrator layer 213
incorporating user customizations 242
maintaining 242
outer join syntax consistency in 249
saving 210
storage 212
user changes 184
user layer 23
spreadsheet view
updating records 274, 276
SQL
permitting user modification 185
statement length 236
SQL scripts
creating tables with 116
SQL, for metadata queries
columns queries 196
editable queries 195
query structure 195
sample metadata queries 195
tables queries 196
SSL. See Secure Sockets Layer protocol
storing split data model layers 212
Submit to Named Table permission 188
Submit to Table permission 188
subqueries
correlated 111
suggested permissions
advanced users 184
novice users 183
T	

tables
attribute description, creating 202
attribute description, populating 204
controlling access by users 172
creating with buttons 156
creating with SQL scripts 116
database 18
denormalizing, database preparation 71
differences between database and data models 244
examples of structure 73
loading from the database 79, 80
merging data objects 107
optimum number for users 65
ownership, changing 269
permit naming 188
populating 116
reloading 116
reorganizing structures 72
structures, examples 73
user permission to create 188
using more than once 101, 103
wide versus narrow 73
tables in the database
   created by BI Query 217
   creating 113, 115
to increase efficiency 236
Technical Support 283
Teradata databases
   database names, metadata 195
   editing metadata template queries 198
   estimating database resources 238
terminology 20
testing
   connection files 47
text files, adding to records 277
text on buttons 154
text ornaments 135
   special strings in 136
text strings
   && 137
   &d 136
&dn 136
&t 136
Time Governor permission 237
Time Governor preference 44
toolbars 26
tools
design 123
Fill Color 141
grid 123
Line Color 141
tips, metadata displayed as 193
transparent objects 143
troubleshooting
   connection files 47
   missing outer join operators 99
typeface in design 131
types of joins 92
U
Uniform Resource Locators (URL)
   definition 162
   linking buttons to 166
   supported protocols for 164
unmerging data objects 108
Update Database permission 187, 274
   assigning 274
update indicators 279
updating 276
   attribute permissions 274
   control by Query administrator 272
data models 244, 245
database 18
database, permission 187
Index

errors 279
importing data with Clipboard 278
importing files 278
join conditions 245
links, automatically and manually 148
object qualifiers 245
preparing text files 277
query data objects 245
Update Database Permission 274
updating records
adding from other applications 276
error message 276
in form view 274, 276
in spreadsheet view 274, 276
more than one updating at a time 274
requirements 271
retrieving results 276
without retrieving results 276
URL. See Uniform Resource Locators
user customizations
  incorporating into split data models 242
  maintaining split data models 242
user layer, split data model 23
user name
  BI Server 34
user name variables, associating with data 180
user needs
  and data model structure 63
  balancing information needs and computer skills 63
  maintaining data models 241
  providing additional information 191
user-layer files, saving 210
users
  controlling access to information 169
  licenses, Connect per Query 235
  skills 65
  suggested permission settings 183
V
variables
  «table_name» 195
  BIGroupID 179
  BIUserID 179
  creating join conditions with 98
  DBUserID 179
  path 160
  using in an object qualifier 182
view permission 109
viewing
  links 148
  viewing database names 204
W
whole-word searches 255
wide tables 73
Window Fonts 131
windows
  Attribute 20
  linking to buttons 157
  resizing 124
  specifying colors 140
  working offline 31, 34
  working online 34
X

XML
  publishing roadmap 225