About using Microsoft Query to retrieve external data

This topic contains information about:

What is Microsoft Query?

Setting up data sources

Defining your query

Working with the data in Microsoft Excel

What is Microsoft Query?

Microsoft Query is a program for bringing data from external sources into other Microsoft Office programs — in particular, Microsoft Excel. By using Query to retrieve data from your corporate databases and files, you don't have to retype the data you want to analyze in Excel. You can also update your Excel reports and summaries automatically from the original source database whenever the database is updated with new information.

Types of databases you can access You can retrieve data from several types of databases, including Microsoft Access, Microsoft SQL Server, and Microsoft SQL Server OLAP Services. You can also retrieve data from Excel lists and from text files. See a list of types of databases you can access.

In Excel, you can also retrieve data from Web pages, but you don't need Query to do this. For information about retrieving data from Web pages, see Excel Help.

Selecting data from a database You retrieve data from a database by creating a query, which is a question you ask about data stored in an external database. For example, if your data is stored in an Access database, you might want to know the sales figures for a specific product by region. You can retrieve a part of the data by selecting only the data for the product and region you want to analyze and omitting the data you don't need.

Query Wizard - Choose Columns		
Available tables and co	lumns:	Columns in
	▲ >	Category
Sales by Region Salesperson Customer ID	_ <<	Amount

🔀 Microsoft Excel - Sales by Region				
123		A	В	С
2	1	Category	Region	Amount
	2	Beverages		
	11		East Total	\$35,233
+	22		North Total	\$51,832
+	30		South Total	\$30,172
+	38		West Total	\$40,392
	39	Dairy		
+	47	200 00	East Total	\$39,444
+	59		North Total	\$63,418
+	65		South Total	\$27,889
+	75		West Total	\$44,110
	76			
-	77			\$332,490

With Query, you can select the columns of data that you want and bring only that data into Excel.

Updating your worksheet in one operation After you have external data in an Excel workbook, whenever your database changes, you can refresh the data to update your analysis — without having to recreate your summary reports and charts. For example, you can create a monthly sales summary and refresh it every month when the new sales figures come in.

Using Query to retrieve data Bringing external data into Excel with Query takes three steps: First you <u>set up a data source</u> to connect to your database, then you <u>use the Query Wizard</u> to select the data you want, and finally you <u>return the data to Excel</u> where you can format it, summarize it, and create reports from it.

Return to top

Setting up data sources

What is a data source? A data source is a stored set of information that allows Microsoft Excel and Microsoft Query to connect to an external database. When you use Query to set up a data source, you give the data source a name, and then supply the name and location of the database or server, the type of database, and your logon and password information. The information also includes the name of an ODBC driver or a data source driver, which is a program that makes connections to a specific type of database. For some types of external databases (OLAP cubes), all you need to do to retrieve the data is to <u>set up a data source</u>.

How Query uses data sources After you set up a data source for a particular database, you can use it whenever you want to create a query to select and retrieve data from that database — without having to retype all of the connection information. Query uses the data source to connect to the external database and show you what data is available. After you create your query and return the data to Excel, Query retrieves the data and provides the Excel workbook with both the query and data source information so you can reconnect to the database when you want to refresh the data.



A data source is the means through which Query and Excel connect to a specific database and retrieve data.

Return to top

Defining your query

Use the Query Wizard for most queries The Query Wizard, which is part of Microsoft Query, is designed for users who are new to creating queries. The wizard makes it easy to select and bring together data from different tables and fields in your database. After you select the data, you can also use the wizard to filter and sort the results of your query. You can <u>use the Query Wizard</u> either to create a complete query or to start a query that you make further changes to directly in Query.

Work directly in Query for other types of queries If you are familiar with creating queries or you want to create a more complex query, you can <u>work directly in Query</u>. You can use Query to view and change queries you start creating in the Query Wizard, or you can create new queries without using the wizard.

Use Query directly when you want to create queries that do the following:

Select specific data from a field In a large database, you might want to choose some of the data in a field and omit data you don't need. For example, if you need data for two of the products in a field that has information for many products, you can use criteria to select data for only the two products you want. Learn about <u>using criteria</u>.

Retrieve data based on different criteria each time you run the query If you need to create the same Excel report or summary for several areas in the same external data — such as a separate sales report for each region — you can create a type of query called a parameter query. When you run a parameter query, you are prompted for the data to use as the criteria when the query selects records. For example, a parameter query might prompt you to enter a specific region, and you could reuse this query to create each of your regional sales reports. Learn about parameter queries.

Combine data in different ways You can use Query to combine, or join, the data from different tables in your database. For example, if you have a table of product sales information and a table of customer information, you might join these tables in ways that show which

customers have not made purchases recently. Learn about joining tables.

Creating an OLAP cube from the data in a query On-Line Analytical Processing (OLAP) is a way of organizing data for querying and reporting instead of processing transactions. In an OLAP database, data is organized hierarchically and stored in cubes instead of tables.

You can use Query to create an OLAP cube from the data in a query so that it takes less time and effort to create Microsoft Excel reports based on the data. To create an OLAP cube, you run the OLAP Cube Wizard from the Query Wizard or directly from Query. You can return the data to Excel as a PivotTable report. Learn about <u>creating an OLAP cube from queried data</u>.

More information

When to use the Query Wizard versus Microsoft Query

Return to top

Working with the data in Microsoft Excel

After you create a query in either the Query Wizard or Microsoft Query, you can return the data to a Microsoft Excel worksheet. The data becomes an external data range or a PivotTable report that you can format, analyze, and refresh.

Formatting and analyzing the data In Excel, you can use familiar tools such as the Chart Wizard or automatic subtotals to present and summarize the data. You can format the data, and your formatting is retained when you refresh the external data from the source database. If you add columns of formulas to the external data range, the formulas can be copied automatically to any new rows that are added to the range when you refresh the data. You can use your own column labels instead of the field names, and add row numbers automatically. For information about controlling the formatting and layout of an external data range, see Excel Help.

Reusing and sharing queries In both the Query Wizard and Query, you can <u>save a .dqy query file</u> that you can modify, reuse, and share. Excel can open .dqy files directly, which allows you or other users to create additional external data ranges from the same query.

If you want to share an Excel summary or report that is based on external data, you can give other users a workbook that contains an external data range, or you can create a report template. A report template lets you save the summary or report without saving the external data so that the file is smaller. The external data is retrieved when a user opens the report template. For information about creating report templates, see Excel Help.

Refreshing external data When you refresh external data, you run the query to retrieve any new or changed data that matches your specifications. You can refresh a query in both Query and Excel. Excel provides several options for refreshing queries, including refreshing the data whenever you open the workbook and automatically refreshing it at timed intervals. You can continue to work in Excel while data is being refreshed, and you can also check the status while it's being refreshed.

For information about refreshing external data in Excel, see Excel Help.

Return to top

Get program information about Microsoft Query

On the Help menu in Microsoft Query, click About Microsoft Query.

Program information, including the version number, appears in the **About Microsoft Query** dialog box.

Use keyboard shortcuts in Microsoft Query

What do you want to do?

Move in the Microsoft Query panes by using shortcut keys Select data and criteria by using shortcut keys Edit data and criteria by using shortcut keys Use function keys in Microsoft Query Use shortcut keys in the OLAP Cube Wizard Move in the Microsoft Query panes by using shortcut keys

Move in the Table pane

Press	То
ТАВ	Move between tables
UP ARROW or DOWN ARROW	Move to and select field names in a table
SHIFT+UP ARROW or SHIFT+DOWN ARROW	Select contiguous fields in a table or to cancel selection

Move between columns and rows in the Data pane or the Criteria pane

Press	То
F5, type the number of the record in the Record box, and then press ENTER	Move to a specific record (row) in the Data pane
TAB, ENTER, or RIGHT ARROW (from a selected cell)	Move to the next column in the Data pane
SHIFT+TAB, SHIFT+ENTER, or LEFT ARROW (from a selected cell)	Move to the previous column in the current row or the last field in the previous record
END	Move to the last column in the current row
CTRL+END	Move to the last column in the last row
HOME	Move to the first column in the current row
CTRL+HOME	Move to the first column in the first row
DOWN ARROW	Move to the current column in the next row
CTRL+DOWN ARROW	Move to the current column in the last row
UP ARROW	Move to the current column in the previous row
CTRL+UP ARROW	Move to the current column in the first row

Move the insertion point within a value in a cell in the Data pane or the Criteria pane

Press	То
F2	Cancel selection and display the insertion point in a selected cell
RIGHT ARROW	Move one character to the right
CTRL+RIGHT ARROW	Move one word to the right
LEFT ARROW	Move one character to the left
CTRL+LEFT ARROW	Move one word to the left
END	Move to the end of the current line in single-line or multiple-line values
CTRL+END	Move to the end of the value in multiple-line values
HOME	Move to the beginning of the current line in single-line or multiple-line values
CTRL+HOME	Move to the beginning of the value in multiple-line values

Move between screens in the Data pane or the Criteria pane

Press	То
PAGE DOWN	Move down one screen in a pane
PAGE UP	Move up one screen in a pane
CTRL+PAGE DOWN	Move right one screen in a pane
CTRL+PAGE UP	Move left one screen in a pane

Select data and criteria by using shortcut keys

Select values within a column

Press	То
F2	Select the current value, or cancel selection and display the insertion point
TAB, ENTER, or RIGHT ARROW (from a selected cell)	Select the value in the next column
SHIFT+HOME	Make a selection in a value from the insertion point to the beginning of the line
SHIFT+END	Make a selection in a value from the insertion point to the end of the line
SHIFT+RIGHT ARROW	Extend selection or cancel selection of one character to the right
CTRL+SHIFT+RIGHT ARROW	Extend selection or cancel selection of one word to the right
SHIFT+LEFT ARROW	Extend selection or cancel selection of one character to the left
CTRL+SHIFT+LEFT ARROW	Extend selection or cancel selection of one word to the left

Select a row

Press	То
SHIFT+SPACEBAR	Select the current row
DOWN ARROW	Select the next row, if the current row is selected
UP ARROW	Select the previous row, if the current row is selected
SHIFT+DOWN ARROW	Extend or cancel selection down one row
SHIFT+UP ARROW	Extend or cancel selection up one row
SHIFT+PAGE DOWN	Extend selection down one screen
SHIFT+PAGE UP	Extend selection up one screen
CTRL+SHIFT+SPACEBAR	Select the entire result set

Select a column

Press	То
CTRL+SPACEBAR	Select the current column
RIGHT ARROW	Select the column to the right, if the current column is selected
LEFT ARROW	Select the column to the left, if the current column is selected
SHIFT+LEFT ARROW	Extend or cancel selection left one column
SHIFT+RIGHT ARROW	Extend or cancel selection right one column
CTRL+SHIFT+PAGE DOWN	Extend selection right one screen
CTRL+SHIFT+PAGE UP	Extend selection left one screen

Edit data and criteria by using shortcut keys

Copy, cut, paste, or delete

Press	То
CTRL+C	Copy the current selection onto the Clipboard
CTRL+V	Paste the contents of the Clipboard at the insertion point
CTRL+X	Cut the current selection and copy it onto the Clipboard
BACKSPACE	Delete the current selection or the character to the left of the insertion point
DELETE	Delete the current selection or the character to the right of the insertion point

Zoom in to edit in the Data pane

Press To

SHIFT+F2 Open the **Zoom Field** dialog box to enter values in small input areas in the Data pane more conveniently

Function keys in Microsoft Query

Press	То
F1	Switch to Help
SHIFT+F1, and then press ENTER	Get context-sensitive Help for the currently selected item
F2	Select or cancel selection of current value
ALT+SHIFT+F2 or SHIFT+F12	Save a query with the Save command
ALT+F2 or F12	Save a query with the Save As command
SHIFT+F2	Open the Zoom Field dialog box to enter values in small input areas in the Data pane more conveniently
CTRL+F4	Close the active query
ALT+F4	Quit Microsoft Query
F5, type the number of the record in the Record box, and then press ENTER	Go to a specific record (row) in the Data pane
F6	Move to the next pane in the Query window
SHIFT+F6	Move to the previous pane in the Query window
CTRL+F6	Switch to the next open Query window
CTRL+SHIFT+F6	Switch to the previous open Query window
F9	Run the currently open query

Use shortcut keys in the OLAP Cube Wizard

Use these keys to complete Step 2 in the OLAP Cube Wizard.

To move a field from the Source fields list to the Dimensions box

- 1. Press the TAB key to select the **Source fields** list.
- 2. Press the UP ARROW or DOWN ARROW key to select the field you want to move.
- 3. Press the TAB key to select the ">> button, and then press ENTER.
- 4. Repeat steps 1 through 3 to move other fields.

To move a field to a lower or higher level within a dimension in the Dimensions box

- 1. Press the TAB key to select the **Dimensions** box.
- 2. Press the UP ARROW or DOWN ARROW key to select the field you want to move.
- 3. Press CTRL+X.
- 4. Press the UP ARROW or DOWN ARROW key to select where you want to move the field.
- 5. Press CTRL+V.

Install an ODBC driver so that you can access an external data source

The ODBC drivers provided by Microsoft with Microsoft Query are all installed when you install Query. <u>See a list of ODBC drivers installed with Query</u>.

If the external database you want to access is not supported by these ODBC drivers, you need to obtain and install a Microsoft Officecompatible ODBC driver from a third-party vendor, such as the manufacturer of the database. Contact the database vendor for installation instructions.

Note OLAP databases do not require ODBC drivers. When you install Query, drivers and support are installed to access databases that were created by using Microsoft SQL Server OLAP Services. To connect to other OLAP databases, you need to install a data source driver and client software. Learn about third-party OLAP providers.

Install OLAP providers

To connect to third-party products that provide OLAP data and services, you need to install OLAP provider software. This software includes additional data source drivers and client software to allow you to connect to a third-party OLAP database on a server.

To use the Microsoft Office features for working with OLAP data, the third-party OLAP provider must be Office compatible. The provider might or might not support creating and working with offline cube files. You can check for this support by creating a Microsoft Excel PivotTable report from source data that's provided by the third-party OLAP database. In the PivotTable report, the **Client-Server Settings** command on the **PivotTable** menu (**PivotTable** toolbar) will be unavailable if the provider does not support offline cube files.

For information about installing and using a third-party OLAP provider, consult your system administrator or the vendor for your OLAP product. After you install the provider, you can <u>set up data sources</u> in Microsoft Query to access the provider's databases.

Databases you can access

Microsoft Office provides drivers that you can use to retrieve data from the following data sources:

Microsoft SQL Server OLAP Services (OLAP provider) 7.0 and later

Microsoft Access 2000 and later

dBASE

Microsoft FoxPro

Microsoft Excel

Oracle

Paradox

SQL Server

Text file databases

Note You can use ODBC drivers or data source drivers from other manufacturers to get information from other types of databases that are not listed here, including other types of OLAP databases. For information about other drivers that may be available from Microsoft, see the xlreadme.txt file. For information about installing an ODBC driver or data source driver that is not listed here or in the xlreadme.txt file, see the documentation for the database, or contact your database vendor.

Specify a source to retrieve data from

- 1. If the **Choose Data Source** dialog box is not displayed, click **New Query** 🔂.
- 2. Do one of the following:

To specify a data source for a relational database, text file database, or Microsoft Excel list, click the **Databases** tab.

To specify an OLAP cube data source, click the **OLAP Cubes** tab. This tab is available only if you ran Microsoft Query from Microsoft Excel.

- 3. Double-click New Data Source.
- 4. In step 1, type a name to identify the data source.
- 5. In step 2, click a driver for the type of database you are using as your data source.

If you don't see the database you want, you may need to install the ODBC driver or data source driver for your data source.

How?

6. Click **Connect**, and then provide the information needed to connect to your database.

For relational databases, text file databases, and Excel lists, the information you provide depends on the type of database selected. You may be asked to supply a logon name, a password, the version of the database you're using, the database location, or other information specific to the type of database.

For Microsoft SQL Server OLAP Services data sources, <u>follow</u> <u>the steps in the Multi-Dimensional Connection Wizard</u>.

- 7. After you enter the required information, click **OK** or **Finish** to return to the **Create New Data Source** dialog box.
- 8. If your database has tables and you want a particular one to be

displayed automatically in the Query Wizard, click the box for step 4, and then click the table you want.

9. If you don't want to type your logon name and password when you use the data source, select the Save my user ID and password in the data source definition check box. If the check box is unavailable, see the database administrator to determine whether this option has been disabled.

When you complete these steps, the name of your data source appears in the **Choose Data Source** dialog box.
Set up a data source that uses the Microsoft Access driver

- 1. If the **Create New Data Source** dialog box is not displayed, click **New** on the **File** menu, click the **Databases** tab, and then double-click **New Data Source**. Type a name for the data source in step 1, and select **Microsoft Access Driver** in step 2.
- 2. In step 3, click **Connect**.
- 3. Under **Database**, click **Select**, and use the **Select Database** dialog box to locate your database file.

If you want Microsoft Query to open the database as read-only, select the **Read Only** check box.

If you want Microsoft Query to be the only user of the database when you use this data source, select the **Exclusive** check box.

4. Click OK.

If your database requires you to set additional connection parameters, click **Advanced**. For information about whether you need to change any of these settings, see the administrator for your database. For information about the advanced settings, click **Cancel**, and then click **Help** in the **ODBC Microsoft Access Setup** dialog box.

- If you want Microsoft Query to access the system database, click Database under System database, and then click System Database. In the Select System Database dialog box, locate your system database file, and then click OK.
- 6. Click **OK**, and enter your logon name and password under **Authorization**. If you don't use a logon name or password to gain access to the database, leave the boxes under **Authorization** blank.

Note You can use the **ODBC Microsoft Access Setup** dialog box to create, repair, and compact Microsoft Access databases. If you are not familiar with these operations, you should perform them only from within Microsoft Access. For more information, see Microsoft Access Help.

Set up a data source that uses the Microsoft dBASE or FoxPro driver

Before proceeding with these steps, it's a good idea to place all of the database files that you want to use together in the same folder, so that Query can find all of your files and you don't have to type the full path whenever you want to add a table to a query. If you want, you can place the database files in one folder and the index files in another folder.

- If the Create New Data Source dialog box is not displayed, click New on the File menu, click the Databases tab, and then double-click New Data Source. Type a name for the data source in step 1 and select Microsoft dBase Driver or Microsoft Visual FoxPro Driver in step 2.
- 2. In step 3, click **Connect**.
- 3. Do one of the following:

For the Microsoft dBase Driver, click the version of dBASE or FoxPro that was used to create your database files in the **Version** list. If the database files are not in the current folder, clear the **Use current directory** check box, and then click **Select Directory** to locate the folder. If the index files for the database are also in a different folder, click **Select Indexes** to locate the folder.

For the Microsoft Visual FoxPro Driver, click the option for your FoxPro database under **Database type**, and then enter the path to the database in the **Path** box.

Set up a data source that uses the Microsoft Excel driver

By using the Microsoft Excel ODBC driver, you can access data only from workbooks in the same folder. Before proceeding with the following steps, copy all workbooks from which you want to access data simultaneously into one folder, and name each worksheet range that you want Microsoft Query to recognize as a table.

You must set up a separate data source for each folder from which you want to access data.

Before you set up a data source or run a query to access data in an Excel workbook, close the workbook in Excel.

- 1. If the **Create New Data Source** dialog box is not displayed, click **New** on the **File** menu, click the **Databases** tab, and then double-click **New Data Source**. Type a name for the data source in step 1, and select **Microsoft Excel Driver** in step 2.
- 2. In step 3, click **Connect**.
- In the Version box, click the Excel version that was used to create the workbook. If your workbook was created with Excel 97, Excel 2000, or a later version of Excel, click Excel 97-2000. If your workbook was created with Excel 95 (Excel version 7.0), click Excel 5.0/7.0.
- 4. Click **Select Workbook**, and then locate the workbook file.

Set up a data source that uses the Microsoft ODBC for Oracle driver

By using the Microsoft ODBC for Oracle driver, you can connect to Oracle databases that support SQL*Net 2.3 or later. For information about this support, see your Oracle documentation.

Before you can set up data sources that use this driver, you must install Oracle client and networking components version 7.3 or later. These components are available from Oracle or your Oracle vendor.

- 1. If the **Create New Data Source** dialog box is not displayed, click **New** on the **File** menu, click the **Databases** tab, and then double-click **New Data Source**. Type a name for the data source in step 1, and select **Microsoft ODBC for Oracle Driver** in step 2.
- 2. In step 3, click **Connect**.
- 3. Under Database, click Select.
- 4. In the **User name** box, enter your database logon name.
- 5. Enter the name of your Oracle server and any additional information necessary to connect to the server. See your Oracle database administrator for this information. For information about setup options, click **Help** in the setup dialog box.
- 6. Enter your logon ID and password, and then click **OK** twice.

Set up a data source that uses the Microsoft Paradox driver

Before proceeding with these steps, it's a good idea to place all of the database files that you want to use together in the same folder, so that Query can find all of your files and you don't have to type the full path whenever you want to add a table to a query.

- 1. If the **Create New Data Source** dialog box is not displayed, click **New** on the **File** menu, click the **Databases** tab, and then double-click **New Data Source**. Type a name for the data source in step 1 and select **Microsoft Paradox Driver** in step 2.
- 2. In step 3, click **Connect**.
- 3. In the **Version** box, select the version of Paradox that was used to create your database files.
- 4. If the database files are not in the current folder, clear the **Use current directory** check box, and then click **Select Directory** to locate the files.
- If your database requires you to set advanced options, click Options. For information about whether you need to change any of these settings, see the administrator for your database. For information about the option settings, click Help on the ODBC Paradox Setup dialog box.

Set up a data source that uses the Microsoft SQL Server OLAP Services driver

You can use this driver to set up data sources for both server databases and offline cube files (*.cub) that were created from Microsoft SQL Server OLAP Services databases with the Offline Cube Wizard in Microsoft Excel or that were created from relational database records with the OLAP Cube Wizard in Microsoft Query. Learn about <u>OLAP data sources</u>.

Note This topic explains how to set up data sources that use the OLAP provider for the first version of Microsoft SQL Server OLAP Services (version 7.0 of MSMDCUBE.DLL). To set up data sources for later versions of this provider, you can use the Data Connection Wizard instead of Query. For additional information about setting up these data sources, see the Help for your version of Microsoft SQL Server OLAP Services.

1. If the **Create New Data Source** dialog box is not displayed, click **New** on the **File** menu, click the **OLAP Cubes** tab, and then double-click **New Data Source**. Type a name for the data source in step 1 and select **Microsoft OLE DB Provider for OLAP Services** in step 2.

The **OLAP Cubes** tab is available only when you run Query from Excel.

- 2. In step 3 of the **Create New Data Source** dialog box, click **Connect**.
- 3. In step 1 of the **Multi-Dimensional Connection** dialog box, do one of the following:

To connect to a server database, click **Analysis server**, and then supply the server name, logon ID, and password for your server. For information about these and other settings for your database, see your server administrator.

To connect to an offline cube file, click **Cube file**, and enter the path to the .cub file in the **File** box.

4. Do one of the following:

If you're connecting to an offline cube file, click **Finish**.

If you're connecting to a server database, click **Next**, select the server database, and then click **Finish**. In step 4 of the **Create New Data Source** dialog box, select the name of the database cube that you want to retrieve data from.

Set up a data source that uses the Microsoft text driver

You'll need to set up a separate data source for each folder that contains text files from which you want to access data. If you want to use the data source to access data simultaneously from more than one text file, copy all of the text files to the same folder before proceeding with these steps.

- 1. If the **Create New Data Source** dialog box is not displayed, click **New** on the **File** menu, click the **Databases** tab, and then double-click **New Data Source**. Type a name for the data source in step 1 and select **Microsoft Text Driver** in step 2.
- 2. In step 3, click **Connect**.
- 3. If the text files are not in the current directory, clear the **Use current directory** check box, and then click **Select Directory** to locate the folder.

Note You can gain access to data in text files in several formats, including columns separated by blank space or by commas or tab characters. For more information about the text file formats that you can use, click **Help** in the **ODBC Text Setup** dialog box.

Set up a data source that uses the Microsoft SQL Server driver

- 1. If the **Create New Data Source** dialog box is not displayed, click **New** on the **File** menu, click the **Databases** tab, and then double-click **New Data Source**. Type a name for the data source in step 1 and select **SQL Server** in step 2.
- 2. In step 3, click **Connect**.
- 3. In the **Server** box, enter the name of the SQL Server on your network. If you want information about the selections available for **Server**, click **Help**.
- 4. Click **Options**, and set any other options required for your database. For information about whether you need to change any of these settings, see the administrator for your database. For information about the option settings, click **Help**.
- 5. Enter your logon ID and password, and then click **OK** twice.

About creating a new query

A query is a means of finding the records that answer a particular question you ask about the data stored in a database. When you create a query, you can either use the Query Wizard or work directly in Microsoft Query. How you create your query depends on the features you need to select the data you want.

When to use the Query Wizard Use the Query Wizard when you want to create a simple query. By using the Query Wizard, you can select the tables and fields you want to include. An inner join is created automatically when the wizard recognizes a primary key field in one table and a field with the same name in the other table.

You can also use the wizard to sort the result set and do simple filtering. In the final step of the wizard, you can choose to return the data to Microsoft Excel, further refine the query in Query, or create an OLAP cube from the data. After you create the query, you can run it in either Excel or Query. <u>How to use the Query Wizard</u>.

When to use Microsoft Query Use Query when you want to create a more complex query. This method gives you the most control because you can narrow your result set either by creating an inner join or outer join or by using expressions and complex criteria. Expressions are calculations you can make by using operators, functions, field names, and constants. You can do more complex filtering if you use criteria.

In Query, you can view the result set, and when you're sure it contains the data you want, you can return the data to Excel. You can also create OLAP cubes in Query. After you create the query, you can run it in either Excel or Query. <u>How to create a query in Microsoft Query</u>.

When to use a parameter query Use a parameter query when you want to create one query to retrieve data that has different criteria. For example, to retrieve sales data for one product at a time, you can use the product name as the criteria. Parameter queries are also useful for retrieving part of the data from a database that is too large for your system's resources to retrieve all of the data at one time.

A parameter query prompts for criteria when you run it. You can create a

parameter query only in Query, not in the Query Wizard. The criteria you enter are used to retrieve data from the tables in your query. After you create the query, you can run it in either Excel or Query. <u>How to create a parameter query</u>.

Create a new query

What do you want to do?

Create a query by using the Query Wizard

Create a more complex query in Microsoft Query

Create a parameter query

Not sure which choice you want?

Create a parameter query

1. Create a query that includes the tables and fields for the records you want.

How?

- 2. Make sure **Auto Query** 🔃 is not pressed in.
- 3. Make sure **Show/Hide Criteria** is pressed in to display the Criteria pane.
- 4. Click the first cell in the **Criteria Field** row, and then click the arrow in the cell to select a field from the list that you want to use as the parameter for the query.
- 5. Click the first cell in the **Value** row. Type **[** (a left square bracket) followed by the text for the prompt you want Microsoft Query to display when the query is run, and then type **]** (a right square bracket). An example is shown in the following illustration.

	(A	
Criteria Field:	CITY	
Value:	[Type the name of the city]	E

The text of the prompt must be different from the field name, although it can include the field name.

To create more than one parameter for the query, repeat steps 4 and 5 for each parameter you want to add in the next available column in the Criteria pane.

For more information about criteria you can use to refine a query, click .

- 6. Press ENTER.
- 7. To run the query, click **Query Now !**.

Note When you finish retrieving the data, you can return the result set to Microsoft Excel. For more information, click \blacksquare .

Open a query

1. Click **Open Query** , click the query file you want to open, and then click **Open**.

A query file has a .dqy or .qry file name extension.

- 2. If prompted, enter your logon name, password, and any other information required for your data source.
- 3. If prompted, enter parameter values for the query, and then click **OK**. If you do not know what the valid parameters are, contact the person who created the query.

Note You can also open an .oqy cube definition file that was created by using the OLAP Cube Wizard. Opening this type of file starts the wizard so that you can change the cube definition. Learn about <u>creating cubes</u> by using the OLAP Cube Wizard.

Save a query

Save a query when you want to preserve formatting or want to use the same query later or in a different workbook. For information about opening query files in Microsoft Excel, see Help in Excel. You don't have to save a query to return data to Excel.

- 1. Click Save File **E**.
- 2. In the **File name** box, type the name you want.
- 3. In the **Save as type** list, Microsoft Query automatically selects **Query Files (*.dqy)**. If you want to use the query in a version of Query earlier than Microsoft Query 97, click **Query Files (*.qry)**.

A parameter query can be saved only in *.dqy format and can't be used in a version of Query earlier than Microsoft Query 97.

- 4. In the **Save in** list, click the folder where you want to store the query.
- 5. Click Save.

Notes

When you save a query file, Query does not save the current result set displayed in the Data pane. The result set is automatically refreshed, before the **Save As** dialog box is displayed, even when the **Auto Query** button is not pressed in.

If you want to prevent Query from running the query before you save the query or return the data to Excel, click **Options** on the **Edit** menu, and then clear the **Validate queries before saving or returning data** check box.

The default location for query files is your user profile folder in Application Data\Microsoft\Queries. When you save a query in this location, its file name appears on the **Queries** tab when you open a new query in Microsoft Query or when you use the **Run Saved Query** command (**Data** menu, **Get External Data** submenu) in Excel.

Run, view, or edit an SQL command, procedure, or statement

If you know Structured Query Language (SQL), you can use SQL to edit a query's SQL SELECT statement. For more information about SQL, see the documentation for your database.

- Create a new query. Do not return the results to Microsoft Excel. <u>How</u>?
- 2. Click View SQL ^{SQL}.

Microsoft Query displays the **SQL** dialog box with the SQL SELECT statement that corresponds to the query.

3. Type any changes you want to make into the SQL statement, and then click **OK**.

To add a new line in the **SQL statement** box, press ENTER.

Notes

To change or replace the SQL statement, use the same SQL syntax that Microsoft Query uses or the syntax your data source supports. Microsoft Query tries to interpret the syntax you use and displays both the query's design and its result set.

If Microsoft Query can't display the query's design because it can't interpret a portion of the syntax you used, the result set is still displayed. However, you can change only the result set's formatting, such as the display font or column width. You can't edit data, and you must make any change to the query definition, such as adding or removing fields, in the **SQL** dialog box.

Run a query automatically, or wait to run the query at a later time

Microsoft Query automatically runs a query and refreshes the data in the result set each time you add a field to the Data pane or add or change criteria in the Criteria pane. If your query contains a large number of records, you might want to wait to run the query until you've finished making changes to the query.

To run a query automatically each time you add new criteria or a field, make sure that the **Auto Query** 🔃 button is pressed in.

To wait to run a query until after you've finished adding new criteria or fields, make sure that the **Auto Query** 💮 button is not pressed in. When you're ready to run the query, click **Query Now** .

Return a result set to Microsoft Excel

- 1. If the **Returning External Data to Microsoft Excel** dialog box is not displayed, click **Return Data P**, and if prompted, enter any parameter values required by the query.
- 2. Click **Properties** to control formatting and layout options for how the external data range is brought into Microsoft Excel. For more information, see Microsoft Excel Help.
- 3. To put the result set on the selected worksheet, click **Existing worksheet**. On your worksheet, click the cell where you want to place the upper-left corner of the result set, and then click **OK**.

To put the result set on a new worksheet, click **New worksheet**, and then click **OK**. Microsoft Excel adds a new worksheet to your workbook and automatically puts the result set in the upper-left corner of the new worksheet.

To use the external data in a PivotTable report, click **PivotTable report**, and then click **OK**.

Notes

The result set is automatically refreshed in Microsoft Query, before the **Returning External Data to Microsoft Excel** dialog box is displayed, even when the **Auto Query** 1 button is not pressed in. If you want to prevent Query from running the query before you return the data to Excel, click **Options** on the **Edit** menu in Query, and then clear the **Validate queries before saving or returning data** check box.

You can check the status of a query that's taking a long time to return data. In Excel, click **Refresh Status** (1) on the **External Data** toolbar. To cancel returning data, click **Stop Refresh**.

Execute an SQL statement or procedure

If you know Structured Query Language (SQL), you can use SQL to execute an SQL SELECT statement. For more information about SQL, see the documentation for your database.

1. On the File menu, click Execute SQL.

To execute an SQL statement on a table in another data source, click **Data Sources**, click the data source you want, and then click **OK**. You can specify a new data source.

How?

2. In the **SQL statement** box, type the SQL statement you want to run.

If you have an existing SQL statement you want to run, click **Open**, click the SQL file, and then click **Open**.

To run an SQL procedure stored in your external database, click **Procedures**, click the procedure you want in the **Query** box, and then click **Paste** to copy the procedure to the **SQL statement box**.

3. Click Execute.

Microsoft Query runs the SQL statement. If the query is taking a long time to return data, you can cancel the query by pressing ESC.

To save the SQL statement, click **Save**.

Notes

To change or replace the SQL statement, use the same SQL syntax that Microsoft Query uses or the syntax your data source supports. Microsoft Query tries to interpret the syntax you use and displays both the query's design and its result set.

If Microsoft Query can't display the query's design because it can't interpret a portion of the syntax you used, the result set is still displayed.

However, you can change only the result set's formatting, such as the display font or column width. You can't edit data, and you must make any change to the query definition, such as adding or removing fields, in the **SQL** dialog box.

If the query returns more than one result set, Microsoft Query displays the first one in the Data pane. Any other result sets are not displayed.

If the query runs successfully but does not return a result set, Microsoft Query displays a message that the statement was run successfully. If the query affected rows in the underlying tables, the message also specifies how many rows were affected.

If the query didn't run successfully, Microsoft Query displays the SQL error message that corresponds to the problem. To display the **Execute SQL** dialog box and correct the statement, click **OK**.
About OLAP queries

Microsoft Query allows you to create and work with data from On-Line Analytical Processing (OLAP) databases by setting up data sources, creating OLAP cubes from relational data, and saving OLAP cube definition files. This topic explains how to use Microsoft Query to prepare OLAP data for use in Microsoft Excel PivotTable reports and PivotChart reports.

This topic provides information about the following:

What is OLAP?

Connecting to OLAP databases

Creating an OLAP cube from records in a query

Saving OLAP query files

What is OLAP?

OLAP is a way to organize large business databases. OLAP data is organized by a database administrator to fit the way users analyze and manage it, so less time and effort is needed to create your reports.

Levels of detail OLAP databases organize data by level of detail, using the same kinds of categories that you use when you analyze the data. For example, a database of worldwide sales information for a company's products might have separate fields identifying the country, region, city, and site of each sale. In an OLAP database, this geographical information could be organized from lowest to highest level of detail, like this:

Levels in a geography dimension

This organization makes it easy both to display high-level sales summaries, such as sales totals across an entire country or region, and to display the details for sites where sales are particularly strong or weak. In a Microsoft Excel PivotTable or PivotChart report, you can display summarized data at different levels and, when you view the levels that contain greater detail, display only the items of interest.

Dimensions and cubes A set of levels that encompasses one aspect of the data, such as geographic areas, is called a dimension. Similarly, information about when sales were made can be organized in a time dimension with levels for year, quarter, month, and day. OLAP databases are called cubes because they combine several dimensions, such as time, geography, and products, with summarized data, such as sales or inventory figures. An OLAP database calculates a summarized value for every intersection of every level of every dimension, such as the total sales for each day, month, and year of each site, region, and country and each product and product line.

Connecting to OLAP databases

OLAP providers To connect to an OLAP database, you need an OLAP provider that supports that particular type of OLAP database. Microsoft Office includes OLAP provider software for the Microsoft OLAP server product, Microsoft SQL Server OLAP Services. For other products that provide OLAP data and services, you need to install additional drivers and client software. For information about using a third-party OLAP provider, consult your system administrator or the vendor for your OLAP product. Learn about <u>installing OLAP providers</u>.

Data sources OLAP client software includes a data source driver for your OLAP database. Office includes a driver, named **Microsoft OLE DB Provider for OLAP Services**, to access databases that were created by using Microsoft SQL Server OLAP Services. When you create cubes or offline cube files from within Microsoft Query, you also create data sources to access these cubes.

When you run Microsoft Query to connect to an OLAP database or offline cube file, you set up a new data source just as you do for any other kind of data, except that you use the **OLAP Cubes** tab in the **Choose Data Source** dialog box to name the data source, specify the data source driver for your OLAP provider, and connect to the database. <u>How to set up a data source</u>.

Returning data to Excel After you set up an OLAP data source and select it for use, you don't create a query to select tables and fields as you do for other kinds of data. The data source gives you access to all data in the OLAP cube. The data fields are immediately returned to Excel as a PivotTable report.

Creating an OLAP cube from records in a query

By using Microsoft Query and the OLAP client software that's provided with Microsoft Office, you can create your own OLAP cube or offline cube file from data that you query from a relational database, such as Microsoft SQL Server. Creating an OLAP cube can make accessing large amounts of relational data more efficient and can help organize the data for use in reports. You do not need an OLAP server product to create and use this type of OLAP cube.

Retrieving the data for the cube To create your own cube, you first <u>set</u> <u>up a data source</u> for the relational database, <u>create a query</u> to retrieve the data, and then run the OLAP Cube Wizard to define a cube structure for the queried data and, optionally, save an offline cube file.

Creating a cube that's built in memory In the OLAP Cube Wizard, you can create a temporary cube that's rebuilt in memory whenever you open a report that's based on the cube. See Help in the OLAP Cube Wizard for information about when to create this type of cube. Learn about <u>running the wizard to create an OLAP cube</u>.

Creating an offline cube file The OLAP Cube Wizard also lets you create an offline cube (.cub) file that you can store on your local disk or a network share. When stored on your local disk, you can use a .cub file to continue working with the data when you're disconnected from the network or the server is unavailable. If you store the file on a network share, you and other users can set up data sources to access the offline cube file and base reports on it. Learn about <u>running the wizard to create an offline cube file</u>.

Saving OLAP query files

OLAP data sources When you use the **OLAP Cubes** tab of the **Choose Data Source** dialog box to set up a data source, Microsoft Query saves an .oqy file in your user profile folder under Application Data\Microsoft\Queries. In addition to specifying the data source when you create a PivotTable or PivotChart report in Excel, you can open the .oqy file directly from Excel to create a PivotTable report. If the .oqy file connects to an OLAP server database or to an offline cube file on a network share, you can make the .oqy file available to other users, who can then open it to create reports without having to set up their own data sources.

OLAP Cube Wizard cubes After the final step in the OLAP Cube Wizard, you are prompted to save a cube definition file with the extension .oqy. This file contains the information that Excel needs to either build the cube in memory or connect to an offline cube file, if you created one. Instead of returning the records in your query to Microsoft Excel, you can open this file in Microsoft Excel to create a PivotTable report. You do not have to set up a separate data source for the OLAP cube, although you can <u>set up a separate data source</u> for an offline cube file if you want to connect directly to the cube file in order to create reports.

Saving the .oqy file is not the same as saving a copy of the query for the relational database records that you used to create the OLAP cube. If you want to keep a copy of that query so that you can add data to your cube later on, you must <u>save the query in a database query (.dqy) file</u>.

About creating an OLAP cube from relational data

This topic explains how you can create an OLAP cube from a query that retrieves data from a relational database, such as Microsoft Access or Microsoft SQL Server. You do not need an OLAP server product to create and use this type of OLAP cube. To understand and use this capability of Microsoft Query, you should be familiar with managing databases and using Microsoft Query to retrieve data for Microsoft Excel PivotTable reports or PivotChart reports.

This topic provides reference information about the following:

What is an OLAP cube?

Creating a query for the data

Creating a cube

Refreshing and changing a cube

What is an OLAP cube?

On-Line Analytical Processing (OLAP) is a way of organizing data to fit the way you analyze and manage it, so that less time and effort is needed to create your reports. When you create an OLAP cube from a query, you turn the flat set of records into a structured hierarchy, or cube, that allows reports to focus on the desired level of detail. You also predefine the summary values for the reports, which speeds up report calculation.

Create an OLAP cube when you want to provide a large amount of external data for use in Microsoft Excel reports, especially if you have more records to analyze than you can return to Excel without running out of system resources. Because a cube allows Excel to retrieve only the data it needs for the current view of the report, creating and updating a report is usually faster for large amounts of data than if you retrieved the entire set of records from the database.

The data from your OLAP cube can be returned to Excel only as a PivotTable or PivotChart report, unlike the relational database records in your query, which can also be returned to Excel as an external data range on a worksheet.

Creating a query for the data

To create an OLAP cube from relational database records, you <u>set up a</u> <u>data source</u> for the relational database and then <u>create a query</u> to retrieve the fields and records to supply the data for the cube.

Selecting fields for the query To ensure that the query runs as efficiently as possible, include only the tables and fields that you need for the cube in the query. However, make sure you <u>add all the fields</u> you want. After you have created a cube, you cannot add more fields to the query and then incorporate the new fields into the cube. To add more fields to a cube, you must completely re-create the cube.

Adding fields more than once Sometimes, you might want to use a field more than once in a cube. For example, you might want to summarize a Sales field in two different ways, by adding up the sales amounts and by counting the number of sales. Or you might want to include two different versions of a Date field: one that organizes the dates by quarter and month, and another that organizes the dates by quarter and week. To use a field more than once in a cube, you must add the field to the query once for each use. For example, add the Sales field to the query twice if you plan to both add up and count the values in the field.

See Help in the OLAP Cube Wizard for specifics of when to include a field in the query more than once. If you find that you need to add fields to the query while running the wizard, you can cancel the wizard, add the fields, and then run the wizard again.

Saving the query If you think you might want to re-create the cube later with different fields, be sure to <u>save the query in a database query (.dqy)</u> <u>file</u>. You can then open the .dqy file, change the query, and create new cubes from the changed query.

Creating a cube

To create a cube, you <u>run the OLAP Cube Wizard</u>. See Help in this wizard for specific instructions for each wizard step.

Temporary cubes and offline cube files The wizard lets you create two types of cubes. One type is a cube definition that the wizard saves in an .oqy file. When you open a report that's based on this type of .oqy file, the cube is built temporarily in memory. The second type is a separate offline cube file that you store on your local disk or a network share. If stored on your local disk, this type of cube allows you to continue working with the data when you are disconnected from the network. You can set up an offline cube file as a data source for reports.

Saving a cube definition file Whether or not you save a separate offline cube file, the wizard saves a cube definition (.oqy) file. This file contains the information that Microsoft Query needs to load the cube so that you can change it, and the information Excel needs to either build the cube in memory or connect to an offline cube file, if you created one.

Using the cube definition file to create reports Instead of returning the records in your relational database query to Microsoft Excel, you can open the .oqy file in Excel to create a PivotTable report that's based on your OLAP cube. You can then save the PivotTable report in a workbook or create PivotChart reports that are based on the PivotTable report. For instructions to open .oqy files and create reports, see Help in Excel.

Refreshing and changing a cube

Refreshing an OLAP cube To refresh a cube, you open a report that's based on the cube in Microsoft Excel and refresh the report. When you do this, new and changed data from the original database is brought into the cube, if the cube is built temporarily in memory. If the cube is an offline cube file, the file is rebuilt with the new data and the existing file is replaced. To create your own cubes and refresh them, you must have an OLAP provider, such as Microsoft SQL Server OLAP Services, that supports these capabilities. See Help in Excel for the procedure to refresh a PivotTable or PivotChart report that's based on an OLAP cube.

Changing an OLAP cube You can <u>change an OLAP cube</u> that you created in the OLAP Cube Wizard to organize the data differently or remove fields that you don't want to use in reports. If you created an offline cube file, you can save the file with your changes.

After you create an OLAP cube, you cannot add more fields to it from the original relational database. You can, however, create a new OLAP cube from the same database query (.dqy) file so that you can use different fields from the original database.

Create an OLAP cube from relational data

The data from your OLAP cube can be returned to Microsoft Excel only as a PivotTable report, not as an external data range on a worksheet.

- Create a query to retrieve the fields you want to use in the cube. How?
- 2. Run the OLAP Cube Wizard in one of the following ways:

If you created the query in the Query Wizard, click **Create an OLAP cube from this query** in the last wizard step, and then click **Finish**.

If you created the query in Microsoft Query, click **Create OLAP Cube** on the **File** menu.

- 3. Follow the steps in the OLAP Cube Wizard. For specific instructions, see Help in the OLAP Cube Wizard.
- 4. When you finish the wizard, Query prompts you to save a cube definition (.oqy) file. Type a name for this file and then click **Save**.

Notes

If you specified an offline cube (.cub) file in the OLAP Cube Wizard, the .cub file is created after you click **Save**. Depending on the amount of data for your cube, saving this file might be time consuming. You cannot do other work in Microsoft Query until the offline cube file creation is complete.

To base additional Microsoft Excel reports on your cube, do not return the data from your query to Excel. Instead, open the .oqy file by using the **Open** command on the Excel **File** menu.

You can also base Excel reports on an offline cube file by <u>setting up a</u> <u>data source</u> for the file and then specifying this data source when you run the Excel PivotTable and PivotChart Wizard.

If you think you might want to add fields to the cube from the source

database in the future, be sure to <u>save a database query (.dqy) file</u> so that you can use the query file to <u>change the cube</u>.

Change content and settings of an OLAP cube

You can change the organization of the fields in a cube and remove fields that you no longer want in the cube, but you cannot add more fields to an existing OLAP cube. To add more fields to a cube, you must create a new cube.

- 1. On the Microsoft Query File menu, click Open.
- 2. In the Files of type list, click OLAP Cubes.
- 3. Locate and double-click the cube definition (.oqy) file for the cube you want to change.
- 4. Follow the steps in the OLAP Cube Wizard to change the content and settings. For specific instructions, see Help in the OLAP Cube Wizard.
- 5. When you finish the wizard, Query prompts you to save the .oqy file. To replace the existing .oqy file, click **Save**. To save a new .oqy file, type a new name and then click **Save**.

Notes

If you specified an offline cube (.cub) file in the OLAP Cube Wizard, the .cub file is created after you click **Save**. An existing .cub file that you are changing must be completely re-created and replaced. Depending on the amount of data for your cube, saving this file might be time consuming. You cannot do other work in Microsoft Query until the .cub file creation is complete.

You can base Microsoft Excel reports on a .cub file either by opening the .oqy file by using the **Open** command on the Excel **File** menu, or by <u>setting up a data source</u> for the .cub file and then specifying this data source when you run the Excel PivotTable and PivotChart Wizard.

Delete an offline cube file

Deleting an offline cube (.cub) file removes it permanently from the disk where it was stored, and frees up the often substantial disk space that these files use. However, you can no longer change reports that were based on the file. If you or other users created data sources that used the file, you can no longer use these data sources.

In Microsoft Windows, locate and delete the .cub file.

Add a table to a query

- 1. Click Add Table(s) 🛅.
- 2. Select the table you want to add.
- 3. Click Add.

Note You can add as many tables as you want to the query. When you finish, click **Close**.

Delete a table from a query

- 1. Click the table you want to delete.
- 2. On the **Edit** menu, click **Delete**.

Add or remove a field

A field is a category of information, such as a product name or order amount, that's stored in a table of information in a database. When you add or remove a field, your query includes or excludes the information from that field. The underlying database that supplies the information is not changed.

Add a field To add a single field to the result set, double-click the field you want to add in the Table pane.



Remove a field To remove a field from the result set, click the column heading of the field you want to delete in the Data pane, and then click **Remove Column** on the **Records** menu.

Tips

To add several fields at one time from the same table, hold down CTRL while you click each field, and then drag the selected fields to the Data pane.

To add all of the fields from a table to the result set, double-click the * (asterisk) at the top of the table in the Table pane.



Double-click the asterisk to add all the fields in the table.

Retrieve related records from multiple tables with joins

This topic provides information about the following:

Joining tables

Inner joins

Outer joins

Self joins

Full outer joins

Subtract joins

Joining tables

Databases consist of multiple tables of related information. When you add two (or more) tables to a query, Microsoft Query automatically creates a "join" between the tables when it recognizes a primary key field in one table and a field with the same name in the other table. Join lines show the relationships of the data in the tables.



Join lines connect the Products table to the Order Details table by joining the ProductID field in each table. The Order Details table and the Orders table are also joined by their OrderID fields.

There are two basic types of joins that can be used to retrieve records from multiple tables. An inner join retrieves only related records from the tables. An outer join retrieves all of the records from one table and related records from another table.

In addition to inner and outer joins, there are three other types of joins you can create — self, full outer, and subtract joins. A self-join returns a result set that combines information contained in one table. A full outer join returns all of the records from both tables, even if the records are not related. A subtract join returns only those records from the first table that do not match records in a second table.

Inner joins

An inner join — the default type of join — retrieves only records that have the same value in the joined fields in both tables. Microsoft Query combines the matching records from each table and displays them in the result set.

Custom	ег	Orders				
ContactName CustomerID		CustomerID	OrderID			
Ernst Handel	ERNSH	ERNSH	11072			
Familia Arquibaldo	FAMIA	FISSA	10184			
Folies gourmandes	FOLIG ·	FOLIG	10016			
ContactName	CustomerID	When you use an OrderID	inner join			
ContactName Ernst Handel	CustomerID ERNSH	When you use an OrderID 11072	inner join			

More information

Join tables in a query

Change the type of join between tables in a query

Outer joins

An outer join retrieves all of the records from one table and any matching records from another table. When values in the two tables are the same, Microsoft Query combines the matching records and displays them as one record in the result set. When a record from the table that's contributing all of its records can't be matched with a record from the other table, the record still appears in the result set. However, Query displays empty cells where no matching record was present in the other table.

	Custon	ner		_					
	ContactName	Custo	omerID			Order	s		
	Ottilies Käseladen	OTTIK			Cus	tomerID	Ord	erID	
гł	Paris spécialités	PARIS	,		OTTIŀ	< · · · · · · · · · · · · · · · · · · ·	1	1020	
	Piccolo und mehr	PICCO			PICCO	D	1	1015	
E ri	ven though this ecord doesn't matc	h							
а	ny entries in the		Conta	ctNarr	ne	Custom	erID	Orde	erID
lС	rders table		Ottilies K	(äsela	den.			1	1020
	using an outer joir	n still –	Paris spe	écialite	és	PARIS			8
ind	cludes it in the resul	t set.	Piccolo u	Ind me	et ir	PICCO		1	1015

More information

Join tables in a query

Change the type of join between tables in a query

Self joins

When you want to combine information in one table, use a self-join. For example, in a table of employees, each record includes the employee ID for the manager that an employee reports to. Because the manager is also in the same table as the employee, you can join the table to itself to return both the name and the employee ID of the manager.

Err Photo Postal Regio	ployees Code n		E Cor Em Ext	imployees_1 untry ployeelD ension
Repor Title TitleOf	tsTo Courtesy		Firs Hire Ho	:tName eDate mePhone ▼
Las	tName	Reports	sTo	LastName
Buc	hanan	2		Fuller
Cal	ahan	2		Fuller
Dav	olio	2		Fuller
Doc	Isworth	5		Buchanan
King	3	5		Buchanan
Lev	erling	2		Fuller
Pea	icock 🗌	2		Fuller
► Suv	ama	5		Buchanan

By joining the table to itself, you can retrieve both the ID and the name of the manager.

More information

Retrieve multiple fields with a self-join

Full outer joins

When you want to combine all of the data from two tables, you can create a full outer join query. This type of query returns all of the data from both tables, even when the data is not related.



For example, if you have a Customers table and an Orders table, you can use a full outer join to return all of the orders that are pending (from the Orders table) and all of the customers who have not placed an order (from the Customers table).

OrderID	ContactName
	Diego Roel
	Marie Bertrand
10248	Paul Henriot
10249	Karin Josephs
10250	Mario Pontes
10251	Mary Saveley
10252	Pascale Cartrain
10253	Mario Pontes
10254	Yang Wang
10255	Michael Holz
10256	Paula Parente

The first two names in this list have not placed orders.

Note When you create a full outer join, Query displays only the result set of the query, not the tables used within the query.

More information

Combine all the data from two tables

Subtract joins

When you have two tables that have similar data, such as two tables of customers, and each table has some information that is not included in the other table, you can create a subtract join. A subtract join returns only the records from the first table that do not match the records from the second table.

City Con Cor	Customers and Suppl npanyName ntactName	iers by City		Customers Customers Address City CompanyName ContactName ContactTitle
	ContactName			
Sv	en Ottlieb			ContactName
Pa	ula Wilson		N	1aria Anders
Re	ne Phillips		Α	kna Trujillo
Re	gina Murphy		A	Intonio Moreno
Elia	ane Noz		Т	'homas Hardy
				N S C D I I

Two tables with similar customer data

Customers * City CompanyNa ContactNar Relationship	and Suppliers ame ne o	by City	4	Customers Address City CompanyName ContactName ContactTitle
Criteria Field: Value: or:	CompanyNar Is Null	ne		
Conta	actName	Comp	any	Name
📃 Regina Mu				
Eliane Noz	2			
Cheryl Say	lor			
- B-LD-1-3+6-1				

More information

Retrieve records that two tables do not have in common

Retrieve records from multiple tables

If your query contains more than one table, you can retrieve related records from multiple tables by using joins. A join is a link between two tables where a field in one table contains the same data as a field in another table.

What do you want to do?

Learn about retrieving records with inner and outer joins

Create an inner join

Create an outer join

Create a self join

Retrieve all of the records from two tables

Retrieve only the records that two tables don't have in common
Combine all the data from two tables

If you want to combine all of the data from two tables that have similar but not identical data — such as two lists (tables) of customers that each have different information (fields) for each customer, and each list has some customers that aren't included on the other list — you can create a query that includes all of the customers. This type of query is called a full outer join. Learn about full outer joins.

- 1. In Microsoft Query, create a query and add both tables to it.
- 2. Join a field that matches between the two tables (for example, the Customer ID field in two customer lists) by dragging the field in the first table to the second table.

Choose a field that has information for every record in each table. For example, if two customer lists have matching Customer ID fields, join the Customer ID field.

- 3. Double-click the join line between the two tables.
- 4. In the **Joins** dialog box, click option **2**, click **Add**, and then click **Close**.
- 5. Add the fields that you want to the query.
- 6. On the View menu, click SQL.
- 7. Select the data in the **SQL statement** box, press CTRL+C to copy it, and then click **OK**.
- 8. Double-click the join line again.
- 9. In the **Joins** dialog box, click option **3**, click **Add**, and then click **Close**.
- 10. On the View menu, click SQL.
- 11. Click at the end of the data in the **SQL statement** box, press ENTER, type the capitalized word **UNION** and then press ENTER again.
- 12. Press CTRL+V to paste the copied information after the word

UNION.

13. Click **OK**, and when prompted that the query can't be displayed graphically, click **OK** again.

Retrieve records that two tables don't have in common

If you have two tables that have similar data — such as two lists (tables) of customers, and each list has some customers that aren't included on the other list — you can create a query that returns only the records from the first table that do not match the records in the second table. This type of query is called a subtract query or subtract join.

- 1. In Microsoft Query, create a query and add both tables to it.
- 2. Join the fields that match between the two tables (for example, the Customer ID field from two customer lists) by dragging the field in the first table to the second table.
- 3. Double-click the join line between the two tables.
- 4. In the **Joins** dialog box, click option **2**, click **Add**, and then click **Close**.
- 5. Add the joined field to the query.
- Add a field other than the joined field from the second table. Select a field that has data for every record in the second table. For example, if you joined the Customer ID fields in the two tables, you might add the Customer Name field from the second table.
- 7. On the View menu, click Criteria.
- 8. In the first criteria field, click the field you added in step 6.
- 9. In the first **Value** field, type **Is Null**.
- 10. To see the records in the first table that aren't included in the second table, click **Query Now !**.
- 11. Add any additional fields that you want in the query, and then return the data to Microsoft Excel.

Note If you also want to see the records in the second table that don't match records in the first table, create a second query in which you follow these same steps, but reverse the order of the tables (make the second

table the first table, and vice versa).

Create an outer join

An outer join displays all of the records from one table and any records from related records from another table.

- 1. Create an inner join between the tables. <u>How to create an inner</u> join.
- 2. Double-click the join line between the tables.
- 3. Click = (equal sign) in the **Operator** list.
- 4. Do one of the following:

To create an outer join that uses all of the records from the table on the left, click option **2** under **Join Includes**.

To create an outer join that uses all of the records from the table on the right, click option **3** under **Join Includes**.

Notes

Retrieve all records from two tables If you want to combine all of the records from two tables, whether or not the records are related, you can <u>create a full outer join</u>.

Retrieve only the records that two tables don't have in common Instead of retrieving related records, you can <u>retrieve all records that two</u> <u>tables don't have in common</u>.

Create a self-join

Use this procedure to combine information from two copies of one table — for example, when a table of employee information includes the employee ID of each employee's manager and you want to combine the ID number with the name of the manager. This procedure creates a type of query called a self-join. Learn about self-joins.

- 1. Create a query in Microsoft Query, and <u>add the table</u> that you want to join to itself to the query.
- 2. <u>Add the same table</u> to the query a second time.
- 3. When prompted that the table is already in the query, click **OK**.
- 4. If Query automatically joins the two tables, double-click the join line, click **Remove**, and then click **Close**.
- 5. <u>Create an inner join</u> between the two tables. For example, if the Manager field uses the employee ID to identify the managers, you can join the Manager field in the first table to the Employee ID field in the second table.
- 6. <u>Add the fields</u> you want to your query. For example, you can add the Employee Name and Manager fields from the first table, and then add the Employee Name field from the second table to see the name of each employee's manager.

Operators in joins

You can use the following operators in joins.

Operator	Description
= (Equals)	Selects records that contain equal values in the joined fields.
<> (Does not equal)	Selects records that contain values that are not equal in the joined fields.
> (Is greater than)	Selects records from the left table that contain values that are greater than values in records from the right table in the joined fields.
>= (Is greater than or equal to)	Selects records from the left table that contain values that are greater than or equal to values in records from the right table in the joined fields.
< (Is less than)	Selects records from the left table that contain values that are less than values in records from the right table in the joined fields.
<= (Is less than or equal to)	Selects records from the left table that contain values that are less than or equal to values in records from the right table in the joined fields.

Remove a join

If your query contains an inner join or an outer join but doesn't retrieve the records you want, you may want to remove the join.

- 1. In the Table pane, click the join line you want to remove.
- 2. Press DELETE.

Notes

When you remove a join, the records included in the result set of the query will change. Depending on the type of join you remove, records may be added to or removed from the result set.

Deleting a join from a query affects only that query. For example, if Microsoft Query automatically creates an inner join between two tables in your query and you remove the inner join, Microsoft Query will automatically create an inner join between the two tables again if you use them in a different query.

About using criteria

This topic provides information about the following:

What are criteria? Meeting a single condition

Meeting multiple conditions

Finding similar records

Specifying a range of values

Using a calculation

Excluding specific records

Finding incomplete records

What are criteria?

Criteria are conditions you specify to find particular records in a database or list. By using criteria, you limit the records (or rows) that are included in the result set of a query. For example, the criteria **30">UnitPrice > 30** selects records where the value in the UnitPrice field is greater than 30. You add criteria to the Criteria pane in Microsoft Query either by entering the criteria directly in the pane or by using **Add Criteria** on the **Criteria** menu. Fields you use to specify criteria do not have to be included in the result set.

Criteria Field: Value: or:	UnitPrice	 First, select the field that contains the values you want to evaluate. Then type the operator and value you want here.
ProductID	ProductName	
17	Alice Mutton	
60	Camembert Pierrot	
18	Carnarvon Tigers	
38	Côte de Blave	

Because your query may contain multiple tables with similar field names, make sure that you use the field from the correct table. For example, if you want to find all of the suppliers for a particular country, select the Country field from the Supplier table instead of from the Customer table.

Specify what records to retrieve by using criteria

Meeting a single condition

You can find all records with values that meet a single condition — for example, a particular country or product. The following example shows criteria that retrieves all of the records that contain Sweden in the Country field.

Criteria Field:	Country
Value:	Sweden
or:	
ContactName	Country
ContactName Christina Berglund	Country Sweden

To specify a single criteria, select the name of the field in the **Criteria Field** box, and then type the value you want to find in the **Value** box.

Examples of expressions that use logical operators

Meeting multiple conditions

Meet conditions in two or more fields You can find all records with values that meet one condition in one field and another condition in another field. For example, you can find all records for orders placed in the USA in the Washington (WA) region. To select these records, you would type two sets of criteria in the Criteria pane. To specify multiple conditions where all of the conditions are true, you create an AND condition.

Criteria Field:	Country	Region
Value:	USA	WA
or:		
	•	
ContactName	Country	Region
John Steel	USA	WA
John Steel Helvetius Nagy	USA USA	WA WA

To find the records that meet two or more conditions, specify a criteria field for each condition.

Meet multiple conditions in the same field You can find all of the records with values that meet multiple conditions in the same field. For example, you can find all of the records for customers in both the USA and Mexico. To select these records, you would type two sets of criteria in the Criteria pane. To specify multiple conditions where one of the conditions is true, you create an OR condition. In an OR condition, only one of two (or more) conditions must be true.

05			
	Criteria Field:	Country	
	Value:	USA	
	or:	Mexico	
_		•	
T	ContactName	Country	
	Miguel Angel Paolini	Mexico	
	Guillermo Fernánde	Mexico	
	Francisco Chang	Mexico	
	Antonio Moreno	Mexico	
	Ana Trujillo	Mexico	
	Karl Jablonski	USA	
	Helvetius Nagy	USA	
	John Steel	USA	

The criteria above locates all customers from Mexico or the USA.

Note You can also use the **Or** operator within a criteria value. For example, instead of typing **Mexico** in the **or** box you can type **USA Or Mexico** in the **Value** box.

Examples of expressions that use logical operators

Finding similar records

You can find records where the values in one field are similar, like phone numbers that have the same country or area code. For example, you can find all of the customers whose phone numbers begin with the area code (171) or the country codes (1) and (11).

Criteria Field:	Phone
Value:	Like '(1%'
or:	
	•
ContactName	Phone
Thomas Hardy	(171) 555-7788
Victoria Ashworth	(171) 555-1212
Patricio Simpson	(1) 135-5555
Pedro Afonso	(11) 555-7647
Elizabeth Brown	(171) 555-2282
App Devop	(171) 555-0297

To use the **Like** operator, type the word **Like**, and then type the value you want to find within single quotation marks.

Note The percent sign (%) in the example above is a wildcard character that represents any number of characters. You can use an underscore (_) when you want to represent just one character. The wildcard characters that you can use vary, depending on your data source. See the documentation for your data source driver for more information about the wildcard characters that the driver supports.

Examples of expressions that use other types of operators

Specifying a range of values

You can find all of the records that fall within a certain range of values or dates. For example, you can retrieve all records for orders placed between November 1, 1998, and December 31, 1998. To specify two or more conditions that must be met, you create an AND condition. You specify an AND condition to define a range of values.

Criteria Field:	OrderDate	OrderDate
Value:	>10/31/1998	<1/1/1999
or:		
	•	
OrderDate		
UrderDate		
11/16/98		
11/16/98		
11/17/98		
11/18/98		
12/21/98		
12/22/98		

An AND condition that uses two instances of the same field in the Criteria pane.

OrderDate
>10/31/1998 And <1/1/1998
•

An AND condition in which the lower and upper limits are separated by the **And** operator.

Note When you type a date in the Criteria pane, Query converts the date to a format that it understands. For example, when you type the criteria **10/31/1998">>10/31/1998**, Query converts it to **#10/31/98#">>#10/31/98#**.

Examples of expressions that use comparison operators

Using a calculation

You may need to locate records that match the result of a calculation, such as all records where the order total is greater than an amount that is based on the price field multiplied by the quantity field.

Criteria Field: Value: or:	UnitPrice*Quantity >500
UnitPrice	Quantity
42.40	40
42.40	35
64.80	40
27.20	40
14.40	42
16.00	40

To find all of the records where the orders total more than \$500, create a formula to calculate the order total, and then specify the criteria.

Examples of expressions that use arithmetic operators

Excluding specific records

Instead of locating records that match a specific value, you can find records that don't match a value. For example, you can use the **Not** operator to find all of the records for suppliers that are located outside of the United States.

Criteria Field:	Country
Value:	Not USA
or:	
	•
CustomerID	Country
ALFKI	Germany
ANATR	Mexico
ANTON	
ANTON	Mexico
AROUT	Mexico UK

Use the **Not** operator to exclude records.

Examples of expressions that use logical operators

Finding incomplete records

You can find records that contain missing or incomplete values. For example, you can use the **Is Null** operator to find all of the records for employees who are missing a value for their supervisor's name.



Use the Is Null operator to locate records with blank fields.

Examples of expressions that use other types of operators

Limit the records you retrieve by using criteria

You can retrieve specific records by using criteria. Microsoft Query uses your criteria to retrieve the records you want and displays them in the result set. To limit the records you retrieve, you can select a value in the result set and then find other records that match that value. Or, you can create specific criteria in the Criteria pane and then retrieve records that meet only that criteria.

What do you want to do?

Learn about using criteria

Retrieve records that match a specific value

Specify what records to retrieve by using criteria

See examples of expressions you can use as criteria

Specify what records to retrieve by using criteria

1. Click Show/Hide Criteria 🔠.

Microsoft Query displays the Criteria pane.

2. In the Criteria pane, click the first cell of the **Criteria Field** row, and then click the field you want to apply criteria to.

If your query uses multiple tables with similar field names, make sure to click the field you want from the correct table.

- 3. In the **Value** row, click the cell directly under the field you selected in the **Criteria Field** row.
- 4. Type the criteria you want Microsoft Query to use to select records, and then press ENTER.

How?

5. If you haven't already done so, add the fields you want to appear in the result set.

How?

To display records from different tables, make sure that you've added those tables to the Table pane in your query. For more information about adding tables, click >.

Notes

You can further limit your result set by adding additional criteria to the Criteria pane. For example, you can find all the records for suppliers who provide dairy products and who are also from Norway or Italy. To select these records, you would type two sets of criteria in the Criteria pane. First click **Country** in the **Criteria Field** box, and then type **Norway Or Italy** in the **Value** box. Then click **Category** in the next **Criteria Field** box, and type **=Dairy** in the corresponding **Value** box.

If the value you want to use for criteria already appears in the result set, you can copy it to the Criteria pane. In the result set, click the value you want to copy, click **Copy** on the **Edit** menu, click the cell in the **Value** row

where you want to paste the value, and then click **Paste** on the **Edit** menu.

To avoid a long wait while data in each field is being returned, you can wait to retrieve data until after you've added all the fields to your result set. To wait to retrieve data, make sure the **Auto Query** to button is not pressed in. When you're ready to run your query, click **Query Now**.

You can cancel a query that's taking a long time to retrieve data by pressing ESC.
Retrieve records that match a specific value

- 1. Make sure the **Auto Query** 🛄 button is pressed in.
- 2. Add the field for which you want to specify criteria to the Data pane.

How?

For example, if you want to retrieve records that have France in the Country field, add the Country field to the Data pane.

- 3. In the result set, click a cell that contains the value you want to match.
- 4. Click Criteria Equals **v**=.

Examples of expressions

An expression is a calculation that uses a combination of operators, functions, field names, and constants. You can use an expression as criteria to select records. To use an expression in the Criteria pane, you select the field in the Criteria Field that contains the values you want to evaluate, and then you enter an operator and value in the Value box.

Which example do you want to see?

Expressions that use arithmetic operators

Expressions that use comparison operators

Expressions that use logical operators

Expressions that use other types of operators

Not sure which choice you want?

Restore the full result set by removing criteria

If your query contains many records, you can save time by waiting to refresh the result set until you've removed all the criteria. To wait to refresh the result set, make sure the **Auto Query** 🔛 button is not pressed in.

- 1. In the Criteria pane, click the heading of the column that contains the criteria you want to delete.
- 2. Press DELETE.

Repeat steps 1 and 2 for each set of criteria you want to remove.

To remove all criteria at one time, click **Remove All Criteria** on the **Criteria** menu.

Note If the **Auto Query** button is not pressed in, you must click **Query Now** ! to restore the full result set.

Create a calculated field

You can display the results of a calculation you specify by creating a new field called a calculated field. You can create a calculated field by typing your own expression, which is a calculation that uses a combination of operators, functions, field names, and constants. You can also use a built-in function to calculate totals in a field.

What do you want to do?

<u>Create a calculated field by using an expression</u> <u>Create a calculated field by using a built-in function</u>

Create a calculated field by using an expression

- 1. In the Data pane, click a blank column heading.
- 2. Type the expression for the calculation, and then press ENTER. Do not begin the expression with an equal sign (=).

For example, if employees receive a commission that equals 15 percent of their total sales, you can calculate each employee's commission by adding the sales for each employee and then multiplying the total by 0.15.

To see examples of expressions that use arithmetic operators, click \blacksquare .

Create a calculated field by using a function

- Add a field that contains the values you want to total. How?
- 2. Click anywhere in the field you want to total, and then click **Cycle through Totals** Σ .
- 3. Continue to click **Cycle through Totals** until Microsoft Query displays the type of total you want.

Each time you click **Cycle through Totals**, Microsoft Query calculates and displays a different type of total (the function used is indicated in the column heading). To see a description of the functions you can use to create totals in a calculated field, click **>**.

4. After you select the function that generates the type of total you want, click **Query Now** I to display the results of the calculated field.

Notes

To remove a calculation from a field, click **Cycle through Totals** until the column heading contains the original field name.

If your query contains many records, you can save time by waiting to display the results of the calculated field until you select the function you want. Make sure the **Auto Query** 😟 button is not pressed in.

Functions you can use in a calculated field

A function returns a value based on the results of a calculation or other operation. For example, the SUM function returns the sum of two or more values. Microsoft Query supports the following functions. The actual functions available for a data source depend on the ODBC driver for the data source. To see what functions are available for your data source, click a numeric field, and then click **Cycle through Totals** Σ .

To return this value	Use
The average of a set of values in a field	AVG
The total number of records	COUNT
The maximum of a set of values in a field	MAX
The minimum of a set of values in a field	MIN
The sum of a set of values in a field	SUM

Change the layout of a result set

When you insert fields or change the order of fields, your layout changes are retained while you work in Microsoft Query and when you return the result set to Microsoft Excel.

What do you want to do?

<u>Change the order of fields in a result set</u> <u>Insert a field into a result set</u>

Format data in Microsoft Query

Microsoft Query saves the changes when you save your query. If you format data in Query, the formatting is not displayed when you return the result set to Microsoft Excel. If you change the order of the fields or the fields that are included in the result set, those changes are retained when you return data to Excel.

What do you want to do?

Change a field column heading so that it's more descriptive Change the width of a column or the height of rows Change the font, style, or size of text in a result set Hide or display a field in a result set Change the layout of a result set

Go to a specific record in the result set

- 1. On the **Records** menu, click **Go To**.
- 2. Type the number of the record you want to go to.

Sort records in a query

- 1. Make sure the **Auto Query** 🖽 button is pressed in.
- 2. In the result set, click in the field you want to sort.
- To sort records in an ascending sort order, click Sort Ascending [▲]

To sort records in a descending sort order, click **Sort Descending** $\boxed{\textcircled{A}}$.

Notes

To sort multiple columns at one time, arrange the columns from left to right in the Data pane in the order you want to sort them. Press SHIFT and click the column headings to select the columns you want to include in the sort, and then click **Sort Ascending** or **Sort Descending**.

To return the records to their original order, click **Sort** on the **Records** menu. In the **Sorts in Query** box, click the field for which you want to remove the sort order, and then click **Remove**.

If you change the order of columns in the result set after you sort the records, the sort order is not affected.

View different fields in a query

- 1. On the Format menu, click Show Columns.
- 2. To temporarily prevent a column from appearing in the Data pane, click the column, and then click **Hide**.

To redisplay a column, click the column name, and then click **Show**.

Note Hidden fields are included when you return the result set to Microsoft Excel. If you don't want hidden fields to be returned to Microsoft Excel, delete the fields from the Data pane before you return the result set to Microsoft Excel.

About making changes to data in a queried database

Changing a database that you have retrieved data from In some cases, you can use Microsoft Query to <u>change data in an external</u> <u>database</u> or to <u>delete data in an external database</u>. When you make changes to the results of the query, Microsoft Query also changes the data in the underlying table that you used for the query.

Conditions that prevent you from changing a database You cannot make changes to data in a queried database in the following situations:

The data source is protected from changes or doesn't allow changes.

The query contains more than one table.

The query contains totals or custom expressions.

The field you're trying to change is locked or disabled, or the field is in a record that is locked.

The data you're trying to enter is not a valid type for the database. Depending on the database you're trying to change, you may need to enter a specific value in a field.

Your administrator has made the modification features on the database unavailable.

Make changes to data in a queried database

What do you want to do?

Learn about making changes to a queried database

Change data in a queried database

Delete data from a queried database

Undo changes to data in a queried database

Use Dynamic Data Exchange (DDE) with Microsoft Query

Microsoft Query supports Dynamic Data Exchange (DDE), a form of communication that allows two programs to exchange data with each other. You can use DDE methods in Visual Basic for Applications to communicate and send data back and forth between Microsoft Query and another program, such as Microsoft Excel.

Learn more about how to use DDE

Open a DDE channel

Execute DDE commands

Learn about the available DDE commands

Request status and other information

Retrieve data

Exit Microsoft Query from DDE

Close a DDE channel

Use Microsoft Query with other programs

DDE commands

Click a command or request to display reference information.

<u>Activate</u>

<u>AddParameters</u>

<u>AllowEdit</u>

BuildODBC

BuildSQL

<u>Close</u>

Command

EnableParameters

EnableTensor

<u>Exit</u>

Fetch and Fetch.Advise

Fetch.Unadvise

<u>Logoff</u>

<u>Logon</u>

<u>LogonNow</u>

ODBCLogon

<u>Open</u>

ParameterNames2

<u>QueryNow</u>

<u>QueryType</u>

<u>SaveAs</u>

SetParameterName SetQueryName SetQueryType SetWindowState UserControl UserControl2 ViewPane Troubleshoot Microsoft Query
Creating and using data sources

l can't create a new data source.

I can't find or delete a data source.

Why does my data source have an asterisk next to it?

An ODBC driver that I used previously doesn't work.

Creating queries

I can't find the table I want.

I can't create the joins I want.

My joins don't give me the records I expect.

Microsoft Query won't let me edit parameter types for my SQL procedure.

I can't edit my query.

Retrieving data

A message indicates that the path to my database is not valid.

Microsoft Query repeatedly runs out of time without connecting to my database.

Microsoft Query won't give me access to data in a Microsoft Excel list.

Microsoft Query continues to use connection resources while I'm not working with a database.

Microsoft Query does not return all of the available records.

Working in Microsoft Query

Why can't I continue to work in Microsoft Query?

Changes and formatting in my query aren't there the next time I run it.

Microsoft Query won't let me print the result set.

Microsoft Query won't let me make mailing labels.

I can't save the result set as a table in my database.

I can't add, copy, or move a record in my database.

Troubleshoot OLAP queries and cubes

The summary function I want for my cube data is unavailable.

Date or time fields are broken out incorrectly.

When I try to save an offline cube file, I run out of disk space.

I can't add fields to my OLAP cube.

I can't change my OLAP cube.

While changing an OLAP cube, a message tells me that data has been lost.

A message tells me the OLAP Cube Wizard can't open my .oqy file.

How Microsoft Query handles year 2000 dates

Four-digit years are recommended To ensure that year values are retrieved and interpreted correctly, make sure your external databases store year values as four digits (2001 instead of 01) whenever possible.

Query and Microsoft Excel interpret two-digit years consistently Query and Excel both interpret two-digit years in the same manner. When you view data that contains two-digit year values in Query and then return the data to Excel, the year values are the same in both programs.

How Query interprets two-digit years When Query retrieves a twodigit year value from an external database, Query interprets the year as follows:

Year values 00 through 29 Two-digit year values 00 through 29 are interpreted as the years 2000 through 2029. For example, if Query retrieves the date 5/28/19 from an external database, Query interprets the date as May 28, 2019.

Year values 30 through 99 Two-digit year values 30 through 99 are interpreted as the years 1930 through 1999. For example, if Query retrieves the date 5/28/98 from an external database, Query interprets the date as May 28, 1998.

Administrators can change how years are interpreted Your system administrator can customize Microsoft Office so that different rules determine how dates are interpreted. For example, your administrator can change the earliest two-digit year value that's interpreted as a future date from 29 to another number. To find out if your system uses customized date interpretation, see your system administrator.

Create a query by using the Query Wizard

- 1. Click New Query 🖻.
- 2. Select the **Use the Query Wizard to create/edit queries** check box.
- 3. On the **Databases** tab, double-click the database you want to retrieve the data from.

If you need to set up a new data source, double-click **New Data Source**, and then <u>specify the data source</u>.

- 4. Follow the instructions in the Query Wizard.
- 5. In the **Query Wizard Finish** dialog box, do one of the following:

To return the data directly to Microsoft Excel, click **Return data to Microsoft Excel**, click **Finish**, and then <u>select where you</u> <u>want to put the external data</u>.

To save the query in a .dqy query file that you can open and edit in Microsoft Query or open and run in Excel, click **Save Query**. This optional step is recommended if you plan to create an OLAP cube from the query.

To run the OLAP Cube Wizard, where you can create a cube from the data in your query, click **Create an OLAP cube from this query**, click **Finish**, and then follow the directions in the wizard. This option is recommended only for experienced users. Learn about <u>creating OLAP cubes</u>.

To refine the query further, click **View data or edit query in Microsoft Query**, and then click **Finish**. The result set is displayed in the Data pane, where you can <u>make further</u> <u>changes</u> and then <u>return the result set to Microsoft Excel</u> at a later time.

Tip To stop running a query that's taking a long time to retrieve data, press ESC.

Create a more complex query in Microsoft Query

- 1. Click New Query 🖻.
- 2. On the **Databases** tab, click the database from which you want to retrieve the data, or create a new data source.

How?

- 3. Clear the **Use the Query Wizard to create/edit queries** check box, and then click **OK**.
- 4. Add the tables you want.

How?

5. Add the fields you want to appear in the result set.

How?

If you want to narrow your result set to include specific records, enter the criteria to filter the records.

How?

If you want to create a parameter query that prompts for criteria when you run it, specify the parameters.

How?

You can create an inner join or outer join between your tables to change the result set.

How?

Notes

When you finish retrieving the data you want, you can return the result set to Microsoft Excel. For more information, click .

You can stop running a query that's taking a long time to retrieve data by pressing ESC.

If you want to wait to retrieve data until you've added all the fields to your result set, make sure that the **Auto Query** 🚯 button is not pressed in.

When you're ready to run your query, click **Query Now !**.

If you want to prevent Query from running the query before you either save the query or return the data to Excel, click **Options** on the **Edit** menu, and then clear the **Validate queries before saving or returning data** check box.

Change the type of join between tables in a query

- 1. Double-click the join line between the tables.
- To create an outer join that uses all of the records from the table on the left, click option 2 under Join Includes. You must select the = operator to create outer joins.

To create an outer join that uses all of the records from the table on the right, click option **3** under **Join Includes**.

To create an inner join that uses only records in both tables, click option **1** under **Join Includes**.

- 4. Click Add.
- 5. Click Close.

For more information about join types and how they affect the results of a query, click \blacksquare .

Create an inner join

Drag a field from one table to the equivalent field (a field of the same data type that contains similar data) in the field list for the other table.



Note With an inner join, Microsoft Query selects records from both tables or queries only when the values in the joined fields are equal. You can also <u>create an outer join</u> that retrieves all of the records from one table and any records from related records from another table.

Examples of expressions that use logical operators

Logical operators compare two values.

Logical operator	Meaning	Example	In the Criteria Field, click	In the Value box, type
And	This value and another value within the same field must be true for records in the result set.	Finds records for orders placed within a certain range of dates	Order_Date	>6/5/2000 And <6/23/2000
Or	This criterion or another criterion must be true for records in the result set.	Finds records for suppliers in either of two locations	Country	USA Or Brazil
Not	This criterion must not be true for records in the result set.	Finds records for suppliers not located in a specific country	Country	Not Canada

Examples of expressions that use other types of operators

The following table lists operators you can use to select a range of values, find values that are equal to any of several values, determine whether a record does or does not contain a value in a specified field, or find records by using a wildcard character.

Operator	Meaning	Example	In the Criteria Field, click	In the Value box, type
Between	Shows a range of values between and including the two endpoints	Finds records for order amounts between \$1,000 and \$3,000	Order_Amt	Between 1000 And 3000
In	Determines whether a value is equal to any of several values in a list	Finds records for customers in Portland, Seattle, and Denver	City	In('Portland', 'Seattle', 'Denver')
Is Null	Determines whether a record has no value in the specified field	Finds employee records that don't have an entry for the employee's supervisor	Reports_To	Is Null
Is Not Null	Determines whether a record has a value in the specified field	Finds employee records that have an entry for the employee's supervisor	Reports_To	Is Not Null
Like	Uses Like and the wildcard	Finds records for employees	Last_Name	Like 'Sm%'

character % t	o whose last
compare	names begin
values	with the letters
	"Sm", such as
	Smith and
	Small

Examples of expressions that use comparison operators

Comparison operators are used in expressions that are used as criteria to select records and limit a result set.

Comparison operator	Meaning	Example	In the Criteria Field, click	In the Value box, type
= (equal sign)	Equal to	Finds records for customers in Germany	Country	=Germany
<> (not equal to sign)	Not equal to	Finds records for suppliers outside the United States	Country	<>USA
> (greater than sign)	Greater than	Finds records for orders placed after a specific date	Order_Date	>6/30/2000
< (less than sign)	Less than	Finds records for orders placed before a specific date	Order_Date	<6/30/2000
>= (greater than or equal to sign)	Greater than or equal to	Finds records for orders placed on or after a specific date	Order_Date	>=6/30/2000
<= (less than or equal to sign)	Less than or equal to	Finds records for orders placed on or before a specific date	Order_Date	<=6/30/2000

Examples of expressions that use arithmetic operators

Arithmetic operators perform basic mathematical operations, such as addition, subtraction, multiplication and division, and produce numeric results. You can use arithmetic operators in an expression in the Criteria pane or in a calculated field.

Arithmetic operator	Meaning	Example	Syntax
+ (plus sign)	Addition	Total payment due from a customer	Order_Amt+Freight
– (minus sign)	Subtraction	Total price of an order, including any discount	Order_Amt-Discount
* (asterisk)	Multiplication	Total order amount for a specific product	Unit_Price*Quantity
/ (forward slash)	Division	Price per unit	Order_Amt/Quantity

About expressions

An expression is a calculation that uses a combination of operators, functions, field names, and constants.

You can use an expression as criteria to select records. To use an expression in the Criteria pane, you select the field in the Criteria Field that contains the values you want to evaluate, and then you enter an operator and value in the Value box. When you select a criteria field, make sure that you select it from the correct table.

Expressions that use arithmetic operators To retrieve records based on the result of a calculation, create an expression that uses the arithmetic operators for addition, subtraction, multiplication, or division. See examples of <u>expressions that use arithmetic operators</u>.

Expressions that use comparison operators Comparison operators compare one value against another. For example, to find all orders where the number of units sold is greater than 1,000, type **1000">>1000** as the criteria for the Quantity field. See examples of <u>expressions that use</u> <u>comparison operators</u>.

Expressions that use logical operators Logical operators compare values, similar to comparison operators. For example, to retrieve all records for sales outside of France, use the phrase **Not France** as the criteria for the Country field. See examples of <u>expressions that use logical operators</u>.

Expressions that use other types of operators You can use additional operators to find values that are equal to any of several values, determine whether a record does or does not contain a value in a specified field, or find records by using a wildcard character. See examples of <u>expressions that use other types of operators</u>.

Change the order of fields in a result set

- 1. Click the column heading of the field you want to move.
- 2. Move the pointer over the column heading, and then drag the column to the place you want.

Insert a field into a result set

- 1. Click the column heading of the field to the right of where you want to insert the new field.
- 2. On the **Records** menu, click **Insert Column**.
- 3. In the **Field** box, select the column you want to insert.

If you want a more descriptive heading for the column, type it in the **Column Heading** box.

4. Click Insert.

Note To quickly insert a field into a result set, you can drag a field from the table pane to the result set.

Change a field column heading so that it's more descriptive

- 1. Double-click the column heading of the field you want to change.
- 2. In the **Column heading** box, type the heading you want.

Change the width of a column or the height of all rows

Change column width To change the width of a column, point to the right border of the column heading, and then drag the border until the column is the width you want.

Tip To quickly adjust the column width to the width of the largest value in the column, double-click the right border of the column heading.

Change row height To change the height of all rows, point to the lower border of any row heading, and then drag the border until the rows are the height you want.

Note Changing row height affects all rows in the result set.

Change the font, style, or size of text in a result set

- 1. On the **Format** menu, click **Font**.
- 2. Make the changes you want.

Hide or display a field in a result set

- 1. Click the column heading of the field you want to hide.
- 2. On the Format menu, click Hide Columns.

To display a hidden column, click **Show Columns** on the **Format** menu, select the column you want, and then click **Show**.
Change data in a queried database

1. In Microsoft Query, make sure the **Allow Editing** command is checked on the **Records** menu.

In some instances, Microsoft Query will not allow you to change the data. If the **Allow Editing** command appears dimmed or Microsoft Query displays a message when you click the **Allow Editing** command, you can't change the data. For more information, click ...

- 2. In the result set, click the field you want to change.
- 3. Change the information the way you want, and then press ENTER.

Note If you make a mistake, you can reverse the most recent change by clicking **Undo** on the **Edit** menu. You must undo your change before you switch to another record or window.

Delete data from a queried database

Caution Once you delete a record from a database, it is permanently deleted from both the result set and the database.

1. On the **Records** menu, make sure the **Allow Editing** command is checked.

In some instances, Microsoft Query will not allow you to change the data. If the **Allow Editing** command appears dimmed or Microsoft Query displays a message when you click the **Allow Editing** command, you can't change the data. For more information, click .

- 2. Select the entire record you want to delete by clicking the square to the left of the record.
- 3. On the **Edit** menu, click **Delete**.

Undo changes to data in a queried database

On the Edit menu, click Undo.

Note You can reverse the most recent change you made to the data. You must undo your change before you switch to another record or window.

Use DDE

Two programs exchange information by engaging in a DDE conversation on a DDE channel. A program can conduct more than one conversation with Microsoft Query, but each conversation occurs on a different channel.

The program that initiates the conversation is called the destination program; the program that responds to the destination program is called the source program. Microsoft Query is always the source program in a DDE conversation. This means that other programs are always the ones to initiate a DDE conversation with Microsoft Query.

Note Not all programs support DDE. To find out if your program does, consult your program's documentation.

DDE Channels

There are two types of DDE channels: system channels and query channels.

You use a system channel in a DDE conversation between the destination program and Microsoft Query to get general and query-related information. In addition to providing information about queries, the system channel also lets you retrieve information about the system, such as the current connections, open queries, and the status of the destination program.

You use a query channel in a DDE conversation between the destination program and a specific query (for example, Query1) in Microsoft Query. To use a query channel, the query window must have already been opened using a system channel.

How DDE Works

You can compare a DDE conversation to a typical conversation between two people. For example, in a job interview, the interviewer may ask the applicant a series of interview questions. The interviewer (the destination) initiates and requests information from the applicant (the source) and records the information on a piece of paper (the worksheet or document). Of course, the applicant can also ask questions of the interviewer and establish a two-way conversation.

Similarly, the participating programs in a DDE conversation pass information back and forth. For example, to use DDE to communicate with Microsoft Query, a Visual Basic for Applications procedure might perform these steps:

- 1. The procedure opens a channel by using the **DDEInitiate** method.
- 2. Optionally, the procedure sends information, such as what the available data sources or current connections are, to Microsoft Query.
- 3. The procedure activates Microsoft Query. This step enables Microsoft Query to accept commands. At this point, either the user or the procedure can build or modify a query. The procedure can send commands directly to Microsoft Query by using the **DDEExecute** method.
- After the user or procedure exits Microsoft Query, control returns to the destination program. The procedure can retrieve query data and send it back to the destination program by using the DDERequest method, or the procedure can send data to Microsoft Query.
- 5. The procedure closes the channel by using **DDETerminate**.

For details about specific DDE methods in Visual Basic for Applications, see Visual Basic Help.

Open a DDE channel

To open a DDE channel, your procedure uses the **DDEInitiate** method. The **DDEInitiate** method has two arguments: the name of the source program and the "subject" of the DDE conversation (called the topic). The program name for Microsoft Query is "MSQuery."

Microsoft Query topics can be one of the following.

Торіс	Description
query_window_name	The name of an open query window. The window name can be specified only if Microsoft Query is running and a window is open.
System	A general topic that allows the destination program to link to Microsoft Query itself rather than to a query window, a file, or an SQL SELECT statement.

When Microsoft Query receives a request for a DDE conversation about a topic it recognizes, it responds and opens a channel. The **DDEInitiate** method returns a DDE channel number, which can be assigned to a variable and used for subsequent operations on this channel.

To open multiple windows on a single channel, use the System topic. When you use the System topic to open a DDE channel, you can activate only windows that were opened on that DDE channel.

Execute DDE commands

You can send commands to Microsoft Query using **DDEExecute**. Commands are specific to Microsoft Query and may or may not correspond to Microsoft Query menu commands. For example, the **Open** and **SaveAs** commands perform the same functions as equivalent menu commands in Microsoft Query, but the **ODBCLogon** and **UserControl** commands do not. For a list of the available commands, click **.**

The **DDEExecute** statement has two arguments.

Argument	Description
channel	The DDE channel value returned by DDEInitiate .
command	The command to send to Microsoft Query.

To let a user build queries in Microsoft Query, you execute the **UserControl** command. The following example sets **My Exit Command** as the **Exit** command on the **File** menu in Microsoft Windows or the **Quit** command on the **File** menu on the Macintosh.

DDEExecute Chan,"[UserControl('&My Exit Command',3,True)]"

The DDE channel value has been previously assigned to Chan, and the command being sent to Microsoft Query is **UserControl**. Notice that the **UserControl** command and its parameters are enclosed in double quotation marks ("") and square brackets ([]).

Request status and other information

Once you've opened a DDE channel to Microsoft Query, you can return a result set or other information using **DDERequest**.

The **DDERequest** method has two arguments.

Argument	Description
channel	The DDE channel value returned by DDEInitiate .
request_item	The data to be returned. This data can be the contents of a field or range of fields in a query window, or other types of information, such as the number of records in a query window.

Request items can be requested on both the query channel and the system channel. On the query channel, you can return information only about the query, such as the number of rows or columns, or the query definition. On the system channel, you can return general information about Microsoft Query, such as the current connections, topic names, and program status, as well as information specific to the active query.

Request items for a query channel or the active query on a system channel can be one of the following.

Request item	Information returned
ColumnTitles	An array of column titles. The column title is the same as the field name, unless the user has modified the column title in Microsoft Query.
ConnectionString	An array of connection string segments (each of which can be up to 255 characters long) that makes up the connection string used to connect to the active query window. If the query was created from a File Data Source Name (DSN), this item returns a DSN-less connection string.
ConnectionString/n	An array of connection string segments (each of which can be up to n characters long) that

	makes up the connection string used to connect to the active query window. If the query was created from a File DSN, this item returns a DSN-less connection string.
DataSourceName	The data source name (DSN) used by the active query.
ErrorText	The error text, if any, from the most recently executed SQL statement. This item can be requested multiple times to retrieve all the errors that have occurred. After all errors are retrieved, the item returns an empty string. The item returns a null string if the most recently executed SQL statement was successful, even if previously executed SQL statements were not successful.
FieldDef	An array of values that describes the expressions and data types of the columns (fields) in the query window. Each row of the array describes a field in the query. The array returns the following data in columnar format: the field name, the field Q+E data type, the field width, the field precision, and the field SQL data type.
GetUniqueItems	An array of the unique items in a given column of the query results. The actual request item must be the zero-based column index of the column appended to "GetUniqueItems" (for example, the unique items in the second column are returned by "GetUniqueItems1"). You cannot request the unique items for a column not included in the query.
	This item returns the same values as a SELECT DISTINCT statement for the requested column, with any joins and criteria clauses in effect. Any criteria clauses that use parameters are not

included.

NameSeparator	The single character used as the ODBC qualifier name separator.
NumCols	The number of columns (fields) in the query.
NumRows	The number of rows (records) in the query.
ODBCSQLStatement	An array of SQL segments (each of which can be up to 255 characters long) that makes up the ODBC SQL statement for the query. With this string, you can bypass Microsoft Query and send an ODBC SQL statement directly to ODBC for processing.
ODBCSQLStatement/n	An array of SQL segments (each of which can be up to n characters long) that makes up the ODBC SQL statement for the query. You can use this request item to store an ODBC SQL statement in several smaller segments.
ParameterNames	An array of parameters in the query. An empty array is returned if no parameters exist. This item can be requested at any time, even if parameters are disabled for the query.
Query	The name of the query.
QueryDefinition	An array of query definition segments (each of which can be up to 255 characters long) that makes up the definition of the query (an SQL statement), as defined by Microsoft Query. This item can be retrieved and saved for future queries.
QueryDefinition/n	An array of query definition segments (each of which can be up to n characters long) that makes up the query definition of the query (an SQL statement).
Recest	A rough estimate of how many rows can be retrieved at a time.

TierType	A single digit specifying the type of data source:
	1 indicates a file that can contain only a single
	table; 2 indicates a file for a database containing
	one or more tables; 3 indicates the data can't be
	browsed. This request item is used if the
	program needs to provide an Open dialog box
	for browsing the data on a disk.

When you use the above request items on the system channel, information is returned for the query window most recently used in a DDE operation.

For the system channel, in addition to the above request items, you can also request the following items.

Request item	Information returned
Sources	All current data source connections (DSN only).
Logon	All remote databases you can connect to (DSN entries from the registry).
Logoff	All currently connected remote databases (DSN only).
Tables/source/user/database	All tables for the specified DSN connection, user, and database. Note that the source must be a valid DSN.
Users/source/database	All users for the specified DSN connection and database. Note that the source must be a valid DSN.
Database/source	All databases for the specified DSN connection. Note that the source must be a valid DSN.
Username/source/database	The user name for the specified DSN connection and database. Note that the source must be a valid DSN.

Topics	The names of the topics open on the system channel, along with System.
Status	The status of the program, which can be Ready or Busy. Ready means that the program can have a DDE conversation with Microsoft Query. Busy means that Microsoft Query isn't ready to have a DDE conversation.

This example inserts the value returned by the NumRows request item into cell A10 in Microsoft Excel.

ActiveWorksheet.Range("A10").Value = DDERequest(chan, "NumRov

Retrieve data

Whereas **DDERequest** can return data to a program, the **Fetch** and **Fetch.Advise** commands deserve special mention because they retrieve data from Microsoft Query and send it directly to a worksheet or document. **Fetch** and **Fetch.Advise** are used as the command argument to **DDEExecute**.

Fetch.Advise is similar to **Fetch**, except that the channel created by the first retrieval isn't closed immediately. Microsoft Query continues to send any changes to the data until the original DDE channel is closed or until Microsoft Query receives a **Fetch.Unadvise** command.

Exit Microsoft Query from DDE

To exit Microsoft Query, a program can use the **Exit** command, rather than have the user choose **Exit** or **Quit** from the **File** menu.

Command	Description
Exit (False)	Exits Microsoft Query only if no queries are open on other channels
Exit (True)	Exits Microsoft Query regardless of whether other queries are open

Close a DDE channel

To close a DDE channel, use **DDETerminate**. This method has only one argument — the channel number of the DDE channel to be closed. All queries that are opened on this channel are closed automatically when you use **DDETerminate**, and you are logged off from all connected data sources.

An error is returned if the channel number you use is invalid.

Use Microsoft Query with other programs

Typically, other programs use Microsoft Query to accomplish three main tasks:

To create a new query

To update existing data

To modify the query definition or data

Microsoft Query as a data query tool for users

Using DDE, you can have Microsoft Query function as a tool for retrieving data. Once your macro or program opens a DDE channel, it can activate Microsoft Query and specify which command replaces **Exit** on the **File** menu in Microsoft Windows or **Quit** on the **File** menu on the Macintosh, what program state (minimized [Windows only], maximized, or restored) Microsoft Query is in, and whether to display a series of dialog boxes.

A user can then create, open, or edit a query as if Microsoft Query were running as a standalone program. When the user has finished building the query, the user can click the custom **Exit** command or click the **Exit** button on the toolbar.

Once control returns to the destination program (the document, worksheet, or other object) data can be requested from Microsoft Query. For example, the user may request the number of rows or columns in the query, the SQL string defining the query, the connection string for the data source, or the data source name. Data can be returned to the destination program before the DDE channel is closed and the macro or program ends.

Update retrieved data

After you have executed a query and returned a result set, you may later want to update the data with whatever is then current in the data source. Using DDE, you can update data without input from the user: A program can send an SQL statement to Microsoft Query, which can execute the query and then return the result set to the destination program.

Modify a query definition or data

Another way a program can use Microsoft Query is to allow a user to modify an existing query definition in Microsoft Query or to edit data in a data source.

A program starts this process by opening a DDE channel, connecting to a data source, and sending an SQL string (saved from the previous query) to Microsoft Query to build a new query. Once the program has opened the new query, user control begins, and the user can modify the query or edit the data as needed. Once the user exits Microsoft Query, the data is returned to the program, the new query definition is saved, and the query is closed.

For example, a user working in Microsoft Word could choose a custom command called "Get Sales Figures" to start the corresponding macro. Without the user being aware of it, this macro could then start Microsoft Query, build a query, and then return the sales data to the Word document.

Activate Command (DDE)

Activates the specified query window. This command is used to change the active window when more than one window is open. It is available only on the system channel.

Syntax

Activate(queryname)

queryname The name of the query window to activate

Example

This example executes two queries, creating two query windows. After executing the second query, the example makes the window created by the first query the active window.

DDEExecute chan, "[BuildSQL('Select * From Customers')]" DDEExecute chan, "[QueryNow()]" stQueryname1 = DDERequest(chan, "Query") DDEExecute chan, "[BuildSQL('Select * From ORDERS')]" DDEExecute chan, "[QueryNow()]" stQueryName2 = DDERequest(chan, "Query") DDEExecute chan, "[Activate(''' & stQueryname1(1) & ''')]"

AddParameters Command (DDE)

Adds a parameter to the active query for use by PivotTable reports.

Syntax

AddParameters(field1, field2, ...)

field1, field2, … The names of table columns that should be added to the criteria as parameters. The names of the new parameters are defined to be the names of the fields themselves. If you specify a field name that isn't available in the active query, an error occurs.

Remarks

AddParameters can be used even when entering parameters is disabled in Microsoft Query.

The modified SQL string can be returned by requesting the ODBCSQLStatement item.

To reset parameters, you must reset the entire SQL string.

AllowEdit Command (DDE)

Determines whether editing is allowed in a query window. This command is available on both the system channel and the query channel. It is equivalent to the **Allow Editing** command on the **Records** menu.

Syntax 1

AllowEdit(allowedit, locked)

Syntax 2

Allow.Edit(allowedit, locked)

The two forms of syntax are equivalent.

- *allowedit* Determines whether editing is allowed in the query window. Can be True or False.
- *locked* Determines whether the **Allow Editing** command on the **Records** menu is disabled when **UserControl** is in effect. Can be True or False.

Example

This example executes a previously built SQL statement. The example then sends the **AllowEdit** command and the **UserControl** command to Microsoft Query to allow the user to view but not edit data in the data source (the **Allow Editing** menu command is disabled after **UserControl** is in effect).

DDEExecute chan,"[BuildSQL(""" & sql & """)]") DDEExecute chan,"[QueryNow()]") DDEExecute chan,"[AllowEdit(False, True)]") DDEExecute chan,"[UserControl(""&Return Data"", 3, False)]")

BuildODBC Command (DDE)

Builds a connection string of any length one segment at a time.

Syntax

BuildODBC([connectstring])

connectstring A segment of a valid connection string. If *connectstring* is 0 (zero) or omitted, the connection string is cleared.

BuildSQL Command (DDE)

Builds an SQL statement one item at a time to create a new query on the DDE channel. This command is available only on the system channel.

Syntax

BuildSQL[(sqltext)]

sqltext A valid SQL statement. If *sqltext* is 0 (zero) or omitted, the SQL statement is cleared.

Remarks

The **BuildSQL** command is useful for building long SQL statements. For example, if an SQL statement is too long to send in a single instruction, you can send it in several smaller pieces.

After you build an SQL statement, you can execute it using the <u>QueryNow</u> command.

Example

This example uses the QueryDefinition request item to return the SQL statement for a query.

sql = DDERequest(chan,"QueryDefinition")

This example uses the **BuildSQL** command to resend the returned SQL string with an additional WHERE clause.

```
DDEExecute chan, "[BuildSQL(""" & sql & """)]"
DDEExecute chan, "[BuildSQL("WHERE Freight > 10")]"
DDEExecute chan, "[QueryNow()]
```
Close Command (DDE)

Closes the current query. This command is available on both the system channel and the query channel.

Syntax

Close()

Command Command (DDE)

Sends an **Open** command to Microsoft Query. This command is provided for programs that limit the length of strings used with **DDEExecute**.

This command is available only on the system channel.

Syntax

Command(*opnum*, *execstring*)

opnum Determines the way Microsoft Query receives the commands. Can be one of the following values.

Value	Description
1	Initializes the buffer in Microsoft Query with <i>execstring</i> (clearing the buffer if it exists), but doesn't execute the query.
2	Concatenates execstring to the end of the buffer.
3	Concatenates <i>execstring</i> to the end of the buffer and executes the complete command buffer. Because the command buffer is cleared after execution, <i>execstring</i> can't be re-executed.

execstring Segments of an Open command. The entire string must be enclosed in single or double quotation marks.

Microsoft Query supports sending only the <u>Open</u> command using the **Command** command.

Remarks

This command is provided for Q+E compatibility only. It is recommended that you use the <u>BuildSQL</u> command, if possible, to send long SQL statements.

EnableParameters Command (DDE)

Controls whether the user can specify parameters in the criteria pane and SQL window. By default, the user cannot specify parameters. You cannot disable parameter specification after specifications have already been entered by the user.

Syntax

EnableParameters(*flag*)

flag Determines whether the user can specify parameters. Can be 0 (zero) to disable specifying parameters or 1 to enable specifying them.

EnableTensor Command (DDE)

Indicates whether the destination program supports OLAP data sources. This command is available only on the system channel.

Syntax

EnableTensor(flag)

flag Use True to allow OLAP data sources to be returned to the destination program. Use False, or omit, to allow only data sources other than OLAP to be returned to the destination program.

Exit Command (DDE)

Exits Microsoft Query. This command is available only on the system channel.

Syntax

Exit(exitall)

exitall Determines whether Microsoft Query will exit if queries on channels other than the one executing the **Exit** command are open. Can be either True or False.

Use True to exit Microsoft Query regardless of which queries are open. If Microsoft Query is under user control for another channel (that is, the <u>UserControl</u> command has been executed), this setting closes that channel and returns control to the program that started Microsoft Query in user control.

Use False to exit Microsoft Query only if any open queries were opened on the channel executing the **Exit** command.

Remarks

This DDE command is also used to quit Microsoft Query for the Macintosh.

Fetch and Fetch.Advise Commands (DDE)

The **Fetch** command retrieves data from Microsoft Query and sends it directly to the destination program.

The **Fetch.Advise** command updates the destination program automatically as values in Microsoft Query change, not just when the **Fetch.Advise** command is sent.

These commands are available on both the system channel and the query channel.

Syntax 1

Fetch.Advise(appname, topic, range, values)

Syntax 2

Fetch(appname, topic, range, values)

appname The name of the destination program to which the data is to be sent. Here are the *appname* arguments you'd use for Office programs:

Program	Value of appname
Microsoft Access	MSAccess
Microsoft Excel	Excel
Microsoft Word	WinWord

- *topic* The name of the worksheet, document, or control to which the data is to be sent.
- *range* The area of the worksheet, document, or control to which the data is to be sent.
- *values* The values to be retrieved from Microsoft Query. (For valid values, see the table below.)

Note The **Fetch.Advise** command automatically updates only query data, not request items or row#col# references.

Can be one of the following values:

Value	Description	
Request_item	The information to return; for example, QueryDefinition.	
row#col#	A reference to retrieve specific rows and columns, such as R3C5 from Microsoft Excel.	
/Headers	Retrieves column names.	
/Rownum	Retrieves row numbers.	
All	Retrieves all records.	

Remarks

For **Fetch.Advise**, the channel created to return data to the destination program isn't closed until a <u>Fetch.Unadvise</u> or <u>Close</u> command is sent.

Example

This example uses the NumRows and NumCols request items to return the number of rows and columns in the query, and then returns the values with column headings to a Microsoft Excel worksheet named "Sheet1."

```
rows=REQUEST(Chan,"NumRows")
cols=REQUEST(Chan,"NumCols")
DDEExecute chan, "[Fetch(""Excel"", ""Data.xls!Sheet1"", _
""R1C1:R"&rows&"C"&cols&""", ""All/Headers"")]"
```

Fetch.Unadvise Command (DDE)

Removes the automatic updating feature set by <u>Fetch.Advise</u> and closes the channel opened by Microsoft Query to pass data to the destination program. This command is available on both the system channel and the query channel.

Note The arguments used by **Fetch.Unadvise** must match the arguments used previously by **Fetch.Advise**.

Syntax

Fetch.Unadvise(appname, topic, range, values)

- *appname* The name of the destination program to which the data is to be sent. (For examples, see the <u>Fetch.Advise</u> command.)
- *topic* The name of the worksheet, document, or control to which the data is to be sent.
- *range* The area of the worksheet, document, or control to which the data is to be sent.
- *values* The values to be retrieved from Microsoft Query. (For a table of valid values, see the <u>Fetch.Advise</u> command.)

Logoff Command (DDE)

Logs off the specified data source. This command is available only on the system channel.

Syntax

Logoff(source)

source The name of the data source you want to disconnect from.

Logon Command (DDE)

Displays a dialog box that enables you to log onto a data source. This command is available only on the system channel.

Syntax

Logon(source)

source The name of the data source you want to connect to.

LogonNow Command (DDE)

If you've used the <u>BuildODBC</u> command to build the connection string, this command logs on with the connection string in the buffer. If there is no connection string in the buffer, an error occurs.

Syntax

LogonNow([dialog])

dialog Determines whether dialog boxes are displayed at logon time. Can be one of the following values.

Value	Description
0 (zero)	No dialog boxes are displayed. If this argument is omitted, the default value is 0 (zero).
1	Dialog boxes are displayed to obtain additional information from the user not provided by the connection string.
2	Dialog boxes are displayed with values from the connection string filled in as default values.
3	Same as 1, except that information not required to connect to the data source appears dimmed.

ODBCLogon Command (DDE)

Allows a program to connect to a data source. You can specify whether dialog boxes are presented to the user at logon time. This command is available only on the system channel.

Syntax

ODBCLogon(connectstring, dialog)

connectstring A string that supplies information (such as the data source name, user ID, and password) required by a driver to connect to a data source. This argument must follow the format required by the driver you're using. (For more information about the connection string, see the documentation for your data source.)

Note Because Microsoft Query supports File Data Source Names (DSNs), you can use the ConnectionString request item to obtain the DSN-less connection string of a query created from a File DSN. You can then use that connection string to reconnect to the data source from any computer.

dialog Determines whether dialog boxes are displayed at logon time. Can be one of the following values.

Value	Description
0 (zero)	No dialog boxes are displayed.
1	Dialog boxes are displayed to obtain additional information from the user not provided by the connection string.
2	Dialog boxes are displayed with values from the connection string filled in as default values.
3	Same as 1, except that information not required to connect to the data source appears dimmed.

Example

This example uses the ConnectionString request item to return the string needed to connect with a data source. The example saves the string after the user logs onto a data source, and later uses the **ODBCLogon**

command to connect to the same data source without user input.

```
connstr = DDERequest(chan, "ConnectionString")
' intervening code
DDEExecute chan, "[ODBCLogon (connstr(1), 0)]"
```

Open Command (DDE)

Opens and executes a query and displays the data in the Data pane. This command is available only on the system channel.

Syntax

Open(*query*)

query The name of the Microsoft Query file (for Microsoft Windows, the file name extension is .dqy) or Q+E file (Microsoft Windows only; the file name extension is .qef) that contains the query; or the SQL SELECT statement to execute.

Example

This example opens a query named Sales.

DDEExecute chan, "[Open(""c:\msquery\sales.dqy"")]")

This is the same example on the Macintosh.

DDEExecute chan, "[Open(""drivename:msquery:sales"")]")

This statement uses an SQL SELECT statement to execute a new query.

DDEExecute chan, "[Open(""SELECT * FROM pubs.dbo.titles"")]")

ParameterNames2 Request (DDE)

Returns information about the parameters for a parameter query. This request is available only on the system channel.

Syntax

ParameterNames2()

For each parameter, ParameterNames2 returns the name, data type, and most recently used value. ParameterNames2 returns a 3 column by *N* row array in xITable format, as follows:

Column 1	Column 2	Column 3
Parameter name	Column data type	Most recently used parameter value, if available; otherwise, null

For information about xITable format, refer to Visual Basic Help for Microsoft Excel.

The ParameterNames2 request can be used at any time, even if parameters have been disabled for a query by the <u>EnableParameters</u> <u>command</u>. If the query has no parameters, ParameterNames2 returns an empty array.

QueryNow Command (DDE)

Executes the active query for the open DDE channel. If you've used the <u>BuildSQL</u> command to build the query, this command executes the SQL statement in the buffer and opens a query window. This command is available on both the system channel and the query channel.

Syntax

QueryNow()

Example

This example saves an SQL statement from a query and later sends the same statement to re-execute the query.

sql = DDERequest(Chan, "QueryDefinition")
DDEExecute chan, "[BuildSQL(""" & sql & """)]"
DDEExecute chan, "[QueryNow()]"

QueryType Request (DDE)

Returns information about the data source for a query. This request is available only on the system channel.

Syntax

QueryType()

The QueryType request returns a two-element array in xlTable format. The first element contains the type of the query data source, either ODBC or OLE/DB. The second element contains the type of commands accepted by the data source: SQL, Cube, or Cancel, as follows:

QueryType	CommandType	Description
ODBC	SQL	The data source is SQL-compliant ODBC.
OLEDB	Cube	The data source is an OLAP cube. The SQL statement contains the cube name.
ODBC	Cancel	The user has canceled the query and returned to the destination program.

For information about xITable format, refer to Visual Basic Help for Microsoft Excel.

Use the QueryType request only as the first request after a <u>UserControl2</u> <u>command</u> has completed execution.

If QueryType indicates that the data source is an OLAP cube, you can use the ODBCSQLStatement and ConnectionString requests to retrieve the data source information. Learn about <u>how to use DDE requests</u>.

SaveAs Command (DDE)

Saves a new query's design or an existing query's design under a different name. Performs the same function as **Save As** on the **File** menu and is available on both the system channel and the query channel.

Syntax 1

SaveAs(queryname)

Syntax 2

Save.As(queryname)

The two forms of syntax are equivalent.

queryname The new name for the query.

SetParameterName Command (DDE)

Sets the name of a parameter received in a query string passed to Microsoft Query. **SetParameterName** should be sent after a **BuildSQL** command is sent.

Syntax

SetParameterName(index, name)

index The parameter index.

name The name for the parameter.

Remarks

If the SQL statement sent by **BuildSQL** contains parameters but no **SetParameterName** commands are received, Query assigns default parameter names in the form "Parameter n," where n is an incrementing number. If the number of parameter names sent using **SetParameterName** commands is insufficient, the remaining parameters are named "Parameter n". If too many parameter names are sent, the excess names are ignored.

SetQueryName Command (DDE)

Sets the name of the active query, or creates a query with that name if there is no active query. The name appears in the query window title bar.

Syntax

SetQueryName(name)

name The name of the query.

SetQueryType Command (DDE)

Determines the action Microsoft Query takes for queries based on different data source types. This command is available only on the system channel.

Syntax

SetQueryType(QueryType, CommandType)

Valid combinations of the *QueryType* and *CommandType* parameters are as follows:

QueryType	CommandType	Description
ODBC	SQL	Edits the query in Microsoft Query. These values are assumed if the parameters are omitted.
OLEDB	Cube	Runs the OLAP Cube Wizard to change the cube definition.
When you us	e SetOuervTvpe(C	LEDB. Cube). Microsoft Ouerv ignores all

LogonNow, EnableParameters, and SetQueryName commands, and UserControl2 arguments that are not valid within the context of editing a cube definition in the OLAP Cube Wizard.
SetWindowState Command (DDE)

Specifies whether the active window is normal, minimized, or maximized when opened. This command is available on both the system channel and the query channel.

Syntax

SetWindowState(*state*)

state The state of the active window. Can be one of the following values.

Value Description

- 1 Normal
- 2 Minimized (Microsoft Windows only)
- 3 Maximized

UserControl Command (DDE)

Activates Microsoft Query and allows users to create, open, or edit a query as if they were running Microsoft Query as a standalone program.

This command is available only on the system channel.

Syntax

UserControl(exittext, appstate, newquery)

exittext The name of the **Exit** command used to exit Microsoft Query after a DDE conversation. If this argument is omitted, "Return" is used.

appstate The state of the program. Can be one of the following values.

Value	Description
1	Normal
2	Minimized (Microsoft Windows only)

3 Maximized

If this argument is omitted, the current state of Microsoft Query is used.

newquery Specifies whether the **Choose Data Source** dialog box is displayed when Microsoft Query is started during a DDE conversation. Can be True or False.

If *newquery* is True, the **Choose Data Source** dialog box is displayed, asking the user to select a data source. If *newquery* is False (the default), no dialog box is displayed and any open queries for that channel are accessible.

Remarks

User control ends when the user clicks **Exit** on the **File** menu in Windows or **Quit** on the **File** menu on the Macintosh. Microsoft Query then returns to its previous program state and returns control to the destination program.

Important Do not allow a program that includes a time-out feature to time out before user control ends.

While Microsoft Query is running under user control, any new or existing queries are opened on the same DDE channel.

Example

This example starts Microsoft Query as a maximized program and replaces the **Exit** command on the **File** menu (or the **Quit** command on the **File** menu on the Macintosh) with the **Return Data to Microsoft Excel** command. Then it displays the **Choose Data Source** dialog box.

DDEExecute chan, "[UserControl(""&Return To Excel"",3,True)]")

UserControl2 Command (DDE)

Activates Microsoft Query and allows users to create, open, or edit a query as if they were running Microsoft Query as a standalone program.

This command is available only on the system channel.

Note The difference between the **UserControl** command and the **UserControl2** command is that **UserControl2** allows you to specify a command to add to the **File** menu that closes all queries created in Microsoft Query and returns to the destination program without any data.

Syntax

UserControl2(exittext, canceltext, appstate, newquery)

- *exittext* The name of the **Exit** command used to exit Microsoft Query after a DDE conversation. If this argument is omitted, "Return" is used.
- *canceltext* The name of the menu item used to cancel the current session. The command is added to the bottom of the **File** menu in Microsoft Query. If this argument is omitted, no new menu item is created.

appstate The state of the program. Can be one of the following values.

- 1 Normal
- 2 Minimized (Microsoft Windows only)
- 3 Maximized

If this argument is omitted, the current state of Microsoft Query is used.

newquery Specifies whether the **Choose Data Source** dialog box is displayed when Microsoft Query is started during a DDE conversation. Can be True or False.

If *newquery* is True, the **Choose Data Source** dialog box is displayed, asking the user to select a data source. If *newquery* is False (the default), no dialog box is displayed and any open queries for that channel are accessible.

ViewPane Command (DDE)

Specifies which panes in the query window are displayed. This command is available on both the system channel and the query channel.

Syntax

ViewPane(pane)

pane Specifies which panes are displayed. Can be one of the following values:

Value	Description
0 (zero)	Data pane only
1	Data pane and Table pane
2	Data pane and Criteria pane
3	Data pane, Table pane, and Criteria pane

I can't create a new data source.

Check your driver First, make sure you have the right ODBC driver or data source driver for your data source. ODBC drivers and data source drivers make Microsoft Query flexible so that you can connect to new databases as they become available by installing new drivers. However, you must make sure you install and set up the right driver for the type of database you're using.

Make sure the driver works with Microsoft Query In addition to the drivers provided with Microsoft Query, you can use ODBC and data source drivers provided by third-party manufacturers. Before you try to use a third-party driver, it's a good idea to make sure the manufacturer has tested the driver with your version of Microsoft Query. For some databases, the driver supplied with the database software may be the best choice. Contact the administrator for your database to find out what's available and what works best at your site. See a list of the drivers supplied with Query.

Check the server address and logon information Before you set up a data source, make sure you know the address where the database is located on your network and have the necessary permissions to connect to the database and log on. See the administrator for your database for a logon name, password, or any other permissions required, and to make sure the access you've been granted is working properly.

Make sure the driver is properly installed To display the list of available ODBC drivers, click New Query , double-click New Data Source on the Databases or OLAP Cubes tab, type a name in step 1 of the Create New Data Source dialog box, and then click the list in step 2. If the OLAP Cubes tab is not available, run Query from Microsoft Excel by pointing to Get External Data on the Excel Data menu, and then clicking New Database Query. Double-click New Data Source on the OLAP Cubes tab, type a name in step 1 of the Create New Data Source dialog box, and then click the list in step 2. If you don't see the driver you need, you should check to make sure the driver is installed properly. How to install drivers.

Make sure you supplied all of the configuration information After

you've installed the driver and selected it in step 2 of the **Create New Data Source** dialog box, make sure you provide all of the necessary information in step 3 of the dialog box. For information about a Microsoft driver, click **Connect**, and then click **Help** in the setup dialog box for the driver. For third-party drivers, see the Help system or the documentation for the driver.

If you are setting up a data source with an ODBC driver or data source driver provided by Microsoft, click the name of your driver for information about the settings you should make in step 3 of the **Create New Data Source** dialog box.

Note Query cannot access data sources created with the Data Connection Wizard.

ODBC drivers

Microsoft Access driver Microsoft dBASE driver Microsoft Visual FoxPro driver Microsoft Excel driver Microsoft ODBC for Oracle driver Microsoft Paradox driver Microsoft text driver Microsoft SQL Server driver

Data source driver

Microsoft SQL Server OLAP Services driver

I can't find or delete a data source.

Data sources that were created with earlier versions of Microsoft Query Data sources that were created with versions of Microsoft Query earlier than Query 97 have an asterisk next to the data source name. All data sources, including new ones that are created with Query 2000 and later, are stored as files with file name extension .dsn. To list data sources that were created with earlier versions of Query, click **Options** in the **Choose Data Source** dialog box (**File** menu, **New** command), and then select the **Include registry DSNs in list of available databases** check box. If these data sources are stored in the default location, you can select them in the **Choose Data Source** dialog box and then click **Delete**.

Locations for data source files Data source files from earlier versions of Microsoft Query are usually stored in the ODBC Datasources folder in the Common Files folder. Data source files for Query 2000 and later are stored in your user profile folder in Application Data\Microsoft\Queries. To see what folders are used to store data source files, click **New Query** , click the **Databases** tab, and then click **Options**.

Find and delete the data source file You can find and delete data source files by doing the following:

Use the Windows Control Panel If all data source files are in the Common Files\ODBC Datasources folder or your user profile folder, double-click the ODBC Data Sources (32bit) icon in Control Panel, or on Microsoft Windows 2000 systems, double-click Administrative Tools and then double-click Data Sources (ODBC). Click the File DSN tab, and then use the Remove button to delete any .dsn files that you don't want.

Search for .dsn files If data source files are in several folders in Windows, use the Microsoft Windows **Find** command to search for all files with file name extension .dsn, and then delete any files that you don't want.

Why does my data source have an asterisk next to it?

The data source is from an earlier version of Microsoft Query Query 97 and later data sources have a different format from data sources that were created with earlier versions of Query. Data sources and queries that were created with versions earlier than Query 97 can still be used with later versions of Query, but queries that are created by using these data sources cannot be shared with other users. Versions earlier than Query 97 store data source information as part of your Microsoft Windows operating system, and that information is available only on your system.

Identify nonshareable data sources To use data sources that were created by using earlier versions of Query, click **Options** in the **Choose Data Source** dialog box (**File** menu, **New** command), and then select the **Include registry DSNs in list of available databases** check box. All data sources that were created by using a version earlier than Query 97 will appear in the **Choose Data Source** dialog box with asterisks next to their names.

Create shareable data sources If you want to share queries or report templates that use data from the external databases that are specified in your nonshareable data sources, <u>create new data sources</u> for these databases, and use the new data sources to create the queries, query files, and report templates.

Delete the old data sources <u>How to delete nonshareable data</u> <u>sources</u>.

An ODBC driver that I used previously doesn't work.

You may need to use a different driver with Microsoft Query in the following situations:

Make sure you're using a 32-bit ODBC driver Microsoft Query requires 32-bit ODBC drivers, such as the ones supplied with Microsoft Query 97 and later. For example, if you develop a Microsoft Excel program in Microsoft Visual Basic to get external data and distribute your program to Microsoft Excel version 5.*x* users, the program may not run with the 16-bit ODBC drivers on the Microsoft Excel version 5.*x* systems.

Make sure Microsoft drivers are version 3.0 or later If you install Microsoft ODBC drivers other than the ones included with the most recent versions of Microsoft Excel or Microsoft Office, make sure you install version 3.0 or later of these drivers. Earlier versions of the Microsoft ODBC drivers may not work with Query 2000 and later. The drivers supplied with Microsoft Excel 97 and Microsoft Office 97 are version 3.0 or later. When you install these drivers, any old versions on your system should be replaced automatically.

Make sure you have the correct version of a third-party driver If you are working with a third-party ODBC driver, you may also need to install a new ODBC driver for Microsoft Query when you upgrade your database or install a new version of the database. To determine whether this is necessary, check the documentation for your database.

Make sure you set up the data source for the driver with Microsoft Query You can't use Query to access data sources set up with the Data Connection Wizard.

I can't find the table I want.

If the list of tables does not include the tables you expect to see, check the following, depending on the options available for your data source:

Check the owner The tables you want might belong to a different owner. In the **Add Tables** dialog box, click a different owner in the **Owner** box. The person who owns the tables might need to grant you permission before you can use the tables.

Check the displayed table types The tables you want might be of types that aren't selected for display. In the **Add Tables** dialog box, click **Options**. Under **Show**, make sure all of the table types are selected.

Refresh the table list The tables you want might have been added to the database since the last time Microsoft Query connected to the database. To see recently added tables, click **Options** in the **Add Tables** dialog box, and then click **Refresh**.

I can't create the joins I want.

For an outer join, make sure the query has no more than two tables If your query has more than two tables, you cannot create an outer join in the query. Outer joins are options 2 and 3 in the **Joins** dialog box.

A query can have only one outer join You can create as many inner joins (option 1 in the Joins dialog box) as you want, but you can create only one outer join in each query. If you know how to program in the SQL language, you can try changing the SQL statement in the SQL dialog box (View menu, SQL command) to create more than one outer join.

Information about joins For information about using outer joins to combine records, click \blacksquare .

My joins don't give me the records I expect.

Query uses inner joins by default The default join that Microsoft Query uses when it joins tables automatically is an inner join (option 1 in the **Joins** dialog box). When two tables are joined with an inner join, records from the second table that don't have corresponding records in the first table are not included in the result set.

Create an outer join To retrieve all of the records from both tables when one table contains records not present in the other table, you can use an outer join (options 2 and 3 in the **Joins** dialog box). For information about using outer joins to combine records, click **.**

Check multiple joins between the same tables An outer join between two tables sometimes does not return any records from the second table. If you then add an inner join between the two tables, Microsoft Query will not retrieve any records from either table. To display records, delete one of the joins.

Information about joins For more information about joins, click .

Microsoft Query won't let me edit parameter types for my SQL procedure.

To create parameters in a SQL stored procedure and set the data types:

- 1. On the File menu, click Execute SQL.
- 2. In the **SQL statement** box, create your query using fixed arguments for the procedure, and then click **Save**.
- 3. On the View menu, click SQL.
- 4. In the **SQL statement** box, replace each fixed argument with a question mark (?).
- 5. Click **OK**, and then click **Parameters** on the **View** menu.
- 6. To change the data type for a parameter, click the parameter and then click **Edit**.

I can't edit my query.

Make sure editing is enabled On the Edit menu, click Options, and then clear the Disable ability to edit query results check box.

A message states that the path to my database is not valid.

Check for a mapped network drive If your database is on a shared network directory, when you set up the data source and selected the database file, the path to the database may have been recorded in the data source with the mapped drive letter in use by your system at that time. For example, if your database is named Inventory.mdb, and you had drive G mapped to the shared network directory where this database is stored, your data source might record this location as G:\public\Inventory.mdb. When you try to use this data source, or you or other users try to run queries created with this data source, the ODBC driver displays a message that the path is not valid if drive G is not mapped to the shared network directory.

Use an alternative to drive mapping If you are using the Microsoft Access driver or the Microsoft Excel driver, you can correct this problem by <u>creating a new data source</u>. When you specify the location of the database file, don't select the mapped drive for the shared network directory. Instead, type the UNC address of the shared network directory, and then locate the database file. For example, if your database file is stored on a server named Shared, you could type **\\Shared\public** and then select the file Inventory.mdb.

Map the same network drive letter before using the data source For other ODBC drivers, before you use a data source or run a query created with that data source, make sure the same drive is mapped to the shared network directory where the database is located as was mapped when the data source was created.

Microsoft Query repeatedly runs out of time without connecting to my database.

Set a longer connect time By default, Microsoft Query tries to connect to your database for an amount of time predetermined by the ODBC driver for the data source. If connections to your database often require more time, you can set a longer interval. On the **Edit** menu in Microsoft Query, click **Options**, make sure the **Cancel the connection if not connected within** check box is selected, and then enter the number of seconds to allow for the connection.

Have Query try to connect indefinitely If you want to allow Microsoft Query to try to connect for an indefinite amount of time, clear the **Cancel the connection if not connected within** check box. If Microsoft Query can't connect to your database, you can cancel the attempt by pressing ESC.

Microsoft Query won't give me access to data in a Microsoft Excel list.

Close the workbook Before you try to access the data, make sure the workbook is not open in Microsoft Excel.

Check the worksheet Check the following on your Microsoft Excel worksheet:

Check for list format Make sure the list follows Microsoft Excel list format. (For information about guidelines for creating a list on a worksheet, see Microsoft Excel Help.) Name the ranges that are tables, and make sure you give each range a different name.

Name the list Microsoft Query recognizes only named ranges as tables. To name a list, select the entire list in Microsoft Excel, and then type a name in the **Name** box on the formula bar.

Don't access the list as a system table Do not attempt to gain access to Microsoft Excel tables by clicking the **System tables** option under **Show** in the **Table Options** dialog box. If you use this approach, Microsoft Query may recognize blank cells on your worksheets as data, may not recognize some of your actual data, and may not be able to retrieve your data correctly.

Format lists to combine data the same way Use the same format for columns in each list that you want to combine. For example, if each list has a date column, apply the same date format to each column. If you mix text and number formats within a column, the Microsoft Excel driver may not be able to find and combine all of your data.

Don't use column labels such as Table or Database Microsoft Query uses some labels internally and does not recognize them in your data.

Move all workbooks to the same folder A data source that you set up for the Microsoft Excel driver can provide access to workbooks in only one folder at a time. If the lists are in two different workbooks, make sure both workbooks are in the same folder as the workbook that you specified when you set up the data source.

Move workbooks to the folder expected by the data source Before you start Microsoft Query, make sure you copy all the workbooks you want to use to the folder specified in your data source. Microsoft Query does not automatically recognize new workbooks that you add to the folder while you are creating a query. You can gain access to new workbooks by clicking **Options** in the **Add Tables** dialog box and then clicking **Refresh**.

Use Excel to combine data Microsoft Query is useful when you want to combine data from two or more Microsoft Excel lists. If you're working with data from a single worksheet or you want to use simple criteria to select rows from a single worksheet, you may find the Microsoft Excel AutoFilter and Advanced Filter commands (Data menu, Filter submenu) easier to use. For more information about AutoFilter and Advanced Filter, see Microsoft Excel Help.
Microsoft Query continues to use connection resources while I'm not working with a database.

Connections stay open throughout a Query session By default, Microsoft Query makes a single connection to your database and keeps the connection open until you exit from Microsoft Query, regardless of how many query windows you open for the database during that time. If you're creating queries to more than one database during a Microsoft Query session, the open connections can tie up resources and expensive connect time.

Close connections when you close query windows If you want to minimize connect time, you can have Microsoft Query close the connection when you don't have a query window open for a database. To keep the connection open only as long as you have a query window open, click **Options** on the **Edit** menu, and then clear the **Keep connections open until MS Query is closed** check box.

Microsoft Query does not return all of the available records.

Make sure you haven't limited the number of records to return Make sure the number of records to be returned has not been restricted. Click **Options** on the **Edit** menu, and then make sure the **Limit number** of records returned to check box under **User settings** is cleared.

Check your criteria Make sure any criteria you have specified are entered correctly, so that they retrieve the records you want. For more information about specifying criteria, click \blacksquare .

Check your joins Make sure any joins you have created have the desired effect. For more information about joins, click **.**

Why can't I continue to work in Microsoft Query?

When some messages are displayed, either by the Office Assistant or without the Assistant, you may not be able to work in Microsoft Query until you close the message. These are important messages that require a response from you.

To continue to work in Microsoft Query, click an option button in the message that appears. If Microsoft Query no longer displays the message but does display a Help topic, click anywhere in the Microsoft Query window to display the message again.

Changes and formatting in my query aren't there the next time I run it.

Excel doesn't save formatting changes in queries When you return data to Microsoft Excel, you save the query definition as part of the worksheet, but you don't automatically save a separate query file. When you then edit the query for the external data range, changes to formatting and column headings that you made in Microsoft Query aren't preserved.

Formatting changes aren't saved in .dqy query files When you save a file in .dqy format, changes to formatting and column headings aren't preserved.

Formatting changes are preserved in .qry files To preserve formatting in Microsoft Query, make your formatting changes, click Save As on the File menu in Microsoft Query, and click Query Files (*.qry) in the Save as type list. When you reopen the query file, you will see your Microsoft Query formatting.

Microsoft Query won't let me print the result set.

Query has no print capability Microsoft Query is a tool that retrieves data from a database. It cannot create printed reports directly.

Print the records from Excel To print the result set, return it to Microsoft Excel by clicking **Return data to Microsoft Excel** on the **File** menu. Once you have the result set in Microsoft Excel, you can use all of the formatting and printing commands in Microsoft Excel to arrange and print the data. For more information about printing in Microsoft Excel, see Microsoft Excel Help.

Microsoft Query won't let me make mailing labels.

Query retrieves data for the labels Microsoft Query is a tool that retrieves data from a database. It cannot create mailing labels. However, you can use data retrieved by Query with Microsoft Excel and Microsoft Word to create mailing labels.

Create the labels in Word First return the data to Excel by clicking **Return data to Microsoft Excel** on the **File** menu. After you have the result set in Excel, you can use the data as a mailing list in Word. You can also start creating the labels in Word, and then return the data from Query directly to Word. For more information about mailing labels, see Excel Help or Word Help.

I can't save the result set as a table in my database.

Microsoft Query 2000 and later do not allow you to save the result set as a table in the database.

Import the data into the database from Microsoft Excel If your database can import Microsoft Excel data, you can return the result set to Microsoft Excel and then use the commands in your database to import the Microsoft Excel data as a table. For example, Microsoft Access has a spreadsheet import wizard that you can use to create a table from Microsoft Excel data. For more information, see the documentation or Help for your database.

Import a text file into the database If your database can import text files, you can return the result set to Microsoft Excel, save it as a text file, and then use the commands in your database to import the text file as a table. For information about text conversion, see Microsoft Excel Help.

I can't add, copy, or move a record in my database.

In Microsoft Query 2000 and later, you can edit and delete records that are already present in the database; but you can't change the sequence of the records and you can't use Query to add records to databases that you access by using either the Microsoft Excel driver or the Microsoft text driver.

Work directly in the database If you need to copy a record, open the database directly, and use the commands provided by the database. For information and instructions, see the administrator for your database.

Sort the records in Query You can change the order of the records in the result set by using the **Sort** command on the **Records** menu. Sorting in Microsoft Query does not affect the underlying database, only the result set. For more information, click \blacksquare .

The summary function I want for my cube data is unavailable.

When you use the OLAP Cube Wizard to create a cube from records in a relational database, the only summary functions available to calculate values for the data fields are Sum, Count, Min, and Max.

Create the report directly from the database records If you can refine your query sufficiently so that the amount of data it returns is not larger than your system can handle, and you don't need to create an OLAP cube for the data, try to return the records in the query to Microsoft Excel instead of creating a cube.

In Excel, when you create a PivotTable or PivotChart report that's based directly on your relational database query instead of being based on a cube, you'll have access to more summary functions: Sum, Count, Average, Max, Min, Product, CountNums, StDev, StDevp, Var, and Varp. For more information about how these summary functions work and how to use them, see Excel Help.

Consider setting up an OLAP server for the database The Microsoft OLAP server product, Microsoft SQL Server OLAP Services, lets you set up a wider range of summary fields than the OLAP client software included in Microsoft Office.

Date or time fields are broken out incorrectly.

Check the date or time field in the source database If you don't see the levels you expect in the date or time dimensions in your OLAP cube, check the date or time field from the underlying relational database that supplied the records for the query. The dates and times in the records might be stored as text instead of in a date or time format that the OLAP Cube Wizard can recognize. If you suspect this is the case, consult the database administrator for the relational database to verify and correct the date or time format for the field.

Check the top level of the dimension When the OLAP Cube Wizard adds date and time fields as lower levels of an existing dimension, the wizard does not automatically break the data out into year/quarter/month/week and hour/minute/second levels. The wizard does this only when you add the date or time field as the top level of a new dimension. If the date or time field is not the top level, modify the cube.

When I try to save an offline cube file, I run out of disk space.

The relational database for which you're creating a cube might be very large. If you specify a large subset of this data for your offline cube file, you might run out of space on your local hard disk.

Free up disk space or find another disk Delete any files you don't need from your disk before saving the file. Or try to save the file to a network drive.

Include less data in the offline cube file Consider how you can minimize the amount of data in the file and still have what you need for your Microsoft Excel PivotTable and PivotChart reports. Try the following:

Include only the fields you absolutely need In step 1 of the OLAP Cube Wizard, select only the data fields your reports will need to display.

Include only the dimensions you will use In step 2 of the OLAP Cube Wizard, add only the fields that your reports will display as dimensions and levels. For instructions about creating dimensions, see Help in the OLAP Cube Wizard.

Use a temporary cube instead of an offline cube file If you've eliminated all possible data and still cannot save an offline cube file, try to use the first option in step 3 of the OLAP Cube Wizard, which is the option that uses the least disk space. This option rebuilds the cube in memory when you open a report that's based on the cube, and retrieves only the data that's needed to display the report.

I can't add fields to my OLAP cube.

After you define an OLAP cube in the OLAP Cube Wizard, you cannot insert more fields from the original relational database. To add more fields, you'll need to modify the database query (.dqy) file and then recreate the cube.

Open or create the query If you saved the database query that you created when you created the cube, you can <u>open the .dqy file</u>. If you did not save the query, you can <u>create a new query</u>.

Add fields to the query Add the fields you need to the query. If you think you might need a cube with different fields in the future, <u>save a copy</u> of the query so that you don't have to re-create the query each time.

Create a new cube After you have a query with all of the fields you want for the cube, you can <u>create a new cube</u>.

I can't change my OLAP cube.

Make sure the original database is available To make changes to an OLAP cube, you must have access to the original server database that supplied the data for the cube. Check the location of this database to make sure it hasn't been renamed or moved. Make sure the server is available, and that you can connect to it.

Check for changes to the original database Tables in the database might have been renamed or deleted, so the cube can no longer locate the original data. Check for any changes to the organization, or schema, of the database. To do this, you can <u>open the database query</u> if you saved the query that you created when you first created the cube, or you can <u>create a new query</u>. On the **Table** menu, click **Add Tables** to see what tables are available. If you cannot open an existing query or create a new one, contact the database administrator for information about how the database has changed.

While changing an OLAP cube, a message tells me data has been lost.

A field you included in the cube is no longer available in the source database.

Check for changes to the original database You can <u>open the</u> <u>database query</u>, if you saved the query that you created when you first created the cube, or <u>create a new query</u>. In the query, check the Data pane to make sure all the fields you originally included are still in the query. In the Table pane, you can scroll through the fields to see what fields are available.

If you cannot open an existing query or create a new one, contact the database administrator for information about how the database has changed.

Create a new cube If fields have been removed from the database, you can <u>create a new cube</u>.

A message tells me the OLAP Cube Wizard can't open my .oqy file.

Make sure the file was originally created by the wizard You can open and modify .oqy files only if they were created and saved by using the OLAP Cube Wizard. You cannot use the wizard to open and modify other .oqy files that do not contain a cube definition, such as those that connect to an OLAP server database or an offline cube file.

Re-create the .oqy file If you can't open an .oqy file that was created by the OLAP Cube Wizard, the file might have become damaged. Try opening a backup copy of the file, or if you don't have a backup copy, <u>create the cube again</u>.

Q+E data types

Value	Туре
1	SQL_CHAR
2	SQL_BINARY, SQL_VARBINARY, SQL_LONGVARBINARY, SQL_LONGVARCHAR, SQL_VARCHAR
3	SQL_NUMERIC, SQL_DECIMAL
4	SQL_INTEGER, SQL_BIGINT
5	SQL_BIT, SQL_TYNYINT, SQL_SMALLINT
6	SQL_FLOAT
7	SQL_REAL, SQL_DOUBLE
8	SQL_TIME, SQL_DATE, SQL_TIMESTAMP

SQL data types

Value	Туре
0	SQL_UNKNOWN_TYPE
1	SQL_CHAR
2	SQL_NUMERIC
3	SQL_DECIMAL
4	SQL_INTEGER
5	SQL_SMALLINT
6	SQL_FLOAT
7	SQL_REAL
8	SQL_DOUBLE
9	SQL_DATETIME (ODBC 3.0 only)
12	SQL_VARCHAR