# Contents

## Chapter 1  Basic Concepts  

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introducing BI Query</td>
<td>15</td>
</tr>
<tr>
<td>BI Query Applications</td>
<td>16</td>
</tr>
<tr>
<td>Basic Concepts</td>
<td>16</td>
</tr>
<tr>
<td>Database Components</td>
<td>16</td>
</tr>
<tr>
<td>Data Model Components</td>
<td>17</td>
</tr>
<tr>
<td>Queries</td>
<td>20</td>
</tr>
<tr>
<td>Reports</td>
<td>20</td>
</tr>
<tr>
<td>Types of Data Models</td>
<td>20</td>
</tr>
<tr>
<td>Toolbars</td>
<td>24</td>
</tr>
<tr>
<td>BI Query Operation Modes</td>
<td>25</td>
</tr>
<tr>
<td>Running BI Query with BI Server</td>
<td>25</td>
</tr>
<tr>
<td>Personal Versus Shared Files</td>
<td>26</td>
</tr>
</tbody>
</table>

## Chapter 2  Getting Started  

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting BI Query</td>
<td>27</td>
</tr>
<tr>
<td>Opening Data Models</td>
<td>28</td>
</tr>
<tr>
<td>Retrieving Data Models</td>
<td>29</td>
</tr>
</tbody>
</table>
Chapter 3

Building Queries

Building Queries ................................................. 33
Selecting Attributes .................................................. 34
Running and Clearing Queries ....................................... 34
Cancelling Queries .................................................. 35
Reordering Columns Before Running Queries ....................... 35
Sorting Rows .......................................................... 36
The Queries Folder ................................................ 36
Saving Queries ...................................................... 37
Deleting Queries ................................................... 37
Opening Saved Queries ............................................... 38
Qualifying Queries ................................................ 38
The Qualification Tree ................................................ 39
Adding, Modifying, and Deleting Qualifications ...................... 39
Qualifying with Available Data Values .............................. 40
Applying And/Or Operators ......................................... 40
Ordering Qualifications ............................................. 40
Negating Qualifications ............................................. 41
Applying Query Modifiers .......................................... 41
Qualifying One Attribute with Another ............................ 41
Applying Qualification Operators .................................. 42
Comparing Numbers, Strings, and Dates ............................ 42

Finding the Administrator Layer in a Split Data Model .............. 30
Working Online or Offline ............................................ 30
Exiting BI Query .................................................... 31
Connecting to DBMSs ............................................... 31
 Disconnecting from DBMSs .......................................... 32

Contents
Chapter 4 Formatting Data 63

Data Formats in BI Query .............................................. 63
  Data Type Categories .............................................. 64
  Data Format Options .............................................. 65
  Format Order ....................................................... 66

Formatting Data: The Basics ................................. 67
  Opening the Format Dialog Box ......................... 68
  Specifying Default Formats for Data Models .......... 69
  Formatting Null Values ......................................... 70
  Formatting Data as Strings .................................... 70
  Formatting Data as Computed Strings .................. 72

Formatting Specific Data Types ............................ 73
  Formatting Boolean Data ....................................... 73
  Formatting Numeric Data ....................................... 74
  Formatting Date and Time Data ......................... 75
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formatting Data for Data Entry</td>
<td>76</td>
</tr>
<tr>
<td>Setting the Data Entry Format</td>
<td>76</td>
</tr>
<tr>
<td>Entering Date and Numeric Data</td>
<td>76</td>
</tr>
<tr>
<td>Converting Year Formats</td>
<td>77</td>
</tr>
<tr>
<td>Correcting Invalid Entries</td>
<td>78</td>
</tr>
<tr>
<td><strong>Chapter 5</strong> Qualifying with Data Values Files</td>
<td>79</td>
</tr>
<tr>
<td>Data Values Files</td>
<td>79</td>
</tr>
<tr>
<td>Types of Data Values Files</td>
<td>80</td>
</tr>
<tr>
<td>Naming and Storing Data Values Files</td>
<td>80</td>
</tr>
<tr>
<td>Order of Data Values Files</td>
<td>81</td>
</tr>
<tr>
<td>Creating Data Values Results Files</td>
<td>82</td>
</tr>
<tr>
<td>Creating Data Values Results Files with a Text Editor</td>
<td>83</td>
</tr>
<tr>
<td>Opening Data Values Results Files</td>
<td>84</td>
</tr>
<tr>
<td>Creating Data Values Query Files</td>
<td>84</td>
</tr>
<tr>
<td>Qualifying with Data Values Files</td>
<td>85</td>
</tr>
<tr>
<td>Data Aliasing</td>
<td>85</td>
</tr>
<tr>
<td>Creating Data Values Files to Alias Data</td>
<td>86</td>
</tr>
<tr>
<td>Modifying Data Aliases with a Text Editor</td>
<td>86</td>
</tr>
<tr>
<td>Viewing Aliased Data in a Query</td>
<td>87</td>
</tr>
<tr>
<td><strong>Chapter 6</strong> Qualifying with Prompts</td>
<td>89</td>
</tr>
<tr>
<td>Prompts in BI Query</td>
<td>89</td>
</tr>
<tr>
<td>Types of Prompts</td>
<td>90</td>
</tr>
<tr>
<td>Naming Conventions for Prompts</td>
<td>91</td>
</tr>
<tr>
<td>Data Types for Prompts</td>
<td>92</td>
</tr>
<tr>
<td>Required Versus Optional Prompts</td>
<td>93</td>
</tr>
</tbody>
</table>
Prompts for BI Web Reports ................................. 94
  Prompt Display Types for BI Web Reports ................. 95
  Non-Typed Prompts in BI Web Reports ....................... 97
Common Tasks Using Prompts ................................. 97
  Creating Single Prompts .................................. 98
  Qualifying with Prompts .................................. 99
  Creating Group Prompts .................................. 100
  Using Group Prompts ..................................... 100
  Creating Conditional Prompts ............................... 101
  Using Conditional Prompts ................................ 102
  Modifying or Deleting Prompts .............................. 102
  Importing Prompts ........................................ 103
  Showing Prompt Values in Queries ......................... 103
Specifying Web-Reporting Values ......................... 104
  Editing and Deleting Web-Reporting Data Values .......... 105

Chapter 7 Qualifying with Variables 107

Variables in BI Query ...................................... 107
  Predefined Variables ...................................... 108
  Typical Uses for Variables ................................ 108
Expressions for Variables .................................. 109
  Literals in Variables ...................................... 109
  Placeholders in Variables .................................. 110
  The Retain Attribute and Operator Option .................. 111
  Functions, Prompts, and Other Values in Variables ....... 112
Creating and Editing Variables ............................. 113
  Qualifying with Variables .................................. 114
  Importing Variables from Files ............................ 114
Chapter 9  Freehand Queries

Getting Started with Freehand Queries .......................... 142
  The Freehand Query Window .................................. 143
Creating and Editing Freehand Queries ......................... 144
  Creating a New Freehand Query .................................. 144
  Converting a Standard Query to a Freehand Query ............... 145
  Editing a Freehand Query ........................................ 145
Saving Freehand Queries ............................................. 148
Running Freehand Queries ......................................... 148
Sending Non-SELECT SQL to DBMSs from the Query Window .... 150
Working with Stored Procedures .................................... 150
  Calling Stored Procedures from Freehand Queries .............. 151
  Running Stored Procedures ..................................... 152
Using Freehand Queries in a BI Server Environment ............. 153

Chapter 10  Creating Standard Reports

Standard Reports in BI Query ......................................... 155
  Standard Reports Structure ...................................... 156
  Data Formats in Standard Reports .............................. 157
Basic Report Tasks ...................................................... 158
  Creating Standard Reports ...................................... 158
  Report Specifications .......................................... 158
  Saving Report Specifications .................................... 159
  Applying Report Specifications ................................. 160
  Printing and Exporting Standard Reports ....................... 161
Adding Calculations to Standard Reports .......................... 161
  Adding Subtotals ................................................. 162
Contents

Adding Grand Totals .............................................. 163
Adding Calculated Columns ................................... 163
Editing and Deleting Calculated Columns ................... 165

Manipulating Columns in Standard Reports .................. 166
Resizing Report Columns ...................................... 166
Combining Columns ........................................... 167
Banding Columns ............................................. 168
Stacking Columns ............................................ 169
Formatting Column Text and Headings ....................... 170
Customizing Column Settings .................................. 170

Adding Borders to Standard Reports ......................... 171

Adding Ornaments to Standard Reports ....................... 172
Special Text Strings in Standard Reports .................... 172
Adding Text Ornaments to Standard Reports ................ 174
Editing Default Ornaments in Standard Reports ........... 174

Chapter 11 Creating Labels ....................................... 177

BI Query Labels .................................................. 177
The Labels Window ............................................. 178
Creating and Editing Labels .................................... 180
Formatting Label Fields ........................................ 182
Specifying Label Field Font Type and Size .................. 183
Resizing Label Fields .......................................... 183
Positioning Label Fields ....................................... 183
Justifying Text Within a Field ................................ 183
Aligning Fields Within a Label ................................. 184
Aligning Two or More Fields Within a Label ............... 184
Centering Label Fields ......................................... 184
Chapter 12  Advanced Querying Techniques 185

Aggregate Functions and Grouping 185
Built-In Aggregate Functions 188
Types of Grouping 188
Applying Functions to Attributes 190
Creating Group Qualifications 190
Suspending Group Qualifications 192
Grouping Without Functions 192
Creating Calculated Attributes 193
Creating Calculated Attributes 193
Data Types for New Attributes 196
Including Functions in Query Expressions 197
Editing or Deleting Calculated Attributes 198
Qualifying with Subqueries 198
Qualifying with Subqueries 199
Previewing Subqueries 200
Qualifying with Data Sets 200
Data Set Format 201
Qualifying with Data Sets 201
Viewing Data Sets 202
Treating Data Values Results Files as Data Sets 202
Creating Tables in the Database from Query Results 203
Creating Tables in the Database from Query Results 204
Creating Tables Using Buttons 204
Deleting User-Defined Objects 204
Creating Dynamic Relationships 205
Choosing Join Operators .................................................. 205
Creating Dynamic Relationships ........................................ 206

**Existing Relationships in the Choose Relationship Dialog Box** .................................. 208
Deleting Dynamic Relationships ........................................... 209

**Outer Joins in Queries** .................................................... 210
Join Order in Merged Data Objects ........................................ 211
Specifying Join Order .......................................................... 212
Join Conflicts ...................................................................... 213
Outer Join Syntax in Queries ................................................ 215
Changing Outer Join Syntax for Connections .......................... 215

**Modifying the SQL for Queries** ........................................ 216
Modifying SQL in the Query Window ...................................... 216
Including SQL Prefixes and Suffixes ...................................... 217

**Building Circular Queries** ............................................... 218
Circular Queries with a Correlation ........................................ 219
Building Circular Queries with a Correlation ......................... 221
Circular Queries Without a Correlation ................................ 222
Building Circular Queries Without a Correlation .................... 223

Chapter 13  **Exporting Information** .................................. 225

**Exporting Information** .................................................... 225

**Preparing Results for Export** .......................................... 226
Results Options and Separators ............................................. 227
Specifying Results Options and Separators ........................... 229
Export Options ................................................................... 230
Setting Export Options ....................................................... 231

**Export Methods** ............................................................. 232
Exporting Results to Text Files ............................................ 232
Chapter 14 Publishing and Scheduling with BI Server

Publishing with BI Server ............................................. 241
Requirements for Publishing Queries ................................ 241
Publishing Queries .......................................................... 242
Retrieving Queries ........................................................ 243
Deleting Published Queries ............................................. 244
Setting Security for Queries ........................................... 244

Publishing Standard Report Specifications ..................... 245
Setting Security for Standard Reports ............................... 246
Standard Reports in BI Web ........................................... 246

Publishing Data Values Results Files ............................. 247
Retrieving Published Data Values Results Files ................. 247
Deleting Published Data Values Files .............................. 248

Scheduling with BI Server .............................................. 248
Scheduling Queries ....................................................... 249
Retrieving Scheduled Results ....................................... 249

Chapter 15 Accessibility and Technical Support ............... 251

General Accessibility ..................................................... 251
Microsoft Accessibility Options ..................................... 253
Contents

Technical Support ......................................................... 253
Using the Trace Utility ................................................... 254

Index ............................................................................. 255
Chapter 1

Basic Concepts

This section provides information on the following:

- “Introducing BI Query” on page 15
- “Basic Concepts” on page 16

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 “Standard Reports in BI Query” on page 155.

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>• Design windows</td>
<td>• Design 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><em>datamodell.gql</em>:</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>
</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 more information on using BI Query in a BI Server environment, see “Publishing with BI Server” on page 241.

**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 27
- “Connecting to DBMSs” 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.
• 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 on opening new data models, see the BI Query Data Models User’s Guide.

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 the BI Query Data Models User’s Guide.

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:
- 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.

Connecting to DBMSs

In order to query a database, BI Query must connect to the corresponding database management system (DBMS) running on the server. To make the connection to a DBMS, BI Query uses a connection file. Each connection file stores the information about the DBMS and the middleware that is necessary to make the connection.

💡 For information on creating connection files, see the BI Query Data Models User’s Guide.

Typically, the BI Query administrator includes at least one connection file in the data model (the default); the administrator may configure the model so that it automatically connects to the database each time you open the model or run a query. Even if the model has not been configured to connect automatically, BI Query attempts to connect if you run a query and a connection file is available.
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.

**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.

   ![Tip] 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.

**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 ![Disconnect].
This section provides information on the following:

- “Building Queries” on page 34
- “The Queries Folder” on page 38
- “Qualifying Queries” on page 41
- “Applying Qualification Operators” on page 50
- “Querying Multiple Data Objects” on page 56
- “The Query Window” on page 57
- “Updating Queries” on page 60
- “Viewing and Clearing Messages” on page 61
Building Queries

In order to retrieve information from the database, you need to build queries (sometimes called ad hoc queries) that ask the questions you want answered. To build a query, you choose categories of information and then decide how much of the information you actually want to receive in the results. You may want the query to perform calculations on the data it retrieves before providing you with results. You may also want to specify how rows are to be sorted and how columns are to be ordered in the results.

Use the attribute window to build a query. You can also use the Query window (in Query View mode) to build queries, once you have selected attributes for the query.

For more information on the Query window, see “The Query Window” on page 57.

When you run a query (submit it to the database), you choose whether to send the results directly to the results window for use in BI Query, to a text file for use in other applications, or to a new table in the host database for use in creating other queries.
You can also schedule queries to run at certain times if you have access to the BI Server Repository and have the appropriate publishing and scheduling permissions.

💡 For more information, see “Scheduling with BI Server” on page 248.

**Selecting Attributes**

You build a query by selecting attributes in a data object. The selected attributes appear as column headings when your query retrieves the information. As soon as you select one attribute, you have built a query. It is up to you to decide how much more you want your query to include.

For example, if you select the *Order Amount* attribute in a Sales data object and then run the query, BI Query returns the dollar amounts of all sales recorded in the database table, in a column labeled Order Amount. If you also select the *Year* attribute in the same data object and run the query, BI Query returns the order amounts, in a column labeled Order Amount, and the year in which each sale was made, in another column labeled Year.

You can select as many attributes as you like in each data object. And you can select attributes from as many data objects as you like within one data model window. (If your data model has data objects in more than one window, you can build separate queries in each window, and then use one of several methods for combining them.)

💡 For more information on combining query results, see “Combining Results Sets” on page 132.

If your data model contains many data objects, you can find the ones you want using the *Find and Replace* dialog box. For more information on searching a data model, see the *BI Query Data Models User’s Guide*.

**To select attributes:**

1. If a query is already open and you want to build a new one, click New on the Query menu.
2. Double-click the data object for which you want to retrieve information.
3. In the Attribute window, click the attributes that represent the categories of information you want to retrieve. A bullet appears beside the name of each selected attribute.
To select all attributes in the window, press CTRL+A. To deselect all attributes, press CTRL+A again. To toggle the selection of all attributes, press CTRL+SHIFT+A.

4. If an additional data object is connected by a relationship with the first, you can select attributes for your query from that data object as well.

For more information, see “Querying Multiple Data Objects” on page 56.

5. If your query is complete, click Run on the Query menu, or, on the Standard or Query toolbar, click the Run Query button.

Running and Clearing Queries

Once you have built a query, you can run it (submit it to the database). BI Query displays the results of the query in a separate results window named after the query.

You can build your entire query and then run it. Or you can specify some initial qualifications, run the query, review the results, and then revise the query to retrieve exactly the set of results you want.

To run a query:

1. Create, open or retrieve a query by doing one of the following:
   - Build a new query.
   - Open a previously saved query: on the Query menu, click Open and specify a query in the Open Query dialog box.
   - Retrieve an existing query from the BI Server Repository. For more information, see “Retrieving Queries” on page 243.

2. On the Query menu, click Run or, on the Standard or Query toolbar, click the Run Query button.

You can also run a query directly from either the Open Query dialog box or the Retrieve Query dialog box.

Queries remain current even after they have run. This may not be obvious if the query is not on the screen. You should always clear the previous query before starting a new one (unless you want to modify the previous query).

To clear a query:

- Do one of the following:
• On the **Standard** or **Query** toolbar, click the **New Query** button.
  
• On the **Query** menu, click **New**.

**Cancelling Queries**

After running a query, you may realize that you need to make some changes, or you may decide that it is taking too long to run. You can cancel a query only after rows have been returned. Not all databases support this function.

**To cancel a query:**

☐ Do one of the following:

  • On the **Query** menu, click **Cancel**.

  • On the **Standard** or **Query** toolbar, click the **Cancel Query** button.

**Reordering Columns Before Running Queries**

Columns appear in results windows and reports in the order in which you select attributes while creating the query. You can change that order before running the query, after you have results, or in a report.

💡 For more information about reordering columns in BI Query reports, see BI Query Reports Help.

**To specify the column order before running a query:**

1. On the **Query** menu, click **Reorder Attributes**. The **Reorder Query Attribute** dialog box opens. The top-to-bottom order of attributes in the **Attributes** list corresponds to the left-to-right order of columns in the results window or report.

2. From the **Attributes** list, select the attribute whose position in the list you want to change. Use the **Up**, **Down**, **Top**, or **Bottom** buttons to move the selected attribute to the desired position.

3. Click **OK**.
Chapter 3: Building Queries

Sorting Rows

When you run a query, BI Query displays the rows of results in the order that the DBMS returns them from the database, unless you specify a different way of sorting them. You can sort rows in ascending or descending alphabetical, numerical, or chronological order, as appropriate.

Use the Sort boxes in the attribute window to sort results; BI Query sorts results according to the order specified in the Sort boxes. You can sort attributes from more than one attribute window in the same query.

💡 You can also sort results after you have run a query. For more information, see “Sorting Results” on page 129.

For example, suppose that you select the Order Date and Order Amount attributes in a Sales data object, and then click the Sort boxes for Order Date and Order Amount, in that order. When you run the query, BI Query sorts the results chronologically by order date first, and then numerically by order amount.

**To specify a sort order in an attribute window:**

1. In the attribute window, select the attributes you want to sort.
2. Click the Sort box for each attribute in the sort order you want.
3. To reverse the sort method (ascending or descending) for any sorted attribute, click the Sort box again for that attribute.

The Queries Folder

By default, BI Query stores queries in the Queries folder for the data model. You can create subfolders in the Queries folder to organize your queries more efficiently. If you are using a split data model, there are two Queries folders: one for administrator-layer queries and one for user-layer queries. (Only the BI Query administrator can create subfolders in the administrator-layer queries folder.)

When you publish queries, the folder structure you have set up is preserved in the BI Server Repository. Similarly, when you retrieve queries, the folder structure in the Repository is reproduced on your computer.

💡 For more information on publishing and retrieving queries, see “Publishing with BI Server” on page 241.

You can save queries either as personal queries or shared queries. For more information, see “Personal Versus Shared Files” on page 26.
External Queries

You can save queries to any accessible location on your system. Queries that are not stored in the queries folder are known as external queries. You can open and run external queries from an accessible location. However, you cannot publish an external query or link it to a button.

Saving Queries

If the Save Queries permission is assigned, you can save queries in order to use them again in BI Query. You do not need to run a query before saving it. To save queries to the BI Server Repository, you must publish them. For more information, see “Publishing with BI Server” on page 241.

💡 For information on deleting saved queries, see the Help for your BI Query application.

You can also save the SQL for a query to a text file, provided that the query does not manipulate the results. The SQL can then be used in other applications. For instance, you can use the saved SQL in stored procedures in the database.

💡 You cannot save the SQL for a super query to a text file.

To save a query:

1. With a query open, do one of the following:
   - On the Query toolbar, click the Save Query button.
   - On the Query menu, point to Save, and then click Query.

     The Save Query dialog box opens.

2. Do one of the following:
   - To save the query as a shared query, click Shared Queries. This option is available only if you are using BI Query Admin.
   - To save the query as a personal query (not accessible to other users), click My Queries. This option is not available if you are using BI Query Admin.
   - To save the query as an external query, click External Queries. You cannot publish external queries or link them to buttons.

💡 For more information on shared and personal queries, see “Personal Versus Shared Files” on page 26.
3. Navigate to the location where you want to save the query. You can create a subfolder by clicking the Create New Folder button.

4. In the File Name box, type a name for the query.

5. From the Save as Type list, select the file type:
   - Select BI Query Queries to save the query as a BI Query file (with extension .qry).
   - Select BI Query Script Files to save the SQL for the query to a text file (with extension .sql).
   - Select All Files to save the SQL for the query to a text file with whatever file name and file extension you choose.

6. Click Save.

   If you save a query while a results set or report is active, BI Query saves the query that generated the results or report. If any other window is open, BI Query saves the current query.

Deleting Queries

BI Query User/Update users can delete personal queries but not shared queries; shared queries are identified with a lock icon in the Save Query and Open Query dialog boxes. Only BI Query Admin users can delete shared queries.

To delete a query:
   - In the Save Query or Open Query dialog box, select a query; then, click Delete.

Opening Saved Queries

You can run saved queries. Locally stored queries (in a data model that you can access without logging on to BI Server) are opened; queries stored in the BI Server Repository must be retrieved.

For information on retrieving a published query, see “Retrieving Queries” on page 243.

To open a saved query:
   1. Open the Open Query dialog box by doing one of the following:
      - On the Standard or Query toolbar, click the Open Query button.
      - On the Query menu, point to Open, and then click Query.
2. Do one of the following:
   - To open a shared query, click **Shared Queries**. This option is available only if any of the following is true: the data model is split; you are using BI Query Admin.
   - To open a personal query, click **My Queries**. This option is not available if you are using BI Query Admin.
   - To open an external query, click **External Queries**.

   For more information on shared and personal queries, see “Personal Versus Shared Files” on page 26.

3. Do one of the following:
   - To open a standard, super, or freehand query, click **BI Query Queries** from the Files of Type drop-down list. This option is selected by default.
   - To load SQL into the **Freehand Query** window from a text-based file with a .SQL extension, click **BI Query Script Files** from the Files of Type drop-down list.
   - To load SQL into the **Freehand Query** window from a text-based file with an extension other than .SQL, click **All Files** from the Files of Type drop-down list.

   If you have not been assigned the Edit Freehand Query permission, and you attempt to load a text-based file, the file will be loaded into the default application associated with the type of file you selected (such as Notepad for .txt files).

4. Select the file you want to open.

5. Do one of the following:
   - Click **Run** to open the query and run it. This option is not available if you selected a text-based file to open.
   - Click **Open** to open the query without running it.

6. BI Query displays the selected query in the appropriate window (Query window, Super Query window, or Freehand Query window).

**Qualifying Queries**

You can limit query results to a subset of the available information by qualifying one or more attributes. In general, a qualification takes the following form:
attribute operator value
where:

- **attribute** is the name of the attribute you are qualifying
- **operator** is the logical operator that specifies the nature of the comparison
- **value** is the qualification value or values you want to compare against the values of the attribute

For example, if you want to return rows that contain data for sales greater than $10,000, you could qualify the Sales attribute using the following expression:

```
sales > 10000
```

You qualify an attribute by clicking its Qualify box in the attribute window or Query window (in Query View mode), and then specifying the qualification in the qualification tree.

### The Qualification Tree

The qualification tree appears at the bottom of the attribute window or Query window (in both modes) when you click the Qualify box for an attribute.

- **Condition box**
- **Qualification box**
- **Operator list**
- **List button**
- **Data Values button**

💡 For more information on the Query window, see “The Query Window” on page 57.

#### Attribute Name Box

Displays the name of the qualified attribute.

#### Condition Box

Lets you combine multiple qualifications. You can also use the Condition box to reverse the effect of a particular qualification. To remove a qualification from a query, click the Condition box; then, hit the DELETE key.
Operator List

Lets you specify the type of restriction you want. For example, you can apply the \( > \) (Greater Than) operator to find all rows in which the specified attribute has a value greater than the specified qualification value.

Qualification Box

Lets you specify the qualification value(s) for the attribute. For example, by qualifying the Country attribute with the values Canada, France, Germany, and UK, you can restrict the results to those countries. After entering a qualification, click outside the qualification tree. If you entered one value, it appears in the qualification tree; if you entered more than one value, click the List button to view the values.

If your database is case sensitive and the Case-sensitive qualification preference is set, type the qualifications in the case in which they are stored in the database.

List Button

Lets you add, modify, or delete entries in the list of qualification values. Values that you enter in the Qualification box are displayed in a list when you click the List button. By clicking a value in the list, you can add that value to the Qualification box. Alternatively, use the Up or Down arrow keys on your keyboard to display each value in the Qualification box.

Data Values Button

Lets you insert into the qualification one or more of the following:

- Data values. For more information, see “Qualifying with Available Data Values” on page 45.
- Data sets. For more information, see “Qualifying with Data Sets” on page 201.
Chapter 3: Building Queries

- **Prompts.** For more information, see “Qualifying with Prompts” on page 99.
- **Variables.** For more information, see “Qualifying with Variables” on page 114.
- **Subqueries.** For more information, see “Qualifying with Subqueries” on page 199.

### Adding, Modifying, and Deleting Qualifications

You can qualify multiple attributes in a given query and qualify a given attribute more than once. You do not need to select an attribute to qualify it. Selecting an attribute indicates only that the attribute appears in the results.

💡 To cancel a qualification, click the **Condition** box (if there are no values in the **Qualification** box). If there are qualification values, click the **Condition** box; then, press the DELETE key.

**To qualify an attribute:**

1. In the attribute window or **Query** window (in Query View mode), click the **Qualify** box for the attribute. A qualification tree appears at the bottom of the window.

2. In the qualification tree, select an operator from the **Operator** list. (The default operator is =.)

💡 For more information on available operators, see “Applying Qualification Operators” on page 50.

3. Qualify the attribute by doing one of the following:
   - In the **Qualification** box, type the qualification value. To type more than one value, hit the down arrow key between successive values.
   - Click the **List** button and select a previously entered value from the pop-up menu that opens.
   - Click the **Data Values** button and select one or more data values, data sets, prompts, variables, or subqueries from the pop-up menu that opens.

4. Press ENTER (or click outside the qualification tree).

**To add a new comparison value to the qualification:**

1. Click the **List** button; then, click **New Entry** from the pop-up menu that opens.

2. In the **Qualification** box, type the new value; then, press ENTER (or click outside the qualification tree).
To modify existing values:
1. If you have qualified with multiple values, click the List button; then, click the value you want to modify.
2. Click in the Qualification box; then, type over the value to modify it.

To delete values:
☐ Do one of the following:
   • Click in the Qualification box to highlight the value; then, press the DELETE key.
   • If you have qualified with multiple values, click the List button. To delete the entire set of values at once, click Delete All Entries. Otherwise, click the value you want to delete; then, with the value highlighted in the Qualification box, press the DELETE key.

You can also add, modify, and delete qualifications using the Query window (in Query View mode). For more information, see “The Query Window” on page 57.

Qualifying with Available Data Values
The Data Values button in the qualification tree displays a pop-up menu that lets you select a data value (or set of data values) to insert into the qualification for an attribute.

When you click Data Values from the pop-up menu, BI Query runs a default SELECT DISTINCT query that retrieves all the data values in the database for that attribute and displays them in the Data Values dialog box, unless the BI Query administrator has disabled data values for that attribute.

If a data values file has been created for the attribute, BI Query instead displays the list from that file. The list can be a timesaving subset of the data values stored in the database.

For more information on data values files, see “Data Values Files” on page 79.

If you need to access all the data values in the database instead of just those in the subset, you can click Load All in the Data Values dialog box.
To qualify an attribute with a data value:

1. In the attribute window or Query window (in Query View mode), click the Qualify box for the attribute you want to qualify.
   
  💡 For more information on the Query window, see “The Query Window” on page 57.

2. In the qualification tree for the attribute, click the Data Values button. From the pop-up menu that opens, do one of the following:
   - Click the value from the list of available values at the bottom of the menu. These values appear only if the attribute has a data values file associated with it.
   - Click (all) to insert all listed values into the qualification.
   - Click Data Values to open the Data Values dialog box. Use this dialog box to select the value or values that you want to insert into the qualification; then, click Insert.

💡 You can view the data values for any attribute (and insert them into the qualification for any attribute) using the Choose Data Values dialog box: on the Insert menu, click Data Values.

Using Data Values in Freehand Queries

You can include data values (or sets of data values) in freehand queries.

To add data values to a freehand query:

1. Create or open the freehand query you want to edit.

2. In the Freehand Query window, position the cursor at the point where you want to add the data value (or data values).

3. On the Insert menu, click Data Values.

4. In the Choose Data Values dialog box, select the data value (or set of data values) you want to use, then click Insert.

💡 When you insert data values into a freehand query, BI Query does not automatically apply any syntax (such as quotation marks or parentheses). You must manually apply the appropriate syntax for the database you are going to be using.
Applying And/Or Operators
When you qualify more than one attribute, or when you qualify the same attribute more than once, BI Query assumes that the query must satisfy both qualifications. This is reflected in the qualification tree, where the word and appears between two qualifications. You can change and to or.

To specify that the query must satisfy either or both of the qualifications:
- Click the word and. It changes to or.

Ordering Qualifications
You can avoid ambiguities in a query by combining two or more qualifications. BI Query executes combined qualifications as if they were a single qualification. Combining qualifications is equivalent to using brackets in a mathematical expression.
The following qualification is ambiguous:

This qualification may return either of the following results:
- Either all rows for France in which the receivable is greater than 60,000 or all rows for 2002:
  \((\text{Country} = \text{France} \text{ and } \text{Receivable} > 60000) \text{ or } (\text{Year} = 2002)\)
- All rows for France in which either the receivable is greater than 60,000 or the year is 2002:
  \((\text{Country} = \text{France}) \text{ and } (\text{Year} = 2002 \text{ or } \text{Receivable} > 60000)\)
To resolve the ambiguity, combine the qualifications.

To combine qualifications:
1. In the qualification tree, select the qualifications to be combined by holding down the SHIFT key and clicking the corresponding condition boxes.
2. Do one of the following:
   - On the Query menu, point to Qualification; then, click Combine.
   - On the Query toolbar, click the Combine Qualifications button.
3. To change the condition in the new branch from and to or, click the word and.
To separate combined qualifications:

1. In the qualification tree, click the condition box corresponding to the combined qualification that you want to separate. (To select more than one, hold down the SHIFT key and click each box.)

2. Do one of the following:
   • On the Query menu, point to Qualification; then, click Uncombine.
   • On the Query toolbar, click the Uncombine Qualification button.

Negating Qualifications

You can make a qualification have the opposite meaning to the condition specified. This is useful when you wish to exclude certain information from a query, or when it is easier to specify the information you do not want than to specify the information you do (for example, if you want all customers located outside Illinois and Indiana).

To negate a qualification:

1. In the qualification tree, select the qualification by clicking the corresponding Condition box. (To select more than one Condition box, hold down the SHIFT key and click each box.)

2. Do one of the following:
   • On the Query menu, click Qualification; then, click Negate Clause.
   • On the Query toolbar, click the Negate Qualification button.

   The Not indicator appears in front of the condition box.

💡 To remove the Not indicator, click the indicator or click the Negate Qualification button.

Applying Query Modifiers

Query modifiers let you change the way a query operates. There are three query modifiers in BI Query:

• Distinct—Removes duplication to retrieve a list of unique values. For example, the database may record information about stores in many different countries. A query that lists these countries can include duplicates if there are several stores in the same country.
Qualifying Queries

- Count All—Specifies how many times a given item occurs in the database (for example, the number of times each country name occurs in the database). When you apply this modifier, BI Query adds a column, COUNT (*), to the results. This column lists for each row the number of times that row appears in the database.

- Trim Relationships—Removes from a query any relationships to data objects that do not have any attributes selected in the query. You can also set this option in the Preferences dialog box.

  For more information on this option, see “Removing Unnecessary Joins from Queries” on page 57.

**To apply a query modifier:**

Do one of the following:

- With an attribute window active, point to **Modifiers** on the **Query** menu; then, click the modifier.

- Alternatively, to apply the Distinct or Count All modifiers, you can click the **Add Distinct Modifier** or **Add Count All** buttons on the **Query** toolbar. You must apply the modifier before running the query.

Some databases perform better if you group attributes rather than apply the Distinct modifier to the query. For more information on grouping attributes, “Grouping Without Functions” on page 192.

Qualifying One Attribute with Another

You can build queries in which one attribute qualifies another, so that the values of the second limit the values returned by the first. When you use an attribute as a qualifier, you must use the database name of the attribute, not its display name. You can also use calculated attributes as qualifiers.

  For more information, see “Creating Calculated Attributes” on page 193.

**Example**

To get a listing of all profitable products, you define profitability as unit price at least ten per cent higher than unit cost:

Unit Price > (Unit Cost * 1.10)
To qualify the Unit Price attribute with the Unit Cost attribute, you cannot type
Unit Cost * 1.10 in the Qualification box: BI Query would interpret Unit Cost as text, not as an attribute name. You must insert the database name for attribute Unit Cost.

To qualify one attribute with another:
1. In the qualification tree, click the Qualification box for the first attribute.
2. On the Edit menu, click Insert Attribute Name. The Insert Attribute Name dialog box opens. The dialog box lists the database name for each selected attribute.
3. From the Data Objects list, select the data object that contains the qualifier attribute.
4. From the Attributes list, select the qualifier attribute; then, click Insert.
5. From the Operator list for the qualified attribute, select the appropriate operator.
6. If you want to include a calculation, click in the Qualification box, and type the calculation. BI Query automatically delimits the expression with brace brackets (for example, \{Sales.order * 100\}).

Applying Qualification Operators

When you create a qualification, you must select an operator that specifies how BI Query compares the qualification values against the qualified attribute. You select the operator from the Operator list in the qualification tree.

The operator you select is based on the data type of the attribute and the information you want to retrieve from the database. Depending on the operator, you can specify one or more qualification values:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Number of Qualification Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS NULL</td>
<td>No values.</td>
</tr>
<tr>
<td>IS NOT NULL</td>
<td></td>
</tr>
<tr>
<td>Operator</td>
<td>Number of Qualification Values</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------</td>
</tr>
<tr>
<td>&gt;(Greater than)</td>
<td>One value.</td>
</tr>
<tr>
<td>&gt;=(Greater than or equal)</td>
<td></td>
</tr>
<tr>
<td>&lt;(Less than)</td>
<td></td>
</tr>
<tr>
<td>&lt;=(Less than or equal)</td>
<td></td>
</tr>
<tr>
<td>==(Equal)</td>
<td>One or more values.</td>
</tr>
<tr>
<td>!=(Not equal)</td>
<td></td>
</tr>
<tr>
<td>IN</td>
<td>If you use the = operator and supply multiple qualification values, the operator automatically changes to IN in the query. Similarly, the != operator changes to NOT IN for multiple qualification values.</td>
</tr>
<tr>
<td>NOT IN</td>
<td></td>
</tr>
<tr>
<td>Begins with</td>
<td></td>
</tr>
<tr>
<td>Contains</td>
<td></td>
</tr>
<tr>
<td>Ends with</td>
<td></td>
</tr>
<tr>
<td>Does not begin with</td>
<td></td>
</tr>
<tr>
<td>Does not contain</td>
<td></td>
</tr>
<tr>
<td>Does not end with</td>
<td></td>
</tr>
<tr>
<td>BETWEEN</td>
<td>Two values (for an upper and lower range).</td>
</tr>
<tr>
<td>NOT BETWEEN</td>
<td></td>
</tr>
</tbody>
</table>

If you do not select an operator, BI Query uses the = operator by default, and finds all records in the database that match the qualification.
Comparing Numbers, Strings, and Dates

BI Query supplies the following operators that let you compare numbers, character strings, and dates. You can use these operators to find values that are larger or smaller, earlier or later, or equal to the qualification value. For strings and dates, the < operator (Less Than) means earlier in the alphabet or in chronological order; the > operator (Greater Than) means later.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Qualification Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>= (Equal)</td>
<td>Location = France</td>
</tr>
<tr>
<td></td>
<td>Returns all locations in France.</td>
</tr>
<tr>
<td>!= (Not Equal)</td>
<td>Location != France</td>
</tr>
<tr>
<td></td>
<td>Returns all locations except for France.</td>
</tr>
<tr>
<td>&gt; (Greater Than)</td>
<td>Order Amount &gt; 500</td>
</tr>
<tr>
<td></td>
<td>Returns order amounts larger than but not equal to 500.</td>
</tr>
<tr>
<td>&gt;= (Greater Than Or Equal)</td>
<td>Order Amount &gt;= 500</td>
</tr>
<tr>
<td></td>
<td>Returns order amounts of 500 or more.</td>
</tr>
<tr>
<td>&lt; (Less Than)</td>
<td>Last Name &lt; Davis</td>
</tr>
<tr>
<td></td>
<td>Returns names that come before Davis—Anders, Coopers, Dale, and so on.</td>
</tr>
<tr>
<td>&lt;= (Less Than Or Equal)</td>
<td>Last Name &lt;= Davis</td>
</tr>
<tr>
<td></td>
<td>Returns Davis or names that come before it.</td>
</tr>
</tbody>
</table>

Matching Character Patterns

If you want to find records that have (or do not have) a particular pattern, or if you are not sure how a value is stored in the database, you can match character patterns using the following operators:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Qualification Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Begins With</td>
<td>Last name Begins with Gre</td>
</tr>
<tr>
<td></td>
<td>Returns Green, Gregory.</td>
</tr>
</tbody>
</table>
Character Pattern Options

When you use the character pattern operators, you can include wildcard characters in the search string to make less specific searches of the DBMS. There are two wildcard characters:

- `%` represents a character string of any length (including no length)
- `_` represents a single character

The following table provides some examples:

<table>
<thead>
<tr>
<th>Qualification</th>
<th>Possible Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Begins with <code>_r</code></td>
<td>Anything that has r as the second letter</td>
</tr>
<tr>
<td>Ends with Smith%</td>
<td>Ms Smith, Smithers, Smithsonian</td>
</tr>
<tr>
<td>Contains L%t%d%</td>
<td>Limited, Ltd., Ltd, Laminated</td>
</tr>
</tbody>
</table>

If you supply multiple qualification values when using a character pattern operator, you can choose to match any of the values or all of them. When you build the query, select either **Match Any** or **Match All** from the List box in the qualification tree.
Chapter 3: Building Queries

Matching Items in a List

When you qualify an attribute with a list, BI Query automatically applies the **IN** operator to the list. The **IN** operator lets you find all records that match any one item in the list. The **NOT IN** operator lets you find items that do not match those in a list.

**Examples**

The following table describes an example of use of the **IN** and **NOT IN** operators in a qualification.

<table>
<thead>
<tr>
<th>Qualification</th>
<th>Possible Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country <strong>IN</strong> France, Germany, Spain</td>
<td>France, Germany, Spain</td>
</tr>
<tr>
<td>Country <strong>NOT IN</strong> France, Germany, Spain</td>
<td>All countries except France, Germany, Spain</td>
</tr>
</tbody>
</table>

Finding Values in a Range

You can qualify an attribute with a range of values to obtain results within a particular range. The range is inclusive, retrieving the lower value and the upper value as well as the values in between.

To qualify with a range of values, enter the values as a qualification list and apply the **BETWEEN** operator. You can also find results that are outside a particular range by applying **NOT BETWEEN**. If you qualify a character field, be sure you use the same case (upper or lower) in which the value is stored in the database.

**Examples**

The following table describes an example of use of the **BETWEEN** and **NOT BETWEEN** operators in a qualification.

<table>
<thead>
<tr>
<th>Qualification</th>
<th>Possible Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit Limit <strong>BETWEEN</strong> 2,000 and 5,000</td>
<td>Credit limits between and including 2,000 and 5,000</td>
</tr>
<tr>
<td>Credit Limit <strong>NOT BETWEEN</strong> 2,000 and 5,000</td>
<td>Credit limits outside this range</td>
</tr>
</tbody>
</table>
Determining if Data Is Missing

A NULL value occurs when no value has been entered into the database. It does not represent a zero or a character space but is instead a placeholder for missing information. Because NULLs do not match anything, you cannot retrieve them using operators such as =, <, or >. To include or exclude rows for which information has not been recorded, apply the `IS NULL` or `IS NOT NULL` operator to an attribute.

To include or exclude rows containing NULLS:

1. In the attribute window or Query window (in Query View mode), select the attribute to qualify and select the Qualify box in the same row.

   ☀️ For more information on the Query window, see “The Query Window” on page 57.

2. In the qualification tree, from the Operator list for the selected attribute, select the `IS NULL` operator to include rows with NULL values or the `IS NOT NULL` operator to exclude rows with NULL values.

3. On the Query menu, click Run. Alternatively, on the Standard or Query toolbar, click the Run Query button.

The above method does not work if missing information is represented by a space character rather than a NULL. In this case, you have to query for the space character itself.

To find rows containing a character space:

1. In the attribute window or Query window (in Query View mode), select the attribute to qualify and select the Qualify box in the same row.

2. In the qualification tree, from the Operator list for the selected attribute, select the = operator.

3. Type `{' '}` in the Qualification box. Be sure to put a space between the single quotation marks.

4. On the Query menu, click Run. Alternatively, on the Standard or Query toolbar, click the Run Query button.

   ☀️ You can combine the two queries with the OR operator to test if information is missing regardless of whether it is a NULL condition or a space. For more information, see “Applying And/Or Operators” on page 47.
Applying DBMS Operators

In addition to applying the standard SQL operators provided in BI Query, you can restrict a qualification using any other operator that your DBMS supports. (For information about operators your DBMS supports, see the documentation for your DBMS.)

To apply your own operator:

1. In an attribute window or Query window (in Query View mode), select Custom from the Operator list in the qualification tree.
2. In the Custom Operator dialog box, type the operator in the Operator box, and then click OK.

Querying Multiple Data Objects

You can create a query involving more than one data object, as long as a relationship exists between the data objects. BI Query depicts relationships as lines drawn between data objects in a Design window.

💡 The more objects used in a query, the longer the query takes.

As you build your query, the qualification tree at the bottom of the attribute window and the Query window (in Query View mode) displays all qualifications you have added to the query, regardless of the attribute on which they were placed.

If there is no relationship between two data objects, you can create a dynamic relationship between them.

💡 For more information on dynamic relationships, see “Creating Dynamic Relationships” on page 205.

To query multiple data objects:

1. Double-click each object in turn and select the attributes that you want. If the objects are directly joined, BI Query automatically highlights the relationship between them.
   
   If there are any other objects between the two you want to access, select the attributes from the first object, click the lines linking it to the second data object, and then select the attributes from the second object. BI Query makes the necessary adjustments to your query.
2. When you have finished building the query, click the Run Query button on the Standard or Query toolbar.

Removing Unnecessary Joins from Queries

The relationship between two data objects consists of one or more join conditions that specify how the data objects are related. When you run a query that involves both objects, any rows returned from the DBMS must satisfy the join conditions between the objects. The join conditions affect the number of rows returned even if you do not include in the query the attributes from one of the joined objects.

💡 For more information on join conditions, see the BI Query Data Models User’s Guide.

You can set the Trim Relationships preference to remove such excess joins from the SQL for a query. If a query contains a relationship to a data object that has no attributes selected, the Trim Relationships preference removes the relationship when you run the query.

If you do want the join conditions to apply to the query, but do not want to include both objects in the query, you must select their relationship and disable this preference before running the query.

To set the Trim Relationships preference:

- Do one of the following:
  - On the Tools menu, click Preferences. In the Preferences dialog box, select Trim Relationships.
  - Open or create a particular query. On the Query menu, point to Modifiers; then, click Trim Relationships (if it is not already selected).

The Query Window

The Query window records the total information you have specified for a query, including the following:
- the attributes you have selected and created
- the qualification tree
- the group and sort operations you have applied to the attributes
As you build a query by selecting attributes from data objects and applying qualifications and other operations, the information in the Query window automatically updates.

**Query Window View Modes**

The Query window has two views:

**Query View**

Presents the total query information in the same format as any attribute window. Only selected attributes appear in the window. You can edit the query in the Query view as you would in any attribute window. However, you must still use attribute windows to select attributes and to qualify non-selected attributes.

**SQL View**

Presents the actual SQL for the query, if the Show SQL String preference is set. If the Edit SQL String permission is also assigned, you can use the SQL view to modify the SQL (anything after the `SELECT` key word). If you have already run the query, you can display the values of any prompts used in the query by selecting Show Prompt Values.

💡 Changing the SQL in the SQL view is subject to any restrictions placed on you by the DBMS. For more information, see “Modifying the SQL for Queries” on page 216.

**To switch between the Query and SQL views:**

- Do one of the following:
  - On the Query toolbar, click the Toggle Query Window button 🔁.
  - On the Query menu, point to Window, and then click Query or SQL.
Query Window States

You can view the Query window in one of two states: docked or floating. When the Query window is docked, it occupies a fixed space on any edge of the main program window. When the Query window is floating, it appears in its own window within the main program window or the desktop.

To turn the docked window into a floating window, drag it into the main program window. To turn the floating window into a docked window, drag its title bar to an edge of the main window.

💡 To move the floating Query window over an edge of the main window without docking the Query window, press the CTRL key before moving the window.

To view the Query window:

- Do one of the following:
  - On the Query menu, click Show.
  - On the Query or Standard toolbar, click the Show Query button 🔄.
### Updating Queries

After the administrator has updated a data model, some of the saved queries may not work—for example, they may include attributes that no longer exist in the database. When you open 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 you to refresh the queries.

The administrator may have also changed the syntax type for outer joins in the data model, in which case the saved queries that include those joins may no longer work correctly. If you enable the **Preserve Legacy Query Outer Joins** preference, BI Query treats outer joins as legacy joins when you run a saved query.

### Refreshing Queries

When you refresh a query, BI Query compares the queries with the data model and displays information on each query that has been affected by changes to the data model. BI Query can detect if attributes have been deleted, but not if they have been changed.

**To refresh queries:**

1. Ensure that BI Query is not in Design mode.
2. On the **Tools** menu, click **Refresh Queries**.
3. In the **Refresh Queries** dialog box, do one of the following:
   - To refresh shared queries (or shared data values queries), click **Shared Queries** (or **Shared Data Values**). This option is available only if you are using BI Query Admin.
     - For more information on shared and personal files, see “**Personal Versus Shared Files**” on page 26.
   - To refresh personal queries (or personal data values queries), click **My Queries** (or **My Data Values**). This option is not available if you are using BI Query Admin.
     - For more information on data values queries, see “**Types of Data Values Files**” on page 80.
4. Select the queries you want to refresh; then, click **Refresh**.
5. If BI Query detects any changes, do one of the following in the dialog box that opens:
• Click Change to update the query.
• Click Change All to update all the queries selected.
• Click Skip to leave the query unchanged and view the next one.
• Click Cancel to return to the Refresh Queries dialog box without comparing the remaining queries.

If you modify any queries, BI Query makes the changes, and then saves the queries in the Queries folder. BI Query also saves the original queries with the extension .bak.

The Message window summarizes the results of the refresh operation.

6. Run any modified queries to make sure they still produce the desired results.

Preserving Legacy Joins As Is

When you 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 Preferences.
2. In the Preferences dialog box, select Preserve Legacy Query Outer Joins.
3. Click OK.

The Preserve Legacy Query Outer Joins preference does not override the Preserve Legacy Query Outer Joins permission set by the administrator.

Viewing and Clearing Messages

BI Query automatically logs and displays messages (such as completion messages, failure messages, and output parameters) in the Message window when you do any of the following:
• Refresh queries
• Run freehand queries
• Run stored procedures within freehand queries
You can display or hide the Message window. When it is hidden, the window will remain hidden until you select the command to enable it again. Regardless of whether it is hidden or displayed, new messages will continue to be added to the window when any of the listed actions occurs.

Messages are cleared automatically when you close BI Query. You can also manually clear the messages at any time.

**To view or hide the Message window:**
- On the **View** menu, click **Message**.

**To clear the Message window:**
- On the **View** menu, click **Clear Messages**.

**To print the Message window:**
- With the Message window active, on the **File** menu, click **Print**.
Chapter 4

Formatting Data

This section provides information on the following:

- “Data Formats in BI Query” on page 63
- “Formatting Data: The Basics” on page 67
- “Formatting Specific Data Types” on page 73
- “Formatting Data for Data Entry” on page 76

Data Formats in BI Query

A data format specifies how BI Query represents data in a result set or report. You can apply a data format to any of the following:

- Attributes—Whenever you run a query that involves the formatted attribute, BI Query applies the format to the corresponding column in the results set (and report, if you create a report based on the results set).
- Columns in a results set or report—In this case, you format the data after you have run a query. The format you specify is limited to the current result set or report and does not apply to any other.
Data Type Categories

For formatting purposes, BI Query organizes the various data types in the data model into general categories. For example, BI Query treats the distinct integer data types (Tinyint, Smallint, Integer, Largeint) as one formatting category.

The available categories are as follows:

<table>
<thead>
<tr>
<th>Category</th>
<th>Data</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>Boolean or bit data</td>
<td></td>
</tr>
<tr>
<td>Currency</td>
<td>Numeric data that represents currency values</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Date-and-time, date-only, and timestamp data</td>
<td></td>
</tr>
<tr>
<td>Fixed</td>
<td>Numeric data with a fixed scale (a fixed number of digits after the decimal point)</td>
<td></td>
</tr>
<tr>
<td>Float</td>
<td>Floating-point (real) data</td>
<td></td>
</tr>
<tr>
<td>Integer</td>
<td>Integer data of any standard byte size</td>
<td></td>
</tr>
<tr>
<td>Null</td>
<td>NULL data</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>Time-only data</td>
<td></td>
</tr>
</tbody>
</table>

With the exception of the Boolean category, BI Query takes the initial default format for each category from the Regional Settings item in the Windows Control Panel. (There is no Regional Settings value for boolean data.)
In Windows 2000, default data formats are defined in the **Regional Options** item in the Control Panel.

Use the **Format** dialog box to set the format for a category. The appearance and functionality of the dialog box depends on the category you want to format.

### Data Format Options

For each data type category, BI Query provides the following formatting options:

<table>
<thead>
<tr>
<th>Format Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category-specific options</td>
<td>The formatting options that are specific to the category. For example, for numeric data types, you can specify the scale—the number of digits after the decimal point. BI Query also provides a set of predefined formats for each category that you can use to format a particular attribute, column, or data model default. The predefined formats include the Regional Settings defined in the Control Panel. If you are formatting an attribute or result/report column, you can also select the default data model format.</td>
</tr>
<tr>
<td>String</td>
<td>A template for the category in the form of a string. Using special control characters in the string, you can add or remove characters in all values of the selected attribute or result column. For example, you could format a string of 10 digits into a recognizable telephone number. This option is not available for data model formats.</td>
</tr>
<tr>
<td>Null values</td>
<td>The format for any NULL values in the category. This format is specific to NULL values in the category; it is not the same as the default format for the Null data type category.</td>
</tr>
</tbody>
</table>
Format Order

You can set more than one formatting option for a given attribute, column, or data model default. However, the particular options that apply depend on the value of the data.

Format Order for NULL Data

If the data is NULL, BI Query applies the Null Values formatting only; no other formatting options apply. The format that BI Query uses in this case is one of the following (in order of precedence):

1. The Null Values format for the attribute or result column (if specified).
2. The data model format for Null Values in the corresponding data type (if specified).
3. The data model format for the Null data type category.

Format Order for Non-NULL Data

If the data is non-NULL, BI Query applies the specified options in the following order:

1. If you have used the Computed String option, BI Query evaluates the conditions based on the value of the data. If the data meets the conditions, BI Query applies the Computed String formatting; no other options apply in this case.
2. If you have not used the Computed String option, or if the data does not match the conditions defined for it, BI Query applies one of the following category-specific formats (in order of precedence):

<table>
<thead>
<tr>
<th>Format Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computed string</td>
<td>A conditional replacement string for values in the category. You can change the formatting color and value of data in the selected attribute or result column based on a comparison between the original data and one or more specified values. For example, you could replace all values less than a certain number with the word “Unsatisfactory”.</td>
</tr>
</tbody>
</table>

This option is not available for default data model formats.
a. the formatting for the attribute or result/report column (if specified)
b. the default format for the corresponding data type

3. If you have specified String formatting, BI Query applies it to the result from step 2.

**Formatting Data: The Basics**

The following procedure outlines the basic steps to follow in setting a data format for one of the following targets:

- an attribute
- a result/report column
- a data model default

You can find more detailed information on each step later in this chapter.

**To specify a data format for a particular target:**

1. Open the **Format** dialog box. The topmost item in the **Format Type** list displays the data type category for the selected target. For more information, see “Opening the Format Dialog Box” on page 68.

2. Do one of the following:
   - To use a custom format, specify the category-specific options for the target. For more information, see “Formatting Specific Data Types” on page 73.
   - To use the Regional Settings for the category, select `<Regional Settings>` from the **Predefined Formats** list.
   - To use your database settings for the scale, and the Regional Settings for all other data formatting properties, select `<Classic>`.
   - To use the default data model format for the category, select the **Default** check box (available only for attributes and result/report columns).

3. To format NULL values for the target, select **Null Values** from the **Format Type** list; then, specify the format. For more information, see “Formatting Null Values” on page 70.

4. To format the target as a string, select **String** from the **Format Type** list; then, specify the format. For more information, see “Formatting Data as Strings” on page 70.
5. To format the target according to certain conditions, select **Computed String** from the **Format Type** list; then, specify the format. For more information, see “Formatting Data as Computed Strings” on page 72.

6. Click **OK**. BI Query formats the data for the target according to the options you have specified.

You can use more than one format option, but BI Query applies them according to its format order rules. For more information, see “Format Order” on page 66.

**Opening the Format Dialog Box**

The procedure for opening the **Format** dialog box depends on the target you want to format.

**To format an attribute:**

1. Do one of the following:
   - Open the attribute window for the data object that stores the attribute. (BI Query must not be in Design mode).
   - Open the **Query** window for a query that uses the attribute. Make sure the window is in Query View mode.

   For more information, see “Query Window View Modes” on page 58.

2. Click the question mark button  beside the attribute name. The **Attribute** dialog box opens.

3. Click **Edit**. The **Format** dialog box opens.

**To format a column in a results set:**

1. Click the column you want to format.

2. On the **Results** menu, click **Format**. The **Format** dialog box opens.

**To format a column in a standard report:**

1. In the standard report, click the column you want to format.

2. On the **Report** menu, click **Format**. The **Format** dialog box opens.
To format a data model default:

1. On the **Tools** menu, click **Preferences**.
2. In the **Preferences** dialog box, click **Data Formatting**. The **Format Preferences** dialog box opens.
3. Click the ellipsis (...) button for the data type category you want to format. The **Format** dialog box opens.

### Specifying Default Formats for Data Models

Use the **Format Preferences** dialog box to specify the default formats for the entire data model. These defaults apply to any target (attribute or result/report column) that does not have its own format.

To specify the data model format for a data type category:

1. On the **Tools** menu, click **Preferences**.
2. In the **Preferences** dialog box, click **Data Formatting**.
3. In the **Format Preferences** dialog box, do one of the following:
   - To use a predefined format, select it from the list that corresponds to the category.
   - To use the Regional Settings defined in the Control Panel, select `<Regional Settings>`.
   - For date, time, and null values, to display nothing at all, select `<Blank>`.
   - For currency, fixed point, and floating point values, to use your database settings for the scale, and the Regional Settings for all other data formatting properties, select `<Classic>`.
   - To specify a custom format, click the ellipsis (...) button. In the **Format** dialog box, specify the custom setting; then, click **OK**. For more information, see “Formatting Data: The Basics” on page 67.

4. Repeat step 3 for any other category you want to format.
5. Click **OK**.

A data model can connect to more than one DBMS. The data type for a given attribute is determined by the DBMS and the connection to it. However, the default formats you specify for the data model are not connection specific; they apply to all Design windows and, therefore, all connections.
Formatting Null Values

You can format NULL values for a particular attribute, result/report column, or data type. You can also set the default format for all NULL values in the data model.

To specify NULL formatting for an attribute, column, or data type:
1. Open the Format dialog box for the target you want to format. For more information, see “Opening the Format Dialog Box” on page 68.
2. From the Format Type list, select Null Values.
3. Do one of the following:
   • To use a predefined format, select it from the Predefined Formats list.
   • To specify your own display format for NULL values, type it in the Display box.
4. Click OK.

To specify the data model format for NULL values:
1. On the Tools menu, click Preferences.
2. In the Preferences dialog box, click Data Formatting.
3. In the Format Preferences dialog box, do one of the following:
   • To use a predefined format, select it from the Null Values list.
   • To display nothing at all, select <Blank>.
   • To specify a custom format, click the ellipsis (...) button ... . In the Format dialog box, specify the custom setting; then, click OK.
4. Click OK.

Formatting Data as Strings

Use the String option to format any data type as a string. This option lets you include additional information in the data (such as a punctuation mark or a code) and exclude unwanted information.
To use the **String** option, you create a template string for the data; the template can contain special control characters that dictate the data format. The following special characters are available:

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>Inserts a character from the original data (in sequence).</td>
</tr>
<tr>
<td>_</td>
<td>Suppresses a character in the original data (in sequence).</td>
</tr>
<tr>
<td>*</td>
<td>Inserts the remaining characters in the original data.</td>
</tr>
<tr>
<td>\</td>
<td>Inserts the next character as is. Use this character to include one of the special formatting characters as itself. For example, type # to insert the # symbol.</td>
</tr>
</tbody>
</table>

To insert any other character, type it where you want it to appear in the string.

**To specify the String option:**

1. Open the **Format** dialog box. The **String** option is available only for formatting attributes and result/report columns. For more information, see “Opening the Format Dialog Box” on page 68.
2. From the **Format Type** list, select **String**.
3. In the **Format String** box, type the template string you want to format the data. Use the special characters described above.
4. Click **OK**.
5. If you are specifying the option for an attribute, click **OK** in the **Attribute** dialog box.

**Example**

A database attribute stores telephone numbers as 13-character strings (for example, 6135554355x26). To format the data in this attribute as a recognizable phone number, you can create the following template:

```
(###) ###-####_ Ext. \#*
```

If you apply the **String** option using this template, BI Query formats the value 6135554355x26 as

```
(613) 555-4355 Ext. #26
```
Chapter 4: Formatting Data

Formatting Data as Computed Strings

The **Computed String** option lets you replace values in the target (an attribute or result/report column) based on certain conditions. For example, to identify areas that require further attention, you could replace certain values in the target (all sales below a certain amount) with the word “Unsatisfactory”. If the target is a standard report, you could also apply a color to these values.

Each condition for the **Computed String** option has four components:

- A test value—BI Query compares each value in the target against the test.
- An operator—The operator specifies the nature of the comparison. For example, if you select `<` as the operator, BI Query checks if each value in the target is less than the test value.
- A replacement value—If the result of the comparison is true, BI Query substitutes the replacement value for the value in the target.
- A replacement color (optional)—If the target value meets the condition and the results appear in a standard report, BI Query applies the specified color to the replacement value.

BI Query evaluates and applies each condition separately.

**To specify the Computed String option:**

1. Open the **Format** dialog box. The **Computed String** option is available only for formatting attributes and result/report columns.
   
   ![Tip icon] For more information, see “Opening the Format Dialog Box” on page 68.

2. In the **Format** dialog box, select **Computed String** from the **Format Type** list.

3. Place the cursor in the first empty row of conditions.

4. From the list in the **Operator** column, select the operator you want to use to compare the results value against the test value.
   
   ![Tip icon] You can apply the **Else** operator to the last condition only.

5. In the box in the **Original Value** column, type the test value.

6. To specify a replacement value, type it into the box in the **Replacement** column.

7. To specify a color for the new value (in standard reports only), click the color button in the same row and specify a color. If you have not specified a replacement value, BI Query applies the color to the value in the target.
8. To specify more than one substitution condition, repeat steps 3–7 for each condition.

9. Click **OK**.

To insert a row, place the cursor in the row below which you want to insert the new one; then, click **Insert Row**. To delete a row you have specified, place the cursor in the row; then, click **Delete Row**.

### Formatting Specific Data Types

Attributes and result/report columns have specific data types. However, for formatting purposes, BI Query collects related data types into general data categories. When you format an attribute or a result/report column, the available formatting options depend on the category for the corresponding data type. Similarly, you specify the default formats for the data model in terms of categories and not separate data types.

For more information on setting the default formats, see “Specifying Default Formats for Data Models” on page 69.

The following sections describe the formatting options available for each category.

#### Formatting Boolean Data

The boolean or bit data category stores one of two values. The names applied to the two values depend on the context: yes or no, true or false, 1 or 0, and so on. You can set these names as part of the format.

To format boolean data:

1. Open the **Format** dialog box for the target you want to format. For more information, see “Opening the Format Dialog Box” on page 68.

2. Do one of the following:
   - From the **Predefined Formats** list, select the format you want to apply.
   - In the **True** and **False** boxes, type the formats for the corresponding boolean values.

3. Apply any other options you want. For more information, see “Formatting Data: The Basics” on page 67.

4. Click **OK**.
5. In the **Attribute** dialog box or **Format Preferences** dialog box, click **OK** (if necessary).

### Formatting Numeric Data

The numeric data categories (Currency, Fixed, Float, and Integer) use the same version of the **Format** dialog box. When you set the format for numeric data, you can either use a predefined format or customize your own.

#### To use a predefined format for numeric data:

1. Open the **Format** dialog box for the target you want to format. For more information, see “Opening the Format Dialog Box” on page 68.
2. Do one of the following:
   - From the **Predefined Formats** list, select the format you want to use. To use the Regional Settings for the category, select `<Regional Settings>`.
   - To use the data model format for the category, select **Default** (available only if you are formatting an attribute or result/report column).
3. Apply any other options you want. For more information, see “Formatting Data: The Basics” on page 67.
4. Click **OK**.
5. In the **Attribute** dialog box or **Format Preferences** dialog box, click **OK** (if necessary).

#### To specify a customized format for numeric data:

1. Open the **Format** dialog box for the target you want to format.
2. To modify an existing format, select it from the **Predefined Formats** list.
   
   ![Tip](image)
   
   For more information on these options, see the Help for your BI Query application.
4. Apply any other formatting options.
5. Click **OK**.
6. In the **Attribute** dialog box or **Format Preferences** dialog box, click **OK** (if necessary).
Formatting Date and Time Data

The **Date** data category encompasses date-and-time, date-only, and timestamp data. The **Time** category is for time-only data. You can format both categories using the same version of the **Format** dialog box.

**To format date or time data:**

1. Open the **Format** dialog box for the target you want to format.
   
   For more information, see “Opening the Format Dialog Box” on page 68.

2. Do one of the following:
   - To use a predefined time format, select it from the **Predefined Time Formats** list.
   - To remove the time component from date-and-time data, select `<Blank>` from the **Predefined Time Formats** list.
   - To use the Regional Settings for the category, select `<Regional Settings>` from the **Predefined Time Formats** list.
   - To use the data model format for the category, select **Default** (available only if you are formatting an attribute or result/report column).

3. If you are formatting Date-category data, do one of the following:
   - To use a predefined date format, select it from the **Predefined Date Formats** list.
   - To remove the date component from date-and-time data, select `<Blank>` from the **Predefined Time Formats** list.
   - To use the Regional Settings for the category, select `<Regional Settings>` from the **Predefined Time Formats** list.
   - To use the data model format for the category, select **Default** (available only if you are formatting an attribute or result/report column).

4. To modify the selected format, edit the text in the **Format String** box.
   
   For more information, see “Formatting Data: The Basics” on page 67.

5. Apply any other options you want.

6. Click **OK**.

7. In the **Attribute** dialog box or **Format Preferences** dialog box, click **OK** (if necessary).
Formatting Data for Data Entry

The data format you specify for a given attribute governs how BI Query formats values for that attribute in a results-set column or report column. This format also applies when you enter data in a qualification or prompt for that attribute, or when you enter data to update the database values for that attribute.

BI Query can recognize multiple formats for a given data type. When you enter data, if BI Query is able to interpret the format you use, it automatically converts the data to the specified format for the attribute. If you enter data in a format that BI Query is unable to interpret, it displays an alert message.

Avoid using an attribute format for numbers or dates that “hides” information. For example, if a date attribute has the format MMM/DD and you type Sept 15, 2001 as the qualification value, BI Query inserts ‘2001-09-25’ into the SQL for the query, but displays Sep/15 in the qualification tree. If you change the value to Sept 26, BI Query interprets your change as being a complete value (Sept 26, not Sept 26, 2001). The “hidden” year is lost, and the date value in the SQL (‘-09-26’) becomes invalid.

Setting the Data Entry Format

By default, BI Query takes the data entry format for an attribute from the data format for that attribute. However, you can set the data entry format to the data model default for the corresponding data type.

To set the data entry format:
1. On the Tools menu, click Preferences. The Preferences dialog box opens.
2. In the Qualifications area, do one of the following:
   • To use attribute formats for data entry, select Attribute.
   • To use the data model formats for data entry, select System Default.
3. Click OK.

Entering Date and Numeric Data

When entering data, you can type numbers (including currencies) without formatting them. BI Query converts them to the format you have specified for the corresponding attribute.
You can also type dates using any format comparable to the one you have specified. For example, if the date format you have specified is MM/DD/YYYY, you can type dates in the following formats:

- 2003.1.14
- 1.14.2003
- January 14, 2003
- 2003 Jan 14

As long as you include the day, month, and year and BI Query is able to unambiguously interpret the format, it converts the date to the specified format.

BI Query cannot reliably interpret ambiguous dates such as 04/05/2003, in which the day can be confused for the month, or 04/05/03, in which the day, month, and year are not obvious. To ensure that BI Query interprets dates correctly, make sure that you type dates using all four digits of the year and, if necessary, use the name of a month rather than its number.

### Converting Year Formats

You can specify a date window to control how BI Query converts two-digit years to four-digit years.

The date window consists of a start year and end year. If you specify a date window with a start year of 1918 and an end year of 2017, BI Query adds the century 19 to all two-digit years greater than or equal to 18 and the century 20 to all two-digit years less than or equal to 17. When you enter 45, BI Query converts it to 1945; when you enter 15, BI Query converts it to 2015.

- If you change the date window after you've entered two-digit years, BI Query does not apply the new date window to those dates.
- Changes you make to the date window in one Hummingbird BI product affect all other Hummingbird BI products.

**To specify the date window:**

1. On the Tools menu, click Date Entry. The Date Entry dialog box opens.
2. In the Start Year Is box, type the start year for the date window. (The end year is calculated automatically.)
3. Select **Sliding** or **Fixed**. A sliding date window adds one to the start and end year on January 1st every year. For example, if the date window is 1918 to 2017, it becomes 1919 to 2018 when the current year changes. A fixed date window is not incremented automatically.

4. Click **OK**.

**Correcting Invalid Entries**

BI Query alerts you if you type data in a format that it cannot interpret.

**To return to an entry and correct it:**
- Click **Cancel** in the alert window.

**To force BI Query to use the entry you typed:**
- Click **Add { }**. The entry appears in curly brackets in the qualification tree. When you run a query, BI Query inserts the data in the SQL string as is.

💡 This feature is for users with advanced knowledge of SQL database access.

**To determine the format for an attribute:**

1. If you have configured the data model to use the default data model formats for data entry, open the **Format Preferences** dialog box:
   a. On the **Tools** menu, click **Preferences**.
   b. In the **Preferences** dialog box, click **Data Formatting**.

   Note the current default format for the relevant data category.

💡 For more information on configuring data entry formats, see “**Setting the Data Entry Format**” on page 76.

2. If you have configured the data model to use attribute formats for data entry, open the **Attribute** dialog box for the relevant attribute. (In the attribute window, click the question mark button ☐ beside the name of the attribute).

   The **Format** box displays sample data in the attribute format.
This section provides information on the following:

- “Data Values Files” on page 79
- “Creating Data Values Results Files” on page 82
- “Creating Data Values Query Files” on page 84
- “Qualifying with Data Values Files” on page 85
- “Data Aliasing” on page 85

Data Values Files

One way to limit what a query retrieves is by qualifying attributes with data values. Before you run the query, you can insert one or more data values into a qualification, either by typing them or by selecting them from a list. Lists of data values are provided by data values files.

A data values file contains a subset of the possible values in the database. These values appear as a list when you click the *Data Values* button. Selecting data values from a list is a more accurate and efficient way to insert values than typing them.

💡 Data values files can also be used to set up data aliasing. For more information, see “Data Aliasing” on page 85.
If no data values file is available for a particular attribute, when you click the **Data Values** button and select **Data Values**, BI Query runs the default data values query and retrieves all the values.

The administrator can disable access to the default data values query. Disabling it is recommended when 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 simply numbers and dates.

**Types of Data Values Files**

There are two types of data values files:

**Data Values Query Files**

Query the database and retrieve a subset of values to create the list. Data values query files provide the most accurate lists; the first time you use them in a session, they query the database to supply the values. Because they can return different results depending on when they are run, they are sometimes called dynamic data values files.

**Data Values Results Files**

Display a stored set of values. Data values results files are produced either manually or by storing the results of a query. BI Query does not update the lists of values they contain, so they are sometimes called static data values files. Results files reduce the load on the database server because they eliminate the need to query the database.

If your querying environment includes BI Server, you can let other users access your data values results files by publishing them to the BI Server Repository.

Data values results files are useful only when data values change infrequently or not at all (for example, sales data from previous years). If they must be updated often, it is more efficient to create data values query files instead.

For more information, see “Publishing Data Values Results Files” on page 247.

**Naming and Storing Data Values Files**

When you name a data values query or results file, adhere to the following naming convention:

- To use the data values file for a prompt *only*, name the file after the prompt.
To use the data values file in the qualification for a particular attribute (and possibly in a prompt for the same attribute), use the following format:

\[ \text{data\_object.\_attribute} \]

where:

- **attribute** is the display name of the attribute
- **data\_object** is the display name of the data object that stores the attribute

For example, the data values file for the country attribute in the Retailers data object would have the name `Retailers\_country`

Do not use database table and column names for **data\_object** and **attribute**.

Prompt, attribute, and data object names are case sensitive.

💡 For more information, see “Naming Conventions for Prompts” on page 91.

Data values query and results files are stored in the **DataVals** folder. BI Query stores data values results files in two separate files: the `.qrd` file, which stores results in text format, and the `.qrr` file, which stores additional information that the application needs to read the text file. Without the `.qrr` file, BI Query cannot open the `.qrd` file. Data values query files have the same file extension as regular queries, `.qry`.

The file names you give these files determine the order in which BI Query selects the files in the **DataVals** folder as well as whether the corresponding data values are available in a qualification or in a prompt.

### Order of Data Values Files

The following table lists the order in which BI Query uses data values files.

<table>
<thead>
<tr>
<th>Order</th>
<th>Data Values File</th>
<th>Available In</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Results file named after the corresponding prompt</td>
<td>Prompt dialog boxes</td>
</tr>
<tr>
<td>2</td>
<td>Query file named after the corresponding prompt</td>
<td>Prompt dialog boxes</td>
</tr>
<tr>
<td>3</td>
<td>Results file named after the data object and attribute used in the query</td>
<td>Qualification tree and prompt dialog boxes</td>
</tr>
</tbody>
</table>
When you click the **Data Values** button in a prompt dialog box in order to display a list of data values that can be used in the prompt, BI Query checks to see if there is a data values file with the same name as the prompt. If there is, BI Query uses it.

If there is not a data values file named after the prompt, BI Query checks to see if there is a data values file named after the data object and attribute used in the query. If there is, BI Query uses it.

💡 For more information on prompts, see “Prompts in BI Query” on page 89.

If BI Query does not find a data values file, and if data values for the qualified attribute have not been disabled, you can click **Data Values** in the pop-up menu to open the **Data Values** dialog box. The dialog box lists all the unique values in the database for the attribute.

### Creating Data Values Results Files

A data values results file stores a set of data values for an attribute, so that you do not need to query the database to obtain data values, thus reducing the load on the network.

You can create data values results files from BI Query results, or customize the results using a text editor or spreadsheet. You can qualify attributes by inserting these data values into qualifications and prompts.

💡 For information on retrieving and opening stored data values results files, see the Help for your BI Query application.

**To create a data values results file:**

1. In an attribute window or the **Query** window (in Query View mode), click an attribute; then, click the **Sort** box and apply any other restrictions you want.
2. On the **Query** menu, point to **Modifiers**; then, click **Distinct** to eliminate duplicate values.
3. If you want to retrieve only specific values, qualify the attribute accordingly.
4. Run the query.
5. If you plan to customize your file using a text editor or spreadsheet, on the **Results** menu, click **Options**. In the **Results Options** dialog box, select `<CR>` `<LF>` from the **Record** list and **<None>** from the **End of File** box; then, click **OK**.

6. On the **Results** menu, click **Save**. The **Save Results** dialog box opens.

7. Do one of the following:
   - If you are using BI Query Admin, click **Shared Data Values**.
   - If you are using BI Query User/Update, click **My Data Values**.

8. In the **File Name** box, type the file name you want. BI Query adds the extension `.qrd` to the file. Remember to name the file according to the naming convention for data values files.
   
   📢 For more information, see “Naming and Storing Data Values Files” on page 80.

9. Click **Save**.

### Creating Data Values Results Files with a Text Editor

You can use any text editor or spreadsheet program to create a data values results file that contains a set of customized data values. To do so, you first create a data values results file to serve as a template for the customized data values. (This sets up the `.qrd` file, which stores results in text format, and the `.qrr` file, which stores additional information that the application needs to read the text file.) Once you create these files, you replace the data values in the `.qrd` file with the customized data values.

When you type values into the text editor or spreadsheet program, type them in the order and case you want them to appear in BI Query, pressing ENTER after each value.

**To create a data values results file using a text editor:**

1. Create the template data values results file using the usual procedure.
   
   📘 For more information, see “Creating Data Values Results Files” on page 82.

2. In a text editor or spreadsheet program, do one of the following:
   - Open the `.qrd` file you created in step 1, modify the file, and save it under the same name.
Chapter 5: Qualifying with Data Values Files

- Type a list of values in a new document file, adding a newline (carriage return + line feed) after each one; then, save the file using the same path and file name as the .qrd file in step 1.

To test the file in BI Query, select the attribute used to create the original results, click its **Qualify** box, and then click the **Data Values** button in the qualification tree.

### Opening Data Values Results Files

You can open an existing data values results file as a results set in BI Query. You can use this results set as you would any other.

**To open a data values results file:**

1. On the **Results** menu, click **Open**.
2. In the **Open Results File** dialog box, do one of the following:
   - To open a shared file, click **Shared Data Values**. This option is available only if any of the following is true: the data model is split; you are using BI Query Admin.
   - To open a personal data values results file, click **My Data Values**. This option is not available if you are using BI Query Admin.
3. Select the data values results file.
4. Click **Open**. The selected file opens as a results set.

### Creating Data Values Query Files

A data values query file stores a query that retrieves values of an attribute from the database. You can use such a file to qualify attributes in other queries (either in a qualification or in a prompt).

**To create a data values query file for an attribute:**

1. Double-click the data object that contains the attribute. In the attribute window, select the attribute; then, click its **Sort** box and apply any other restrictions you want.
2. To eliminate duplicates, on the **Query** menu, point to **Modifiers**; then, click **Distinct**.
3. Click in the **Qualify** box beside the attribute, and then enter the values you want in the **Qualification** box for the selected attribute.
4. On the **Query** menu, point to **Save**, and then click **Data Values Query**. The **Save Data Values Query** dialog box opens.

5. In the **File Name** box, type the file name you want. A file name with the extension .qry is created by BI Query. Remember to name the file according to the **naming convention for data values files**.

  💡 For more information, see “**Naming and Storing Data Values Files**” on page 80.

6. Click **Save**.

### Qualifying with Data Values Files

You can use a data values file to qualify an attribute in a query.

**To use a data values file in a query:**

1. Create a query that involves the attribute the data values file is based on.

2. To access the data values list, click the attribute's **Qualify** box. Then, click the **Data Values** button ☐ in the qualification tree and select **Data Values** from the pop-up menu.

   If you created a data values results file, BI Query loads it and displays the values as a list. If you created a data values query file, BI Query runs it, and then displays the values in the **Data Values** dialog box.

   You can also access the data values list by clicking the **Data Values** button ☐ in the prompt window and selecting **Data Values** from the pop-up menu.

### Data Aliasing

Data aliasing lets users select from a list of common-language names for data values while BI Query inserts the corresponding actual values into the query.

To set up data aliasing, you create a data values results file or data values query file that contains two columns; in the first column are the actual values, in the second are the aliases. Only the data in the second column is displayed in the list of data values.

💡 For more information on alias data values, see the Help for your BI Query application.
For example, instead of selecting a value for a qualification that may be difficult to understand or remember, such as an employee’s ID number, you will be able to select a more meaningful value, such as the person’s full name. You select the name from a list of data values and BI Query inserts the ID number into the **Qualification** box for the query.

**Creating Data Values Files to Alias Data**

You can create a data values file to alias data so that a user can see values in the data values list that are meaningful (such as last names), while still using values from the database (such as ID numbers).

To create a data values file that aliases data:

1. In the attribute window as you construct a new query, select the attribute that is to be sent to the database and the attribute you want to serve as the alias (whose values are displayed in the data values list).
2. Click the **Sort** box corresponding to the alias attribute.
3. Do one of the following:
   - Save the query as a data values query file, naming it appropriately in the **Save Query** dialog box.
   - Run the query and save the results as a data values results file.
4. Name the file after the first attribute (the attribute sent to the database). Remember to use the naming convention for data values files.

💡 For more information, see “Naming and Storing Data Values Files” on page 80.

**Modifying Data Aliases with a Text Editor**

You may need a customized data alias list if the aliases you want to see are not in the database. You can do this using a data values results file and a text editor.

If you have not already done so, you must create a data values results file.
To modify a data alias data values file:

1. In the text editor (not a word processor), open the data values results file. (Data values results files are stored in the DataVals folder of the data model you are using.)
2. Enter or modify the list of values, pressing ENTER after each one. (Be careful you do not add blank rows to the end of the list.)
3. Save the modified results file with the same name in the same location.

Viewing Aliased Data in a Query

You can use a data values file to see values in the data values list that are meaningful (such as last names), while still using values from the database (such as ID numbers).

To view aliased data in a query:

1. Construct a query that displays the aliased attribute.
2. Click the aliased attribute's Qualify box; then, click the Data Values button in the qualification tree.

If you created a data values results file, BI Query loads it and displays the aliases as a list. If you created a data values query file, BI Query runs it, and then displays the aliases in the Data Values dialog box. When a user chooses an alias, BI Query inserts the value that was aliased into the qualification tree and then into the SQL string.
This section provides information on the following:

- “Prompts in BI Query” on page 89
- “Prompts for BI Web Reports” on page 94
- “Common Tasks Using Prompts” on page 97
- “Specifying Web-Reporting Values” on page 104

Prompts in BI Query

A prompt is a request to enter a value or set of values. Typically, you can insert a prompt into a query qualification instead of specifying data values. Each time the user runs the query, the query can retrieve information for different values, depending on what response the user makes to the prompt. You can also include prompts in other contexts, including join conditions and object qualifiers.

A prompt consists of a name and a prompt string. The name distinguishes the prompt from others you may create (and may indicate if it belongs to a special aggregate known as a group prompt). The prompt string is the text that appears in the Enter Value dialog box when users run a query involving the prompt (for example, “Please enter a city name”).
The name of the Enter Value dialog box changes depending on the type of prompt in use. For more information, see “Naming Conventions for Prompts” on page 91.

You can also associate a particular data type with a prompt and specify whether users must supply a value for the prompt. If the prompt appears in a report for BI Web users, you can specify the appearance of the prompt and, in some cases, an optional list of allowed data values.

BI Query stores prompts in a single Prompts file in the Queries folder. BI Query does not save the prompts created for a given data model until you close the data model.

If you insert a prompt into a query and then convert that query into a query data object (QDO), all queries using that QDO will require users to respond to the prompt, although the prompt will not be visible in the data object or the SQL for the query. For more information on QDOs, see the BI Query Data Models User’s Guide.

By default, when you reuse a prompt, BI Query offers the same values used for the prompt the last time it was used in a query.

Types of Prompts

There are three types of prompts: single prompts, group prompts, and conditional prompts.

Single Prompts

A single prompt retrieves a value (or set of values) for a given query. Single prompts usually qualify a single attribute. A single prompt appears in the Enter Value dialog box.

Group Prompts

A group prompt is made up of two or more distinct prompts that are grouped by their names. When you run a query that involves multiple prompts in a given group, they all appear in the same dialog box. For example, a group prompt may consist of prompts for the attributes Country and City; a query that qualifies those attributes prompts you to select a country and a city using the same dialog box.
A group prompt can display up to five prompts in one dialog box. BI Query presents the distinct prompts in a group in alphanumerical order. If a query contains more than five prompts in a group, the first dialog box displays the first five prompts (in order), a second displays the next five, and so on. A query can contain more than one group prompt.

Using a group prompt in a query lets users see all the prompts at once and reduces the number of steps in the query. A group prompt also relates one prompt to another.

**Conditional Prompts**

A conditional prompt is a group prompt in which the available data values for each successive prompt in the group depend on the values you specify for the preceding prompts. For example, a conditional prompt involving Country and City attributes may list only data values for those cities located in the country that you specify.

Conditional prompts can eliminate inappropriate data values and make it easier for users to run a given query. Conditional prompts are also useful in a data model that doesn’t let users build their own queries, because they provide a means of drilling down in the data to obtain increasingly detailed information.

**Naming Conventions for Prompts**

Single prompts can have any name; prompts that qualify an attribute are usually named after the attribute. (For example, if you create a prompt that qualifies the City attribute, the prompt can be named City.) Group prompts (including conditional prompts) have a special naming convention.

**Group Prompt Names**

The name for each prompt in a group takes the following form:

```
group_name!prompt_name
```

where `group_name` is the name of the group and `prompt_name` is the name of the prompt in that group. For example, the city prompt in the stores group has the name stores!city. `group_name` is optional; it lets you create several groups of prompts.
Chapter 6: Qualifying with Prompts

Group prompts can also have a special Title prompt associated with them. This prompt stores the title of the group prompt dialog box. If you do not supply a Title prompt, the dialog box for the group is called **Enter Values**. Title prompt names take the following form:

\[ \text{group\_name}!!\text{Title} \]

where `group_name` is the name of the group and `!!Title` is the literal string as written (case is important). The prompt string for a Title prompt represents the title for its group. For example, if you create a Title prompt for the stores group (stores!!Title) and specify its prompt string as “Stores by City”, the dialog box that opens for that group is called **Stores by City**.

### Data Types for Prompts

Prompts that qualify attributes automatically use the data type of those attributes. However, if you are creating a prompt that does not qualify an attribute, such as a prompt in a join, object qualifier, or variable, you need to set the data type of the prompt. When users respond to the prompt, the value they enter must conform to the specified data type. The data type for a prompt ensures that users enter values that are acceptable to the database.

BI Query groups prompt data types into **general categories**. When you create a prompt, you select the category that corresponds to the data type that the database expects, rather than the specific type itself.

#### Example

You want to use a prompt as an argument in a DBMS function that requires Smallint values. When you create the prompt, you select the prompt data type category that corresponds to Smallint data type (in this case, Integer). When the users run a query involving that prompt, they must supply integer values; if they enter character data or floating-point values, BI Query displays an alert box to warn them of the problem, and then lets them either modify the value to the correct format or enclose the value in curly braces (\{\}) to send it to the database as is.

#### Data Type Categories

You can set the data type for a prompt using the **Data Type** list in the **Edit Prompt** dialog box. If you are using a prompt to qualify an attribute, the prompt automatically uses the data type of the attribute, even if you assign a different data type to the prompt.
The following table lists the data type categories for prompts:

<table>
<thead>
<tr>
<th>Data Type Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Automatic Formatting)</td>
<td>Ensures that a prompt has the same data type as the attribute it qualifies. Do not select this type for any of the following:</td>
</tr>
<tr>
<td></td>
<td>• prompts that do not qualify an attribute</td>
</tr>
<tr>
<td></td>
<td>• prompts that qualify a date, time, or datetime attribute in a Teradata DBMS</td>
</tr>
<tr>
<td></td>
<td>• prompts that you intend to use in freehand queries</td>
</tr>
<tr>
<td>Text</td>
<td>Includes the following DBMS data types: c, char, character, long, long varchar, text, varchar, varchar2, vchar, varcharacter.</td>
</tr>
<tr>
<td>Integer</td>
<td>Includes the following DBMS data types: integer, smallint, number(p), int, smallint, tinyint, bit, serial, integer1, integer2, integer4, byteint, smint, decimal(p).</td>
</tr>
<tr>
<td>Real</td>
<td>Includes the following DBMS data types: dec, decimal(p,s), numeric(p,s), float, real, double precision, largeint, number, number(p,s), money, smallmoney, smallfloat, float4, float8.</td>
</tr>
<tr>
<td>Date</td>
<td>Includes the following DBMS data types: date, datetime (for example, year to day).</td>
</tr>
<tr>
<td>Time</td>
<td>Includes the following DBMS data types: time, datetime (for example, hour to second).</td>
</tr>
<tr>
<td>Date and Time</td>
<td>Includes the following DBMS data types: date, datetime, timestamp, smalldatetime.</td>
</tr>
<tr>
<td>(No Formatting)</td>
<td>Lets users enter any type of data. This is equivalent to placing curly braces ({} ) around the prompt name in the qualification tree.</td>
</tr>
</tbody>
</table>

**Required Versus Optional Prompts**

When you create a prompt, you can specify whether it is required or optional.
Chapter 6: Qualifying with Prompts

Required Prompts
These are useful to prevent queries from returning too much data. A required prompt forces users to supply at least one value for the prompt. If users leave a required prompt empty, the query does not run. BI Query marks required prompts with a red asterisk in the prompt dialog box.

⚠️ If the prompt you are creating may be used in a freehand query, make sure that you specify that it is a required prompt.

⚠️ Users can remove any prompt (required or otherwise) from the qualification tree for a query. Therefore, do not use required prompts to control user access to the database.

Optional Prompts
These are useful if a query contains different prompts for different users, not all of whom want to use each prompt. Users can choose to enter no values for an optional prompt. In this case, if the prompt qualifies an attribute, the prompt has no effect on the query; BI Query removes the qualification involving the prompt from the WHERE clause of the query. However, if the prompt does not qualify an attribute (for example, if it is an argument to a function), BI Query does not remove the prompt; in this case, an empty prompt value may produce a DBMS syntax error.

_inches

Each prompt retains the last value selected by you and displays that value as the default the next time it opens. If you leave an optional prompt empty, you effectively erase the last value.

Prompts for BI Web Reports
For BI Web users who access published standard reports or BI Query reports, you can specify allowed data values for prompts in the corresponding queries. These values are known as web-reporting values. When BI Web users open and refresh a report whose query has a prompt, they can select a value for the prompt from the web-reporting values you have specified.

As an alternative to specific values, you can also make available the data values from a data values query file, if one exists for the prompt or the attribute (if the prompt qualifies an attribute).
For more information, see “Creating Data Values Query Files” on page 84.

If data values query files exist for both the prompt and the attribute, only the file associated with the prompt supplies data values.

**Prompt Display Types for BI Web Reports**

When you create a prompt for a BI Web report, you must design its interface. For a given prompt, you can select one of the following display objects in the *Edit Prompt* dialog box:

<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Text Box" /></td>
<td><strong>Text Box</strong>—lets the user type one or more values. This is the default object for web-reporting values. You can use a data values query file for web-reporting values only if the display object is a text box. If there is no data values query file associated with the prompt or the attribute it qualifies, then the prompt uses the data values from the default SELECT DISTINCT query for the attribute.</td>
</tr>
<tr>
<td><img src="image" alt="List Box" /></td>
<td><strong>List Box</strong>—lets the user select one or more values from a list.</td>
</tr>
<tr>
<td><img src="image" alt="Combo Box" /></td>
<td><strong>Combo Box</strong>—lets the user type a value or select one from a list.</td>
</tr>
<tr>
<td><img src="image" alt="Option Buttons" /></td>
<td><strong>Option Buttons</strong>—let the user select exactly one value from a group. If you use option buttons for a prompt, it must be a required prompt.</td>
</tr>
<tr>
<td><img src="image" alt="Check Boxes" /></td>
<td><strong>Check Boxes</strong>—let the user select multiple values from a group.</td>
</tr>
<tr>
<td><img src="image" alt="Date Entry" /></td>
<td><strong>Date Entry</strong>—lets the user enter a date value in the form YYYY-MM-DD.</td>
</tr>
<tr>
<td><img src="image" alt="Time Entry" /></td>
<td><strong>Time Entry</strong>—lets the user enter a time value in the form hh:mm AM.</td>
</tr>
<tr>
<td><img src="image" alt="Date &amp; Time Entry" /></td>
<td><strong>Date &amp; Time Entry</strong>—lets the user enter a date-and-time value in the form YYYY-MM-DD hh:mm AM.</td>
</tr>
</tbody>
</table>
The display object that you specify for a prompt determines how the BI Web user’s browser displays the web-reporting values. For example, if you specify 7 web-reporting values and select check boxes as the display object, the BI Web user can select the prompt value(s) from among 7 check boxes.

The display object also determines how many values the user can select. If in the previous example you had selected option buttons as the display object, the BI Web user could select only a single value for the prompt.

In some circumstances BI Query will override your choice of display object. For example, if you had selected check boxes as the display object for a prompt based on a qualification using a greater-than (>) operator, BI Web will instead use the option button display object.

You can also override the properties of the display object and force users to select a single value.

If the display object is a text box, date entry, time entry, or date & time entry, you cannot specify particular web-reporting values. You must specify these values for all other display objects.

The following table summarizes the available prompt options based on the display object:

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Required or Optional Prompt?</th>
<th>Multiple Selections Allowed?</th>
<th>Data Values Query File Allowed?</th>
<th>Specific Data Values Allowed?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text box</td>
<td>Either</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>List box</td>
<td>Either</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Combo box</td>
<td>Either</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Option buttons</td>
<td>Required</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Check boxes</td>
<td>Either</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Date entry</td>
<td>Either</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Time entry</td>
<td>Either</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Non-Typed Prompts in BI Web Reports

If a prompt in a BI Web report does not qualify an attribute and does not have a defined data type, the behavior of the prompt when the user refreshes the report depends on the context:

- If the prompt is in a join condition or an object qualifier, the BI Web user is prompted and the prompt data type defaults to (No Formatting).
- If the prompt is in a component query of a super query involving multiple connections, and the prompt relies on the default `SELECT DISTINCT` query for data values, the prompt is not supported. To resolve this problem, publish a data values query or data values results file for the prompt and name the file the same as the prompt.
- For other prompts of type (Automatic formatting), BI Web attempts to assign a data type and uses a text object as the display object. In this case, if the assigned data type is date/time/datetime/timestamp, and the prompt appears in a standard report, BI Web supplies a recommended format for entering the prompt value. (No recommendation is supplied for such prompts in a BI Query report.)

It is strongly recommended that you define data types for prompts that do not qualify attributes.

Common Tasks Using Prompts

A typical application of a prompt is to improve the scope of a given query. For example, you could create a query about customer payment information that prompts you for a country name each time you run the query. When you want information on customers located in Argentina, you run the query and type Argentina in response to the prompt; when you want information on customers located in Australia, you run the query and type Australia in response to the prompt, and so on.
For tables that contain a large number of rows, you can use prompts to improve query performance and prevent users from running queries that retrieve more data than they need. For example, regional sales managers may need sales information only for their regions. When they run queries against the table containing sales information, the queries prompt them to specify the regions they want.

**Other Uses for Prompts**

In addition to using prompts to qualify attributes, you can include a prompt in any context where you want the user to supply a value. For more information on prompts and their applications, see the following topics:

- “Functions, Prompts, and Other Values in Variables” on page 112
- “Adding Calculated Columns” on page 163
- “Creating Group Qualifications” on page 190

For information on using prompts in join conditions and object qualifiers, see the *BI Query Data Models User's Guide*.

**Creating Single Prompts**

Use the **Prompts** and **Edit Prompt** dialog boxes to create a prompt and specify its properties.

For information on modifying and deleting prompts, see the Help for your BI Query application.

**To create a single prompt:**

1. On the **Edit** menu, click **Prompts**. The **Prompts** dialog box opens.

2. In the **Prompts** dialog box, click **New**. The **Edit Prompt** dialog box opens.

3. In the **Edit Prompt** dialog box, type the name of the prompt into the **Name** box.

4. In the **Data Type** list box, select the **data type** for the prompt.

5. In the **String** box, type the phrase that you want to prompt the user (for example, “**Please enter an employee ID**”).

6. To force users to supply a value for the prompt, select **Required**; to make the prompt optional, clear **Required**.
7. If the prompt will appear in a published standard report or BI Query report for BI Web users, specify its web properties in the **Web Reporting** area. These properties apply only to prompts that appear in reports in the BI Web Portfolio. Do the following:

   a. From the **Prompt Type** list, select the object for displaying the allowed data values.

   b. To let BI Web users enter multiple values for the prompt, select **Allow Multiple Selections**. This check box is enabled only if you have selected **Text Box, List Box, or Check Boxes** from the **Prompt Type** list.

   c. To make data values from a data values query file available to the user, select **Allow Data Value Queries**. This check box is enabled only if you have selected **Text Box** from the **Prompt Type** list. (You must create the file if it does not exist or does not have the correct name.)

      ![Tip]
      For more information, see “**Naming and Storing Data Values Files**” on page **80**.

   d. To specify allowed data values explicitly, click **Values**. The **Web Reporting Values for Prompt** dialog box opens. Specify the values you want to make available to users; then, click **OK**. (You cannot specify values if you have selected **Text Box, Date Entry, Time Entry, or Date & Time Entry** from the **Prompt Type** list.)

      ![Tip]
      For more information, see “**Specifying Web-Reporting Values**” on page **104**.

8. Click **OK**; then, click **Close** to close the **Prompts** dialog box.

**Qualifying with Prompts**

You define a qualification with a prompt in the same way you would a qualification with a value. If you qualify more than one attribute in a query with the same single prompt, the value you specify for the prompt applies to each attribute.

For information on using prompts with freehand queries, see “**Adding Prompts to Freehand Queries**” on page **145**.

![Tip]
Curly braces ({}) around a prompt value let the database receive formats that BI Query does not support. If you place curly braces around the prompt name in the qualification tree, then when the query runs and the user enters values as prompted, each value is enclosed in curly braces, making it unnecessary for the user to remember to add them.
To insert a prompt when forming a qualification:
1. In an attribute window, click the **Qualify** box for an attribute. BI Query moves the cursor to the **Qualification** box for the attribute in the qualification tree.
2. Click the **Data Values** button; then, select **Prompts** from the list.
3. In the **Prompts** dialog box, select a prompt. If the prompt you want is not listed, you can create it.
4. Click **Insert**.
5. If your query is complete, click the **Run Query** button on the **Standard** toolbar, or save it for later submission.

Creating Group Prompts

To make a group prompt, you create two or more prompts and name them with the group prompt naming convention.

To create a group prompt:
1. Create the prompts you want. You will require at least two. Name each using the group prompt naming convention (for example, *stores!city*, *stores!country*).

   ![💡] For more information, see “Naming Conventions for Prompts” on page 91.

   The prompts are listed in alphanumeric order in the prompt dialog box (*stores!city* before *stores!country*, but *stores!1country* before *stores!2city*), so name them according to the order you want them to have.

2. To specify a title for the group prompt dialog box, create a Title prompt with the name *group_name!!Title*, where *group_name* is the name of the group from step 1 and **!!Title** is the literal string as written (case is important). Type the title in the **Prompt String** box.

   ![💡] If you do not create a Title prompt for a group, the dialog box for the group prompt is called **Enter Values**.

Using Group Prompts

When you run the query, the group prompt dialog box displays the prompts in the specified order and with the specified title.
Common Tasks Using Prompts

To use a group prompt:

1. In an attribute window, select the attributes you want.
2. For each attribute, do the following:
   a. Click the **Qualify** box for the attribute.
   b. On the **Edit** menu click **Prompts**, and then specify one of the prompts that have been grouped with the naming convention from the list in the **Prompt** dialog box.
3. Run the query.
4. In the group prompt dialog box, specify a value in the first box. (Type it or use the **Data Values** button to select a value from a list.)
5. Specify values in the subsequent boxes.
6. Click **OK**.

Creating Conditional Prompts

A conditional prompt is a group prompt that uses a data values query to restrict the available data values for successive prompts in the group. The second prompt in a conditional prompt uses a data values query that includes the first prompt. In this way, the values supplied by the first prompt qualify the query that supplies a choice of values for the second prompt.

💡 For more information on data values query files, see “Creating Data Values Query Files” on page 84.

To create a conditional prompt:

1. Create the group prompts you want. Use the group prompt naming convention so that BI Query presents the prompts in the order you want.
2. To restrict data values for the first prompt in the group, create and save a data values query file that returns the data values you want for the first prompt. Name this file after the name of the first prompt (such as Stores!1country.qrd). When you insert this prompt into a qualification, BI Query makes the results of the query file available in the corresponding prompt dialog box.
   - If you do not create a data values file for the prompt, BI Query uses the default **SELECT DISTINCT** query to generate the available data values.
3. Create a new query to return the data values you want for the second prompt. In this query, use the first prompt to qualify the attribute that corresponds to the first prompt, and then apply the appropriate operator.
4. Save the query as a data values query file using the name of the second prompt (such as Stores!2city.qrd).

**Using Conditional Prompts**

**To use a conditional prompt:**

1. In an attribute window, click the attributes you want.
2. Use the first prompt to qualify the corresponding attribute.
3. Use the second prompt to qualify the corresponding attribute.
4. Run the query.
5. To enter a value for the first prompt, do one of the following:
   - In the group prompt dialog box, type a value into the first box.
   - If data values have not been disabled for the attribute, click the **Data Values** button and select a value from the list.
6. Click the **Data Values** button for the second prompt. From the pop-up menu that opens, click **Data Values**.
7. In the **Data Values** dialog box the values displayed should be limited by your choice for the first prompt. Select the value(s) you want. (Depending on the operator you applied to the qualification, you can select one or more values.) Click **Insert**.
8. In the group prompt dialog box, click **OK**.

**Modifying or Deleting Prompts**

In a combined data model, you can edit or delete any prompt if the **Edit Data Model** permission is assigned. In a split data model, you can edit or delete only prompts that you have created. Those created by your BI Query administrator are identified with a lock icon and can be edited only by the administrator.

**To edit a prompt:**

1. On the **Edit** menu, click **Prompts**.
2. In the **Prompts** dialog box, select the prompt you want; then, click **Edit**.
3. In the **Edit Prompt** dialog box, make the changes.
4. Click **OK**; then, **Close**.
To delete a prompt:
1. On the Edit menu, click Prompts.
2. In the Prompts dialog box, select the prompt you want; then, click Delete.
3. Click Close.

Importing Prompts
To use prompts created by someone else, import them from that person’s Prompts file (stored in that person’s Queries folder).

To import prompts:
1. On the Edit menu, click Prompts.
2. In the Prompts dialog box, click Import.
3. In the Select Import File dialog box, select the file that contains the prompts you want to import; then, click Open. The Import Prompts dialog box opens.
4. Select the prompts you want under Available prompts; then, click the right arrow button to move the selected prompts to the Current prompts box.
5. Click Import, and in the Prompts dialog box, click Close.

Showing Prompt Values in Queries
Values that have been typed into prompts at run time can be viewed in the SQL string area of the Query window. To view the SQL string, the Show SQL String preference must be set.

To show a prompt value in a query:
1. Open the Query window. Make sure the window is in SQL View mode.
2. In the Query window, select Show Prompt Values.

For more information, see “Query Window View Modes” on page 58.

If you have entered prompt values, they are usually displayed in the WHERE clause of the SQL string. (For calculated attributes, they appear in the SELECT clause.) If no values were entered for the prompt, the prompt does not appear in the SQL string. If the prompt has never been run, the Show Prompt Values check box will be grey.
Specifying Web-Reporting Values

If you are creating a prompt for a BI Web report and the prompt does not use a text box, date entry, time entry, or date & time entry, you can specify the web-reporting values explicitly. The values you specify can come from the database (in which case, you must specify a particular data object and attribute in that object). You can also enter your own values directly.

To specify web-reporting values for a prompt:

1. Open the **Web Reporting Values for Prompt** dialog box for the prompt. (In the **Edit Prompt** dialog box, click **Values**).

2. To select values from the database, do the following until all the values you want appear in the list:
   a. From the **Data Objects** list, select an object. From the **Attributes** list, select an attribute.
   b. Click **Load**. The **Source: Database** list displays all distinct values in the database for the selected table and attribute.
   c. From the **Source: Database** list, select the web-reporting values that you want to use for the prompt; then, click the right arrow button ➔. The selected values appear in the **Web Reporting Values** list.

3. To specify your own values, do the following until all the values you want appear in the list:
   a. Click **Create**. The **New Web Reporting Value** dialog box opens.
   b. In the **Data Value** box, type a data value. Click **OK**.

4. Set the order for the values. The top-to-bottom order of the values in the **Web Reporting Values** list is the order in which they appear in the BI Web prompt form. Use the available buttons:

   ![Up](up.png) ![Down](down.png) ![Top](top.png) ![Bottom](bottom.png) ![Ascending Sort](ascending.png) ![Descending Sort](descending.png)

   BI Query sorts numerical values as text strings.

   For more information on sorting and editing web-reporting values, see the Help for your BI Query application.
5. Click **OK**.

**Editing and Deleting Web-Reporting Data Values**

Once you have selected allowed data values for a prompt, you can edit or delete them in the **Web Reporting Values for Prompt** dialog box.

**To edit an allowed data value:**

1. Open the **Web Reporting Values for Prompt** dialog box. (In the **Edit Prompt** dialog box, click **Values**.)
2. From the **Web Reporting Values** list, select the value you want to edit; then, click **Edit**.
3. In the **Edit Web Reporting Value** dialog box, edit the value in the **Data Value** box; then, click **OK**.
4. In the **Web Reporting Values for Prompt** dialog box, click **OK**.

**To delete an allowed data value listed in the Web Reporting Values for Prompt dialog box:**

1. From the **Web Reporting Values** list, select the value you want to delete; then, click **Remove**.
2. Click **OK** to close the dialog box.
Variables in BI Query

A variable (or automatic variable) is a predefined operation that you can use to qualify a query. Typically, variables qualify attributes, but you can also use them in object qualifiers. Using variables can simplify potentially complex queries.

A variable is defined by an expression. When you run a query that includes a variable, BI Query evaluates the corresponding expression and adds it to the WHERE clause for the query.

A variable can contain attributes, operators, values in the database, calculations, functions, subqueries, prompts, and other variables (anything that is valid in an SQL WHERE clause). You can also use special placeholders in the expression for a variable to increase its scope.
Chapter 7: Qualifying with Variables

Variables are stored in the Prompts file in the Queries folder and therefore must have names that are different from those given to prompts. You can import variables stored in another user’s Prompts file.

If a query using a variable is converted into a query data object, the variable is preserved within the object. For more information on query data objects, see the BI Query Data Models User’s Guide.

Predefined Variables

BI Query includes three predefined variables:

**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 cannot edit the predefined variables.

Typical Uses for Variables

Variables can carry out a variety of operations. In their simplest form, they can retrieve values in the database so you don’t have to specify them in queries. For example, when you need information about retailers in a particular country, you can use a variable to determine the states or provinces that are included in that country, rather than specify each state or province yourself.

Variables can also perform calculations based on database functions to work out date, time, and other values. For example, you can use a variable to retrieve retailer sales that were placed today, before today, and so on. The variable supplies what today’s date is, and BI Query returns the corresponding results.

Variables are especially useful for queries that are run on a regular basis or are scheduled to run at specific intervals because they can automatically supply data values for the time when they are run.
Expressions for Variables

When you run a query qualified with a variable, BI Query evaluates the expression for the variable and adds the result to the \texttt{WHERE} clause for the query. Each variable usually contains information about what attribute to qualify, what operator to use, and what value to test against the attribute.

There are three ways to provide attribute and operator information in a variable:

- Use an actual attribute name and operator.
- Use placeholders for attributes and operators.
- Use the \texttt{Retain Attribute And Operator} option to ensure that the \texttt{WHERE} clause includes the attribute and operator specified in the qualification.

Literals in Variables

One of the three ways to provide attribute and operator information in a variable is to use an actual attribute name and operator in the expression.

A user-defined variable that names an actual attribute and operator qualifies that attribute using that operator, even if you insert it into the qualification for a different attribute in the query. Unless you define the variable with placeholders or enable the \texttt{Retain Attribute And Operator} option, BI Query removes the qualified attribute and qualification operator from the \texttt{WHERE} clause for the query and replaces them with the attribute and operator defined in the variable.

Example

A query includes the Store Name and State attributes of the Store data object. The State attribute is qualified with the user-defined variable Current Ratio.

The expression for Current Ratio is

\texttt{Store.Receivable > Store.Payable}

When you run the query, BI Query substitutes the variable for the qualification of the State attribute. In other words, the original qualification,

\texttt{Store.State = «Current Ratio»}

becomes
Store.Receivable > Store.Payable

The syntax for variable expressions depends on the DBMS you’re using; the examples provided here are based on ODBC.

Placeholders in Variables

As an alternative to actually specifying attribute and operator information in a variable, you can create a variable that uses placeholders for the attribute and operator. The placeholders let you create variables that return results based on the particular attribute qualified and the operator used in the qualification.

There are two placeholders for variables:

^Attribute^  
The placeholder for the qualified attribute. When you run a query containing the ^Attribute^ placeholder, BI Query replaces the placeholder with the qualified attribute; this attribute becomes part of the WHERE clause for the query.

^Operator^  
The placeholder for the operator in the qualification. When you run a query containing the ^Operator^ placeholder, BI Query replaces the placeholder with the operator from the qualification; this operator becomes part of the WHERE clause for the query.

Use the ^Attribute^ and ^Operator^ buttons in the Edit Variable dialog box to insert placeholders in the expression for a variable.

If you want your variable to start with the ^Attribute^ and ^Operator^ placeholders in that order (such as ^Attribute^ ^Operator^ 1000), select the Retain Attribute and Operator check box instead.

The predefined variables, DBUserID, BIUserID, and BIGroupID, use placeholders for attributes and operators. This means you can directly qualify any attribute using these variables.

Example

The variable Exceeding Credit is defined as follows:

^Attribute^ ^Operator^ Retailers.Credit

If you qualify the attribute Receivable in the Retailers data object with this variable,
Retailers.Receivable > «Exceeding Credit»
then the actual qualification becomes
Retailers.Receivable > Retailers.Credit
If you qualify the Invoice_amount attribute of the Retailers object with the same variable,
Retailers.Invoice_amount < «Exceeding Credit»
then the query returns the list of retailers who have been invoiced for amounts less than their credit limit:
Retailers.Invoice_amount < Retailers.credit
You can use the placeholders as building blocks to create complex expressions based on the qualified attribute and qualification operator:
{fn week(^Attribute^)} ^Operator^ 12 AND
{fn year(^Attribute^)} ^Operator^ 2002
You do not need to use both placeholders in the same expression. The following expression lets you control whether the query returns rows for a Receivable value greater than, less than, or equal to $50,000:
Retailers.Receivable ^Operator^ 50000

The Retain Attribute and Operator Option
To ensure that a variable always applies to the attribute and operator specified in the qualification, you can enable the **Retain Attribute And Operator** option in the **Edit Variable** dialog box. Use this option whenever you want BI Query to treat a variable as a value or set of values rather than a complete expression.

Example
The Threshold variable is defined as follows:
50000
In this case, the expression contains no attribute or operator information. If you enable the **Retain Attribute And Operator** option when you create this variable, BI Query evaluates the following qualification
Store.receivable >= «Threshold»
as
Store.receivable >= 50000
If you do not enable the **Retain Attribute And Operator** option, BI Query replaces the entire qualification with the contents of the variable, which may produce unexpected results.
Functions, Prompts, and Other Values in Variables

Typically, the expression for a variable takes the form of a comparison between an attribute and a value. An operator specifies the nature of the comparison (for example, `Sales.order_id = '10002'`). The expression can also include functions, prompts, or other variables. These additional terms often supply the comparison value for the attribute.

**Functions**

Use a function in a variable to create an expression that performs a calculation. For example, you can include in an expression the DBMS function for the current date:

```
ATTRIBUTE OPERATOR {fn now()} - {fn dayofmonth({fn now()})}
```

This variable determines the date for the beginning of the month. You can use this variable to find all retailer sales received since that date.

You can also apply functions to attributes in the expression:

```
{fn abs(Store.receivable - Store.payable)} OPERATOR 50000
```

For more information on functions, see “Including Functions in Query Expressions” on page 197.

**Prompts**

Use a prompt in a variable if you want to qualify the query but do not want to restrict the comparison to a specific value. For example, in the following expression,

```
{fn week(ATTRIBUTE)} OPERATOR «weekofyear!1week» AND
{fn year(ATTRIBUTE)} OPERATOR «weekofyear!2year»
```

the variable includes the weekofyear group prompt. This variable returns results based on the week and year that you specify.

**Other Variables**

Use other variables in a variable as the building blocks for a more complex expression. For example, the following expression,

```
ATTRIBUTE OPERATOR «Today» - 1
```

includes the Today variable, which determines today’s date. The resulting expression returns yesterday’s date.
Creating and Editing Variables

Before you create the expression for a variable, you can test that it returns the results you want by creating a calculated attribute using the same expression. Use the attribute in a query, and then verify the results. If the expression is correct, you can use the **Edit Variable** dialog box to create the variable.

💡 For more information, see “Creating Calculated Attributes” on page 193.

To create or edit a variable:

1. On the **Edit** menu, click **Variables**. The **Variables** dialog box opens.
2. Click **New** to create a new variable or, to edit an existing variable, select the variable; then, click **Edit**. The **Edit Variable** dialog box opens.

    💡 You cannot edit any variables with lock icons beside them.

3. Type or edit the name and description of the variable in the **Name** and **Description** boxes.
4. In the **Expression** box, type the expression that supplies the value for the variable. (You can include database values and functions, calculations, subqueries, other variables, and prompts.)

You can also use the available buttons to build the expression:

- To insert an attribute into the expression, select a data object from the **Attributes** list; then, select the attribute from the list below. Click **Insert**.
- To specify a placeholder for an attribute, place the cursor in the **Expression** box; then, click ^Attribute^.
- To specify a placeholder for an operator, place the cursor in the **Expression** box; then, click ^Operator^.
- To insert a prompt, click **Prompts**; then, use the **Prompts** dialog box to create or select the prompt and insert it into the expression.

💡 For more information on prompts, see “Prompts in BI Query” on page 89.

- To insert a variable, click **Variables**; then, use the **Insert Variable** dialog box to select the variable and insert it into the expression.
- To insert a function, click **Functions**; then, use the **Functions** dialog box to create or select the prompt and insert it into the expression.
Chapter 7: Qualifying with Variables

If you are typing the expression yourself, use curly braces ({} ) to enclose functions or formats supported by your DBMS that BI Query does not support. If you are using the buttons, BI Query automatically inserts the curly braces for you as necessary.

5. If you want BI Query to evaluate the variable as a value rather than a complete qualification, select Retain Attribute And Operator.

6. Click OK. In the Variables dialog box, click Close.

Qualifying with Variables

You can qualify an attribute so that a variable specifies the search conditions for the query. A variable can contain anything that is valid in an SQL WHERE clause (attributes, operators, values in the database, calculations, functions, and subqueries). Variables are especially useful for queries that run on a regular basis or are scheduled to run at specific intervals using BI Server’s scheduling facility.

💡 For more information on scheduling, see the BI Server Scheduling Guide.

To qualify an attribute with a variable:

1. Click the Qualify box for the attribute. The cursor will move to the Qualification box for the attribute in the qualification tree.

2. On the Edit menu, click Variables.

3. In the Variables dialog box, select the variable you want (or create or edit one).

4. Click Insert.

Importing Variables from Files

When you want to use variables created by another user, you can import them from that user’s Prompts file.

To import variables from a file:

1. On the Edit menu, click Variables.

2. In the Variables dialog box, click Import.

3. In the Select Import File dialog box, open the Prompts file containing the variables you want to add to the current data model. The Import Variables dialog box opens.
4. Select the variable(s) you want from the Available Variables list; then, click the right arrow button to move the selected variables to the Current Variables list.

5. Click Import; then, in the Variables dialog box, click Close.

Supplying User and Group Names with Variables

The creator of the data model can include a security table—a data object that associates data with the value(s) returned by one of the predefined variables (DBUserID, BIUserID, or BIGroupID). If such an object exists in the data model, you can use the predefined variables to qualify a query so that it retrieves results customized for you. If you distribute that query to other users, it retrieves results customized for each of them. You can also use these variables to restrict data to certain users or groups.

💡 For more information on security tables, see the Bi Query Data Models User’s Guide.

To qualify a query using a user or group name variable:

1. Double-click the data object that serves as the security table.

2. In the attribute window, click the Qualify box for the attribute you want to qualify. This attribute must contain data that corresponds to the data returned by the variable you want to use. (For example, BIUserID returns the BI Server user name for the current user. The attribute that you qualify with this variable must store BI Server user names.) The cursor will move to the Qualification box in the qualification tree.

3. Click the Data Values button; then, select Variables from the list.

4. In the Variables dialog box, select the appropriate variable (DBUserID, BIUserID, or BIGroupID).

5. Click Insert.

⚠️ Users must be logged on to BI Server when running queries that execute the BIUserID or BIGroupID variables. Otherwise, the variable cannot supply the user or group name(s). This is true even if the user is running a local copy of the query that would not otherwise require logging on to BI Server.
Chapter 8

Fine-Tuning Results

This section provides information on the following:

- “Results Sets” on page 117
- “Modifying Queries for Selected Results” on page 120
- “Super Queries” on page 120
- “Filtering Results” on page 128
- “Combining Results Sets” on page 132

Results Sets

Results are the information returned from your database after you run a query. Query results can be used “as is” or as a source of data for a report. Results can be returned to a results window on your computer screen, where they can be combined, filtered, and formatted, as well as stored and distributed, without further database access. You can also run a query so that the results are stored in a text file or in the database, and then load the results at a later time.

💡 For more information on saving results as text, see “Exporting Results to Text Files” on page 232.
If your querying environment includes BI Server and you have the necessary system permissions, you can schedule a query to run automatically. In that case, the results are returned to the BI Server Repository, where they can be retrieved by you as well as by other users with appropriate permissions. All the results returned from a query, taken together, are called a results set.

**Formatting Results**

You can format the results that appear in a results window, either by formatting the attributes involved in the query or by formatting the result columns directly.

💡 For more information on formatting results, see “Data Formats in BI Query” on page 63.

**Spreadsheet and Form Views**

By default, results are returned in a spreadsheet view in the results window. It displays multiple records (each record is displayed as a row in the spreadsheet). The columns reflect the attributes selected in the query. The name of the results window corresponds to the name of the query.

💡 You can search for specific content in query results. For more information, see the Bi Query Data Models User’s Guide.

You can also display a row in form view by double-clicking the row number in the results window. Form view displays one row at a time and provides a way to navigate through the rows you have retrieved.

**Opening and Closing Results**

BI Query automatically creates a results set whenever you run a query. The name of the results set matches the name of the query. If you run several queries in a row, BI Query adds a numerical suffix to the name of each successive results set.

You can also open a saved results set. BI Query saves results as two files: one with the extension .qrd (ASCII text format for use in other applications) and one with the extension .qrr (BI Query-specific information, such as formats for each column and the SQL used to retrieve the results). When you open results in BI Query, only the .qrd file appears in the Open Results File dialog box.
To open a results file:
1. On the Results menu, click Open.
2. In the Open Results File dialog box, click External Results.
3. Navigate to the file you want to open and select it.
4. Click Open.

To close all open results sets:
- On the Results menu, click Close All Results.

Selecting Results
You can select all or a portion of the results in the results window in order to copy them for export into another application or to update them.

To select results:
- Do one of the following:
  - To select partial results, click the cell or the row number you want, or hold down the SHIFT key while you click each of several cells or row numbers.
  - To select a block of cells, drag through the cells you want or click the first cell or row number, hold down the SHIFT key; then, click the last cell or row number.
  - To select all the results, click the total row number in the upper left corner of the spreadsheet. Alternatively, on the Edit menu, click Select All.

Changing Column or Cell Width
Results columns are sized to fit the longest value up to 15 characters. When a column is too narrow to view all the data in it, or when it is wider than necessary, you can change its width. You can also change the width of individual cells.

To change the column width in a results window:
1. Position the pointer on the vertical line between two columns. It changes to a double-headed arrow.
2. Drag the line to the right to widen the column or to the left to narrow it.

To expand a cell:
- Double-click the cell.
Modifying Queries for Selected Results

If you want to retrieve more detailed data about specific results you have retrieved, you can do so in Ad Hoc Drill-Down Mode. In this mode, you select one or more results cells, and then run a query that returns results only for those values. BI Query automatically adds \texttt{IN} clauses to your query.

To modify a query for selected results:

1. In the results window, on the Results menu, click \textit{Ad Hoc Drill Down}.
2. Select the cell(s) on which to base your query.
   - When you select a single cell, \texttt{[drilldown]} appears in the qualification tree instead of the contents of the cell.
   - When you select more than one cell, the \texttt{List} button appears; you can use it to view and modify the list of cells you selected.
3. Do one of the following:
   - To retrieve results that match these items, re-run the current query.
   - To retrieve different results for these items, modify the current query, and then run it again.
4. To work with the same subset of results later, save the new query.

To cancel Ad Hoc Drill-Down Mode:

- In the results window, on the Results menu, click \textit{Ad Hoc Drill Down}.

Super Queries

A super query is a query that either returns a results set and then filters it or returns multiple results sets and combines them. You can save, open, and edit these queries in BI Query. However, you cannot use them in data values query files and you cannot save their SQL as a text file.

💡 For more information about editing super queries, see “Guidelines for Editing Super Queries” on page 123.

Super queries are made up of component queries and operations. A component query can be either a standard query or a freehand query.
The Super Query Window

Use the **Super Query** window to combine queries. You can also use it to sort results, reorder columns, and specify a range of results. The **Super Query** window is divided into two panes separated by a splitter bar, which you can use to resize the panes.

[Diagram of Super Query window with operations and components]

💡 For more information on combining results, see “Combining Results Sets” on page 132.

Top Pane

The top pane displays the structure of the query using a tree list. Each component query and each operation is included in the tree and is represented by an icon. The operations performed on a component query’s results appear above the component query in the tree. For example, if the results of a component query are reordered, and then sorted, the tree list shows three branches: first the component query, and then the reorder operation, and then the sort operation above it.

💡 If a join or append operation is performed on two of the component queries, it is shown in the window above the queries and their associated operations. Any operation that appears above a join or append operation is applied to the results of the join or append.
Chapter 8: Fine-Tuning Results

Bottom (Message) Pane

The bottom, message pane displays information about the currently selected item in the query. When you select a component query, the bottom pane displays the query’s SQL string. When you select an operation, the bottom pane shows information about what the operation does to the query results. If there is an error in the query, the bottom pane displays information about why the error occurred and how to correct it.

Super Query Toolbar

Use the Super Query toolbar to create and edit a super query. You can also perform these operations using the Super Queries submenu of the Query menu.

💡 For more information on the Super Query toolbar, see the Help for your BI Query application.

Example

The following example shows the Super Query window for the Retailer Accounts query:

![Super Query Window]

When you are working with complex queries, you can see where a join or append occurs by collapsing the tree structure, and then expanding the branches one by one.

This super query performs the following sequence of operations:

1. Retrieves the results of the Order Amounts query.
2. Applies a range to the results.
3. Sorts by Store ID.
4. Retrieves the results of the Retailer Receivables query.
5. Reorders the columns.
6. Sorts by Store ID.
7. Joins the results from step 3 with the results from step 6.
8. Reorders the columns to produce the final query.

Opening the Super Query Window

To open the Super Query window for the current query or active results set:

- On the Query menu, point to Super Queries, and then click Show Super Query. The new query appears in the Super Query window where you can combine it with saved queries.

  The Super Query window also appears when you open a saved query that was created using the Super Query window.

Guidelines for Editing Super Queries

When you want to change the results a combined query retrieves, you can edit it in the Super Query window. You can change each component query (each query that makes up the larger query) and each operation performed on the results it returns. For example, you can add an attribute to a component query, and then change a sort operation to use the new attribute.

When you edit a query in the Super Query window, follow these guidelines:

- Start with the innermost query and work up through the tree. This makes it easier to correct any errors.
- Save your work regularly.
- Do not edit a combined query by modifying the SQL string. Your changes will not be checked for errors.

Editing Component Queries

Component queries (the individual queries that are part of the larger super query) can be either standard queries or freehand queries. How you edit a component query depends on what type of component query it is.
Instead of editing a component query, you can also replace a component query with another saved (standard, freehand, or super), open (freehand or super), or current (standard) query.

**Editing a Standard Component Query**

When you edit a standard component query, BI Query copies it from the Super Query window into the data model. The component query then becomes the current query, allowing you to open the attribute windows you want to use and make your changes. When you have finished editing the query, replace the component query in the super query with the current query you have just edited. If any errors occur as a result of making the changes, you need to correct them before you can save and run the query.

**To edit a standard component query:**

1. In the Super Query window, click the component query you want to edit.
2. On the Query menu, point to Super Queries, then click Edit Query.
3. Minimize the Query window for the component query.
4. In the Design window, click the appropriate data object(s) and make your changes to the component query.
5. In the Super Query window, click the component query you have edited.
6. On the Query menu, point to Super Queries, then point to Replace Query With, then click Current Query.
   
   If you cannot find the Super Query window in the list, on the Window menu, click More Windows; then, use the Select Window dialog box to open the window.
7. Correct any errors.
8. Save the query.

**Editing a Freehand Component Query**

When you edit a freehand component query, BI Query opens it in the Freehand Query window where you can make whatever edits you require. When you have finished editing the query, replace the component query in the super query with the open freehand query you have just edited. If any errors occur as a result of making the changes, you need to correct them before you can save and run the query.
To edit a standard component query:
1. In the **Super Query** window, click the component query you want to edit.
2. On the **Query** menu, point to **Super Queries**, then click **Edit Query**.
3. In the **Freehand Query** window, make your changes to the component query.
4. In the **Super Query** window, click the component query you have edited.
5. On the **Query** menu, point to **Super Queries**, then point to **Replace Query With**, then click **Open Query**.
6. In the **Choose Open Query** dialog box, click the freehand query you just edited, then click **OK**.
7. Correct any errors.
8. Save the query.

Replacing a Component Query

When you edit a component query in a super query, you are actually opening a new instance of the query and making your changes to the new query. To have those changes included in the super query, you must then replace the original component query with the edited one. Although this editing scenario is likely the most common use of the **Replace With** command, you can also use the functionality to replace a component query with any current, saved, or open query.

<table>
<thead>
<tr>
<th>Select</th>
<th>To replace the component query with...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Query</td>
<td>The standard query currently open in BI Query. (There can only ever be one standard query open at one time in BI Query.)</td>
</tr>
<tr>
<td>Saved Query</td>
<td>Any saved query, regardless of type (standard, super, or freehand) or location (internal or external). If you have been assigned the <strong>Edit Freehand Query</strong> permission, you can also specify a text-based file containing SQL.</td>
</tr>
<tr>
<td>Open Query</td>
<td>Any open freehand or super query (other than the super query you are already editing).</td>
</tr>
</tbody>
</table>
To replace a component query:
1. In the Super Query window, click the component query you want to replace.
2. On the Query menu, point to Super Queries, then point to Replace Query With, then do one of the following:
   • Click Current Query to replace the component query with the open standard query.
   • Click Saved Query to replace the component query with a saved query. The Open Query dialog box appears which you can use to select the saved query (or text-based file containing SQL) you want to use.
   • Click Open Query to replace the component query with an open freehand or super query. The Choose Open Query dialog box appears which you can use to select the freehand or super query you want.
3. Correct any errors.
4. Save the query.

Editing Operations in the Super Query Window
You can change how an operation manipulates query results by editing the operation in the Super Query window. (You cannot edit append operations.)

When you select the operation you want to edit, BI Query opens the same dialog box you used to create the operation. (The Join Columns dialog box is slightly different. You cannot change the results sets that are being joined.) When you click OK in the dialog box, the changes are applied to the query. If any errors occur as a result of the changes, you need to correct them before you can save or run the query.

To edit an operation:
1. In the Super Query window, click the operation you want to edit.
2. On the Query menu, point to Super Queries, and then click Edit Operation.
3. In the dialog box that opens, make your changes to the operation. (The type of operation you are editing determines which dialog box appears.)
4. Click OK.

To delete an operation:
1. In the Super Query window, select the operation you want to delete.
2. On the Query menu, point to Super Queries, and then click Delete Operation.
3. If an alert dialog box opens, select the branch of the join or append operation that you want to delete.

To delete a component query, you must delete its join or append operation. Each branch in a join or append operation corresponds to a different component query.

Correcting Errors in Super Queries

You must correct any errors in the Super Query window before you can save, run, or debug the query. You can correct errors by editing either the operation or its associated component query. Most errors occur because an attribute used in an operation is missing from the component query on which the operation is performed.

To correct an error in the Super Query window:

1. In the Super Query window, click the operation causing the error. When an error occurs, BI Query displays an error icon beside the corresponding operation.
2. Read the information in the message pane. The message pane provides information about why the error occurred and how to correct it.
3. Edit the component query or operation. If you are editing more than one operation, start with the operation immediately above its corresponding query and work up through the tree.
4. Save the query.
5. To test the query, on the Query menu, click Run.

Troubleshooting Super Queries

Troubleshooting a super query is useful if the query returns unexpected results. To help you understand the operations that BI Query performs to process a super query, you can run the super query in Troubleshooting mode.

In Troubleshooting mode, BI Query halts the processing of the super query after performing each operation; you can then view the outcome of each operation by itself. Upon your prompting, BI Query will either resume the super query or cancel it.
To troubleshoot a super query:

1. With the Super Query window active, on the Super Query toolbar, click the Troubleshoot Super Query button . The button will remain raised, indicating that the query is in Troubleshooting mode.

2. On the Standard toolbar, click the Run Query button to run the super query. BI Query performs the first operation in the super query.

3. When BI Query prompts you to continue, do one of the following:
   - Click Cancel to stop the super query at the current operation and investigate the results.
   - Click OK to perform the next operation in the super query; then, repeat step 3 as necessary.

As BI Query runs the super query, it highlights the current query or operation. To return the super query to normal operation mode, click the Troubleshoot Super Query button again.

Filtering Results

You filter results in order to see different views of your data. You can specify a sort order before you run the query or after you have results. When you specify the sort order before running the query, the DBMS does the sorting. You place fewer demands on the DBMS by specifying the sort order in the results, since BI Query sorts the results. However, if you specify the sort order in the results, you cannot save the query as a data values query file, and you cannot save the SQL for the query as a text file.

All forms of filtering results—sorting results after the query has been run, setting a range, and performing calculations on results—cause your query to become a super query when saved from a results window. You can also filter results by adding operations in the Super Query window.

For information about narrowing the range of results and performing calculations on results, see “Narrowing the Range of Results” on page 131.
Sorting Results

The rows in any column of results can be sorted into alphabetical, numerical, or chronological order (or the reverse), as appropriate. You can determine which columns to sort and the order in which they are to be sorted. Using the Sort dialog box, you can specify a sort order before you run the query or after you have results. For example, if you have a column of customer names and a column of delivery dates, you can specify whether they should be organized alphabetically by customer name (with delivery dates given for each customer given chronologically) or chronologically by delivery date (with an alphabetized list of customers who received delivery for each date).

To sort results:

1. Open the Sort dialog box by doing one of the following:
   - Open the Super Query window. On the Query menu, point to Super Queries; then, click Sort.
   - On the Results menu, point to Filter; then, click Sort.
   - With the attribute window or the Query window (in Query View mode) active, click Sort on the Query menu.

2. From the Available Columns list, click each column or attribute that you want to sort; then, click the right arrow button. The selected items move to the Sort Rows By list. The top-to-bottom order of items in the Sort Rows By list corresponds to the sort order (the topmost item is sorted first).

3. To change the position of an item in the Sort Rows By list, select the item; then, click the Up, Down, Top, or Bottom buttons as appropriate.

4. To sort an item in reverse alphabetical, numerical, or chronological order, select the item; then, click the Descending Sort button.

5. To sort an item in alphabetical, numerical, or chronological order, select the item; then, click the Ascending Sort button.

6. Click OK.

Reordering Columns

Columns appear in results windows and standard reports in the order in which you chose attributes while creating the query. You can change that order before running the query or after you have results.
For more information on ordering columns, see “Reordering Columns Before Running Queries” on page 37.

To reorder columns after you have results:

1. To open the Reorder Columns dialog box, do one of the following:
   - Open the Super Query window. On the Query menu, point to Super Queries; then, click Reorder Columns.
   - In the Results window, on the Results menu, click Reorder Columns.
   - With a Standard Report active, on the Report menu, click Reorder Columns.

   In the Reorder Columns dialog box, the top-to-bottom order of columns in the Columns list corresponds to the left-to-right order of columns in the results window.

2. From the Columns list, select the column whose position in the list you want to change. Use the Up, Down, Top, or Bottom buttons to move the selected column to the desired position.

3. To hide a result column, select it from the Columns list; then, click Hide. A red X icon appears beside the column in the list. Hidden columns do not appear in the results window or in any report based on that results window.

4. To display a hidden column, select it from the Columns list; then, click Show.

   You cannot hide or show columns in a standard report directly. You must hide or show the column in the results set or super query, and then generate the report.

5. Repeat steps 2–4 until you have achieved the desired column order.

6. Click OK.

Hiding and Showing Columns

You can temporarily remove one or more columns of results from the results window. You may hide a column in order to exclude it from a report.

To hide a column:

1. Select the column you want to hide.

2. On the Results menu, click Hide Column.
To make a column visible:
1. On the **Results** menu, click **Reorder Columns**.
2. In the **Reorder Columns** dialog box, select a hidden column in the **Columns** list.
3. Click **Show**; then, click **OK**.

Narrowing the Range of Results

To help you analyze your data, you can focus on a particular range of results. Specifying a range for results includes only the rows within that range. For example, you can specify retailer credit limits between $200,000 and $600,000, sales received between June 30 and December 31, and salespeople’s names between the letters A and S. You can also focus on results outside a range.

To narrow a range of existing results:
1. In a **Results** window, point to **Filter** on the **Results** menu; then, click **Range**. The **Range Filter** dialog box opens.
2. From the **Where** list, select the column you want to filter.
3. In the **Is > Than Or =** box, type the minimum value for the range.
4. In the **Is < Than Or =** box, type the maximum value for the range.
5. Do one of the following:
   - To display the results that lie within the specified range, select **Include Rows**.
   - To display the results that lie outside the specified range, select **Exclude Rows**.
6. Click **OK**.

You can also specify a range for results before running a query using the **BETWEEN** and **NOT BETWEEN** operators in the qualification tree. For more information, see “Finding Values in a Range” on page 54.

Performing Calculations on Results

You can perform some basic calculations on results. You can also create an attribute that performs more complex calculations you define as part of a query.

To display basic calculations for a results set:
- On the **Results** menu, point to **Filter**; then, click **Compute**.
  
  In the **Compute Results** information box, minimum, maximum, sum, and averages for each column appear.
Combining Results Sets

In order to enlarge the scope of your data, you can combine results from multiple queries into one results set. You can do the following operations:

- append the rows of one results set to another
- join one or more columns of one results set to another

For information on troubleshooting combined results, see the Help for your BI Query application.

The results sets can be from queries run in different Design windows within the model. Because different windows can connect to different databases, you can combine information from two different databases.

You can combine the results in two ways:

- Before you run the corresponding queries. In this case, you use the Super Query window to combine the results of the two queries.
- After you run the queries. In this case, you specify the results sets you want to combine directly.

If you save the query after combining results, you can open and edit it in the Super Query window. (However, you cannot save the corresponding SQL string as a text file.) You can also link the combined query to a button to perform the combine operations automatically.

For more information on buttons, see the BI Query Data Models User’s Guide.

Rules for Appending Rows

When you have similar sets of results (for instance, identical kinds of data from two different years) that you want to present together, you can append the rows of one results set to another. If the two sets have different column names, BI Query copies the name(s) from the first set to the combined set.

Adhere to the following rules when you append the rows of one results sets to the rows of another:

- The data types of the columns in the first results set must match the data types of the corresponding columns in the second results set. For example, if the first results set contains two columns, and their data types are character and numeric, respectively, the first two columns of the second results set must also be character and numeric.
• The order of columns in one results set must match the order of columns in the other. One set may have additional columns that the other does not have as long as these extra columns are at the end (on the right).

Example

In the following example, the rows from Query Results 2 have been appended to Query Results 1 to form a new results set.

<table>
<thead>
<tr>
<th>Query Results 1</th>
<th>Query Results 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name</td>
</tr>
<tr>
<td>Sales ($)</td>
<td>Sales ($)</td>
</tr>
<tr>
<td>State</td>
<td>State</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Sales ($)</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abicon Inc.</td>
<td>12,000</td>
<td>NULL</td>
</tr>
<tr>
<td>Textal Ltd.</td>
<td>113,000</td>
<td>NULL</td>
</tr>
<tr>
<td>Unisystems</td>
<td>10,145</td>
<td>NULL</td>
</tr>
<tr>
<td>Ziptech</td>
<td>6,430</td>
<td>NULL</td>
</tr>
<tr>
<td>Bainstream Inc.</td>
<td>10,000</td>
<td>New York</td>
</tr>
<tr>
<td>Knatl Corp.</td>
<td>2,000</td>
<td>California</td>
</tr>
<tr>
<td>MonoLitho</td>
<td>2,100</td>
<td>Ohio</td>
</tr>
<tr>
<td>Textronics</td>
<td>60,000</td>
<td>Illinois</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Sales ($)</th>
<th>State</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Sales ($)</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abicon Inc.</td>
<td>12,000</td>
<td>NULL</td>
</tr>
<tr>
<td>Textal Ltd.</td>
<td>113,000</td>
<td>NULL</td>
</tr>
<tr>
<td>Unisystems</td>
<td>10,145</td>
<td>NULL</td>
</tr>
<tr>
<td>Ziptech</td>
<td>6,430</td>
<td>NULL</td>
</tr>
<tr>
<td>Bainstream Inc.</td>
<td>10,000</td>
<td>New York</td>
</tr>
<tr>
<td>Knatl Corp.</td>
<td>2,000</td>
<td>California</td>
</tr>
<tr>
<td>MonoLitho</td>
<td>2,100</td>
<td>Ohio</td>
</tr>
<tr>
<td>Textronics</td>
<td>60,000</td>
<td>Illinois</td>
</tr>
</tbody>
</table>
Combining Results by Appending Rows

Use the Append Rows dialog box to append one results set to another.

To append rows using existing results sets:

1. Open the results sets you want to combine, either by running the corresponding queries or by opening saved results set files.
2. On the Results menu, point to Combine; then, click Append Rows. The Append Rows dialog box opens.
3. From the First Results list, select the first results set. By default, this is the active results set.
4. From the Second Results list, select the results set you want to append to the first set. By default, this is the first results set (other than the active results set) to be opened in the current session.
5. Click OK.

To append rows using queries:

1. Specify the first query by doing one of the following:
   - Open an existing super query. With the query active, point to Super Queries on the Query menu, and then click Show Super Query.
   - Run a query. With the results set active, point to Super Queries on the Query menu, and then click Show Super Query.
   The Super Query window opens.
2. On the Query menu, point to Super Queries; then, click Append Rows. The Append Rows dialog box opens.
3. Do one of the following:
   - To append the current query, select Current Query.
   - To append a saved query, select Saved Query. In the Choose Query dialog box, select the query and click Open.
   - To append an open freehand or super query, click Opened Query. In the Choose Open Query dialog box, select the query and click OK.
4. In the Append Rows dialog box, click OK. In the Super Query window, an append operation appears in the tree list above the component queries.
Rules for Joining Columns

You can join columns from different results sets to produce a new results set. The new results set can contain data from more than one data object (including actual relationships and query data objects) from more than one Design window. Typically, the BI Query administrator specifies the join operations between data objects when the model is created. However, you can produce the same results by joining columns from different results sets. In effect, you are creating a relationship between results sets that is equivalent to a relationship between joined data objects.

💡 For more information on creating joins between data objects, see the BI Query Data Models User’s Guide.

By convention, a column join has a “left” results set and a “right” results set. You specify which column in the left set joins to which column in the right. (You can join more than one pair of columns.) The combined results set represents each pair of joined columns in a single column.

Adhere to the following rules and guidelines when joining a column of one results sets with a column of another:

- The joined columns must have the same data type. However, you can join columns containing floating point and money data types.
- If the joins you need already exist in the data model, you may want to create a query that retrieves the results rather than a super query that combines results sets. The DBMS—which can handle a single query at a time—may be able to optimize the join operations more efficiently than BI Query.

Reasons for Joining Columns

You may want to join columns of results sets for the following reasons:

- The Design window does not provide joins for the attributes you need to join.
- The data model does not include permission to create relationships between tables.
- The database does not support outer joins.
- You need to combine results from more than one database.
- After examining recent results sets, you want to join certain columns but do not want to rerun the component queries (for example, they may require too much processing time).
Types of Joined Columns

When you join columns from two results sets, you can use one of four types of join: an inner join, a full outer join, a left outer join, or a right outer join.

- **Inner Join**: An inner join includes only those rows that have an exact match in the joined columns. (BI Query actually performs a type of inner join called a natural join, which displays the joined columns only once in the combined results set.)

- **Full Outer Join**: A full outer join includes all rows from both results sets, even if they do not match in the joined columns.

- **Left Outer Join**: A left outer join includes all rows from the left results set and those from the right that match in the joined columns.

- **Right Outer Join**: A right outer join includes all rows from the right results set and those from the left that match in the joined columns.

💡 For more information on types of joins, see the *BI Query Data Models User’s Guide*.
Unmatched data in a full, left, or right outer join is represented by a \texttt{NULL} in the joined column.

**Example**

In the following example, the first set of query results shows the total sales for products that a company has produced in the U.S. The second set of results shows the total payments received to date for all products sold by the company, including those produced outside the U.S.

An inner join of both results sets on Product ID shows all American products by the company that have been sold and received some payment to date. A left outer join on the same column shows the sales and payments of all American products, regardless of payment.

<table>
<thead>
<tr>
<th>Query Results 1</th>
<th>Query Results 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product ID</strong></td>
<td><strong>Sales ($)</strong></td>
</tr>
<tr>
<td>101</td>
<td>15,000</td>
</tr>
<tr>
<td>102</td>
<td>52,303</td>
</tr>
<tr>
<td>213</td>
<td>995</td>
</tr>
<tr>
<td>331</td>
<td>2,561</td>
</tr>
<tr>
<td>511</td>
<td>8,000</td>
</tr>
<tr>
<td>514</td>
<td>2,303</td>
</tr>
<tr>
<td>818</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inner Join on Product ID</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product ID</strong></td>
</tr>
<tr>
<td>101</td>
</tr>
<tr>
<td>102</td>
</tr>
<tr>
<td>331</td>
</tr>
</tbody>
</table>
Combining Results by Joining Columns

Use the Join Columns dialog box to specify which results sets to join on which columns. You must also specify the type of column join. The joined columns must have the same (or similar) data type. You can join one or more pairs of columns. (Typically, you should join columns that represent primary keys.)

You can join columns either by specifying the results sets directly, or by specifying the queries that generate the results sets.

To join columns using existing results sets:

1. Open the two results sets you want to join.
2. On the Results menu, point to Combine, and then click Join Columns.
3. Select the “left” results set from the Left Results list. By default, this is the active results set.
4. Select the “right” results set from the **Right Results** list. By default, this is the first results set (other than the active results set) to be opened during the current session.

5. Do one of the following:

   - To create an inner join, clear the **Include All Left Rows** and **Include All Right Rows** check boxes. (This is the default setting.)
   - To create a left outer join, select **Include All Left Rows**.
   - To create a right outer join, select **Include All Right Rows**.
   - To create a full outer join, select both check boxes.

6. Select the columns you want to join (up to eight pairs) in the order in which you want to join them.

   A joined column appears only once in the combined results set.

7. Click **OK**.

**To join columns using queries:**

1. Open or run one of the queries whose results you want to join. (This query will produce the “left” results set.)

2. With the query or results set active, point to **Super Queries** on the **Query** menu, and then click **Show Super Query**. The **Super Query** window opens.

3. On the **Query** menu, point to **Super Queries**, and then click **Join Columns**. The **Join Columns** dialog box opens.

4. Specify the query that will produce the “right” results set (or the “bottom branch” with respect to the super query) by choosing one of the options from the **Query for the bottom branch** list:

   - If you want to use the current standard query, click **Current Query**.
   - If you want to use a saved query, click **Saved Query**. In the **Choose Query** dialog box, select the query and click **Open**.
   - If you want to use an open freehand or super query, click **Opened Query**. In the **Choose Open Query** dialog box, select the query and click **OK**.

5. Do one of the following:

   - To create an inner join, clear the **Include All Left Rows** and **Include All Right Rows** check boxes. (This is the default setting.)
   - To create a left outer join, select **Include All Left Rows**.
Chapter 8: Fine-Tuning Results

- To create a right outer join, select **Include All Right Rows**.
- To create a full outer join, select both check boxes.

6. Select the columns you want to join (up to eight pairs) in the order in which you want to join them.

    A joined column appears only once in the combined results set.

7. Click **OK**.

### Troubleshooting Combined Results

If a query contains an error, you need to correct it before you can save or run the query. Some common errors and their solutions are presented below.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Reason</th>
<th>Solution</th>
</tr>
</thead>
</table>
| Append: Column \( n \) does not match | The attributes in the query results you are appending do not match because of one of the following:  
- you selected the attributes in the first query in a different order than the attributes in the second  
- an attribute used in one query is missing from another query | Edit the queries to reselect the attributes in a consistent order; or edit the queries to add the missing attribute. (You cannot edit an append operation.) |
| Reorder Error: Missing Column | One of the attributes used in the operation is missing because of one of the following:  
- the attribute is missing from the component query  
- you hid the column with a reorder operation | Edit the query to add the missing attribute. |
Chapter 9

Freehand Queries

This section provides information on the following:

- “Getting Started with Freehand Queries” on page 142
- “Creating and Editing Freehand Queries” on page 144
- “Saving Freehand Queries” on page 148
- “Running Freehand Queries” on page 148
- “Sending Non-SELECT SQL to DBMSs from the Query Window” on page 150
- “Working with Stored Procedures” on page 150
- “Using Freehand Queries in a BI Server Environment” on page 153
Chapter 9: Freehand Queries

Getting Started with Freehand Queries

A freehand query is simply SQL that is not tied to the graphical data-model-interface in BI Query. You can enter the SQL using the Freehand Query window, either by typing directly in the window or by pasting SQL into the window from another source, or you can open a saved file containing the SQL. Once created, freehand queries behave for the most part as standard queries do; you can save, open, and edit freehand queries in BI Query; publish them to, and retrieve them from, the repository; add them to super queries; link them to buttons in a data model, and so forth. You can use freehand queries to generate data results and then send the results to Standard Reports or BI Query Reports.

Because freehand queries are not integrated with data models in the same way that standard queries are, BI Query does not attempt to regenerate them before running them. This means any customizations you may have introduced are preserved when the data model is updated.

Another significant difference between standard queries and freehand queries is that when you save a freehand query, the connection document information is saved with the query. This allows you to work with multiple connection documents within a single Design window.

All users are able to open and run freehand queries, but only users who have been given the Edit freehand queries permission are able to create or edit freehand queries. If you have not been assigned this permission, and you open a freehand query, the Freehand Query window is read only.

If you have been given the Edit freehand queries permission, you can enter and run any valid SQL statement (or even multiple SQL statements) in the Freehand Query window. Thus, while it may be referred to as a ‘query’, the SQL statement might actually be non-SELECT SQL. You can use SQL scripts to do any of the following:

- create tables in the database for use in a data model
- grant database permissions
- drop tables (such as tables that users have defined)
- move tables to a different database
- add and update tables and data in the database
- send an EXEC statement to the database to call a procedure stored in the DBMS

💡 You should consult your database administrator before sending SQL scripts to the DBMS.
If you open a stored procedure in BI Query, the stored procedure is displayed in the **Freehand Query** window, where you can edit it if necessary. For more information on stored procedures, see “Working with Stored Procedures” on page 150.

The **Allow non-Select SQL** permission and the **Edit SQL string** permission only apply to standard queries; they do not apply to freehand queries. For example, if a user has been assigned the **Edit freehand queries** permission, but denied the **Allow non-Select SQL** permission, they can still enter and run non-Select SQL statements from the **Freehand Query** window.

One limitation you should be aware of when using freehand queries is that BI Query is not able to determine the data types of objects referenced in the SQL until the query has been submitted to the database. This means that while you can insert prompts (and variables) in a freehand query, you must ensure that the prompt data type matches the database attribute data type.

In BI Web, you can open and run freehand queries that you have created and published in BI Query, but you cannot create new freehand queries or edit existing freehand queries.

### The Freehand Query Window

Use the **Freehand Query** window to view and run freehand queries. If you have been assigned the **Edit Freehand Queries** permission, you can also use this window to create and edit freehand queries. The upper area of the **Freehand Query** window displays the SQL for the query. You can type in this area, as well as copy from and paste into this area.

Below the SQL pane, there are buttons that you can use to add prompts, variables, or functions to the query, or to run the query. You can also use the **Available Connections** drop-down list to specify a different connection file to use with the query.

You can have multiple **Freehand Query** windows open at the same time, and you can have the **Freehand Query** window open at the same time as the **Query** window or the **Super Query** window.

### Opening the Freehand Query Window

The **Freehand Query** window opens automatically when you do any of the following:
• Create a new freehand query. For more information, see “Creating a New Freehand Query” on page 144.
• Convert a standard query to a freehand query. For more information, see “Converting a Standard Query to a Freehand Query” on page 145.
• Open a saved freehand query. For more information, see “Opening Saved Queries” on page 40.
• Open a text-based file containing SQL. For more information, see “Opening Saved Queries” on page 40.
• Open a stored procedure. For more information, see “Opening Saved Queries” on page 40.
• Edit a component query of a super query where the selected component query is a freehand query. For more information, see “Editing Component Queries” on page 123.

Creating and Editing Freehand Queries

There are multiple ways to create freehand queries. You can:

• Start with a blank query and either type or paste SQL into it. For more information, see “Creating a New Freehand Query” on page 144.
• Open an existing standard query and have BI Query convert it into a freehand query. For more information, see “Converting a Standard Query to a Freehand Query” on page 145.
• Prepare a text-based file containing the SQL and then open it in BI Query. For more information, see “Opening Saved Queries” on page 40.

Once you have a freehand query open, you can edit it in the Freehand Query window. For more information, see “Editing a Freehand Query” on page 145.

Creating a New Freehand Query

When you create a new freehand query, you are basically just opening an empty Freehand Query window. You can then either type the SQL into the window or paste it in from another source via the Clipboard. You can also use the buttons in the Freehand Query window to add prompts, variables, and functions to your query.

You must have the Edit Freehand Query permission in order to create new freehand queries.
To create a new freehand query:

1. On the Query menu, point to Freehand, and then click New.
2. In the Freehand Query window, prepare your SQL statement.
3. When the SQL statement is complete, you can save or run the freehand query.

Converting a Standard Query to a Freehand Query

Rather than start with a blank freehand query, you can create a new freehand query based on a standard query. BI Query automatically transfers the SQL from the standard query into the Freehand Query window where you can edit it, run it, or save it.

You must have the Edit Freehand Query permission in order to convert a standard query to a freehand query.

To convert a standard query to a freehand query:

1. In BI Query, create a new standard query or open a saved standard query.
2. On the Query menu, point to Freehand, and then click Convert to. The SQL from the current standard query appears in the Freehand Query window.
3. Edit the freehand query as necessary, then save or run it.

Editing a Freehand Query

With the Freehand Query window open, you can type or paste in any SQL you want. In that sense, the Freehand Query window behaves like a basic text editor. In addition to simply typing the SQL, you can use the buttons provided in the Freehand Query window to add prompts, variables, and functions to your query. You can also specify the connection document you want the query to use.

Before you can edit a freehand query, you must have the Edit freehand queries permission.

Adding Prompts to Freehand Queries

Adding a prompt to a freehand query is similar to adding a prompt to a standard query. For more information on adding prompts, see “Prompts in BI Query” on page 89. However, with the greater flexibility of freehand queries comes added responsibility on behalf of the query author to ensure that the prompts behave as expected. When adding prompts to freehand queries, please follow these general rules:
Any prompt you add to a freehand query should exist in the data model's Prompts file, otherwise a generic Enter Value dialog box will appear and the prompt will be assigned a data type of Automatic Formatting.

Prompts you add to freehand queries should have a specific data type and not rely on Automatic Formatting.

Prompts in freehand queries should be mandatory.

A prompt in a freehand query should be linked to a data values query or a data values result set if you want to ensure the end user has a list of values from which to choose. Do not rely on the default select distinct data values list.

To add a prompt to a freehand query:
1. Open the freehand query you want to edit.
2. In the Freehand Query window, position the cursor at the point where you want to add the prompt.
3. Click Prompts. The Prompts dialog box appears.
4. Select the prompt you want to add.
5. Click Insert.

Adding Variables to Freehand Queries

Adding a variable to a freehand query is similar to adding a variable to a standard query. For general information on adding variables, see “Variables in BI Query” on page 107. However, with the greater flexibility of freehand queries comes added responsibility on behalf of the query author to ensure that the variables behave as expected. When adding variables to freehand queries, please follow these general rules:

- Any variable you add to a freehand query must exist in the data model's Prompts file.
- Variables you add to freehand queries should have a specific data type and not rely on Automatic Formatting.

To add a variable to a freehand query:
1. Open the freehand query you want to edit.
2. In the Freehand Query window, position the cursor at the point where you want to add the variable.
3. Click Variables. The Variables dialog box appears.
4. Select the variable you want to add.
5. Click Insert.

Adding Functions to Freehand Queries
Adding a function to a freehand query is similar to adding a function to a standard query. For general information on adding functions, see “Aggregate Functions and Grouping” on page 185.

To add a function to a freehand query:
1. Open the freehand query you want to edit.
2. In the Freehand Query window, position the cursor at the point where you want to add the function.
3. Click Functions. The Functions dialog box appears. BI Query automatically uses the freehand query’s connection document to connect to the database and retrieves the functions supported by the database.
4. Select the function you want to add.
5. Click Insert.

Using Calculated Attributes in Freehand Queries
You can add calculated attributes to freehand queries either by typing the information directly in the Freehand Query window, or by converting a standard query that contains a calculated attribute. Be aware that in the results, the column name corresponding to the calculated attribute may not be meaningful, or may change each time you run the query. If you want, you can specify a name for the calculated attribute column.

To specify a column name for a calculated attribute:
□ In the Freehand Query window, insert the text “as <column_name>” immediately after the calculated attribute expression (where <column_name> is the name you want to appear in the query results). For example, if the original query was:
   select (table.price - table.cost) from table;
   You could modify the SQL to include the column name Profit as follows:
   select (table.price - table.cost) as Profit from table;
**Specifying Connection Documents for Freehand Queries**

Each freehand query must have a connection document associated with it. By default, the connection document associated with new freehand queries is the default connection document for the current data model, but you can specify any of the connection documents available in your current connection folder when you are editing the query. When you save a freehand query, this association is saved with the query.

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.

**To specify a connection document for a freehand query:**

1. Open the freehand query you want to edit.
2. In the Freehand Query window, from the Available Connections list, choose the connection file you want to use.
3. Save or run the query.

**Saving Freehand Queries**

You save freehand queries the same way you save standard queries or super queries.

**To save a freehand query:**

- With the freehand query you want to save displayed in the active Freehand Query window, on the Query menu, click Save.

**Running Freehand Queries**

You run freehand queries in much the same way as you do standard BI queries or super queries, but there are a few minor differences in the execution of the query that you should keep in mind:

- You can specify a different connection document for each freehand query. When BI Query runs a freehand query, if the current connection is different from the connection defined for the freehand query, BI Query disconnects the current connection, establishes the connection specified for the query, runs the query, then reestablishes the original connection.
• Freehand queries can be composed of multiple SQL statements. As each statement is run, BI Query highlights the current statement in the Freehand Query window.

• Although you can include multiple SELECT statements in a freehand query, BI Query will only return results for the first SELECT statement in the query.

• Freehand queries can include non-SELECT SQL statements. With no results to indicate that such statements have completed, BI Query instead displays a message box to indicate when a freehand query that contains a non-SELECT SQL statement has completed. If you want, you can turn this message off or on by changing the Show Completed Freehand Dialog preference.

• When you run a freehand query, some additional status information (such as the start and finish time of the query, and the actual SQL sent to the DBMS) is sent to the Message window. For more information, see “Viewing and Clearing Messages” on page 61.

• You can cancel a freehand query before it has completed all of the SQL statements in the query. BI Query highlights the last command executed in the Freehand Query window.

To run a freehand query:

1. Create or open the freehand query you want to run.
2. Do one of the following:
   • On the Query menu, click Run.
   • In the Freehand Query window, click Run.

To cancel a freehand query:

□ When the freehand query is running, do one of the following:
   • On the Query menu, click Cancel.
   • On the Standard toolbar, click the Cancel Query button.
Sending Non-SELECT SQL to DBMSs from the Query Window

As an alternative to using freehand queries to send SQL to the DBMS, you can also use the Query window to send SQL commands to the DBMS, even commands that are not queries (that is, commands that do not include the SELECT statement). For example, you can send an EXEC statement to call a procedure stored in the DBMS. To send non-SELECT SQL using the Query window, the Edit SQL String and Allow Non-Select SQL permissions must be assigned; you also need to enable the Show SQL String preference.

💡 For more information on modifying the SQL in the Query window, see the BI Query Queries User’s Guide.

To send non-SELECT SQL to the DBMS:

1. Ensure you have the necessary BI Query permissions and database permissions. Enable the Show SQL String preference.
2. Open any query.
3. Open the Query window: on the Query menu, click Show. Alternatively, click the Show Query button on the Standard or Query toolbar.
4. Make sure the Query window is in SQL View mode. If necessary, click the Toggle Query Window button on the Query toolbar.
5. In the box at the bottom of the Query window, delete the SQL for the query you opened. Type the SQL statements you want to send to the database.
6. Run the query: on the Query menu, click Run or, on the Standard or Query toolbar, click the Run Query button.

Working with Stored Procedures

You can use BI Query to run stored procedures if you are connected to any of the following databases:

- ODBC
- Oracle (via SQL*Net)
- Teradata (via CLI)
If you have been assigned the **Edit Freehand Query** permission, you can also add a stored procedure call to a freehand query, modify any input parameters, and save the freehand query. Freehand queries that contain stored procedure calls behave the same as other freehand queries.

The **Allow non-Select SQL** permission and the **Edit SQL string** permission only apply to standard queries; they do not apply to freehand queries or to stored procedures. For example, a user who has been denied the **Allow non-Select SQL** permission can still run non-Select SQL statements from the Freehand Query window, or run stored procedures that include non-Select SQL statements.

If you run a stored procedure that requires input parameters (prompts), a corresponding prompt for each parameter must exist in the Prompts file in BI Query. When you run the stored procedure, BI Query attempts to create the required prompts based on the variable names in the stored procedure. If a prompt with that name already exists, BI Query uses the existing prompt. If the prompt doesn’t exist, BI Query automatically creates it. When you save the data model, any newly created prompts are also saved. For information on the naming conventions for prompts, see “**Naming Conventions for Prompts**” on page 91.

**Calling Stored Procedures from Freehand Queries**

You can use the **Stored Procedures** dialog box to select a stored procedure from your database, and have BI Query convert it to a procedure call in a freehand query. As part of the conversion, BI Query determines what parameters are required by the procedure, and sets up the appropriate prompts to collect the information for any input parameters. You can have BI Query prompt you (as the freehand query author) for the input values (in which case the prompts will be replaced in the freehand query with the values you supply, and these values will become the default values when other users run the query), or you can leave the prompts for the query users to fill in when they run the freehand query.

You must have the **Edit Freehand Query** permission in order to convert a stored procedure to a call in a freehand query.
To convert a stored procedure to a call in a freehand query:

1. On the Host menu, click Stored Procedures.

2. If you have not opened any stored procedures during the current session, the List Stored Procedures dialog box opens. If you are already connected with the connection document you want, use this dialog box to specify parameters for filtering the list of stored procedures, then click List and proceed to step 5. If you want to use a different connection document, proceed to step 3.

3. If you want to use a connection document other than the one displayed at the top of the Stored Procedures dialog box, use the drop-down list to select the connection document you want.

4. If the List Stored Procedures dialog box did not appear automatically, or the stored procedure you want is not displayed, or you have selected a different connection document, you can click List to open the List Stored Procedures dialog box, where you can specify new filter criteria.

5. Select the stored procedure you want the freehand query to call.

6. Click Prompt for input values when converting if the stored procedure requires input parameters, and you want BI Query to prompt you for those values immediately. BI Query will automatically replace the prompts in the stored procedure call with the values you provide. You can subsequently edit the values in the Freehand Query window if you want.

7. Click Add to current freehand query if you want to add the converted procedure call to the active open freehand query. Clear this check box if you want to add the converted procedure call to a new freehand query.

8. Click Convert. BI Query converts the selected stored procedure to a procedure call, and adds the converted call to the Freehand Query window.

Running Stored Procedures

You can run stored procedures from the Stored Procedures dialog box, or as part of a freehand query. For information on running a stored procedure from a freehand query, see “Calling Stored Procedures from Freehand Queries” on page 151.

When you run a stored procedure, information is sent to the Message window. For more information, see “Viewing and Clearing Messages” on page 61.
To run a stored procedure from the Stored Procedures dialog box:

1. On the Host menu, click Stored Procedures.

2. If you have not opened any stored procedures during the current session, the List Stored Procedures dialog box opens. Use this dialog box to specify parameters for filtering the list of stored procedures.

3. In the Stored Procedures dialog box, select the connection document you want to use.

4. If the stored procedure you want is not displayed, click List to open the List Stored Procedures dialog box, where you can specify new filter criteria.

5. Click Run. BI Query runs the stored procedure.

6. If the stored procedure requires input parameters, BI Query prompts you for the necessary information.

7. If the stored procedure returns results, the results are displayed in the Results window. Any other output from the stored procedure (such as completion messages, failure messages, or output parameters) is logged and displayed in the Message window.

   If the stored procedure includes multiple SELECT statements, BI Query may return unpredictable results.

Using Freehand Queries in a BI Server Environment

If your Hummingbird BI environment includes BI Server, you can publish, retrieve, and schedule freehand queries. Publishing, retrieving and scheduling freehand queries is done the same way as it is with standard queries. For more information, see any of the following:

- “Publishing Queries” on page 242
- “Retrieving Queries” on page 243
- “Scheduling Queries” on page 249

In BI Web you cannot create freehand queries, but you can open and run freehand queries that have been published. You can also open and refresh standard reports and BI Query reports created from freehand queries, and use buttons in data models in BI Web. As with standard queries, if you use a prompt in a published freehand query, you must also publish the prompts.
Creating Standard Reports

This section provides information on the following:

- “Standard Reports in BI Query” on page 155
- “Basic Report Tasks” on page 158
- “Adding Calculations to Standard Reports” on page 161
- “Manipulating Columns in Standard Reports” on page 166
- “Adding Borders to Standard Reports” on page 171
- “Adding Ornaments to Standard Reports” on page 172

Standard Reports in BI Query

BI Query includes a fully integrated reporting facility called Standard Reports. Using Standard Reports, you can produce fully formatted, professional reports from the results of your queries. Each standard report you create appears in a report window in BI Query, where you can format it in a variety of ways. The results in a standard report correspond to the results in a results window.

A standard report consists of two components:

- a query that generates the results that appear in the report
• a report specification that specifies how the results are formatted and what additional calculations are applied to them

If the BI Query administrator has provided report specifications with the data model, you can use them to create a report from query results. You can also create reports using your own specifications or the default specification.

If you are working in a BI Server environment, you can publish your standard reports to the BI Server Repository and share them with BI Web users (who can access and run the reports over the Web) and other BI Query users (who can access the reports when they retrieve the data model from the Repository).

💡 For more information, see “Publishing Standard Report Specifications” on page 245.

💡 Standard Reports is an older and less sophisticated reporting facility than BI Query Reports. However, you can convert standard reports into BI Query Reports format. For more information, contact Technical Support.

**Standard Reports Structure**

Standard reports are divided into the following sections:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>Contains the title and date and appears at the top of each page of the report. You can enlarge it, change its appearance and content, and add ornaments to it. For more information on using ornaments, see the <em>BI Query Data Models User’s Guide</em>.</td>
</tr>
<tr>
<td>Column headings</td>
<td>Correspond to the headings in the results set. You can change their appearance, content, width, and depth. You can also add ornaments to them.</td>
</tr>
<tr>
<td>Results</td>
<td>Contain the actual results of the query. You can add subtotals, grand totals, and ornaments to them. You can reorder them, replace their values, and change their widths, appearance, and data formats.</td>
</tr>
</tbody>
</table>
When you click in a section, a border appears around it. When a section is too small to accommodate the information in it or when it displays too much empty space, you can resize it to improve the layout and ensure that all the information appears when you print the report.

### Data Formats in Standard Reports

When BI Query returns results from the database, the format in which they appear may not be the way you want them represented in a standard report. You can change the default display format for each attribute in a standard report.

For more information on formatting standard reports, see “Formatting Data: The Basics” on page 67.

If you save a standard report specification with a query and subsequently change the display format for an attribute in the query, the original display format in the corresponding standard report does not change. To apply the new format to the report, change the format for the corresponding attribute in the report window, and then save the report specification with the query.
Basic Report Tasks

This section describes the basic tasks you need to perform to create and save a standard report. Later sections describe how to add calculations to a standard report, how to manipulate its columns, and how to change its appearance.

Creating Standard Reports

When you create a standard report, the report window displays the results of a query in a default format. You can change the appearance of a report in a variety of ways and add information to it. The title of the report corresponds to the query that generated the results.

To create a standard report:

1. Retrieve, open, or generate the results you want to present in a standard report.
2. On the Results menu, point to Show as Report, and then click BI Query Standard.

Report Specifications

Report specifications help you present information consistently and adhere to corporate standards (such as corporate logos and fonts). If you create similar reports on a regular basis, you can use a report specification to avoid formatting the data each time.
If you create a button that links to the query and sends output to a results window, BI Query runs the query whenever you click the button, and then creates the standard report using the associated specification.

When you save a report specification locally, BI Query automatically adds the extension .rpt to the file name and stores the specification file in the Queries folder (unless you specify otherwise). By default, BI Query also saves the corresponding query and stores it in the Queries folder with the same name but with the extension .qry. You can create subfolders in the Queries folder to organize your reports more efficiently.

If the report specification and the query are in the same folder and have the same name, BI Query automatically applies the specification to the results each time you run the query. You can remove the specification associated with a query to avoid generating a report each time you run the query. (With the query open, click Remove Report Specification on the Query menu.)

Specifications that are not stored in the Queries folder are known as external specifications. You can apply external specifications from an accessible location to any query. However, you cannot publish an external specification.

**Saving Report Specifications**

You must have the Save Queries permission to save report specifications.

**To save a report specification:**

1. Create and format the report. Add any required calculations to it.
3. Do one of the following:
   - To share the report specification, click Shared Reports. This option is available only if you are using BI Query Admin.
   - For more information on shared and personal specifications, see “Personal Versus Shared Files” on page 26.
   - To save the specification as a personal specification (not accessible to other users), click My Reports. This option is not available if you are using BI Query Admin.
   - To save the specification as an external specification, click External Reports. You cannot publish external specifications.
4. Navigate to the location where you want to save the specification. You can create a subfolder by clicking the **Create New Folder** button.

5. In the **File Name** box, type the file name for the specification. Do not include the `.rpt` extension (BI Query adds it automatically).

6. If you do not want to save the corresponding query, clear **Save Query Also**.

7. Click **Save**.

You can also publish standard reports to the BI Server Repository. For more information, see “Publishing Standard Report Specifications” on page 245.

**Applying Report Specifications**

After you have saved a report specification, you can apply it to the results of other queries. When you select a specification, BI Query compares the columns in the specification with the new results. If the corresponding tables and attributes match, BI Query formats the columns accordingly. If they do not match, BI Query formats the columns based on their position in the report. (For example, it applies the format of the first column in the report specification to the first column of results.)

To apply a report specification to results:

1. Retrieve, open, or generate the results you want to present in a standard report.

2. On the **Results** menu, point to **Show as Report**, and then do one of the following:
   - To use a specification that is stored on your local machine or network, click **Open Report Specification**. The **Choose Report Specification** dialog box opens.
   - To use a specification stored in the BI Server Repository, click **Retrieve Report Specification**. (You must be in a BI Server environment to use this menu item.) The **Retrieve Report Specification** dialog box opens.

3. In the dialog box that opens, do one of the following:
   - To apply a shared specification, click **Shared Reports**. This option is available only if any of the following is true: the data model is split; you are using BI Query Admin; you are using BI Query User/Update in a BI Server environment.

     For more information on shared and personal specifications, see “Personal Versus Shared Files” on page 26.

   - To apply a personal specification, click **My Reports**. This option is not available if you are using BI Query Admin.
• To apply an external specification, click **External Reports**. This option is not available in the **Retrieve Report Specification** dialog box.

4. Navigate to the specification you want to apply to the current results set and select the specification.


### Printing and Exporting Standard Reports

You can print a standard report to a printer or a file, or export it directly to a text file. Standard reports print exactly as they appear on screen.

💡 When you print a report, you can include a title page that provides the date, time, number of pages, and list of prompt values used in the query. Before printing the report, click **Title Page** on the **Report** menu.

**To print a report:**

1. Create the report or run the query associated with it.
2. On the **File** menu, click **Print**. The **Print** dialog box opens.
3. Do one of the following:
   • To print to a printer, specify the options you want; then, click **OK**.
   • To print to a file, select **Print to File**; then, click **OK**. In the **Print To File** dialog box, specify a name for the file; then, click **OK**.

**To export a report to a text file:**

1. Create the report or run the query associated with it.
2. On the **Report** menu, click **Export Report**.
3. In the **Export Report to File** dialog box, specify a location and name for the file.
4. Click **Save**.

### Adding Calculations to Standard Reports

You can add subtotals and grand totals to numeric columns in standard reports to perform standard calculations (sum, average, and so on) or custom calculations (such as squaring the sum of a column). You can also add calculated columns—similar to calculated attributes—that perform user-defined calculations.
For more information on calculated attributes, see “Creating Calculated Attributes” on page 193.

Unlike calculated columns, totals do not appear in a separate column; instead, they appear on separate subtotal or grand total lines.

Adding Subtotals

Subtotals allow you to analyze data after results are returned. A subtotal line appears below each group of information in the numeric column, displaying a label and default calculation. The label is an ornament; you can change its content and appearance (font, size, and so on). You can also move the label to a different location or delete it. When you add a subtotal, a blank header is added above each group of information. You can add ornaments to the header and resize it.

To add subtotals:
1. Click the column on which to base the subtotal (usually a column that has been sorted).
3. Click the default calculation box in one of the subtotals.
4. From the pop-up menu that opens, do one of the following:
   - To apply a standard calculation, click it.
   - To specify your own calculation, click Calculation; then, create the calculation in the Edit Subtotal Calculation dialog box.
Adding Calculations to Standard Reports

For more information on calculated columns, see “Adding Calculated Columns” on page 163.

5. Click outside the subtotal to run the calculation.

The Count operator counts only the visible values in a column. That is, if you have suppressed duplicates, the count does not include the suppressed values.

To insert a page break after each subtotal in a report:

1. Click in a subtotal line.

Adding Grand Totals

You can add multiple grand totals to a report to help you analyze data after results are returned. For example, you may wish to calculate both the sum and the average for a numeric column.

To add grand totals:

2. For each column, do the following:
   a. Click the default calculation at the bottom of the report.
   b. From the pop-up menu that opens, click a standard calculation to apply to the column. Alternatively, to specify your own calculation, click Calculation, and then create the calculation in the Edit Grand Total Calculation dialog box.

   For more information on calculated columns, see “Adding Calculated Columns” on page 163.
3. Click outside the grand total to run the calculation.

Adding Calculated Columns

You can add a column to a report to perform a specific calculation based on information in the report (such as values in a numeric column, subtotals, and grand totals).

If your report contains a list of countries, order amounts for each country, and a grand total of those orders, you can create a calculated column to determine what percentage of the grand total each order amount represents.
You can also use multi-pass reporting to create a column that performs a calculation on other calculated columns. In this case, BI Query must make two or more passes through the data to arrive at the final calculation.

For example, suppose your report contains a list of products, the cost per unit of each product, the number of units sold, a calculated column that determines the value of each sale (the cost per unit multiplied by the number of units sold), and a grand total of sales. You can add another calculated column to determine what percentage of the grand total each sale represents.

<table>
<thead>
<tr>
<th>Item</th>
<th>Cost/Unit</th>
<th>Units Sold</th>
<th>Total Sale</th>
<th>% Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>bolts</td>
<td>$2</td>
<td>1</td>
<td>$2</td>
<td>8.33</td>
</tr>
<tr>
<td>nails</td>
<td>$3</td>
<td>3</td>
<td>$9</td>
<td>37.50</td>
</tr>
<tr>
<td>nuts</td>
<td>$3</td>
<td>2</td>
<td>$6</td>
<td>25.00</td>
</tr>
<tr>
<td>pins</td>
<td>$1</td>
<td>6</td>
<td>$7</td>
<td>29.17</td>
</tr>
</tbody>
</table>

*Grand Total: $24*

The Total Sale column calculates the value of each sale (cost per unit multiplied by the number of units sold).

The % Total column divides each value in the Total Sale column by the grand total and multiplies by 100 to determine the percentage of the total that each sale represents.

If a calculated column references a column that contains NULL values, use the Report Options dialog box to specify whether you want BI Query to convert the NULL values to zero values for the purposes of the calculation, or maintain them as NULL values.

**To add a calculated column:***

2. In the Name box, type the name of the new column.
3. In the Expression box, type the calculation for the new column. You can also use the Items list box to select an item on which to base the calculation (Columns, Subtotals, Grand Totals, Prompts). From the list below the Items list, select the calculation, attribute, or prompt that you want to add to the calculation. Click Insert.
Your calculation can involve more than one item in the report or corresponding query.

4. Click OK.

If you have applied more than one calculation to a column (such as a subtotal or grand total), BI Query adds a number prefix to the name of each calculation in the Items list.

For example, if you have added a grand total to sum the values in the Total Sale column, and added another grand total to average the amounts, the Items list in the Add Calculated Column dialog box lists the following grand totals:

- <1> Sum of Total Sale
- <2> Avg of Total Sale

Editing and Deleting Calculated Columns

Use the Edit Calculated Column dialog box to edit or delete a calculated column.

To edit a calculated column:

1. In a standard report, click in the calculated column you want to edit.
3. In the Edit Calculated Column dialog box, make the necessary modifications. You can change the name of the column; you can also alter the calculation by selecting different items on which to base the calculation or by changing the numbers and operators.
4. Click OK.

To delete a calculated column:

1. In a standard report, click in the calculated column you want to delete.
3. In the Edit Calculated Column dialog box, click Delete Column.
4. Click OK.
Manipulating Columns in Standard Reports

When you generate a standard report from the results of a query, BI Query displays the results in the order in which they appear in the results window. If you want to reorganize the results or save space, you can change the column order and combine, band, or stack columns.

Resizing Report Columns

If a standard report contains more columns than fit on one page, BI Query spreads the columns over the required number of pages; a gray-shaded line separates each page. To view each column, scroll to the right. To prevent columns dividing over more than one page, resize or reorder them as necessary.

💡 When data appears as a sequence of asterisks, the corresponding column is not wide enough to display the data.

You can also reorder columns. For more information, see “Reordering Columns” on page 129.

To resize columns:

1. Click a column.
2. Do one of the following:
   - Drag the right-hand border of the column to the width you want.
   - On the Report menu, click Column Settings. The Column Settings dialog box opens. From the Columns list, select the column you want to resize; then, modify its character width in the Column Width box. Click OK.
Combining Columns

When related data (such as a person’s name and title) are in different columns and you want to present them together, you can combine the columns. You can combine two or more columns into a single column, and you can merge combined columns with another column. When BI Query combines columns, the new column displays the heading of the first selected column.

To combine columns:

1. Hold down SHIFT, and then click the columns you want to combine.
3. From the Separator list, select the separator you want to appear between the items in each combined column (for example, a space). You can also type a separator in the box.
4. Click OK.
Banding Columns

To create a standard report in which one column occupies the entire width of the page, you can band the column. This feature is useful if a report contains several columns with short pieces of data and one column that contains a lengthy description. You can band the descriptive column so that it appears across the page, either above or below the other columns. Banding columns also lets you create subheadings in a report.

<table>
<thead>
<tr>
<th>Sales Report</th>
<th>Sales Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order#</td>
<td>Amount</td>
</tr>
<tr>
<td>089</td>
<td>$1,200</td>
</tr>
<tr>
<td>269</td>
<td>$5,500</td>
</tr>
<tr>
<td>348</td>
<td>$3,500</td>
</tr>
<tr>
<td>402</td>
<td>$2,800</td>
</tr>
</tbody>
</table>

Original

<table>
<thead>
<tr>
<th>Sales Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order#</td>
</tr>
<tr>
<td>089</td>
</tr>
<tr>
<td>269</td>
</tr>
</tbody>
</table>

Original

<table>
<thead>
<tr>
<th>Sales Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order#</td>
</tr>
<tr>
<td>089</td>
</tr>
<tr>
<td>269</td>
</tr>
</tbody>
</table>

Banded (with Before selected)
To band a column:

1. On the **Report** menu, click **Column Settings**. The **Column Settings** dialog box opens.
2. From the **Columns** list, select the column you want to band.
3. Select **Band Column**; then, select **Before** or **After** to display the column above or below the other columns.
4. Click **OK**.

**Stacking Columns**

Instead of displaying columns horizontally across a page, you can stack them into a single vertical column, with each item appearing on a separate line; the columns that you do not stack appear in their normal position. The stacked column is as wide as the widest column in the stack. Only the heading of the left-most column in the stack is used.

<table>
<thead>
<tr>
<th>Sales Report</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Country</strong></td>
</tr>
<tr>
<td>Canada</td>
</tr>
<tr>
<td>France</td>
</tr>
<tr>
<td>USA</td>
</tr>
<tr>
<td>Australia</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sales Report</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Country</strong></td>
</tr>
<tr>
<td>Canada</td>
</tr>
<tr>
<td>Ottawa</td>
</tr>
<tr>
<td>France</td>
</tr>
<tr>
<td>Paris</td>
</tr>
<tr>
<td>USA</td>
</tr>
<tr>
<td>Chicago</td>
</tr>
<tr>
<td>Australia</td>
</tr>
<tr>
<td>Sydney</td>
</tr>
</tbody>
</table>

A standard report can contain only one stacked column. You can add columns to a stacked column by un-stacking the column, and then re-stacking. You can include both banded and stacked columns in a report. Stacked columns appear to the left of banded columns. The stacked column displays the heading of the first column you select.
To stack columns:
1. SHIFT+click the columns you want to stack.

Formatting Column Text and Headings

Varying the appearance of text in a report provides variety and contrast, improves readability, and helps distinguish between different pieces of information. A unique typeface can make a title or masthead stand out. Bolding and italicizing add further contrast and emphasize important information.

You can format the text in columns and column headings in terms of font, size, style, alignment, effects (strikeout and underline), and color. You can also change the actual text of a column heading (which is initially the same as the column heading in the corresponding results set).

To format text in a column or column heading:
1. Click the column or column heading you want to format.
3. Specify the text format for the selected column or heading.
4. If you are formatting a column heading and want to change the text of the heading, type the new heading in the Text box.
5. Click OK.

Customizing Column Settings

In addition to resizing and banding a column, you can apply the following customizations:
• modify the column name
• change the text wrap within the column
• make the column fit the page
• add a special border to the column

To customize a column:
2. From the Columns list, click the column you want to customize.
3. To change the title for the column, type the new title in the **Column Heading** box.

4. To apply a special border to the left side of the column, click **Special Column**. You can format this special border using the **Report Borders for Standard Reports** dialog box.

   For more information on border formats, see “Adding Borders to Standard Reports” on page 171.

5. To wrap column data that exceeds the width of the column, select **Wrap Column**. This check box is enabled only for columns containing character data.

6. Click **OK**.

---

### Adding Borders to Standard Reports

Using the **Report Borders for Standard Reports** dialog box, you can add horizontal and vertical borders to report components to help you organize and highlight information. For example, you can add vertical lines between columns or horizontal lines below column headings or between records. You can set a default border format for all standard reports; you can also specify borders for a particular report.

You can also create borders manually using the Line tool on the Drawing toolbar. For more information on report borders, see the Help for your BI Query application.

**To add borders to a standard report:**

1. Open the **Report Borders for Standard Reports** dialog box by doing one of the following:
   - On the **Tools** menu, click **Preferences**. In the **Preferences** dialog box, click **Report Borders**.
   - Create the report. On the **Report** menu, click **Borders**.

   If you open the **Report Borders for Standard Reports** dialog box for a particular standard report, the border format you specify applies only to that report. If you open the dialog box from the **Preferences** dialog box, the border format applies to all standard reports that you subsequently create.

2. From the **Report Component** list, select the component you want to format.

3. From the **Style** list, select the line style for the border.
4. To specify a color for the border, click the Color button and select the color from the palette that opens. To specify a custom color, click Custom Color in the palette.

5. Repeat steps 2–4 for all other report components that you want to format.

6. To add a border below any primary and secondary subtotals, select the Primary Subtotal and Secondary Subtotal check boxes.

7. Click OK.

Adding Ornaments to Standard Reports

Some default report elements are text ornaments and appear in specific sections of a report by default. The title and date are text ornaments that appear in the header of a report. The title is based on the name of the corresponding results window. The date is the same as the date on your computer. The page number is a text ornament that appears in the footer of a report.

You can change the appearance and content of each default ornament, move it to a different position, or remove it altogether. For example, you can change the default title of your report to reflect the information it contains, and you can change the date to display a different date format.

You can also add text and graphical ornaments to a standard report. You create these ornaments in much the same way as you create ornaments in a Design window in BI Query.

💡 For more information on ornaments, see the BI Query Data Models User’s Guide.

Special Text Strings in Standard Reports

The content of a text ornament in a standard report can contain any of the special text strings that BI Query supports. For example, the default date ornament contains the special string &d, which displays the current date.

BI Query supports the following lowercase strings in standard reports:

<table>
<thead>
<tr>
<th>String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;t</td>
<td>Displays the current time.</td>
</tr>
</tbody>
</table>
Adding Ornaments to Standard Reports

<table>
<thead>
<tr>
<th>String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;d</td>
<td>Displays the current date.</td>
</tr>
<tr>
<td>&amp;dn</td>
<td>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, &amp;d0 displays the current date using the first predefined date format, &amp;d1 displays the date using the second predefined format, and so on. &amp;d is equivalent to &amp;d0. $n$ corresponds to a named predefined format only. The &lt;Regional Settings&gt;, &lt;Custom&gt;, and &lt;Blank&gt; formats do not apply in this case.</td>
</tr>
<tr>
<td>&amp;p</td>
<td>Displays the current page number. The default page number ornament contains the text Page &amp;p.</td>
</tr>
<tr>
<td>&amp;pp</td>
<td>Displays the total number of pages. To display a page number in the form “Page 1 of 3”, type Page &amp;p of &amp;pp in the text ornament.</td>
</tr>
<tr>
<td>&amp;q</td>
<td>Displays the name of the query associated with the report. If the query has not been saved, the ornament displays the word “Untitled”.</td>
</tr>
<tr>
<td>&amp;«prompt_name»</td>
<td>Displays the values of the prompt called prompt_name. For example, &amp;«Country» displays the values entered for the Country prompt. This string is useful if you need to include an ornament that identifies the information on which the report is based (for example, Shipments to: Singapore, where Singapore is the value supplied for the prompt). Prompt values also appear on the report title page.</td>
</tr>
</tbody>
</table>

You can change the content and appearance of these strings as you would any other text in an ornament.
To create a text ornament or button that contains a literal version of one of the above strings, precede the string with an extra ampersand (for example, “peas&&queues” displays as “peas&queues”).

Adding Text Ornaments to Standard Reports

When you add an ornament to a page in a multi-page report, it appears on all the pages above and below that page. If you add an ornament to a particular section of a report, such as a subtotal or the blank header above a subtotal, it is reproduced in each corresponding section.

For more information on report sections, see “Standard Reports Structure” on page 156.

To add a text ornament to a standard report:

1. In the report window, right-click where you want the ornament to appear.
2. On the pop-up menu that opens, click Create Ornament.
3. In the Create Ornament dialog box, click Edit Text. The Text Style dialog box opens.
4. In the Text box, type the text for the ornament. You can include any of the special text strings supported in Standard Reports. For more information, see “Special Text Strings in Standard Reports” on page 172.
5. To add prompt values, click Prompts. In the Prompts dialog box, double-click the name of the prompt used in the query.
   The prompt appears in the Text box preceded by an ampersand (&) and delimited by chevron characters (» and «).
6. Specify the format for the ornament; then, click OK.
7. In the Create Ornament dialog box, click OK.

Editing Default Ornaments in Standard Reports

Use the Edit Ornament dialog box to edit the default title, date, and page number ornaments.

To edit a default ornament in a standard report:

1. In the report window, double-click on the ornament.
2. In the Edit Ornament dialog box, click Edit Text. The Text Style dialog box opens.
3. In the **Text** box, edit the text for the ornament as necessary. You can include any of the special text strings supported in Standard Reports.

4. Specify the format for the ornament; then, click **OK**.

5. In the **Edit Ornament** dialog box, click **OK**.
Chapter 11

Creating Labels

This section provides information on the following:

- “BI Query Labels” on page 177
- “The Labels Window” on page 178
- “Creating and Editing Labels” on page 180
- “Formatting Label Fields” on page 182

BI Query Labels

BI Query lets you create labels from results sets. For example, you can run a query to retrieve results for a group of customers and use the information to produce a variety of labels, including address and shipping labels, as well as labels for files, diskettes, name tags, and index cards. Each row in the results set forms the content of one label.

To format labels, you can choose from a range of standard Avery label formats available from Avery Dennison Corporation. You can also create your own customized labels and save their specifications for use with other results sets.
The Labels Window

The **Labels** window, which uses Avery (5167, 5267) Return Address as the default label format, displays each row in the results set as a separate label.

View and Layout Mode

There are two modes associated with the **Labels** window:

**View Mode**

Displays all labels generated from the results set. Use this mode to view all labels and to set their dimensions. View mode is the default.

**Layout Mode**

Displays the current layout of label fields in the **Labels** window. Use this mode to edit the appearance of label fields such as font type and justification. Any formatting changes you make to these fields are automatically applied to all labels in the **Labels** window.

Regardless of the mode you select, a toolbar appears in the **Labels** window that lets you edit label dimensions or label properties.

The following toolbar buttons are common to View and Layout mode:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Layout" /></td>
<td><strong>Layout</strong>—Displays a label in <strong>Layout</strong> mode, which lets you format label properties (font, size, and style) and move label fields.</td>
</tr>
<tr>
<td><img src="image" alt="Dimensions" /></td>
<td><strong>Dimensions</strong>—Displays the <strong>Label Dimensions</strong> dialog box, which lets you choose from a range of Avery label formats. You can also customize your own labels.</td>
</tr>
</tbody>
</table>

The following toolbar buttons appear only in View mode:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="To Start" /></td>
<td><strong>To Start</strong>—Displays the first page of labels. Alternatively, select <strong>To Start</strong> from the <strong>Report</strong> menu.</td>
</tr>
</tbody>
</table>
The Labels Window

The following toolbar buttons appear only in **Layout** mode:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="bold.png" alt="Bold" /></td>
<td><strong>Bold</strong>—Applies boldface to the selected fields.</td>
</tr>
<tr>
<td><img src="italics.png" alt="Italics" /></td>
<td><strong>Italics</strong>—Italicizes the selected fields.</td>
</tr>
<tr>
<td><img src="underline.png" alt="Underline" /></td>
<td><strong>Underline</strong>—Underlines the selected fields.</td>
</tr>
<tr>
<td><img src="left-justification.png" alt="Left Justification" /></td>
<td><strong>Left Justification</strong>—Justifies the selected text at the left border of a field. Text is left-justified by default.</td>
</tr>
<tr>
<td><img src="center-justification.png" alt="Center Justification" /></td>
<td><strong>Center Justification</strong>—Centers the selected text within a field.</td>
</tr>
<tr>
<td><img src="right-justification.png" alt="Right Justification" /></td>
<td><strong>Right Justification</strong>—Justifies the selected text at the right border of a field.</td>
</tr>
<tr>
<td><img src="left-alignment.png" alt="Left Alignment" /></td>
<td><strong>Left Alignment</strong>—Aligns the field (or the last field selected) with the left border of the label.</td>
</tr>
<tr>
<td><img src="right-alignment.png" alt="Right Alignment" /></td>
<td><strong>Right Alignment</strong>—Aligns the field (or the last field selected) with the right border of the label.</td>
</tr>
<tr>
<td><img src="top-alignment.png" alt="Top Alignment" /></td>
<td><strong>Top Alignment</strong>—Aligns the field (or the last field selected) with the top border of the label.</td>
</tr>
<tr>
<td><img src="bottom-alignment.png" alt="Bottom Alignment" /></td>
<td><strong>Bottom Alignment</strong>—Aligns the field (or the last field selected) with the bottom border of the label.</td>
</tr>
</tbody>
</table>
Chapter 11: Creating Labels

Creating and Editing Labels

Use the Labels window to create labels and the Label Dimensions dialog box to create or edit existing label formats and dimensions.

💡 For more information about formatting fields within a label, see “Formatting Label Fields” on page 182.

To create labels from query results:

1. Run the query.
2. With the results set active, point to Show As Report on the Results menu and then click Labels. The Labels window opens. The query results are automatically displayed as individual labels.

To create or edit label dimensions:

1. Do one of the following:
   - In the Labels window, click the Dimensions button.
     The Label Dimensions dialog box opens, displaying the current label specifications.

2. To create a new label format, select Custom Label from the Format drop-down list. Custom Label in the Format drop-down list is automatically selected whenever you edit the label dimensions or Rows Per Page or Columns Per Page options.

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td>Horizontal Centering—Centers the selected field horizontally within a label.</td>
</tr>
<tr>
<td><img src="image2.png" alt="Image" /></td>
<td>Vertical Centering—Centers the selected field vertically within a label.</td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td>Horizontal Spacing—Evenly spaces the selected fields horizontally within a label.</td>
</tr>
<tr>
<td><img src="image4.png" alt="Image" /></td>
<td>Vertical Spacing—Evenly spaces the selected fields vertically within a label.</td>
</tr>
</tbody>
</table>
3. Edit label formats by doing any of the following:

- To specify label dimensions, type the desired label dimensions in the dimension text boxes and select the appropriate dimension units.

- To specify page offsets, type the desired values in the **Top** and **Left** text boxes to specify where you want labels to print on a page. This may not be necessary because printers typically print with a small margin of error and do not center output perfectly on the page. To offset printer output down and to the right, specify positive values. To offset printer output up and to the left, specify negative values.

- To specify label formats, select the desired Avery label format from the **Format** list. The number of rows and columns per page are automatically adjusted.

  - **Tip:** Certain label formats extend beyond the printable area on a page. Non-printable areas are identified with a gray border.

- To specify rows and columns per page, type the desired number of rows and columns you want displayed per page in the **Rows Per Page** and **Columns Per Page** text boxes.

- To specify rounded corners, select **Rounded Corners** if the label sheet you are using has labels with rounded corners.
4. Click **OK**. The **Label** window displays the labels using the new format.

   Borders serve as a visual aid when you adjust label formats and layouts. To display borders on printed labels, click **Print Label Borders** on the **Report** menu.

**Formatting Label Fields**

You can modify Avery or custom labels by changing the placement and appearance of label fields. Each field in a label corresponds to a single results cell, or attribute, in the results window.

   For more information about creating and editing labels, see “Creating and Editing Labels” on page 180.

**To edit the appearance of label fields:**

1. Do one of the following:
   - In the **Labels** window, click the **Layout** button.
   - In the **Labels** window, double-click any label.
   - On the **Report** menu, click **Edit Layout**.

   A label displays in Layout mode.

2. Select the field(s) that you want to edit. To select more than one field, press the **SHIFT** key while selecting subsequent fields. A border appears around the selected field(s).

3. Use the toolbar to make the desired changes to the appearance and placement of label fields. You can do any of the following:
   - specify label field font type and size
   - resize label fields
   - position label fields
   - justify text within a field
   - align a field within a label
   - align two or more fields within a label
   - center label fields
   - space label fields with respect to each other

   See the following sections for more information.
4. Click the **Layout** button to exit Layout mode. Alternatively, double-click the white area outside the label. The **Label** window displays the labels using the new label format.

**Specifying Label Field Font Type and Size**

To specify label field font type and size:

- Use the toolbar to specify the font type and size of field labels. To resize a field in a label to accommodate changes in font type and size, click **Fit Field to Font** on the **Report** menu.

For example, if you change the Address field from 10 point Arial to 14 point Helvetica bold, click **Fit Field to Font** to resize the field to accommodate the new font type and size.

**Resizing Label Fields**

To resize label fields:

1. Select the label field you want to resize and position the crosshair cursor over one of the four square handles until a double arrow appears.
2. Drag the handle to adjust the size of the label field.

**Positioning Label Fields**

To position label fields:

1. Select the label field you want to move and position the crosshair cursor over the field.
2. Drag the label field to the desired location. Alternatively, use the keyboard cursor keys to position the field one pixel at a time.

💡 Because you can move fields outside the border of a label, be sure to keep the text within the printable area.

**Justifying Text Within a Field**

To justify text within a field:

- Select the desired label field and click the appropriate buttons on the toolbar. The text in the field automatically justifies to the corresponding field border.
Aligning Fields Within a Label

To align fields within a label:

- Select the label field and click the appropriate button on the toolbar. The field automatically aligns with the corresponding border of the label.

Aligning Two or More Fields Within a Label

To align two or more fields within a label:

1. Select the fields you want to align. The last field selected is the field with which other fields are aligned.
2. Click the appropriate button on the toolbar. The fields automatically align with the last field you selected at the corresponding border.

Centering Label Fields

To center label fields:

- Select the fields you want to center and click the appropriate button on the toolbar. The fields automatically center.

💡 Use the spacing and centering tools to provide maximum flexibility for positioning text in labels.

Spacing Label Fields with Respect to Each Other

To space label fields with respect to each other:

- Select the fields you want to space and click the appropriate button on the toolbar. The fields automatically space out evenly with respect to each other.

💡 To enable the spacing tools, you must select at least two fields. To use the spacing feature to best effect, select a minimum of three fields.
Aggregate Functions and Grouping

You can apply an aggregate function to an attribute to perform a basic calculation such as summing or averaging results. You can also apply a user-defined calculation to an attribute to carry out more sophisticated calculations.
Aggregate functions are mathematical operations that let you calculate summary values from a set of values for an attribute. For example, you can retrieve retailers’ total sales by applying the SUM function to an Invoice Amount attribute. Different functions are available depending on the attribute, its data type, and the DBMS you are using.

**Automatic Grouping**

When you apply an aggregate function to an attribute, BI Query automatically groups all the other attributes in the query in the order in which you selected them. (If this order is not appropriate for your needs, you can change it.) Grouping organizes the data into sets and retrieves a summary value for each set. All selected attributes to which functions have not been applied are grouped to ensure that the results set does not calculate a summary value for individual members of the group.

**Example**

A database contains the following data about retailer sales:

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>Invoice Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sterling</td>
<td>Ottawa</td>
<td>$3000</td>
</tr>
<tr>
<td>Triteck</td>
<td>Toronto</td>
<td>$1000</td>
</tr>
<tr>
<td>Billings</td>
<td>Toronto</td>
<td>$3000</td>
</tr>
<tr>
<td>Sterling</td>
<td>Ottawa</td>
<td>$1000</td>
</tr>
<tr>
<td>Billings</td>
<td>Toronto</td>
<td>$4000</td>
</tr>
<tr>
<td>Saddler</td>
<td>Toronto</td>
<td>$5000</td>
</tr>
<tr>
<td>Triteck</td>
<td>Ottawa</td>
<td>$2000</td>
</tr>
<tr>
<td>Triteck</td>
<td>Toronto</td>
<td>$3000</td>
</tr>
</tbody>
</table>

If you want to retrieve a list of retailers, cities, and total retailer sales, you could form a query in which you select the Name, City, and Invoice Amount attributes (in that order) and apply the SUM function to Invoice Amount.
When you run the query, the DBMS organizes the data into groups: first by Name; then, by City. Each row of results in a group has identical values except the attribute to which the function is applied.

The table below shows how the DBMS groups the data. The rows for “Billings, Toronto” represent one group, the row for “Saddler, Toronto” represents another group, and so on.

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>Invoice Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Billings</td>
<td>Toronto</td>
<td>$3000</td>
</tr>
<tr>
<td>Billings</td>
<td>Toronto</td>
<td>$4000</td>
</tr>
<tr>
<td>Saddler</td>
<td>Toronto</td>
<td>$5000</td>
</tr>
<tr>
<td>Sterling</td>
<td>Ottawa</td>
<td>$1000</td>
</tr>
<tr>
<td>Sterling</td>
<td>Ottawa</td>
<td>$3000</td>
</tr>
<tr>
<td>Triteck</td>
<td>Ottawa</td>
<td>$2000</td>
</tr>
<tr>
<td>Triteck</td>
<td>Toronto</td>
<td>$1000</td>
</tr>
<tr>
<td>Triteck</td>
<td>Toronto</td>
<td>$3000</td>
</tr>
</tbody>
</table>

The DBMS then applies the SUM function to each group. The DBMS returns one row for each group. Most DBMSs sort each group before returning the results. In this case, groups are sorted first alphabetically by name, and then alphabetically by city.

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>Invoice Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Billings</td>
<td>Toronto</td>
<td>$7000</td>
</tr>
<tr>
<td>Saddler</td>
<td>Toronto</td>
<td>$5000</td>
</tr>
<tr>
<td>Sterling</td>
<td>Ottawa</td>
<td>$4000</td>
</tr>
<tr>
<td>Triteck</td>
<td>Ottawa</td>
<td>$2000</td>
</tr>
<tr>
<td>Triteck</td>
<td>Toronto</td>
<td>$4000</td>
</tr>
</tbody>
</table>
Chapter 12: Advanced Querying Techniques

**Built-In Aggregate Functions**

BI Query provides the following built-in functions:

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVERAGE</td>
<td>The average of all the values of an attribute or of each group of the attribute’s values.</td>
</tr>
<tr>
<td>MAXIMUM</td>
<td>The maximum value of an attribute or of each group. The maximum is the last alphabetical item, the most recent date, or the highest number.</td>
</tr>
<tr>
<td>MINIMUM</td>
<td>The minimum value of an attribute or of each group. The minimum is the first alphabetical item, the earliest date, or the lowest number.</td>
</tr>
<tr>
<td>SUM</td>
<td>The sum of all the values of an attribute or of each group.</td>
</tr>
<tr>
<td>COUNT</td>
<td>The number of values that are not NULL (missing) values of an attribute or group.</td>
</tr>
<tr>
<td>AVERAGE DISTINCT</td>
<td>The average of all the values of an attribute or of each group, ignoring any duplicates.</td>
</tr>
<tr>
<td>SUM DISTINCT</td>
<td>The sum of all the values of an attribute or of each group, ignoring any duplicates.</td>
</tr>
<tr>
<td>COUNT DISTINCT</td>
<td>The number of values that are not NULL (missing) values of an attribute or group, ignoring any duplicates.</td>
</tr>
</tbody>
</table>

**Types of Grouping**

Grouping can be done either before or after the calculation, depending on the results you want.

**Grouping Before Calculating**

When you group results before applying a function, BI Query eliminates the records that do not meet the specified qualification, groups the results, and then performs the calculation.
Aggregate Functions and Grouping

**Grouping After Calculating**

When you group results after applying a function to an attribute, you qualify the attribute with a group qualification. The qualification is applied to the entire query, after all aggregate functions have been calculated (including any calculated attributes that are treated as aggregates) and the appropriate attributes have been grouped. BI Query lets you apply a group qualification to query results obtained using aggregate functions such as SUM, AVERAGE, and COUNT.

When you group results before calculating, the SQL for the query contains a *WHERE* clause. When you group results after calculating, the SQL for the query contains a *HAVING* clause.

💡 For more information on calculated attributes, see “Creating Calculated Attributes” on page 193.

**Examples**

**Grouping Before Calculating**

In order to retrieve retailers’ largest sales, you might include the Retailer Name and Invoice Amount attributes in a query, qualify Invoice Amount to be greater than $50,000, and then apply the SUM function to it. In this case, BI Query finds all sales that are greater than $50,000, groups them by Retailer Name, and then sums them.

**Grouping After Calculating**

If you wanted a query to return only those salespersons whose total sales exceeded their sales quota, you could apply a group qualification to produce that result. BI Query would first sum the sales for each salesperson and then filter for those sums that exceed the quota. You could not achieve this result if you restricted the query to a value of “greater than the sales quota” and then applied the SUM function. In this case, the query would restrict results to only those individual sales that by themselves exceeded the quota (if any); then, BI Query would apply the SUM function to these results.
Applying Functions to Attributes

When you apply an aggregate function to an attribute, BI Query first groups all the other attributes in the query in the order in which you selected them. You can also specify your own group order by clicking the appropriate Group boxes in the attribute window or Query window.

To apply a function to an attribute:

1. In an attribute window or Query window (in Query View mode), click the Function box for the attribute; then, select the function from the pop-up menu that opens.
2. To group the data in a different order than the order in which you selected the attributes, click the Group box for the remaining attributes. (The function is applied after the results are grouped.)

Creating Group Qualifications

If you need to apply a function before the results are grouped, you can create a group qualification.

To create a group qualification:

1. Construct the query; apply the aggregate function to the attribute that you want to use as the basis for the calculation.
2. On the Query menu, point to Qualification; then, click Group. The Group Qualification dialog box opens.
3. From the Aggregated Attributes list, select the attribute and aggregate function to which you want to apply the group qualification; then, click Insert. The attribute and function appear in the Expression box.

For example, if you have applied the AVERAGE function to the order amount attribute and inserted this combination, the Expression box reads as follows:

\[
\text{AVG(Sales_S.order\_amount)}
\]

4. Type the qualification in the Expression box.

For example, to find only averages greater than $2000, you would add “> 2000” so that the complete entry in the Expression box reads as follows:

\[
\text{AVG(Sales_S.order\_amount)} > 2000
\]

If you had grouped before calculating, only amounts greater than $2000 would have been averaged.

You can also use the available buttons to build the expression:
• To include a prompt in the qualification, click Prompts; then, use the Prompts dialog box to select and insert it.

• To include a variable, click Variables; then, use the Variables dialog box to select and insert it.

• To include another attribute in the qualification, click Attributes. In the Insert Attribute Name dialog box, select the data object and attribute name; then, click Insert.

• To include a built-in or DBMS-specific function, click Functions; then, use the Functions dialog box to select and insert the function.

5. Click OK.

6. Run the query. BI Query retains the group qualification with the current query unless you remove the expression from the Group Qualification dialog box or click New on the Query menu. If necessary, you can temporarily suspend a group qualification from a query without removing it from the query. For more information, see “Suspending Group Qualifications” on page 192.

When you apply group qualifications to restrict the number and types of groups returned, you can specify a group order as well.

Identifying Queries with Group Qualifications

You can easily determine whether or not the current query has a group qualification by looking at either the Apply Group Qualification entry on the Query menu (under Qualification), or the Group Qualification button on the Query toolbar:

• If the menu entry is not enabled, or if the button does not display a check mark, the query does not have a group qualification.

• If the menu entry or button are displayed with a check mark, the query has an active group qualifier.

• If the menu entry is enabled and displayed without a check mark, the query has a suspended group qualifier.
Chapter 12: Advanced Querying Techniques

Suspending Group Qualifications

There may be times when you have applied a group qualification to a query, but want to temporarily suspend the group qualification. For example, you might want to test that the query is returning the correct records. Suspending a group qualification does not permanently remove it from the query; when you save a query that has a suspended group qualification, the group qualification will automatically become active again when you reopen the query.

To suspend a group qualification:

1. Open the query containing the group qualification you want to suspend.
2. On the Query menu, point to Qualification; then, click Apply Group Qualification. BI Query removes the check mark from both the Apply Group Qualification menu entry and the Group Qualification button on the Query toolbar.

To activate a suspended group qualification:

- On the Query menu, point to Qualification; then, click Apply Group Qualification. BI Query displays the check mark on both the Apply Group Qualification menu entry and the Group Qualification button on the Query toolbar.

Grouping Without Functions

To eliminate duplicates, you can group attributes without applying functions. For instance, if you select the Country attribute for a query, the query retrieves all names of countries in the database table, listing each in the order found and as many times as it is found. If you group by Country, your results set contains the name only once for each group. For example, no matter how many times Canada appeared as a country, the name “Canada” appears only once in the list, as the name for the whole group. You can use this method to produce results sets for data values results files.

Grouping attributes in this way is similar to applying the Distinct modifier to eliminate duplicates. Some databases perform better if you group attributes rather than apply the Distinct modifier to the query.

💡 For more information on the Distinct modifier, see “Applying Query Modifiers” on page 48.
When you group attributes, most DBMSs also sort the results (alphabetically, numerically, and chronologically). However, grouping attributes is not the same as sorting them, because the sort process does not remove duplicate rows. For more information on sorting, see “Sorting Rows” on page 38.

To group without applying a function to an attribute:
1. In an attribute window, select the attributes you want.
2. To group the data, click the Group box for the remaining attributes in the attribute window or the Query window (in Query View mode).

Creating Calculated Attributes

BI Query lets you add calculated attributes to a data object. A calculated attribute performs a user-defined calculation in a query. The calculation can be based on more than one attribute from more than one data object in a Design window. You can also apply a function to an attribute in the calculation.

For example, a data object may include Unit Price and Unit Cost attributes that retrieve a product’s market price and production cost, respectively. By creating a new attribute that calculates Unit Price minus Unit Cost, you can determine the profit made on each product.

If you are using BI Query User or Update, you can edit or delete any calculated attribute in a combined data model. However, in a split data model, you can edit or delete only calculated attributes that you have created, not those provided by the administrator.

For more information on editing and deleting calculated attributes, see the Help for your BI Query application.

Creating Calculated Attributes

You can create a calculated attribute that performs a calculation you define. The calculation can be based on more than one attribute from more than one data object in a data model.

When you create a calculated attribute, you can either set a specific data type for the attribute or let BI Query automatically set the data type based on the calculated values.
If you select an automatic data type for a calculated attribute, BI Query cannot run the default `SELECT DISTINCT` query for the attribute. In this case, if you want to qualify the attribute, you cannot select the qualification value(s) from a list of data values retrieved from the database.

To create a calculated attribute:

1. Ensure that BI Query is not in Design mode.
2. In an attribute window or the Query window (in Query View mode), click the Function box for the attribute below which you want to add the calculated attribute.
3. On the pop-up menu that opens, click Calculation. The Edit Calculated Attribute dialog box opens.
4. In the Name dialog box, type the name you want to give to the calculated attribute.
5. From the Data Type list, select the data type for the attribute.
   
   For more information on available data types for calculated attributes, see the Help for your BI Query application.

If you want the type to automatically match the data type of the calculation, select either Automatic Numeric or Automatic Character, depending on the calculation. You can also select a specific data type for the attribute.

If you are connecting to a Teradata DBMS and you are not using ODBC, you must select a specific data type for the calculated attribute. Do not use the defaults Automatic Numeric or Automatic Character.

6. In the Expression box, type the calculation. Add a space between each item in the calculation. You can also build the calculation using the available buttons, as follows:
   
   • To enter the database name of an attribute, select a data object from the Attributes list, then select an attribute from the list below the Attributes list, then click Insert.
   
   • To insert a prompt into the expression, click Prompts. In the Prompts dialog box, select the prompt you want to insert from the Prompts list; then, click Insert.
The following restrictions apply to prompts in calculated attributes:

- they must be mandatory.
- they must have an explicitly set data type.
- they can only contain one value (they must be linked to either a single-value data value query or data values result set).
- if the calculated attribute is in a query that may be used in BI Web, the prompt must accept a single value only; it can not be linked to a data values query.

Furthermore, if the same prompt is used elsewhere and contains multiple values, or if someone changes the prompt’s data type, the calculated attribute may fail.

- To insert a variable into the expression, click **Variables**. In the **Variables** dialog box, select the variable you want to insert from the **Variables** list; then, click **Insert**.

- To insert a function into the expression, click **Functions**. In the **Functions** dialog box, select the function you want to insert from the **Functions** list or **Other Functions** list; then, click **Insert**.

If you select a built-in function from the **Functions** list, it automatically applies to the selected attribute; the function and the database name of the attribute appear in the **Expression** box and the **Aggregate** option is selected. (This ensures that when the calculated attribute is used in a query, the other attributes in the query are grouped.)

For more information on using functions in calculated attributes, see “Including Functions in Query Expressions” on page 197.

7. To ensure that the other attributes in the query are grouped when the calculated attribute is used in a query, select **Aggregate**.

   For more information on grouping, see “Types of Grouping” on page 188.

8. To save the attribute for future sessions, select **Save in Data Model**. This option is unavailable if you are running BI Query Update or BI Query User and you do not have the **Save calculated attribute in model** permission.

9. Click **OK**. The calculated attribute is added below the attribute you selected in step 2.
Data Types for New Attributes

When you create a new attribute for a data object, you must assign a data type to it. The following table describes the possible data types for attributes:

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>Stores Boolean (True/False, yes/no) data.</td>
</tr>
<tr>
<td>Char($n$)</td>
<td>On most databases, stores $n$ bytes of character data, including letters, numbers, and symbols.</td>
</tr>
<tr>
<td>Date</td>
<td>Stores a calendar date.</td>
</tr>
<tr>
<td>Decimal($p$, $s$)</td>
<td>Stores numbers with $p$ significant digits and $s$ places after the decimal point. For example, Decimal(6, 2) stores numbers up to +/- 9999.99.</td>
</tr>
<tr>
<td>Double</td>
<td>Stores a 64-bit approximation of real numbers. The number can be zero or can range from -1.79769E+308 to -2.225E-307, or from 2.225E-307 to 1.79769E+308.</td>
</tr>
<tr>
<td>Float</td>
<td>Stores double-precision floating point numbers with up to 16 significant digits.</td>
</tr>
<tr>
<td>Integer</td>
<td>Stores whole numbers that range from -2,147,483,647 to 2,147,483,647.</td>
</tr>
<tr>
<td>Long Varchar</td>
<td>Stores a varying-length string of up to 32,000 or 2E+30 bytes (depending on the DBMS). If you are using an Informix DBMS, BI Query supports Long Varchar strings of up to 64K.</td>
</tr>
<tr>
<td>Numeric($p$, $s$)</td>
<td>Stores numbers with $p$ significant digits and $s$ places after the decimal point. Numeric is the same as Decimal.</td>
</tr>
</tbody>
</table>
Including Functions in Query Expressions

Use the **Functions** dialog box to include a function in the following types of expression:

- the expression used in a group qualification
- the calculation for a calculated attribute
- the expression that defines a variable

The **Functions** dialog box displays a list of built-in aggregate functions in the **Functions** list and, for certain DBMSs, a list of **scalar functions** in the **Other Functions** list. (A scalar function takes a predefined number of parameters—usually one or two—and produces a value for each result in turn. Aggregate functions are often used with a **GROUP BY** clause; scalar functions are not.)

You can use most scalar functions available in your DBMS even if they do not appear in the **Other Functions** list. Manually type the function as part of the expression. (Check your DBMS documentation for the correct format.)

When you run the query, BI Query evaluates the function and inserts its return value into the expression.
Editing or Deleting Calculated Attributes

In a split data model, users can edit or delete only the calculated attributes they create, not those provided by the BI Query administrator.

To edit a calculated attribute:

1. In the attribute window or Query window (in Query View mode), click the Function box for the calculated attribute.
2. On the pop-up menu that opens, click Edit Calculation.
3. In the Edit Calculated Attribute dialog box, make the necessary changes to the attribute.
4. Click OK.

To delete a calculated attribute:

1. In the attribute window or Query window (in Query View mode), click the Function box for the calculated attribute.
2. On the pop-up menu that opens, click Delete Calculation.

Qualifying with Subqueries

You can dynamically qualify an attribute in a query with another query—known as a subquery. A subquery must return a single column of data; in other words, a subquery must be based on a single attribute. A subquery does not have to qualify the attribute on which it is based, but the values it returns must have the same data type as the qualified attribute.

Example

To retrieve information about retailers with the highest credit limit, you could build and save a query that includes the Receivable attribute with the MAXIMUM function applied to it.

When you build other queries, you can insert that query as a subquery into a qualification to obtain results based on the maximum receivable amount. When you run such a query, BI Query runs the subquery first and then uses the results to qualify the query. In this way, the query is always qualified with the current maximum value in the database.
Guidelines for Using Subqueries

Keep in mind the following points when you use subqueries:

- When you qualify an attribute with a subquery, you cannot include any other qualifiers on that attribute (such as other subqueries, data values results file, data sets, and prompts).
- If you use a subquery to qualify an attribute that is already qualified with other data values, those values are removed from the qualification.
- You cannot delete a subquery from a qualification without removing the entire qualification.
- Subqueries cannot supply sorted data values.

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

Qualifying with Subqueries

To qualify a query with a subquery, you need to create the subquery first and then insert it into the qualification tree.

To qualify an attribute with a subquery:

1. In an attribute window or the Query window (in Query View mode), click the Qualify box for the attribute you want to qualify; then, click the Data Values button in the qualification tree.
2. On the menu that opens, click Subqueries.
3. In the Insert Subquery dialog box, do one of the following:
   - To insert a shared query, click Shared Queries. This option is available only if any of the following is true: the data model is split; you are using BI Query Admin.
   🌟 For more information on shared and personal queries, see “Personal Versus Shared Files” on page 26.
   - To insert a personal query, click My Queries. This option is not available if you are using BI Query Admin.
   - To insert an external query, click External Queries.
4. Select the query you want to open; then, click Insert.
5. From the Operator list in the qualification tree, select an operator. (If the subquery is returning a single value, select =. If it is returning a list of values, select IN or NOT IN.)

The Begins With, Contains, Ends With, Does Not Begin With, Does Not Contain, and Does Not End With operators are invalid for subqueries.

6. Run the query or save it for later submission.

Previewing Subqueries

When you include a subquery in a query qualification, you can preview its SQL before running the query.

To preview a subquery:

1. In the qualification tree, click the List button for the attribute qualified by the subquery.

2. From the pop-up menu that opens, click View Subquery. The View Subquery dialog box opens, displaying the SQL statement that defines the subquery.

Qualifying with Data Sets

A data set is a text file that stores one or more data values. You can use data sets to qualify attributes in a query. Unlike data values results files, which you usually produce using BI Query, data set files can come from any source and can contain any values. For example, you can use an application like Microsoft Excel to create a file containing a series of comma-separated values.

When you include a data set in a query, all of the values stored in the data set qualify the query. Data sets let you include large blocks of values in a query without entering them one by one. They also let you edit those values using a text editor or spreadsheet program.

Like data values results files, data sets have an associated .qrr file. BI Query creates the .qrr file when you first include the data set in a query. You can use comma-separated files (.csv) and data values results files (.qrd) as data sets.

💡 For more information on .qrr files, see “Naming and Storing Data Values Files” on page 80.
Data Set Format

Data set files have a specific format that you must follow if you create them in a text editor.

Distinct values must be delimited throughout the file by one of the following delimiters:

- a tab (ASCII 9)
- a comma (ASCII 44)
- a newline (ASCII 13 + 10)

Use only one type of delimiter throughout a given file. Make sure that the final value in the file is followed by a delimiter character.

Use the standard HTML/XML syntax for comments, as follows:

```html
<!-- comment_text -->
```

where `<!--` and `-->` are literal strings as written and `comment_text` is the actual text of the comment. Data set comments appear on a line by themselves; a given comment cannot extend beyond one line.

Remember to store the data set file in the `DataVals` folder for the data model. When you publish data values files to the BI Server Repository, the data set files are also published. For more information on publishing data values files and data sets, see the `BI Query Data Models User’s Guide`.

Example Data Set File

The following example of a data set file includes two comments and uses carriage returns to delimit data values:

```html
<!-- North American sales centers -->
Albany
Peterborough
Winnipeg
<!-- European sales centers -->
Bristol
Paris
Verona
```

Qualifying with Data Sets

When you qualify an attribute with a data set, BI Query applies all of the values in the data set to the qualification. You can insert more than one data set in the qualification for a given attribute; in this case, all of the values in all of the files qualify the attribute. You can also combine data sets with other qualifiers, such as variables, prompts, and data values.
To qualify an attribute with a data set:

1. Click the Qualify box for the attribute.
2. In the qualification tree, click the Data Values button beside the attribute. On the menu that opens, click Data Sets.
3. In the Open Data Set dialog box, select the data set file that you want to apply to the attribute. Click Open. The qualification tree displays the name of the data set file in the form \{Data set: dsname\}, where dsname is the name of the file.

When you run the query, BI Query inserts the contents of the data set file into the qualification.

Viewing Data Sets

Once you have inserted a data set into a query, you can view the values it contains and the number of values in the set.

To view the contents of a data set:

1. In the qualification tree, click the List button beside the attribute that is qualified by the data set.
2. On the menu that opens, click View Data Set. (If you have applied more than one data set to the attribute, the Data Set dialog box opens; in this case, select the data set file from the list and click OK.)

The View Data Set dialog box opens. The top of the dialog box displays the name of the selected data set and the number of values it contains. The list in the center of the box displays the actual values.
3. Click OK to close the dialog box when you have finished viewing the values.

Treating Data Values Results Files as Data Sets

You can use .qrd files as data sets, because they are essentially the same thing: a text file containing one or more delimited values. There are several advantages to using a .qrd file as a data set rather than as a data values file:

• All values in the .qrd file apply to a query. You do not need to select the values as a group. This makes it easy to apply a large set of values to a qualification.
• You can edit multiple values in the query by editing the file: you do not need to edit the SQL or the qualification tree to change the values.
• The values in the data set are applied to the qualification at run time; they are not saved with the query. This means that some other person or process can update the data set; any scheduled query that includes the data set will automatically use the updated qualification values.

💡 For more information on scheduling a query, see “Scheduling with BI Server” on page 248.

Using a .qrd file as a data set also lets you include values that contain delimiter characters (for example, 1,330.00, which includes a comma). Each .qrd file automatically comes with a .qrr file that BI Query uses to interpret the data. If you create a data values results file for an attribute whose values contain delimiters, BI Query lets you choose other delimiters for the file.

💡 For information on editing data values results files, see “Creating Data Values Results Files with a Text Editor” on page 83.

Creating Tables in the Database from Query Results

Most of the time, the tables you need already exist in the database. However, any query that retrieves information from a number of tables takes time because it uses multiple joins. Summarizing the information in one table eliminates the joins and reduces the number of rows that the DBMS has to process for subsequent queries. This gives you faster access to the information and reduces the load on the database server. Instead of querying the Stores, Sales and Sales Detail (three tables and two joins) tables to get a store’s order history, you could create an order history table that combines those records together in one table.

💡 You can also reduce the processing time for complex queries by creating super queries or query data objects (QDOs). For more information on QDOs, see the BI Query Data Models User’s Guide.

If the database administrator grants permission to create tables, and if the Edit Data Model, Save Data Model, and Submit to Table permissions are assigned, you can create tables in the database from query results.

This method works best when you need to analyze information that does not change regularly. For example, by creating a table that summarizes sales for a particular year, you can query the table and perform calculations on the results without the data becoming out of date.
When results are saved to a table in the database, BI Query creates in the active Design window a data object that corresponds to the table. If the Submit to Named Table permission is assigned, you can name the new table and the corresponding data object. Otherwise, BI Query generates the name for the table; you name the data object.

Creating Tables in the Database from Query Results

If you have the appropriate permissions, you can create a table in the database from a query.

💡 For information on deleting user-defined data objects (and the corresponding tables), see the Help for your BI Query application.

To create a table in the database from query results:
1. Create a query that retrieves the information you want to include in the table.
2. On the Query menu, point to Submit To; then, click Table. The Temporary Table dialog box opens.
3. Type a name for the new table; then, click OK. When you save the data model, the new table is saved with it.

Creating Tables Using Buttons

A query that submits results to a table can be linked to a button so that the process of creating new tables in the database is automated. If you have the Submit to Named Table permission, BI Query overwrites the table with the current results each time you click the button. If the linked query is run frequently, the Submit to Named Table permission prevents the database from filling up with outdated tables. If you have the Submit to Table permission, BI Query creates a new table in the database each time you click the button.

💡 For more information on creating buttons, see the BI Query Data Models User’s Guide.

Deleting User-Defined Objects

When you create a table in the database, BI Query automatically creates the corresponding data object for you. You can delete these user-defined data objects (and corresponding tables) using the Delete User Objects dialog box.
To delete a user-defined data object:

1. On the Tools menu, click Delete User Object. The Delete User Objects dialog box appears.
2. From the Data Objects box, select the user-defined data objects you want to delete.
3. Click Delete.

Creating Dynamic Relationships

The relationships between data objects in a data model let you form queries using more than one object. Rather than using all possible relationships between data objects, the data model typically provides only the most important ones. When data objects do not have a relationship, or when existing relationships do not relate the attributes the way you need them to be related, you can create dynamic relationships. Dynamic relationships are typically created “on the fly” for temporary use.

For more information on relationships, see “Data Model Components” on page 17.

In order to create a dynamic relationship between two tables, you need to decide the condition under which the data in one table is related, or “joined,” to data in the second. That is, you need to specify a join condition.

For example, you can create a relationship between the Retailer and Sales data objects, where the Retailer Number attribute in one object is made equal to the Retailer Number attribute in the other object. A query that includes both objects will then return data only for retailer numbers that appear in both.

For more information on join conditions, see the BI Query Data Models User’s Guide.

Choosing Join Operators

When you relate two data objects, you must specify an operator to determine the join condition. The available operators depend on the DBMS you are using.
The default operator, =, is the most common type of relationship; it specifies that the value of an attribute from one table equals the value of an attribute from another. You can also use other comparison operators (greater than, less than, greater than or equal to, less than or equal to) for the join condition. You may need to consult your administrator to ensure that the relationships you create use the appropriate join conditions and produce the results you want.

Depending on the DBMS you are using, you can specify outer joins in a dynamic relationship using the *= and =* operators or the outer join check boxes in the Choose Relationship dialog box. For more information on outer joins, see “Types of Joined Columns” on page 136.

Creating Dynamic Relationships

Creating dynamic relationships requires the Dynamic Relationships permission and the Allow Dynamic Relationships preference. If you have the Save Data Model and Save Dynamic Relationships permissions, you can save dynamic relationships with the model; otherwise, if you have the Save Queries permission, you can save the relationships with the queries that use them. (When you reopen queries involving saved dynamic relationships, BI Query recreates the relationships temporarily for those queries.)

You can either have BI Query attempt to create the join automatically using database keys and matching names, or you can specify the join conditions manually.

To create a dynamic relationship automatically:

1. Open the Choose Relationship dialog box by doing one of the following:
   - Select the two data objects you want to join (SHIFT+click). On the Query menu, click Relationship.
   - CTRL+click and drag from one data object to the other.
2. In the Choose Relationship dialog box, click Auto Join.
3. After the join has been created, you can edit the join conditions if you want.

To create a dynamic relationship manually:

1. Open the Choose Relationship dialog box by doing one of the following:
   - Select the two data objects you want to join (SHIFT+click). On the Query menu, click Relationship.
   - CTRL+click and drag from one data object to the other.
2. In the **Choose Relationship** dialog box, from the **Relationship** list, select `<dynamic>`. If no relationship currently exists between the data objects, `<dynamic>` is selected by default in the **Relationship** list. If a relationship already exists, the name of the existing relationship is selected and the dialog box displays the properties of the relationship. You can create a dynamic relationship between any data objects, even if they are already related.

💡 For more information on existing relationships in the **Choose Relationship** dialog box, see the Help for your BI Query application.

3. From the **Left Side** list, select a data object; then, click an attribute in the list below **Left Side**.

4. From the **Right Side** list, select a data object; then, click an attribute in the list below **Right Side**.

   The **P** and **F** columns identify primary and foreign keys respectively. These columns do not appear unless you have selected the **Show primary key** preference in the **Keys** dialog box.

5. 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>From the <strong>Operator</strong> list, select a join operator. Ensure that the <strong>Include All Left Rows</strong> and <strong>Include All Right Rows</strong> check boxes are cleared and that the join operator is not <code>*=</code> or <code>=*</code>.</td>
</tr>
<tr>
<td>Half Outer join</td>
<td>From the <strong>Operator</strong> list, do one of the following:</td>
</tr>
<tr>
<td>(Classic syntax)</td>
<td>• Select <code>*=</code> to make the <strong>Left Side</strong> data object row preserving.</td>
</tr>
<tr>
<td></td>
<td>• Select <code>=*</code> to make the <strong>Right Side</strong> data object row preserving.</td>
</tr>
</tbody>
</table>

The connection for the **Design** window must support Classic syntax.
Chapter 12: Advanced Querying Techniques

The Include All Left Rows and Include All Right Rows 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.

6. Click the down arrow button.
7. Click OK.
8. To save a relationship and name it, on the Query menu, click New.
9. In the BI Query dialog box, click Save.
10. In the Save Dynamic Relationship dialog box, specify a relationship name, and click Save. The name should reflect how the two data objects are related and should be unique.

If you have created more than one relationship, repeat for each. To remove a relationship, click Do not Save in the Save Dynamic Relationships dialog box; otherwise, when you save the data model, the relationship(s) are saved with it.

### Existing Relationships in the Choose Relationship Dialog Box

Using the Choose Relationship dialog box, you can create a dynamic relationship between any data objects, even if they are already related.

If no relationship currently exists between the data objects, <dynamic> is selected by default in the Relationship list. If a relationship already exists between the objects, the dialog box represents the existing relationship in the following ways:

- the relationship is selected in the Relationship box by default
Existing Relationships in the Choose Relationship Dialog Box

- the list of attributes for each selected data object is dimmed
- the **Join Conditions** box displays the join condition(s) for the relationship
- the **Include All Left Rows** and **Include All Right Rows** check boxes are dimmed and display the row preservation for the relationship (if the Design window uses the ODBC, Oracle, or SQL/92 join syntax)

If the existing relationship is an **actual relationship**, the dialog box displays two pairs of dimmed check boxes to represent row preservation. The outside check box in each pair represents one of the data objects that you want to join; the inside check box represents the actual relationship. For example, the **Include All Left Rows** pair of check boxes can represent the following types of row preservation:

- **Inner join**—The join returns only those rows from the data object and the actual relationship that match in the joined column.
- **Left outer join**—The join returns all rows from the data object and only those from the actual relationship that match in the joined column.
- **Right outer join**—The join returns all rows from the actual relationship and only those from the data object that match in the joined column.
- **Full outer join**—The join returns all rows from both objects.

### Deleting Dynamic Relationships

You can delete dynamic relationships you have created either by not saving them, or by deleting them after they have been saved. BI Query Admin users can also delete dynamic relationships by removing them from the **Design** window.

**To delete an unsaved dynamic relationship:**

1. Do one of the following:
   - Open another query.
   - On the **Query** menu, click **New**.
• On the **Standard** or **Query** toolbar, click the **New Query** button.

The **BI Query** dialog box displays the message “Save dynamic relationships before deleting current query?”

2. In the **BI Query** dialog box, do one of the following:
   - Click **Delete** to delete all new relationships.
   - Click **Save**. In the **Save Dynamic Relationship** dialog box, click **Don’t Save** for each relationship you want to delete.

**BI Query** deletes the relationships that you have created but not saved. Relationships will still be part of the query if you have saved the query.

To **delete** a saved dynamic relationship:

1. On the **Tools** menu, click **Delete User Object**. The **Delete User Objects** dialog box appears.

2. From the **Saved Dynamic Relationships** box, select any dynamic relationships you want to delete.

3. Click **Delete**.

### Outer Joins in Queries

For queries containing outer joins, the order in which the DBMS executes the joins can significantly affect the query results. For this reason, once you add an outer join to a query, **BI Query** sets an order for every join. If you have selected the **Show Join Order** preference, **BI Query** displays the join order for the connections (if the join order is relevant to the current query).

💡 For more information on outer joins, see “**Types of Joined Columns**” on page 136.

áticas can be two or more joins with the same join order. This is because join order is based on the level of nesting. Two joins that are nested at the same level within other joins have the same join order.
Join Order in Merged Data Objects

If there are merged data objects in the query, and the **Automatically Order Outer Joins** preference or permission is assigned, BI Query treats each join within the merged object as any other join and orders it automatically. If the preference and permission are not assigned, the join order for each join in the merged object matches the order in which the join was added to the object. If the **Show Join Order** preference is set, the ordinal number for joins on either side of a merged object will not likely be consecutive. (The missing ordinals belong to joins inside the merged object.)

💡 For more information on merged data objects, see the *BI Query Data Models User’s Guide*.

Join Order Example 1

In the following example, the inner join between the actual relationship PersonAddress and the data object Address takes place first (1). Person is then left-outer-joined to the first join (2). The result is then left-outer-joined to AddressType (3) using the embedded relationship Type.

💡 For more information on join types, see “Types of Joined Columns” on page 136.

The SQL (ANSI SQL/92) for this example is:

```sql
SELECT Person.*, Address.*, AddressType.*
FROM
  (Person LEFT OUTER JOIN
   (PersonAddress INNER JOIN Address
     ON Address.ID = PersonAddress.AddressID)
  )
```
JOIN Order Example 2

In the following query, both of the inner joins have an order of ‘1’, whereas the outer join has an order of ‘2’:

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

This query does not tell the DBMS how to order the inner joins with respect to each other because they occupy the same level of nesting.

Specifying Join Order

You can define the order of joins either automatically or manually.

To specify join order automatically:

Select either the **Automatically Order Outer Joins** preference or the **Automatically Order Outer Joins** permission. (You can set permissions using BI Query Admin only.) BI Query automatically chooses the join order for you. When selected, the **Automatically Order Outer Joins** permission overrides the **Automatically Order Outer Joins** preference.

The join order that BI Query automatically determines for a given query depends on the order in which you add the joins to the query. If you deselect and reselect joins within a query, the automatic join order may change but the outcome will be the same.

To specify join order manually:

- Clear the **Automatically Order Outer Joins** preference and permission. Select (click) the joins in the desired order.

  If you are not satisfied with a join order chosen manually, you can change it by deselecting the connection lines and then selecting them again in the order you want.
Join Conflicts

A join conflict occurs when more than one join could supply NULLs to the same materialized relation (either a nested join that is performed in advance of the joins in question or an actual table).

BI Query cannot resolve join conflicts on its own; instead, it can highlight the conflicting joins. If you want BI Query to display join conflicts, select the Show Join Conflicts preference. Conflicting joins appear as dashed connection lines:

Join conflicts do not appear in BI Web.

A query containing a join conflict may fail with a DBMS error, or the DBMS may treat one of the outer joins as an inner join. To resolve a join conflict, create two separate queries—one for each outer join; then, in a super query, combine the results of the two queries using a full outer join. The component queries will return results with the correct row preservation; the full outer join in the super query will combine those results and maintain the preserved rows.

For more information, see “Super Queries” on page 120.

Join Conflict Example

Consider three tables, A, B, and C, with the following data:

<table>
<thead>
<tr>
<th>Table A</th>
<th>Table B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td><strong>Country</strong></td>
</tr>
<tr>
<td>Jane Smith</td>
<td>UK</td>
</tr>
<tr>
<td>Hector Alvarez</td>
<td>Canada</td>
</tr>
<tr>
<td>Ella Bruce</td>
<td>Mexico</td>
</tr>
<tr>
<td>Francis Coulomb</td>
<td>Japan</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jane Smith</td>
<td>100</td>
</tr>
<tr>
<td>Hector Alvarez</td>
<td>200</td>
</tr>
<tr>
<td>Ella Bruce</td>
<td>300</td>
</tr>
<tr>
<td>Francis Coulomb</td>
<td>400</td>
</tr>
</tbody>
</table>
Tables A and B each form outer joins with C such that A and B are both row-preserving; C in this case is the materialized relation:

<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Manager</td>
<td>France</td>
</tr>
<tr>
<td>200</td>
<td>Writer</td>
<td>Mexico</td>
</tr>
<tr>
<td>250</td>
<td>Manager</td>
<td>Germany</td>
</tr>
<tr>
<td>500</td>
<td>Assistant</td>
<td>Australia</td>
</tr>
</tbody>
</table>

If the join between A and C has order 1, and you create a query involving both A and B, the join conditions between A and C are tested first. The results of this first join contain all rows from A, plus those from C that match A on the joined column (ID). Table A supplies NULLs for its unmatched rows:

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Title</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jane Smith</td>
<td>100</td>
<td>Manager</td>
<td>France</td>
</tr>
<tr>
<td>Hector Alvarez</td>
<td>200</td>
<td>Writer</td>
<td>Mexico</td>
</tr>
<tr>
<td>Ella Bruce</td>
<td>300</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>Francis Coulomb</td>
<td>400</td>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>
The results of the second join contain all rows from B, plus those from the results of the first join that match B on the joined column (Country). However, the NULL values supplied by A do not match any values in B in the joined column, so that the corresponding rows from the first join are not part of the second set of join results. In other words, the final result does not preserve all rows from table A. If the join between B and C has order 1, a similar problem occurs (not all rows from B are preserved).

**Outer Join Syntax in Queries**

BI Query supports the ODBC, Oracle, ANSI SQL/92, and “Classic” join syntax types. “Classic” (ANSI SQL/89) syntax uses the legacy operators *= and =*. No two syntax types can appear in the same Design window.

💡 For more information on outer join syntax types, see the [BI Query Data Models User’s Guide](#).

The particular outer join syntax in use in a Design window (or across the data model) depends on the connection assigned to that window (or model). If you want to use a particular type of outer join syntax in a query, you do not have to edit the existing joins; instead, you can change the syntax for the connection. In this case, BI Query applies the specified syntax to all outer joins that use the connection.

💡 Because outer join syntax information is specific to a given connection, it is stored in a connection file. When you change the syntax for a connection, it applies to all Design windows and data models that use the corresponding connection file. If you want your changes to be local to a particular window or data model, create a new connection file, assign it to the window or model, and then make the necessary changes to it. For more information on creating connection files, see the [BI Query Data Models User’s Guide](#).

**Changing Outer Join Syntax for Connections**

Use the Join Syntax dialog box to select an outer join syntax for a connection file. The syntax you select must be supported by the connection.

To change the outer join syntax for a connection:

1. On the Host menu, click Connections.
2. In the Connections dialog box, select the connection file you want to use; then, click Edit.
3. In the **Edit Connections** dialog box, click **Join Syntax**. The **Join Syntax** dialog box opens.

4. From the **Join Syntax** list, select the outer join syntax type that you want to apply to the connection. The dialog box displays the default settings for the selected syntax.

5. Click **OK**. The **Join Syntax** dialog box closes.

6. In the **Edit Connections** dialog box, click **Save** or **Save As** to save your changes.

Once you save the connection file with your changes, BI Query uses the selected syntax for all Design windows and data models that use that connection file.

## Modifying the SQL for Queries

You can modify the SQL string generated by BI Query when you formulate a query. However, you cannot directly modify the SQL for those queries (such as super queries) in which the results have been modified or combined with other results.

You must have the **Edit SQL String** permission to modify the SQL. You also need the **Show SQL String** preference to display the SQL in the **Query** window. You can then modify anything after the **SELECT** keyword (subject to any restrictions placed on you by the DBMS). If you modify the SQL string incorrectly, the DBMS may return an error message.

Changes you make to the SQL string should be the last changes you make to a query; BI Query replaces any changes you make in the **Query** window with any changes you make using attribute or Design windows.

BI Query expects results using the data format of the query selected in the Design window. The modified query must return compatible results from the database. If it does not, the results may have missing, misaligned, or otherwise incorrect columns. To avoid this problem, do not change anything before the **WHERE** clause: do not change the attributes you have selected; do not delete, add, or change the function you have applied to them.

## Modifying SQL in the Query Window

The SQL string in the **Query** window uses attribute names that appear in the database. When you edit the string, you must use these database names, not the display names that may appear in the attribute window.
To edit a standard query by modifying its SQL string:

1. Make sure that the **Edit SQL String** permission is assigned and that the **Show SQL String** preference is set.

2. Create, open, or retrieve the query.

3. Do one of the following:
   - On the **Query** menu, click **Show**.
   - On the **Query** toolbar or the **Standard** toolbar, click the **Open Query** button.

4. In the **Query** window (in SQL View mode), make the changes to the SQL string.

   - If you select **Show Prompt Values**, BI Query displays in the SQL string the information provided for prompt qualifications the last time the query was run. You cannot edit the SQL string while **Show Prompt Values** is selected.

If you subsequently modify the query in an attribute or **Design** window after changing the query’s SQL, BI Query displays an alert box to warn you that the modified SQL will be overwritten.

If the alert box appears, do one of the following:
   - Click **Change** to overwrite the SQL string you modified.
   - Click **Keep** to undo your changes in the attribute or Design window.

---

**Including SQL Prefixes and Suffixes**

You can fine-tune the SQL you send to the DBMS by specifying a prefix and/or a suffix for every query and subquery that you run.

For example, if you're using a DB2 DBMS on a VAX machine, you can improve database performance by appending the suffix `WITH UR` to every query to allow "uncommitted reads." If you're using Sybase, you can use the suffix `FOR BROWSE`.

You can also specify whether the suffix occurs before or after the semicolon that terminates the SQL for queries. This feature requires the **Edit SQL String** permission.

- Use the **Custom Check** permission to improve the efficiency of the query. For more information, see the Help for your BI Query application.

**To specify a prefix and suffix:**

1. In the **Preferences** dialog box, click **Query Options**.

2. In the **Query Options** dialog box, specify a prefix and/or a suffix.
3. To place the suffix after the semicolon that terminates the SQL, clear **Place Suffix Before Semi-Colon**.

4. Click **OK**.

**Building Circular Queries**

When the data model you are using contains two or more data objects whose relationships form a circular path, you can build queries that use all the data objects and all the relationships in that path. For example, if your data model contains the Employees and Departments data objects, and they are connected by the Work In and Managed By relationships, you can build queries that use both data objects and both relationships.

Because a data object in a circular path may be used in two roles, BI Query prompts you to specify whether or not to create a correlation name (or “alias”) for the object. If you choose to do so, BI Query creates a temporary copy of the object using the correlation name.

You can also choose not to create a correlation. The choice you make determines how BI Query applies join conditions in your query and the extent to which your results are restricted. A query that uses a correlation name lets you ask two questions of the same data object and is less restrictive than a query that does not use a correlation.
Circular Queries with a Correlation

The last relationship you select when building a circular query determines which objects can be aliased: you can create a correlation for one of the two objects that form that last-selected relationship. The last-selected relationship also determines which join conditions apply to the correlation.

BI Query creates the correlation name for an object using the following format:

$object__relationship$

where

- $object$ is the first five characters of the name of the aliased object
- $__$ is the literal string as written (two consecutive underscore characters)
- $relationship$ is the first five letters of the last-selected relationship

For example, if the last relationship you select is called Managed By and you create a correlation for an object named Employees, BI Query creates a second, temporary copy of Employees called Emplo__Manag.

If you are using BI Query Admin, you can also use a correlation name to create a permanent copy of a data object. Users of the data model can then create a query involving the aliased data object more than once without creating a circular query at the same time.

💡 For more information on correlation names, see the BI Query Data Models User’s Guide.
Example

If you select the Managed By relationship last and create a correlation for Employees, BI Query creates a second, temporary copy of Employees called Emplo__Manag. The result is two copies of Employees: one connected to Departments through the Work In relationship, the other, temporary copy connected to Departments through Managed By.

When you double-click the original data object in the Design window, you are prompted to specify which attribute window you want to use—the one for the original data object or the one for its correlation. You can use one attribute window to ask one question and the other attribute window to ask the second question.

For example, suppose you want to use the Employees and Departments data objects to identify employees that work for managers who make over $60,000. A correlation is needed in this case since you require two types of information from the Employees data object—the name of all employees and the name of managers who earn over $60,000. Use the Employees data object to first identify the names of the employees, and then use its correlation to identify managers who make over $60,000. (You use the correlation for the second question because it is the one that uses the join condition in the Managed By relationship.)

When you run the query, the DBMS processes it in two stages. Behind the scenes, the DBMS first obtains a preliminary results set (names of all employees and the departments they work in) by joining each row in Employees with each row in Departments, and then applying the join conditions for Work In.
The DBMS then applies the join condition for Managed By to the preliminary results set and Emplo__Manag (the copy of Employees) to retrieve the final results (employees who work for managers making over $60,000).

The SQL for this query is as follows:

```sql
SELECT Employees.LastName
FROM Employees Emplo__Manag, Departments, Employees
WHERE
  (Emplo__Manag.Salary > 60000.00
   AND Departments.DepartmentID = Employees.DepartmentID
   AND Departments.Manager = Emplo__Manag.EmployeeID)
);
```

Building Circular Queries with a Correlation

A circular query with a correlation is less restrictive than one without a correlation. Because circular queries can be complex, if you build one with a correlation, it is good practice to build it initially with an extra qualification to limit the number of rows returned. In this way, you can verify the design of the query without retrieving a possibly large amount of data.

To build a circular query with a correlation:

1. Decide which data object you want to create a correlation for and which relationship you want to use for that correlation.
2. Click the relationships joining the data objects you want to include in the query, making sure that the last relationship you click is the one you want to use for the correlation.
3. In the Circular Query dialog box, select the data object for which you want to create a correlation.
4. Click OK. In the Design window, double-click the data object you selected in step 3.
5. In the Open Attribute Windows dialog box, click the attribute window(s) you want to open.
6. Click OK. In the attribute window(s), select the attributes you want to include in the query; then, close the window(s).
7. Select any other attributes you want to use from the other data objects in the circular path.
8. To maintain all the join conditions in the query, on the Query menu, point to Modifiers; then, deselect Trim Relationships (if necessary).
For more information on this option, see “Removing Unnecessary Joins from Queries” on page 57.


Circular Queries Without a Correlation

When you build a query without a correlation, BI Query does not create a temporary copy of a data object in the circular path. This allows you to ask one question using all the data objects and all the join conditions in the circular path. The results are usually more restricted because the rows returned from the database must meet all the join conditions in the circular path.

Example

Suppose you want to use the Employees and Departments data objects to find out which employees make over $60,000 and manage the department they work in. In this case, you need to create a circular query, but you do not need a correlation because you are using each data object only once.

When you run the query, the DBMS processes it in two stages. Behind the scenes, the DBMS first obtains a preliminary results set (names of all employees who make over $60,000 and the departments they work in) by applying the join condition for Work In to Departments and Employees.

The DBMS then applies the join condition for Managed By to the preliminary results set to obtain the final results (employees who make over $60,000 and manage the department they work in).

Without a correlation, this circular query returns only the rows that meet the join conditions in both the Work In and Managed By relationships.

The SQL for this query is as follows:
SELECT Employees.LastName
FROM Departments, Employees, Departments
WHERE
  (  
    Employees.Salary > 60000.00 
  )
  AND Departments.DepartmentID = Employees.DepartmentID
  AND Departments.Manager = Employees.EmployeeID
);

Building Circular Queries Without a Correlation

Building a query without a correlation is similar to building a query with a correlation except that the order in which you select the relationships is not important. When BI Query prompts you to specify whether or not to create a correlation, choose not to create one.

If the circular query contains an outer join, you must create a correlation. (The No Correlation check box in the Circular Query dialog box is dimmed.)

To build a query with a circular path that does not use a correlation:
1. Click the data objects and relationships you want to include in the query.
2. In the Circular Query dialog box, click No Correlation.
3. Click OK.
4. In the Design window, double-click the data objects in the circular path.
5. In the attribute window(s), select the attributes you want to include in the query; then, close the attribute window(s).
6. On the Query menu, click Run.
Chapter 13

Exporting Information

This section provides information on the following:

- “Exporting Information” on page 225
- “Preparing Results for Export” on page 226
- “Export Methods” on page 232
- “Exporting Results to Microsoft Excel” on page 236

Exporting Information

By exporting query results to other applications, you can extend the usefulness of your queries. BI Query exports results as text files that can be used by any application. For example, you can augment a report prepared in Microsoft Word by including actual data that you have retrieved from a query, or you can transfer results to a worksheet in Microsoft Excel to perform advanced statistical analyses. In addition to exporting query results directly to text files, you can do the following:

- export the SQL for queries that have not been filtered or combined
- export results automatically using menu commands or Microsoft’s Dynamic Data Exchange (DDE)
e-mail the query SQL files and results files that you have created
• copy the results from a result set in BI Query and paste them directly into a document in another application

💡 For more information on saving the SQL for a query, see “Saving Queries” on page 39.

If the appropriate permissions are assigned, you can also send query results to a new table in the database. For more information on this procedure, see “Creating Tables in the Database from Query Results” on page 203.

Preparing Results for Export

Before exporting, you need to prepare your results for export. There are three things to consider:

Display Format
How numbers, characters, and dates are represented. For example, Microsoft Excel does not accept some date formats allowed in BI Query, like YY/MM/DD.

💡 For more information on data formats, see “Data Formats in BI Query” on page 63.

Results Options
Settings required by the receiving application. For example, a worksheet application may require double quotation marks around textual data. (If you are not sure what the receiving application requires, check its documentation.)

💡 For more information, see “Results Options and Separators” on page 227.

Export Options
The receiving application, the export mechanism, and the commands to be executed.

💡 For more information, see “Setting Export Options” on page 231.
Results Options and Separators

When you save results to files or export them to other applications, be sure to specify the results options and separator characters required by the receiving application. You should also set these options if you are importing data to update the database. Use the **Results Options** dialog box.

**Results Options**

You can set the following result options:

<table>
<thead>
<tr>
<th>Results Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save Query Name</td>
<td>Includes the name of the query that generated the results. This option is useful if you export results to applications that use macros to perform certain operations depending on which query generated the data.</td>
</tr>
<tr>
<td>Save Column Headings</td>
<td>Includes results column headings.</td>
</tr>
<tr>
<td>Save Prompts</td>
<td>Includes the number of prompts used in the query (followed by a record separator), plus the following information for each prompt:</td>
</tr>
<tr>
<td></td>
<td>• the prompt ID (followed by a field separator)</td>
</tr>
<tr>
<td></td>
<td>• the number of values used in the prompt (followed by a field separator)</td>
</tr>
<tr>
<td></td>
<td>• the prompt values (each separated by field separators)</td>
</tr>
<tr>
<td>Text in Quotes</td>
<td>Inserts double quotation marks around textual data.</td>
</tr>
</tbody>
</table>
Chapter 13: Exporting Information

Separators

Separator characters delimit each field in a row (normally a tab character) and each row (record) of data (normally a carriage return). A separator can also indicate the end of the file.

A separator must be a valid ASCII character. BI Query provides the following codes to represent common ASCII characters:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;TAB&gt;</td>
<td>The tab character (ASCII 9).</td>
</tr>
<tr>
<td>&lt;LF&gt;</td>
<td>The line feed character (ASCII 10).</td>
</tr>
<tr>
<td>&lt;CR&gt;</td>
<td>The carriage return character (ASCII 13).</td>
</tr>
<tr>
<td>&lt;Space&gt;</td>
<td>The space character (ASCII 32).</td>
</tr>
</tbody>
</table>

Results Options Scope

You can specify results options and separators for the following targets:
Preparing Results for Export

- The entire data model. If you open the Results Options dialog box from the Preferences dialog box, the options you specify apply to all results sets and buttons you subsequently create, and to all data you import for updating the database. These options are the default for the model.
- A particular results set. If you open the Results Options dialog box for the active results set, the options apply only to that results set; they override the data model defaults.
- The results generated by a button linked to a query. If you open the Results Options dialog box for such a button, the options apply only to the results sets created when you click the button; the options override the data model defaults.

Specifying Results Options and Separators

Use the Results Options dialog box to set the options and separators for the entire data model, a particular results set, or a button.

To specify results options and separators:

1. Open the Results Options dialog box by doing one of the following:
   - To specify default results options for the data model, on the Tools menu, click Preferences. In the Preferences dialog box, click Results Options.
   - To specify results options for a particular results set, on the Results menus, click Options. (The results set must be active.)
   - To specify results options for a button linked to a query, open the Edit Button dialog box for the button, select File from the list in the Output To area; then, click Results Options.

2. In the Results Options dialog box, specify the option(s) you want.

3. In the Separators area, specify the separator(s) you want from the Field, Record, and End of File lists. You can either type the separators or select them from the lists. Use the caret character (^) to represent the CTRL key. For example, type ^D to represent CTRL+D (ASCII 4).

   🕵️‍♀️ For more information on options and separators, see the previous topic.

4. Click OK.
Export Options

To export results directly to another application, you need to specify the following export options:

- the application to which the results will be exported
- the export mechanism you are using (DDE or menu commands)
- the commands to be executed in the receiving application

When you export results to another application, BI Query copies the results, checks whether the receiving application is already running, and, if not, starts it. Depending on the export mechanism, BI Query then calls the appropriate menu commands in the receiving application or sends the DDE commands to the receiving application.

DDE lets you pre-program commands in the target application’s programming language (such as Excel’s macro facility) and further process the results. For example, you could open an existing Excel spreadsheet and run a macro to perform a calculation not available in BI Query.

Export Mechanisms

Whether you use menu commands or DDE depends on what you want the receiving application to do:

Menu Commands

Let you export results by specifying the menu commands to be executed in the receiving application. For example, to paste the contents of the Clipboard into a Microsoft Excel worksheet, you can specify the Paste command on the Edit menu in Excel.

DDE

Lets you include arguments in your commands that instruct the receiving application to carry out more specific tasks than simply using menu commands (for example, to open a particular worksheet and run a macro).

To determine the requirements of the receiving application, see its user documentation. You may also need the help of your administrator to find out whether an application supports DDE and what service name to use.
You must set up reports properly in order to export it directly to an application. For more information, see “Specifying Results Options and Separators” on page 229.

Setting Export Options

You can specify export options as a preference or when you create a button. Specifying export options in the Preferences dialog box applies the options to buttons created thereafter and to any export operations you perform using the Export command on the Results menu. Export options specified from the Create Button or Edit Button dialog boxes override the options specified as preferences.

To set export options:

1. Open the Export Options dialog box using one of the following methods:
   - To specify default export options, open the Preferences dialog box (on the Tools menu, click Preferences). In the Preferences dialog box, click Export Options.
   - To specify export options for a button, open the Create Button or Edit Button dialog box. In the Output To area, select Application from the list; then, click Export Options.

2. In the Export Options dialog box, from the Application list, select the application to which you want to export. If the application is not in the list, select User Defined.

3. Do one of the following:
   - If the path for the application is defined in your system PATH variable, type the name of the application in the Execute Command Line box. Use the name that appears in the title bar of the main window for the application.
   - If the executable for the application is not in your system PATH variable, type the full path and file name of the executable file in the Execute Command Line box.
     Alternatively, click the browse button to the right of the Execute Command Line box. In the Select Application dialog box, browse to and select the executable file; then, click Open.

4. If you want to paste results to a specific document, type the full path and file name of the document in the Execute Command Line box (after the application file).
5. If you are exporting using DDE, and the service name is not the same as the executable name, type the service name in the **Service Name** box. (The service name is the name that the application registers with the DDE server.)

6. In the **Commands** area, select **DDE** or **Menu Item**.

7. In the box to the right of the **Commands** area, type the commands to be executed in the receiving application.

   ![Tip]
   For examples of DDE commands for Excel, see “DDE Command Examples for Excel” on page 237.

   To send two DDE commands from BI Query, add a vertical bar (|) between the DDE command sequences you type. The first sequence is sent to the application only if it is not already running; BI Query must start it. You must enter all arguments in each DDE command.

   The menu command mechanism requires the following format:

   `menucommand|command`

   For example, to run the **Paste** command on the **Edit** menu, type `Edit|Paste`.

8. Click **OK**.

### Export Methods

When you export query results, you can either save the results or SQL to a text file, or send the results directly to an application.

### Exporting Results to Text Files

You can export query results to a text file in order to use them again in BI Query or retrieve them in other applications such as word processors or spreadsheet programs.

BI Query saves results as two files: one with the extension `.qrd` (ASCII text format for use in other applications) and one with the extension `.qrr` (BI Query-specific information, such as formats for each column and the SQL used to retrieve the results).

![Tip]
When you open results in BI Query, only the `.qrd` file appears in the **Open Results File** dialog box.
To save the results of a query:

1. Specify the results options required by the external application. For more information, see “Results Options and Separators” on page 227.

2. Do one of the following:
   - Create or open the query that generates the results you want to save. On the Query menu, point to Submit To; then, click File. In the Save Query Result to File dialog box, type a name into the File Name box. BI Query displays the corresponding file names with the .qrd extension.
   - With the results window active, click Save on the Results menu. In the Save Results dialog box, click External Results, and then navigate to the location where you want to save the file. In the File Name box, type the name of the file. (BI Query automatically adds the .qrd extension when you save the file.)

3. Click Save.

4. In the external application, open the .qrd file to view the saved results.

When you delete an exported results set, be sure to delete both the .qrd and the .qr.f files.

Exporting Results to Applications

When you export results to an application, BI Query formats the results according to the results options you have specified, copies the results to the Clipboard, and starts the target application (if it is not already running). If you have specified any export options—such as opening a particular document and pasting the Clipboard contents into it—BI Query also executes those commands.

To export results:

- With the results window active, click Export on the Results menu.

Exporting Results Using Buttons

If the Edit Data Model and Save Data Model permissions are assigned, you can create a button that sends query results to an application. When you click the button, BI Query first runs the linked query before exporting the results.

💡 For more information on buttons, see the BI Query Data Models User’s Guide.
If you can share queries to other users, you can create shared and personal versions of a given query. Although the two versions of the query have the same name, the actual SQL for each may differ. When you link a button to the query, you must specify which version of the query you want to use.

💡 For more information on shared and personal queries, see “Personal Versus Shared Files” on page 26.

⚠️ You cannot link a button to an external query (one that is not located in the query folder or its subfolders).

**To use a button to export results to an application:**

1. Build and save the query (if necessary). If BI Query is not in Design mode, click the **Design Mode** button on the **Standard** toolbar.

2. On the **Drawing** toolbar, click the **Button** button; then, click and drag the mouse over the area where you want the button to appear.

3. Double-click the button that appears. The **Edit Button** dialog box opens.

4. In the **Appearance** area, specify the appearance for the button.

5. In the **Link To** area of the **Edit Button** dialog box, select **Query** from the list box on the left.

6. 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**.

   If you have selected a query that has shared and personal versions, you can switch between the two by selecting **Shared Query** or **My Query** in the **Edit Button** dialog box.

7. In the **Output To** area of the **Edit Button** dialog box, select **Application**; then, click **Export Options**. The **Export Options** dialog box opens.

8. Set the export options. For more information, see “Setting Export Options” on page 231.

9. Click **OK**. In the **Edit Button** dialog box, click **OK**.
Exporting Results Using E-Mail

If you are running an e-mail application on your computer, you can export BI Query files as attachments with your e-mail messages. E-mail provides an easy way to distribute new and revised queries and results to others. As long as your e-mail application supports Microsoft’s Messaging API (MAPI), you can e-mail files directly from BI Query.

For more information on using your e-mail client, see your client documentation.

When you send e-mail from within BI Query, it starts your e-mail client and automatically attaches the query, results, or report files to the mail message, depending on the context in which you open the client:

- If you open or build a query before starting your e-mail client, BI Query automatically attaches the corresponding .qry file in your message—regardless of which window you are working in.
- If you open, retrieve, or generate a results set, and then start your e-mail client, the corresponding .qrd and .qrr files are attached only if the results window is the active window. To automatically attach a results set, display the results window before starting your e-mail client.
- If you open or save a standard report specification before starting your e-mail client, BI Query automatically attaches files corresponding to the current report (.rpt).

For more information on standard report files, see “Standard Reports in BI Query” on page 155.

If you do not want to include the current query or results set in an e-mail message, remove the file from the attachment.

To send files using e-mail:

1. On the File menu, click Send. The e-mail application dialog box appears with a list of the attached files.
2. Attach additional files or remove any files you do not want to send.
3. Supply a subject for the message and the address of the recipient; then, send the message.
When you retrieve exported files that have been sent to you by another user, store query and standard report files in the Queries folder so that they appear in the corresponding directory dialogs when you open them. Store results in the folder that contains the data model. Be sure you retrieve both .qrd and .qrr results files. To use retrieved files, close the data model if it is open, reopen it, and then open the files.

**Exporting Results to Microsoft Excel**

When you set the options for exporting query results to Microsoft Excel, the default DDE command is [Paste ()], which starts up Excel (if it is not already running) and pastes the contents of the Clipboard (the query results) into a worksheet.

Using the [Run (...) command, you can run any macro that is stored in any worksheet in your XLSTART folder. Because Excel 97/2000 automatically opens the worksheets in your XLSTART folder, you do not need to explicitly open these files using a DDE command. However, to run a macro in any other worksheet, you must first open that worksheet.

For more information on the XLSTART folder and Excel add-in programs, consult your Excel documentation.

You can also run macros that are supplied by an add-in program. However, before you can run such a macro in a worksheet, you must install the add-in (if necessary) and load it into the worksheet.

**Specifying DDE Commands**

When you create a button that sends query results to Excel, you can specify up to two DDE commands. If the first of the two commands is to be executed only once (such as opening a specific worksheet on startup), use the pipe symbol (|) between the two commands:

[Open (...) | [Run (...)]

(The pipe symbol has a different purpose for DDE commands than it does for menu commands.)

You can also use the pipe symbol to prevent Excel from doing the following:
- opening a worksheet that is already open
- pasting results into an open worksheet and overwriting the contents
If you use the pipe symbol, the first command is sent only if the application is not already running and BI Query must start it up; the second command is always sent, even when the application is already running. Remember to specify all arguments for each command.

![Note] The pipe symbol is not necessary if it is logical for both commands to run each time you export results.

**DDE Command Examples for Excel**

The following examples of DDE command sequences send results to Microsoft Excel. For other commands, see your Excel user documentation.

<table>
<thead>
<tr>
<th>Command Sequence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Paste()]</td>
<td>Pastes the results into a worksheet already open in Excel (or, if Excel is not already running, runs Excel, opens a new worksheet, and pastes the Clipboard contents into it).</td>
</tr>
<tr>
<td>![New()]![Paste()]</td>
<td>Pastes the results into a new worksheet in Excel.</td>
</tr>
<tr>
<td>![Open(&quot;c:\mydocs\results.xls&quot;)][Paste()]</td>
<td>Opens the specified worksheet (if it is not already open) and pastes the results into it. If you often use the same format for a worksheet, you can set up a worksheet template, and then use this command sequence to paste results into it. Once the results have been pasted, you can save the worksheet under a different name.</td>
</tr>
<tr>
<td>![Open(&quot;c:\mydocs\results.xls&quot;)][Run(&quot;macro1&quot;)]</td>
<td>Opens the <code>results.xls</code> worksheet and then applies the <code>macro1</code> macro to that sheet. The macro must reside on the worksheet.</td>
</tr>
</tbody>
</table>
Combining Multiple Results Sets

You can combine the results of more than one query in a Microsoft Excel worksheet. This is useful when you are using data from different tables. There are two methods you can use to combine multiple results sets.

Method A

1. In BI Query, create a button for each results set. Link each button to the appropriate query. Set each button to send the query results to a macro. The macro must position the cursor in Excel and then paste the results into the appropriate location in the default worksheet.

For more information, see “Exporting Results to Applications” on page 233.

For example, to combine three results sets, you can set up three buttons to execute the following commands:

<table>
<thead>
<tr>
<th>Button</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Button A</td>
<td>[open(&quot;c:\mydocs\results.xls&quot;)]</td>
</tr>
</tbody>
</table>
Exporting Results to Microsoft Excel

In this case, you would have to click button A first. However, if the macros you want to use reside on a worksheet in your XLSTART folder, you can accomplish the same result using the following commands:

<table>
<thead>
<tr>
<th>Button</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Button B</td>
<td>[run(&quot;macro2&quot;)]</td>
</tr>
<tr>
<td>Button C</td>
<td>[run(&quot;macro3&quot;)]</td>
</tr>
</tbody>
</table>

For more information on the XLSTART folder, see your Excel documentation.

2. Click each button in turn.
   - Do not close Excel until all three buttons have sent their results; otherwise, you will create three separate worksheets.

3. If necessary, save the worksheet under a different name.

**Method B**

Create a button that selects the current query and then exports the results to Excel using the default [Paste()] command. This method reduces the number of buttons needed, but requires you to open each query in BI Query before running it. You must also reposition the cursor in Excel each time so that each successive set of results does not overwrite the previous one in the worksheet.
Chapter 14

Publishing and Scheduling with BI Server

This section provides information on the following:

- “Publishing with BI Server” on page 241
- “Publishing Standard Report Specifications” on page 245
- “Publishing Data Values Results Files” on page 247
- “Scheduling with BI Server” on page 248

Publishing with BI Server

If you are in a BI Server environment, you can take advantage of BI Server’s publishing, retrieving, scheduling, and security features. The requirements for using these features depend on the item you want to publish or otherwise access.

Requirements for Publishing Queries

When you publish and secure data models, queries are automatically published as well. You can also publish and secure queries individually. Published queries that have been shared are available to other users and can be scheduled.

💡 For more information on scheduling queries, see “Scheduling with BI Server” on page 248.
Before you can publish a query to the BI Server Repository, the following requirements must be met:

- The data model on which the query is based must have been published. If it has not been published, BI Query requires you to publish it before you can publish a query based on it. In order to be published, a query must also be the current query. You do not have to save a query before you publish it.
- You must have the Save Queries permission and at least one of the following BI Server system permissions:
  - Publish Queries from Admin
  - Publish Queries from User/Update
- If you are using BI Query User/Update, and you want to share the query to other users, you must have the Share queries and reports permission. (All queries published by BI Query administrators are effectively shared.)

In addition to queries, you can also publish data values results files and standard report specifications.

### Publishing Queries

When you publish a query to the Repository, BI Query also saves the query locally in the Queries folder where the data model is stored. (Data values queries are stored in the DataVals folder.) If you want to publish a query as a data values query file for a particular attribute, follow the naming convention for data values files.

💡 For more information, see “Naming and Storing Data Values Files” on page 80.

You can create subfolders in the Repository to organize your queries more efficiently. When you publish queries, the folder structure you have set up in the Repository is preserved in the local Queries folder.

**To publish a query to the Repository:**

1. In a saved and published data model, create or open a query.
2. On the Query menu, point to Publish, and then do one of the following:
   - To publish the query as a standard query, click Query. The Publish Current Query dialog box opens.
   - To publish the query as a data values query, click Data Values Query. The Publish Data Values Query dialog box opens.
For more information on shared and personal queries, see “Personal Versus Shared Files” on page 26.

3. Do one of the following:
   • To share the query (or data values query), click **Shared Queries** (or **Shared Data Values**).
   • To publish the query as a personal query (or personal data values query), click **My Queries** (or **My Data Values**). This option is not available if you are using BI Query Admin.

4. Navigate to the location where you want to save the query. You can create a subfolder by clicking the **Create New Folder** button.

5. In the **File Name** box, type the file name for the query.

6. To set security for the query, click **Set Security**; then, use the **Set Security** dialog box to grant or deny access to the query. For more information, see “Setting Security for Queries” on page 244.

7. Click **Publish**.

8. Test the query to be sure that it's been published. (Close it; then, retrieve it from the Repository.)

**Retrieving Queries**

Before you can retrieve a query from the BI Server Repository, you must either open a local copy of the published data model on which the query is based or retrieve the published data model from the Repository.

For information on deleting published queries, see the Help for your BI Query application.

To retrieve a query from BI Server:

1. On the **Query** menu, point to **Retrieve**, and then do one of the following:
   • To retrieve a standard query, click **Query**. The **Retrieve Query** dialog box opens.
   • To retrieve a data values query file, click **Data Values Query**. The **Retrieve Data Values Query** dialog box opens.

2. Do one of the following:
   • To retrieve a shared query (or shared data values query), click **Shared Queries** (or **Shared Data Values**).
To retrieve a personal query (or personal data values query), click **My Queries** (or **My Data Values**). This option is not available if you are using BI Query Admin.

💡 For more information on shared and personal queries, see “Personal Versus Shared Files” on page 26.

3. Select the query you want to open; then, do one of the following:
   - To retrieve the selected query, click **Retrieve**.
   - To retrieve and run the query, click **Run**.

### Deleting Published Queries

When you delete a published query from the Repository, you can delete the local copy at the same time.

**To delete a published query:**

1. In a saved and published data model, on the **Query** menu, point to **Retrieve**, and then do one of the following:
   - To delete a standard query, click **Query**. The **Retrieve Query** dialog box opens.
   - To delete a data values query file, click **Data Values Query**. The **Retrieve Data Values Query** dialog box opens.

2. In the dialog box, select a query; then, click **Delete**. The **Confirm File Delete** dialog box opens.

3. If you want to delete the local copy of the selected query at the same time, select **Also Delete Local Copy**.

4. Click **Yes**.

💡 Only BI Query administrators can delete shared files.

### Setting Security for Queries

When you publish data models, you include queries with them and set security for the queries as part of setting security for the data model as a whole.

If you have the appropriate permissions, you can also add a query to a previously published data model, whether the model is your own or someone else’s. When you publish such a query, it automatically has the security settings of the data model with which it is published.
For more information on BI Server security, see the BI Query Data Models User’s Guide.

If you’re working in BI Query Admin, once you’ve published a query, whether as part of a data model or separately, you can change its security settings.

To set security for an individual query:

1. In the Publish Current Query dialog box or Publish Data Values Query dialog box, do one of the following:
   - To secure an unpublished query (or data values query), type its name in the File Name box.
   - To secure a published query (or data values query), select it.

2. Click Set Security.
3. In the Set Security dialog box, review the security assignments currently in effect for the query. Make any changes as necessary.

   The Set Security dialog box is part of another BI program module and has its own Help system.

4. When you’ve finished setting security, click OK.
5. Click Publish to publish the query.

Publishing Standard Report Specifications

If you are in a BI Server environment, you can publish a standard report specification to the Repository. By default, the corresponding query for the report is also published. You cannot publish a standard report to the Repository unless the corresponding data model has also been published.

To publish a report specification:

1. Create and format the report. Add any required calculations to it.
3. Do one of the following:
To share the report specification, click **Shared Reports**. If you are using BI Query User/Update, this option is available only if you have the **Share Queries and Reports** permission.

To publish the specification as a personal specification (not accessible to other users), click **My Reports**. This option is not available if you are using BI Query Admin.

For more information on shared and personal queries, see “Personal Versus Shared Files” on page 26.

4. Navigate to the location where you want to save the specification. You can create a subfolder by clicking the **Create New Folder** button.

5. In the **File Name** box, type the name for the specification you want to publish. Do not include the `.rpt` extension.

6. If you do not want to publish the corresponding query with the specification, clear **Publish Query Also**.

7. To control access to the specification, click **Set Security**; then, use the **Set Security** dialog box to grant or deny access to the corresponding query. The **Set Security** button is available only if **Publish Query Also** is selected and only if you are using BI Query Admin.

8. Click **Publish**.

**Setting Security for Standard Reports**

Published standard report specifications inherit the security setting applied to the data model. You cannot set security on a specification independently of the model. However, you can effectively grant or deny access to a standard report by granting or denying access to its query, because a specification is visible to BI Web users only if its query is also visible. Similarly, BI Query users who have retrieved a data model can refresh its standard reports only if they have been granted access to the corresponding queries.

**Standard Reports in BI Web**

Published standard reports appear in the BI Web Portfolio in a special subfolder for each data model. When a BI Web user opens a standard report in the Portfolio, the report is automatically refreshed.

BI Web users can open a standard report only if the following conditions are true:

- the report specification has the same name as the corresponding query
• the query has been published
• the users have been granted access to the query

**Publishing Data Values Results Files**

When you publish a data values results file, you can make its data values available in qualifications and prompts for a particular attribute. To do so, follow the **naming convention for data values files**. For more information, see “Naming and Storing Data Values Files” on page 80.

💡 For information on retrieving and deleting published data values files, see the Help for your BI Query application.

**To publish a data values results file:**

1. Create a data values results file or open an existing one.

2. On the **Results** menu, click **Publish Data Values**. The **Publish Data Values Results** dialog box opens.

3. In the **File Name** box, type the name of the file. BI Query automatically adds the extension `.qrd` to the file when you publish it.

4. Click **Publish**.

**Retrieving Published Data Values Results Files**

If you are working in a BI Server environment, you can retrieve a published data values result file from the Repository, and then open it as a results set in BI Query.

**To retrieve and open a data values result file from the Repository:**

1. On the **Results** menu, point to **Retrieve**; then, click **Data Values**. The **Retrieve Data Value Results** dialog box opens. (You must be working online in order to open this dialog box).

2. Do one of the following:
   • To retrieve a shared data values result file, click **Shared Data Values**.
   • To retrieve a personal data values result file, click **My Data Values**. This option is not available if you are using BI Query Admin.

3. Select the file you want to retrieve and open.

4. Click **Retrieve**.
Deleting Published Data Values Files

When you delete a published data values results file, you can also delete the local copy of the file at the same time.

To delete a published data values results file:
1. In a saved and published data model, on the Results menu, point to Retrieve; then, click Data Values.
2. In the Retrieve Data Values Results dialog box, select a data values file; then, click Delete.
3. In the Confirm File Delete dialog box, select Also Delete Local Copy if you want to delete the local copy of the file in addition to the copy in the Repository. Click Yes to delete the file(s).

Scheduling with BI Server

If your querying environment includes the BI Server Repository and you have the appropriate publishing and scheduling permissions assigned by the BI Server administrator, you can schedule queries.

Scheduling regulates the flow of information from the database by processing the enterprise’s queries and reports in a managed time frame. Jobs can be scheduled to run once or repeatedly, at specific intervals, during less busy times, or when a specific event has occurred. Scheduling also provides a variety of ways to distribute information.

If the queries in a data model you publish will be scheduled to run automatically, you must set a default connection file for the model before you publish it. Since the Scheduler runs queries from BI Server, not from the machine where they were created or last saved, you need to ensure as well that the connection you use is supported from the BI Server machine. Your BI Server administrator can provide you with this information.

💡 For more information on default connection files, see the Bi Query Data Models User’s Guide.

For more information on scheduling, see the BI Server Scheduling User’s Guide.
Scheduling Queries

You must publish your query before you can schedule it, and in order to publish a query, you must first publish the data model on which it is based.

To schedule a query:
1. On the Query menu, point to Retrieve and then click Query.
2. In the Retrieve Query dialog box, select a query to schedule and click Retrieve.
3. On the Query menu, click Schedule.
4. In the Schedule Job dialog box, provide the appropriate information. (The Schedule Job dialog box is part of another BI program module with its own Help system. For information about this dialog box, click its Help button.)

Retrieving Scheduled Results

You can retrieve from the BI Server Repository any results sets generated by scheduled queries. To do so, you must have the BI Server Schedule Queries permission.

💡 For more information on scheduled queries, see the BI Server Scheduling User’s Guide.

To retrieve results for a scheduled query:
1. On the Results menu, point to Retrieve; then, click Scheduled Results.
2. In the Retrieve Results Sets dialog box, browse for and select a results set.
3. Click OK.
This section provides information on the following:

- “General Accessibility” on page 251
- “Technical Support” on page 253

**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.
Index

Symbols

!= operator 51, 52
"and/or" qualifications 47
# formatting character 71
% wildcard character
in qualifications 53
&& text string 174
&«prompt» text string 173
&d text string 173
&dn text string 173
&p text string 173
&pp text string 173
&q text string 173
&t text string 172
* formatting character 71
* = operator 215
.bak files 61
.gqa files 21
.gql files 23
.gqu files 21
.qrd files 80, 232
.qrr files 80, 232
.qry files 39, 80
< operator 51, 52
<= operator 51, 52
= operator 51, 52
=* operator 215
> operator 51, 52
>= operator 51, 52
\ formatting character 71
_ formatting character 71
_ wildcard character 53

A

aborting queries 37
accessibility, general 251
ad hoc
  drill-down mode 120
queries 20, 34
administrator in BI Query 16
administrator-layer files 30
location 30
split data model 21
aggregate functions 185
and grouping 186
applying to attributes 190
AVERAGE 188, 189
AVERAGE DISTINCT 188
COUNT 188, 189
COUNT DISTINCT 188
example of grouping with 186
grouping after calculating 189
grouping before calculating 190
MAXIMUM 188
MINIMUM 188
SUM 188, 189
SUM DISTINCT 188
aliased data 85
data values file 86
in queries 87
aliases for data objects 218
Allow Dynamic Relationships preference 206
Allow Non-Select SQL permission 150
alphabetic sort order 129
ANSI SQL/89 outer join syntax 215
ANSI SQL/92 outer join syntax 215
appending rows 132
applications
exporting results to, overview 233
using buttons to export results to 233
Attribute Name box 42
attribute window
checking available data values 45
creating queries 34
grouping attributes 190
modifying SQL 217
qualification tree 42
sort box 38
sorting rows 38
attributes
definition 18
in variables 109
placeholders in variables 110
qualifying 41
qualifying other attributes 49
selecting 35
attributes, calculated
overview 193
creating 194
automatic variables 107
Automatically Order Outer Joins 211
permission 212
preference 212
AVERAGE aggregate function 188
AVERAGE DISTINCT aggregate function 188
average, results 131
B
Begins With operator 52
BETWEEN operator 51, 54
BI Query
overview 15
administrator 16
disconnecting from DBMS 32
exiting 31
opening a data model 28
Reports 20
starting 28
BI Query Admin 16
    administrator 16
BI Query Reports 20
BI Query Update 16
BI Query User 16
BI Server 25
    environment 27
    logging on 30
    Repository 25
    system permissions 26
    terminology 26
BI Server permissions
    Publish Queries from Admin 242
    Publish Queries from User/Update 242
    Schedule Queries 249
BI Server security
    securing queries 244
    BIGroupID variable 108
        using in a query qualifier 115
    BIUserID variable 108
        using in a query qualifier 115
    borders, adding to standard reports 171
    building queries 35
    buttons 19
        export options 231
        exporting results to applications 233
        for combining results sets in Excel 238
C
    calculated attributes
        overview 193
        creating 194
    deleting 198
    editing 198
    qualifying other attributes 49
    selecting functions for 197
    calculated columns in standard reports 163
        deleting 165
        editing 165
        calculating a function
            grouping after 189
            grouping before 190
        calculations 185
            performing on results 131
    calculations, in standard reports
        adding grand totals 163
        basing one on another 164
        in report columns 163
    cancelling
        qualifications 42
        queries 37
    cells, changing width 119
    characters
        inserting/deleting in results/reports 71
        spaces appearing as nulls 55
        strings containing wildcards 53
    Choose Relationship dialog box
        existing relationships in 208
    chronological sort order 129
    circular queries 218
        outer joins in 223
        with correlations 219, 221
        without correlations 222, 223
Index

Classic (ANSI SQL/89) outer join syntax 215
clearing previous queries 36
column width, changing in results 119
columns
  hiding/showing in results 129, 130
  joining 135, 138
  reordering 37
  reordering in results 129
results, calculations 131
specifying order 37
columns in standard reports
  adjusting size of 166
  banding 168
changing heading style 170
combining 167
headings 156
reordering 129
stacking 169
combined data models 20, 23
combined queries
  exporting 39
  saving 39
combining columns 138
combining qualifications 47
combining results sets 132
  in Excel 238
commands
  Dynamic Data Exchange (DDE) 237
common language names, substitute for database names 85, 86
component queries in super queries
deleting 126
editing 123
Condition box 42
conditional prompts
  overview 91
  creating 101
  naming 101
connection files
  changing outer join syntax 215
  changing the current connection 32
  connecting to a DBMS 32
  connection files, default 32
Contains operator 53
copying results for export 119, 226
correcting errors in super query windows 127
correcting queries 61
correlation names 218
correlations, circular queries without 222
COUNT aggregate function 188
Count All modifier 48
COUNT DISTINCT aggregate function 188
creating
  freehand queries 144
creating tables
  in the database 203, 204
curly braces
  around prompts 99
  for text entry 78
customizing data models 21
customizing results for users 115
D
data aliasing
overview 85
creating a data values file for 86
modifying with text editor 86
using in queries 87
data formats 63
ambiguous 77
converting year formats 77
correcting invalid data entry formats 78
data entry formats 76
data type formats 64
date entry format 77
formatting options 65
in standard reports 157
modifying SQL 216
number-entry format 76
opening the Format dialog box 68
order of formatting options 66
predefined formats 65
target types 67
data formats, options
category-specific options 65
Computed String option 66
Null values option 65
order of options 66
String option 65
data formats, specifying
default formats 69
for boolean data 73
for data entry 76
for numeric data 74
for years (number of digits) 77
the Computed String option 72
the Null Values option 70
the String option 70
data models 18
customizing 21
opening 28
retrieving 29
split 30
types 20
data models, combined 20, 23
data models, passwords 28
data models, split 20, 21
data objects
definition 18
in circular queries 218
querying 56
relating 205, 206
data sets
.qrr files 200
overview 200
character delimiters in 201
comments in 201
file extensions 200
format 201
in qualification trees 202
using .qrd files as a data set 202
using as a qualifier 202
viewing 202
data type formatting categories 64
data values
checking 45
loading all 45
qualifying with 45
data values files 79
overview 79
aliasing data 86
in conditional prompts 101
naming 80
naming for use in prompts 80
order of use 81
storing 80
types 80
using in prompts 81
using in queries 85
data values query files
creating 84
defined 80
location for 39
naming 84
saving 39
data values results files
creating 82
creating with a text editor 83
defined 80
naming 82
using as a data set 202
database
overview 16
connections to 32
creating tables from query results 204
functions in variables 112
records 16
supported operators 56
date formats
automatic conversion of 77

DBMS connections 16, 32
DBUserID variable 108
using in a query qualifier 115
DDE commands
combining 237, 238
for Excel 237
new 237
paste 237
run 237
specifying 236
default data values query 45, 79
default export options, setting and overriding 231
default results options 228
default text ornaments 172
deleting component queries 126
deleting dynamic relationships 209
deleting operations in super queries 126
deleting user-defined objects 205
deleting values in qualification tree 42
Design mode 25
Design windows 19
disconnecting from DBMS 32
displaying data values 45
displaying prompt values in queries 103
displaying results
form view 118
spreadsheet view 118
Distinct modifier 48
Does Not Begin With operator 53
Does Not Contain operator 53
Does Not End With operator 53
duplicate values, eliminating 48, 192
dynamic relationships 205
between unrelated data objects 205, 206
choosing operators 205
creating 206
deleting 209
outer joins in 206
vs. existing joins 208
Dynamic Relationships permission 206

E
Edit Data Model permission 203, 233
Edit SQL String permission 58, 150, 216
editing component queries of super queries 123
ing super queries 123, 126
eliminating duplicate values 48, 192
e-mail using to export BI Query files 235
Ends With operator 53
equijoins 136
error messages
  column n doesn’t match 140
  missing column 140
  modifying SQL 216
range error 140
reorder error 140
sort error 140
elements in super query windows 127
Excel
  combining three results sets 238
  DDE commands for 236
  exporting results to 236
  excess joins, cleaning up 57
excluding rows with nulls or character spaces 55
EXEC statement in SQL scripts 142, 150
executive buttons 19
exiting BI Query 31
Export command (Results menu) 233
export options 230
  DDE commands 236
exporting BI Query files by e-mail 235
exporting queries 225
exporting results
  copy and paste 226
directly 233
  options for 227
  preparations for 226
  specifying DDE commands 236
to an application 233
to Excel 236
  using e-mail 235
external queries 39
external specifications 159

F
file extensions for data sets 200
file types
  .bak 61
  .gqa 21
  .gql 23
  .gqu 21
  .qrd 80, 232
  .qrr 80, 232
  .qry 39, 80
files
administrator-layer 30
personal 26
query 39
results 232
shared 26
files, data values 79, 80
query 39, 80
results 80
filtering results 128
in super queries 120
finding items in query results 118
finding ranges of values 54
fine-tuning SQL 217
footers in standard reports 156
form view 118
formatting characters
# 71
* 71
\ 71
_ 71
freehand queries
canceling 149
creating new 144
ingiving 145
Freehand Query window 143
overview 142
running 149
Freehand Query window 143
opening 143
functions
AVERAGE 188
AVERAGE DISTINCT 188
COUNT 188
COUNT DISTINCT 188
in calculated attributes 197
MAXIMUM 188
MINIMUM 188
SUM 188
SUM DISTINCT 188
G
grand totals, adding to standard reports 163
group prompts
overview 90
creating 100
naming 100
order of execution 100
group qualifications
suspending 192
grouping
after calculating functions 189
before calculating functions 190
with aggregate functions 186
without calculating functions 192
H
headers in standard reports 156
hiding columns in results 129, 130
Hummingbird accessibility 251
I
importing prompts 103
importing variables from a file 114
IN operator 51, 54
invalid data formats, correcting 78
IS NOT NULL operator 55
IS NULL operator 55
J
joining columns 135, 138
joining rows 132
joining tables 205, 206
joins
  conflicts 213
  joining columns 135, 138
  natural joins 136
operators for join conditions 205
order in outer joins 210
removing unnecessary from queries 57
types 136

L
layers
  administrator 21
  user 21
left outer joins 136
legacy outer join operators 215
lists, matching items in 54
local data model 28
lock icon 40
logging on to BI Server 27, 30

M
MAPI, mailing from BI Query 235
matching items in lists 54
materialized relation 213
MAXIMUM aggregate function 188
merged tables in outer joins 211
MINIMUM aggregate function 188
missing column
  range error 140
  reorder error 140
sort error 140
missing data 55
modifiers
  Count All 48
  Distinct 48
  Trim Relationships 48
multi-pass reporting 164

N
naming data values files 80
  in prompts 80
query files 84
results files 82
naming prompts 91, 98
  conditional prompts 101
  group prompts 100
natural joins 136
negative qualifications 48
nested queries 198
  creating 199
  inserting 199
non-SQL operators 56
NOT BETWEEN operator 51, 54
NOT IN operator 51, 54
not indicator 48
NULL operator 55
NULL values 55
  format order 66
numeric sort order 129

O
objects
  user-defined, deleting 205
ODBC
syntax for outer joins 215
offline, working 27
opening
  Freehand Query window 143
opening data models 28
operations
  deleting in super queries 126
  editing in super queries 126
operators
  != 51, 52
  *= 215
  < 51, 52
  <= 51, 52
  = 51, 52
  *= 215
  > 51, 52
  >= 51, 52
applying to qualifications 44
Begins With 52
BETWEEN 51, 54
Contains 53
database supported 56
Does Not Begin With 53
Does Not Contain 53
Does Not End With 53
Ends With 53
for join conditions 205
IN 51, 54
in variables 109
IS NOT NULL 55
matching any/all 53
NOT BETWEEN 51, 54
NOT IN 51, 54
NULL 55
placeholders in variables 110
User defined 56
Oracle
  outer join syntax 215
ordering columns in results 129
ornaments
  in standard reports 172
  special text strings in 172
outer join types
  full outer joins 136
  left outer joins 136
  right outer joins 136
outer joins
  available syntax 215
Classic (ANSI SQL/89) syntax 215
conflicts 213
  in circular queries 223
  in dynamic relationships 206
  in queries 210, 215
join order 210
ODBC, Oracle, and ANSI SQL/92 syntax 215
  syntax, changing per connection 215
using legacy joins as is 61
outer joins, join order 210
  Automatically Order Outer Joins preference and permission 212
  Show Join Order preference 210
specifying join order 212
Index

P

password
  BI Server 30
passwords
data model 28
paste, DDE command 237
performing calculations 108
  using variables 108
permissions
  Allow Non-Select SQL 150
  Automatically Order Outer Joins 211, 212
  Dynamic Relationships 206
  Edit Data Model 203, 233
  Edit SQL String 58, 150, 216
  Save Data Model 203, 206, 233
  Save Dynamic Relationships 206
  Save Queries 206
  Share Queries and Reports 242, 246
  Share queries and reports 26
  Submit to Named Table 204
  Submit to Table 203, 204
personal files 26
placeholders
  examples of use in variables 110
  for attributes and operators 110
preferences
  Allow Dynamic Relationships 206
  Automatically Order Outer Joins 211, 212
  Preserve Legacy Query Outer Joins 60, 61
  Show Join Conflicts 213
Show Join Order 210, 211
Show SQL String 58, 150, 216
preferences
  for results options 227
prefixes, SQL 217
Preserve Legacy Query Outer Joins
  preference 60, 61
printing standard reports 161
prompts
  overview 89
  and curly braces 99
appearance in BI Web 90
automatic formatting 93
creating 98
data types 90, 92
data values in BI Web 94
default values 90
deleting 102, 105
display objects in BI Web 95
ing 98
examples of uses 97
importing 103
in query data objects 90
in variables 112
inserting in queries 99
modifying 102
naming 91, 98
no formatting data type 93
non-typed prompts in BI Web 97
optional versus required 93
order of web-reporting values 104
radio buttons in 96
showing values in queries 103
specifying web-reporting values 104
types 90
using more than once in queries 99
prompts, conditional
overview 91
creating 101
prompts, group
overview 90
creating 100
publishing
data values results files 247
queries 242
standard reports specifications 245
publishing queries 242
deleting published queries 244
requirements for 241

Q
Qualification Edit box 43
qualification operators, user-defined 56
qualification tree 42
combining qualifications 47
modifying values 42
negative qualification 48
qualifications
combining 47
group, suspending 192
matching any/all 53
multiple 47
multiple with "and/or" 47
negative 48
qualifying 41, 44
more than one attribute with a prompt 99
one attribute with another 49
with data values 45, 85
with lists 54
with prompts 99
with ranges of values 54
with subqueries 199
with user name variables 115
with variables 114
queries 20, 34
ad hoc 20, 34
building 35
cancelling 37
circular 218
clearing previous 36
combining previous results 132
combining results 132
correcting 61
creating freehand 144
creating tables from results 204
default data value 79
deleting 40
exporting 39, 225
external 39
file location 39
modifying for selected results 120
multiple, combining 132
nested 198
opening 40
outer joins in 210, 215
qualifying 41, 44
qualifying with variables 114
refreshing 60
removing unnecessary joins 57
retrieving 40
reusing 40
running 36
saving 39
using legacy joins as is 61
viewing SQL 58
with a correlation 221
Queries folder 38
report specifications in 159
subfolders in 38
queries, circular 218
with correlations 219
without correlations 222, 223
queries, saving 39
saving results 232
Query Admin
administrator 16
query data objects
containing prompts 90
query modifiers
Count All 48
Distinct 48
Trim Relationships 48
query results, searching 118
Query window
changing views in 58
docked or floating 59
Query view 58
SQL view 58
view modes 58
viewing 59
querying multiple data objects 56
quitting BI Query 31
R
range error: missing column 140
range of results, narrowing 131
range of values, finding 54
real database name, using to qualify attributes 49
recent data model 28
records 17
reducing network load 203
refreshing queries 60
Regional Options (Control Panel) 64
Regional Settings (Control Panel) 64
relational database management system (RDBMS) 16
relationships 205
between unrelated data objects 206
missing from data objects 205
new 205
trimming 57
reorder error: missing column 140
reordering columns 37, 129
reordering standard reports columns 129
reports
BI Query Reports 20
Standard Reports 20
Repository 25
result sets, exporting 225
results 20, 117
BI Server environment 118
closing all 119
combining results sets 132
deleting 233
files 232
filtering, overview 128
finding text in 118
form view 118
grouping after calculating functions 189
modifying queries for selected results 120
narrowing the range 131
opening 118
performing calculations on 131
preparing for export 226
saving 232
selecting 119
set 117
sorting 128, 129
spreadsheet view 118
troubleshooting combined 140
results options 227
button preferences 229
preferences 227
Save Column Headings 227
Save Prompts 227
Save Query Name 227
specifying for export 228
Text in Quotes 227
Transpose 228
results section in standard reports 156
results sets 117
and super queries 120
combining 132
results, exporting
by copying and pasting 226
options 227
to an application 233
Retain Attribute and Operator option 111
retrieving
data models 29
queries 243
reusing queries 40
reverse qualifications 48
right outer joins 136
rows
appending 132
in databases 16
sorting 38
Run mode 25
run, DDE command 237
running
stored procedure 152
running queries 36
S
Save Column Headings results option 227
Save Data Model permission 203, 206, 233
Save Dynamic Relationships permission 206
Save Prompts results option 227
Save Queries permission 206
Save Query Name, results option 227
saving dynamic relationships 206
saving queries 39
combined queries 39
data values queries 39
subqueries 39, 199
saving results 232
scalar functions 197
scheduling queries by using variables 108
searching, query results 118
selecting
  attributes 35
selecting operators for joins 205
selecting results 119
separators in results options 227
server 27
Share Queries and Reports permission 242, 246
Share queries and reports permission 26
shared files 26
Show Join Conflicts preference 213
Show Join Order preference 210, 211
Show SQL String preference 58, 150, 216
showing columns in results 130
showing prompt values in queries 103
sort error: missing column 140
sort order
  for attributes 38
  for rows 38
specifying 128
sorting
  alphabetical or chronological 129
before running 38
results 128, 129
reverse order 129
rows 38
special text strings
  for standard reports 172
specifications
  applying in standard reports 160
  for standard reports 159
  publishing 245
split data models
  overview 21
  administrator layer 21, 30
  user layer 21
spreadsheet view 118
  of results 118
SQL
  fine-tuning 217
  modifying 216
  permissions required to modify 216
  prefixes and suffixes 217
  sending non-SELECT SQL 150
SQL scripts 142
  EXEC statement 142
Standard Reports 155
  applying specifications 160
  creating 158
  data formats in 157
  default text ornaments in 172
  exporting reports to text files 161
  external specifications 159
  in BI Web 246
  printing 161
  publishing specifications 245
  removing a specification 159
  reordering columns 129
  report specifications 158
  saving specifications 159
special text strings for 172
structure 156
subheadings in 168
Standard Reports, editing
adding borders 171
adding calculated columns 163
adding grand totals 163
adding subtotals 162
adding text ornaments 174
adjusting column size 166
banding columns 168
basing one calculation on another 164
changing heading text style 170
combining columns 167
customizing column settings 170
deleting calculated columns 165
editing calculated columns 165
editing default ornaments in 174
formatting columns 170
stacking columns 169
starting queries 36
stored procedure
running 152
stored procedures
overview 150
Submit to Named Table permission 204
Submit to Table permission 203, 204
subqueries
creating 199
data types 198
exporting 39
inserting 199
invalid operators 200
overview 198
previewing 200
saving 39
substituting common language names 85, 86
subtotals, adding to standard reports 162
suffixes, SQL 217
SUM aggregate function 188
SUM DISTINCT aggregate function 188
sum values, calculated on results 131
super queries 120
combining multiple results sets 120
correcting errors 127
editing 123
editing component queries 123
filtering results sets 120
from filtering results 128
Super Query window 121
testing 127
troubleshooting 127
Super Query window 121
opening 123
suspending
group qualifications 192
T
tables
creating in the database 203, 204
database 16
joining 205, 206
Technical Support 253
Teradata databases
prompt data types 93
terminology 18
testing
   super queries 127
Text in Quotes results option 227
text ornaments
   in standard reports 172
   special strings in 172
text strings
   && 174
   &«prompt» 173
   &d 173
   &dn 173
   &p 173
   &pp 173
   &q 173
   &t 172
toolbars 24
Transpose results option 228
Trim Relationships modifier 48
trimming relationships 57
troubleshooting
   combined results 140
   super queries 127
Troubleshooting mode (super queries) 127
types of column joins 136

user name
   BI Server 30
user-defined objects
   deleting 205

V
variables
   overview 107
   attribute and operator information 109
   BGUserID 108, 115
   BIUserID 108, 115
   creating 109, 113
   DBUserID 108, 115
   editing 113
   examples of placeholder use 110
   importing from a file 114
   in scheduled queries 108
   including database functions 112
   including prompts 112
   naming 107
   naming attributes and operators 109
   performing calculations 108
   placeholders for attributes and operators 110
   Retain Attribute and Operator option 111
   retrieving values 108
   storage 107
   uses 108
   using to qualify attributes 114
   within variables 112
view modes, Query window 58
viewing
Index

data sets 202

W

web-reporting values 104
wildcard characters 53
windows
  Attribute 18
  working offline 27, 30
  working online 30

Y

years, two digit to four digit 77