

Database Toolbox™

Getting Started Guide

R2012a

MATLAB®

How to Contact MathWorks



www.mathworks.com
[comp.soft-sys.matlab](mailto:comp.soft-sys.matlab@mathworks.com)
www.mathworks.com/contact_TS.html

Web
Newsgroup
Technical Support



suggest@mathworks.com
bugs@mathworks.com
doc@mathworks.com
service@mathworks.com
info@mathworks.com

Product enhancement suggestions
Bug reports
Documentation error reports
Order status, license renewals, passcodes
Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

Database Toolbox™ Getting Started Guide

© COPYRIGHT 1998–2012 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

Revision History

September 2007 Online only
March 2008 Online only
October 2008 Online only
March 2009 Online only
September 2009 Online only
March 2010 Online only
September 2010 Online only
April 2011 Online only
September 2011 Online only
March 2012 Online only

New for Version 3.4 (Release 2007b)
Revised for Version 3.4.1 (Release 2008a)
Revised for Version 3.5 (Release 2008b)
Revised for Version 3.5.1 (Release 2009a)
Revised for Version 3.6 (Release 2009b)
Revised for Version 3.7 (Release 2010a)
Revised for Version 3.8 (Release 2010b)
Revised for Version 3.9 (Release 2011a)
Revised for Version 3.10 (Release 2011b)
Revised for Version 3.11 (Release 2012a)

Using the Database Toolbox Software

1

Product Description	1-2
Key Features	1-2
Product Prerequisites	1-3
Software Requirements	1-3
Background Knowledge	1-3
Configuring Your Environment	1-4
About Data Sources and Database Drivers	1-4
Before You Begin	1-5
Preparing Examples	1-5
Setting Up ODBC Data Sources	1-7
Setting Up JDBC Data Sources	1-12
Starting the Database Toolbox and Visual Query Builder Software	1-16
Database Toolbox Workflows	1-17
Importing Database Data into the MATLAB Workspace ..	1-17
Exporting Data from the MATLAB Workspace to a Database	1-17
Database Toolbox Functions Versus Visual Query Builder	1-18
Limitations of Visual Query Builder	1-18
Working with Visual Query Builder	1-20
Using Queries to Import Database Data into the MATLAB Workspace	1-20
Saving Queries	1-25
Running Saved Queries	1-26
Editing Queries	1-26
Clearing Variables from the VQB Data Area	1-26

Using Queries to Export MATLAB Workspace Data to a Database	1-27
Exiting Visual Query Builder	1-30
Learning More	1-31
Next Steps	1-31
Product Help	1-31
MathWorks Online	1-32

Index



Using the Database Toolbox Software

- “Product Description” on page 1-2
- “Product Prerequisites” on page 1-3
- “Configuring Your Environment” on page 1-4
- “Starting the Database Toolbox and Visual Query Builder Software” on page 1-16
- “Database Toolbox Workflows” on page 1-17
- “Database Toolbox Functions Versus Visual Query Builder” on page 1-18
- “Working with Visual Query Builder” on page 1-20
- “Learning More” on page 1-31

Product Description

Exchange data with relational databases

The Database Toolbox™ product enables you to exchange data between MATLAB® and any ODBC/JDBC-compliant database. With the Visual Query Builder tool within the toolbox, you can query stored data without needing to know or learn SQL. This gives you the ability to access, analyze, and store your data quickly and easily from within MATLAB.

Key Features

- Connects MATLAB to ODBC/JDBC-compliant databases, including Oracle®, Microsoft™ SQL Server™, Sybase®, Microsoft Access™, Informix®, and Ingres®
- Directly executes SQL queries against a database from within MATLAB and preserves data types during all data import and export activities
- Lets you import data from one or multiple databases, perform calculations on that data, and then export the data to another database
- Retrieves large data sets in a single transaction or through multiple transactions by discretely segmenting the data
- Streamlines data import and export by keeping the database connection open during the MATLAB session
- Enables you to access and query a database without knowing SQL

Product Prerequisites

In this section...
“Software Requirements” on page 1-3
“Background Knowledge” on page 1-3

Software Requirements

- The Database Toolbox product requires that the MATLAB software be installed.
- This toolbox requires your system to have access to a supported database. For a list of supported databases, see “Supported Databases” in the Database Toolbox User’s Guide documentation.

Background Knowledge

The following technical knowledge is required to use this toolbox:

- This documentation assumes that you know how to work with MATLAB cell arrays and structures.
- You must be familiar with basic database concepts, such as tables, attributes, and queries.
- You do not need to be familiar with Structured Query Language (SQL) to use Database Toolbox functions and Visual Query Builder. However, you do need to be familiar with SQL to perform complex queries and database operations.

Configuring Your Environment

In this section...
“About Data Sources and Database Drivers” on page 1-4
“Before You Begin” on page 1-5
“Preparing Examples” on page 1-5
“Setting Up ODBC Data Sources” on page 1-7
“Setting Up JDBC Data Sources” on page 1-12

About Data Sources and Database Drivers

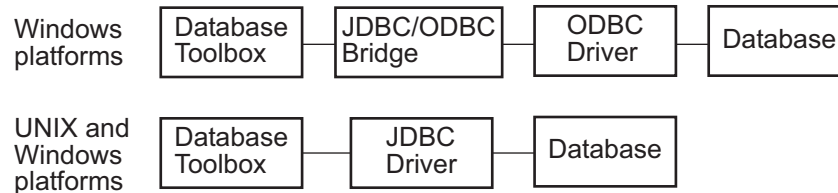
Before you can use the Database Toolbox software to connect to a database, you must set up a *data source*. A data source consists of:

- Data that the toolbox accesses
- Information required to find the data, such as driver, folder, server, or network names

Data sources interact with *ODBC drivers* or *JDBC drivers*. An ODBC driver is a standard Microsoft Windows® interface that enables communication between database management systems and SQL-based applications. A JDBC driver is a standard interface that enables communication between applications based on Sun™ Java™ and database management systems.

The Database Toolbox software is based on Java. It uses a JDBC/ODBC bridge to connect to the ODBC driver of a database, which is automatically installed as part of the MATLAB JVM™.

The following figure illustrates how drivers interact with the Database Toolbox software.



Tip Some Windows systems support both ODBC and JDBC drivers. On such systems, JDBC drivers generally provide better performance than ODBC drivers because the ODBC/JDBC bridge is not used to access databases.

Before You Begin

Before you can use the Database Toolbox software with the examples provided in this documentation, do the following:

- 1 Set up the data sources that are provided with the Database Toolbox product. For instructions, see “Preparing Examples” on page 1-5.
- 2 Configure the data sources for use with your database driver.
 - If you are using an ODBC driver, see “Setting Up ODBC Data Sources” on page 1-7.
 - If you are using a JDBC driver, see “Setting Up JDBC Data Sources” on page 1-12.

Preparing Examples

Setting Up the `dbtoolboxdemo` Data Source

The `dbtoolboxdemo` data source uses the tutorial database located in `matlabroot/toolbox/database/dbdemos/tutorial.mdb`.

To set up this data source:

- 1 Copy `tutorial.mdb` into a folder to which you have write access.
- 2 Confirm you have write access to `tutorial.mdb`.
- 3 Open `tutorial.mdb` from within the MATLAB Current Folder browser by right-clicking the file and selecting **Open Outside MATLAB**. The file opens in Microsoft Access.

Note You might need to convert the database to the version of Access you are currently running. For example, beginning in Microsoft Access 2007, you will see the option to save as `*.accdb`. For more information, consult your database administrator.

Setting Up the SampleDB Data Source

The SampleDB data source uses a Microsoft Access sample database called Nwind. If you do not have the Nwind database on your system, download it from the Microsoft downloads page at <http://www.microsoft.com/downloads/>. The version referred to in this documentation is the Northwind Traders sample database, `Nwind.exe`.

To set up this data source:

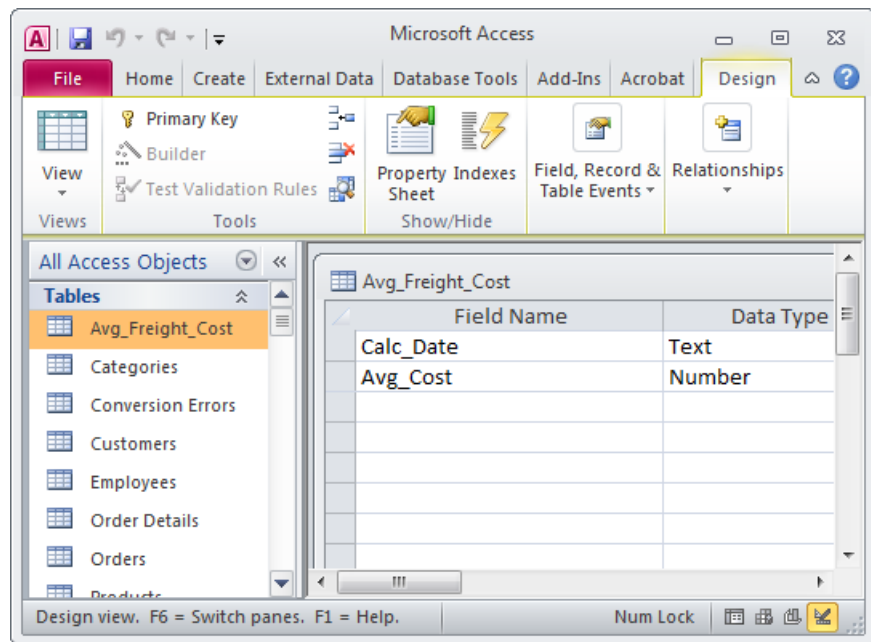
- 1 Run `Nwind.exe` to create the `Nwind.mdb` database.
- 2 Confirm you have write access to `Nwind.mdb`.
- 3 Open `Nwind.mdb` from within the MATLAB Current Folder browser by right-clicking the file and selecting **Open Outside MATLAB**.

Note When using Microsoft Access 2007 or later, the file might convert to `Nwind.accdb`.

- 4 Create a table in the Nwind database into which to export MATLAB data:
 - a Open the Nwind database in the Access application.
 - b Create a new table and save it as `Avg_Freight_Cost`.

Note The Microsoft Access database warns you that there is no primary key, but you do not need one. If you choose to designate a primary key, you can run the example as written only once. Access prevents you from inserting the same record into the database twice.

- c Add two fields, Calc_Date and Avg_Cost.
- d Use the default type, which is Text, for the Calc_Date **Data Type** field.



- e Set the **Data Type** of the Avg_Cost field to Number.
- f Save the table and close it.

Setting Up ODBC Data Sources

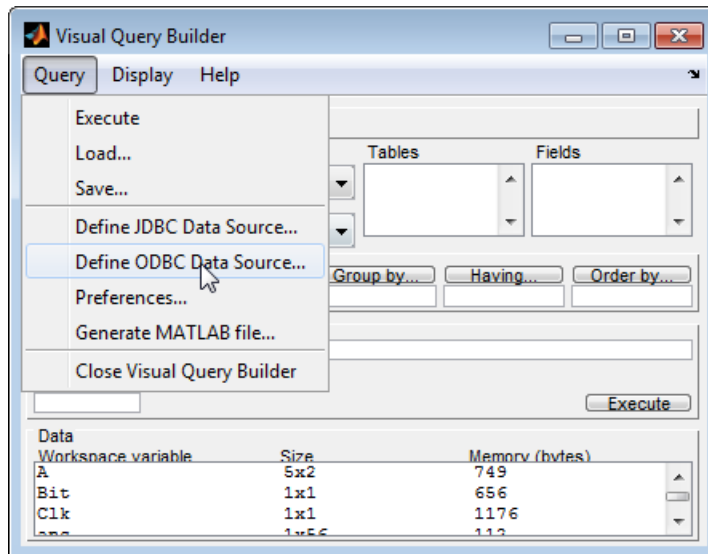
This section describes how to set up a data source whose database is located on a PC running the Windows operating system, or on another system to which the PC is networked, for use with an ODBC driver. These instructions use the Microsoft ODBC Data Source Administrator Version 6.1 for the U.S.

English version of Microsoft Access 2010 for Windows systems. If you have a different configuration, you might need to modify these instructions. For more information, consult your database administrator.

- 1 Close open databases, including tutorial.mdb and Nwind.mdb (or Nwind.accdb) in the database program.
- 2 Open the Windows Data Source Administrator dialog box to define the ODBC data source:
 - a Start Visual Query Builder by entering the command querybuilder at the MATLAB command prompt.

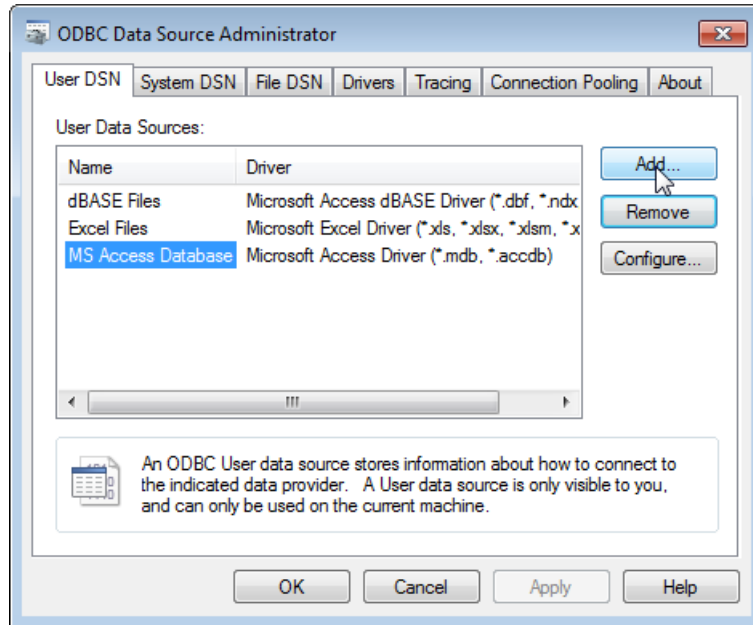
Requirement When using a 32-bit version of Microsoft Office, you must also use a 32-bit version of MATLAB to complete the remaining steps.

- b In Visual Query Builder, select **Query > Define ODBC Data Source**.



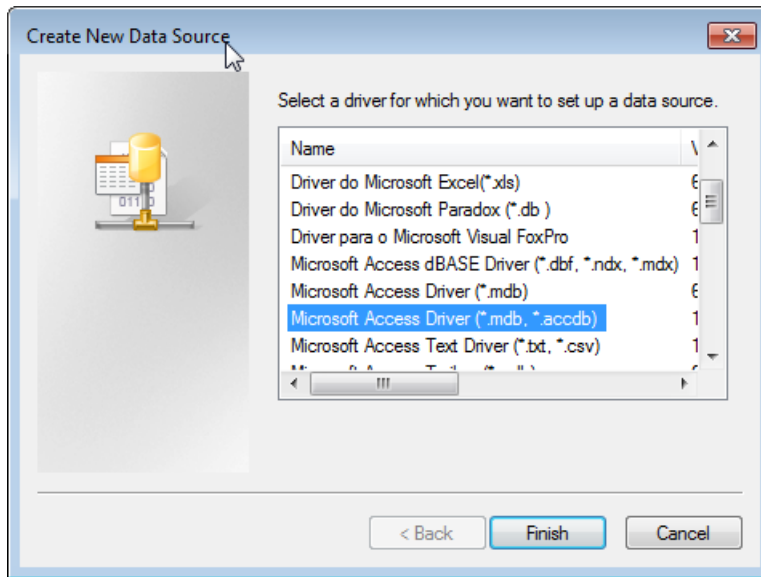
The ODBC Data Source Administrator dialog box appears, listing existing data sources.

- 3 Click the **User DSN** tab.
- 4 Select MS Access Database and click **Add**.

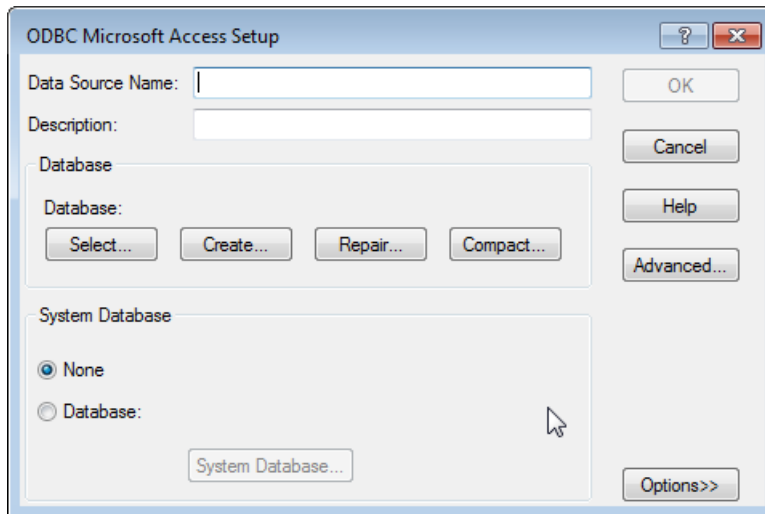


A list of installed ODBC drivers appears in the ODBC Data Source Administrator dialog box.

- 5 Select Microsoft Access Driver (*.mdb, *.accdb) and click **Finish**.



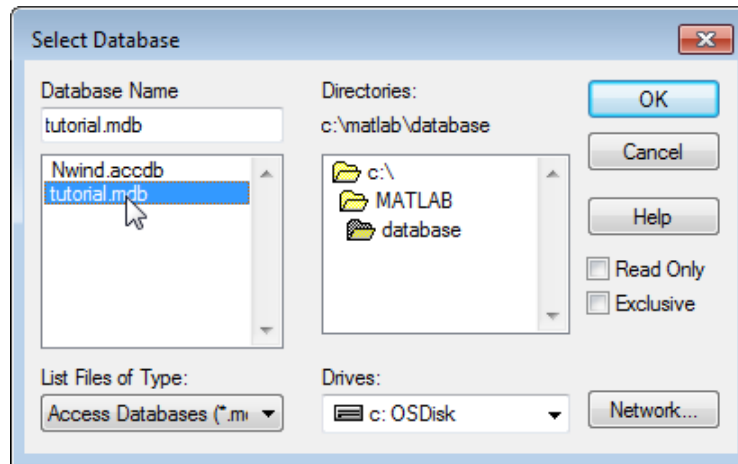
The ODBC Microsoft Access Setup dialog box for your driver appears. The dialog box for your driver might look different from the following.



6 Enter dbtoolboxdemo as the **Data Source Name**.

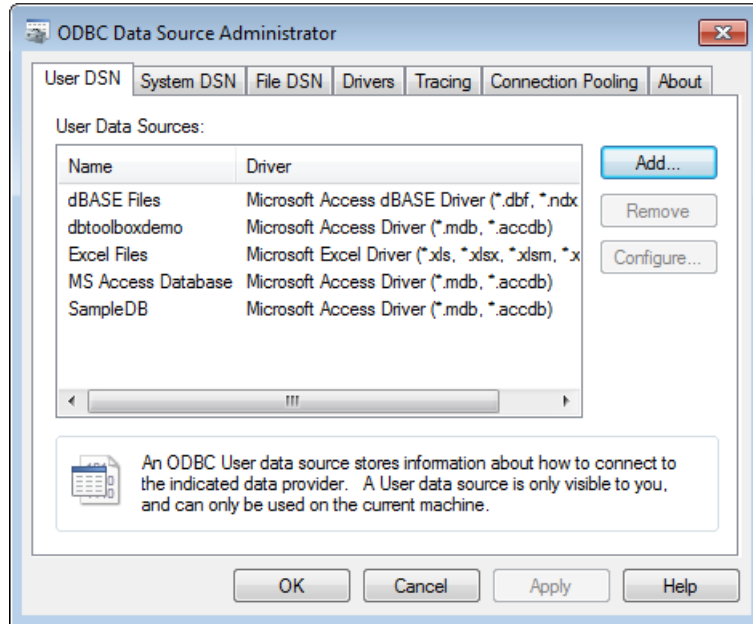
- 7 Enter tutorial database as the **Description**.
- 8 Select the database for this data source to use. For some drivers, you can skip this step. If you are not sure whether you can skip this step, consult your database administrator.
 - a In the ODBC Microsoft Access Setup dialog box, click **Select**.

The Select Database dialog box appears.



- b Specify the database you want to use. For the dbtoolboxdemo data source, select tutorial.mdb.
 - c If your database is on a system to which your PC is connected:
 - i Click **Network**. The Map Network Drive dialog box appears.
 - ii Specify the folder containing the database you want to use.
 - iii Click **Finish**.
 - d Click **OK** to close the Select Database dialog box.
- 9 In the ODBC Microsoft Access Setup dialog box, click **OK**.
- 10 Repeat steps 6 through 9 with the following changes to define the data source for the other sample database, Nwind:
 - a In step 6, enter SampleDB as the **Data Source Name**.

- b** In step 7, enter Northwind database as the **Description**.
 - c** In step 8, select Nwind.mdb or Nwind.accdb and then click **OK**.
- 11** The ODBC Data Source Administrator dialog box now displays the dbtoolboxdemo and SampleDB data sources in the **User DSN** tab.



- 12** Click **OK** to close the dialog box.

Setting Up JDBC Data Sources

- 1** Find the name of the file that contains the JDBC drivers. This file is provided by your database vendor. The name and location of this file differs for each system. If you do not know the name or location of this file, consult your database administrator.
- 2** Specify the location of the drivers file in the MATLAB Sun Java classpath by adding this file's path to the `matlabroot/toolbox/local/classpath.txt` file.

For example, use the following syntax to add a reference to a MySQL[®] JDBC drivers file, `D:/mysql/mysql-connector-java-3.0-bin.jar`, to `classpath.txt`:

```
D:/mysql/mysql-connector-java-3.0-bin.jar
```

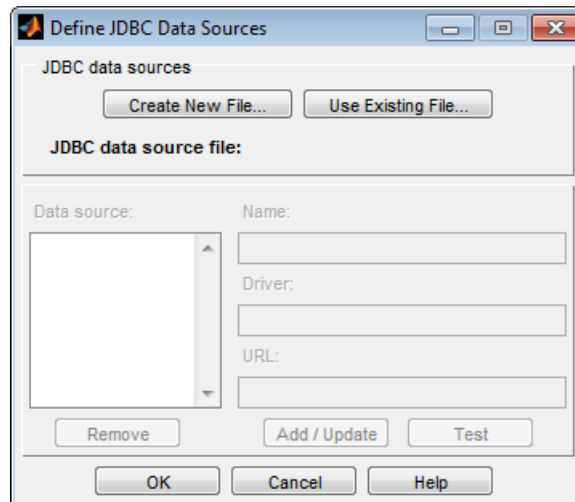
Restart the MATLAB session before you access the database.

If the drivers file is not located where `classpath.txt` indicates, errors do not appear, but MATLAB does not establish a database connection. Make sure that you update `classpath.txt` with the correct location and file name for your drivers file if the path and/or name of this file changes. If a MATLAB session is running while you modify `classpath.txt`, be sure to restart it before you connect to the database.

3 To set up JDBC drivers for use with Visual Query Builder:

- a** Run the `querybuilder` command in the MATLAB Command Window to start Visual Query Builder.
- b** Select **Query > Define JDBC Data Source**.

The Define JDBC data sources dialog box appears.



The following table describes the fields that you use to define your JDBC data source. For examples of syntax used in these fields, see “JDBC Driver Name and Database URL” on the database function reference page.

Field Name	Description
Name	The name you assign to the data source. For some databases, the Name must match the name of the database as recognized by the machine it runs on.
Driver	The JDBC driver name (sometimes referred to as the class that implements the Java SQL driver for your database).
URL	The JDBC URL object, of the form <code>jdbc:subprotocol:subname</code> . <i>subprotocol</i> is a database type; for example, the Oracle Corporation Oracle database. <i>subname</i> may contain other information used by Driver , such as the location of the database and/or a port number. It may take the form <code>//hostname:port/databasename</code> .

Note Your driver manufacturer’s documentation specifies the **Driver** and **URL** formats. You may need to consult your database system administrator for this information.

- c** In the Define JDBC data sources dialog box, click **Create new file**.
- d** The Specify new JDBC data source MAT-file dialog box appears. Use this dialog box to create a MAT-file that saves your specified data source information for future Visual Query Builder sessions.

Navigate to the folder where you want to put the MAT-file, specify a name for it that includes a `.mat` extension, and click **Save**.

- e In the Define JDBC data sources dialog box, enter information for your JDBC data source into the **Name**, **Driver**, and **URL** fields as described in the table in step 3b.
- f Test the connection by clicking **Test**.

If your database requires a user name and password, a dialog box prompting you to supply them appears. Enter values into these fields and click **OK**.

A confirmation dialog box states that the database connection was successful.

- g In the Define JDBC data sources dialog box, click **Add/Update**. The data source now appears in the **Data source** list.
- h To add more data sources, repeat steps c through g for each new data source. You can add more data sources later by editing the MAT-file that contains the data source information.

Note You can create another MAT-file to add new data sources, but Visual Query Builder can only access data sources from one MAT-file at a time. Include multiple data sources in a single MAT-file for easy access.

- i Click **OK** to close the Define JDBC data sources dialog box.

The newly added data source now appears in the VQB **Data source** list.

The defined JDBC data source is only valid for the current MATLAB session. To access the data source you defined in a future MATLAB session, follow the instructions in “Accessing Existing JDBC Data Sources” in the Database Toolbox User’s Guide documentation.

Starting the Database Toolbox and Visual Query Builder Software

To use the Database Toolbox software, enter its functions into the MATLAB Command Window. For example, to start Visual Query Builder, enter the `querybuilder` function into the MATLAB Command Window.

Alternatively, start Visual Query Builder from the MATLAB desktop by selecting **Start > Toolboxes > Database > Visual Query Builder**. For more information about Visual Query Builder, see “Using Visual Query Builder” in the Database Toolbox User’s Guide documentation.

Database Toolbox Workflows

In this section...
“Importing Database Data into the MATLAB Workspace” on page 1-17
“Exporting Data from the MATLAB Workspace to a Database” on page 1-17

Importing Database Data into the MATLAB Workspace

Here is the workflow for importing data from a database into the MATLAB workspace:

- 1 Construct a query to import data from a data source.
- 2 Define a MATLAB variable to store the results of the query.
- 3 Run the query.
- 4 (Optional) Use MATLAB functions to manipulate the data.

For more details about how to import database data into the MATLAB workspace, see “Using Queries to Import Database Data into the MATLAB Workspace” on page 1-20.

Exporting Data from the MATLAB Workspace to a Database

Here is the workflow for exporting data from the MATLAB workspace to a database:

- 1 Construct a query to export data to a data source.
- 2 Specify the MATLAB variable that contains the data to export.
- 3 Run the query.

For more details about how to export data from the MATLAB workspace to a database, see “Using Queries to Export MATLAB Workspace Data to a Database” on page 1-27.

Database Toolbox Functions Versus Visual Query Builder

The table describes tasks you can perform with Database Toolbox functions and tasks you can perform with Visual Query Builder.

Database Toolbox Functions	Visual Query Builder
Retrieve large data sets or partial data sets in a single <code>fetch</code> command, or in discrete amounts using multiple fetches.	Import data from relational databases into the MATLAB workspace by selecting information from lists to build queries.
Dynamically import data into the MATLAB workspace.	Display retrieved information in relational tables, reports, and charts.
Modify SQL queries in MATLAB statements.	Easily build SQL queries and exchange data between databases and the MATLAB workspace.
Write MATLAB files and applications that access databases.	View and edit SQL statements for queries generated with VQB.
Replace existing records in databases with data from the MATLAB workspace.	Automatically generate a MATLAB file that consists of Database Toolbox functions that perform queries you built using VQB.
Export binary data or other data types.	Export data from the MATLAB workspace into new records in a database.
Access database metadata.	

Limitations of Visual Query Builder

You can use Visual Query Builder for most, but not all, of the tasks you can do with Database Toolbox functions. The limitations of Visual Query Builder are as follows:

- You cannot use Visual Query Builder to replace existing data in a database with data from the MATLAB workspace. Use the `update` function instead.

- You cannot use Visual Query Builder to export binary data. Instead, use the `fastinsert` function.
- You cannot use Visual Query Builder to access database metadata.

Working with Visual Query Builder

In this section...

“Using Queries to Import Database Data into the MATLAB Workspace” on page 1-20

“Saving Queries” on page 1-25

“Running Saved Queries” on page 1-26

“Editing Queries” on page 1-26

“Clearing Variables from the VQB Data Area” on page 1-26

“Using Queries to Export MATLAB Workspace Data to a Database” on page 1-27

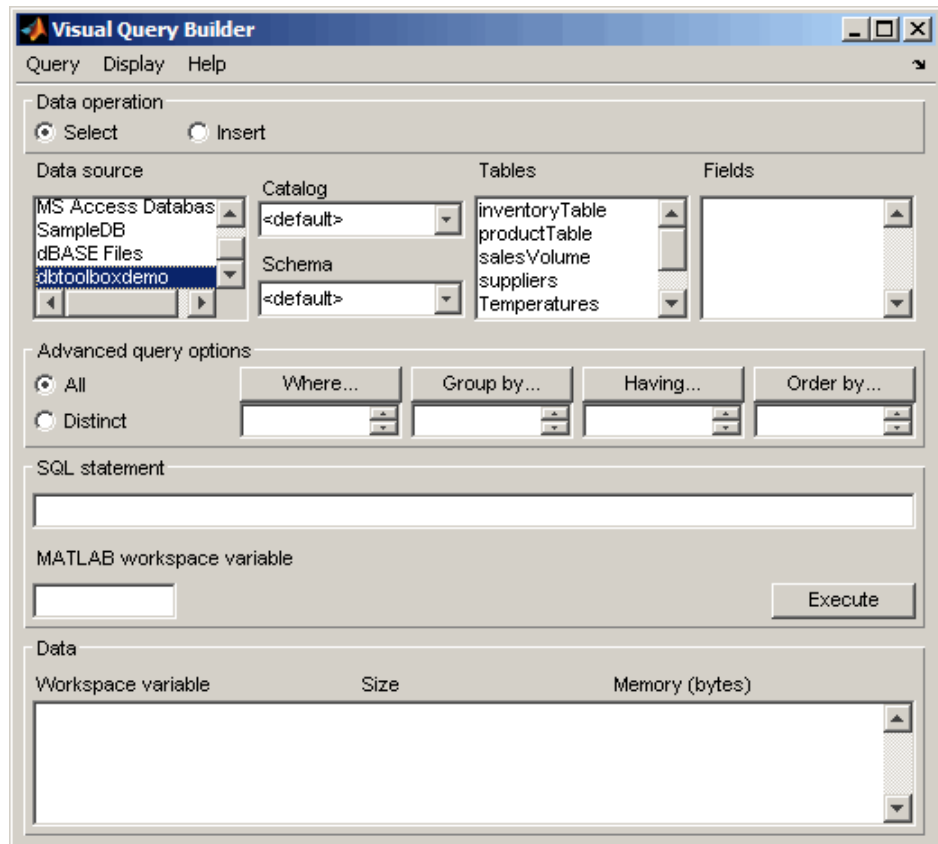
“Exiting Visual Query Builder” on page 1-30

Using Queries to Import Database Data into the MATLAB Workspace

To create and run a query to import data from a database into the MATLAB workspace:

- 1 Select data from a database by clicking the **Select** button under **Data operation**. The data sources that you defined in “Configuring Your Environment” on page 1-4 appear.
- 2 Select dbtoolboxdemo as the data source from which to import data.

After you select a data source, the catalog, schema, and tables for your specified data source appear in the **Catalog**, **Schema**, and **Tables** fields.



- 3 Accept the default values `<default>` for the **Catalog** and **Schema** fields. Setting these fields to the default values indicates that you have not specified a catalog or schema.

Tip To populate the **VQB Schema** and **Catalog** fields, you must associate your user name with schemas or catalogs before starting VQB.

- To specify a **Catalog**, select one from the list, and then select a schema from within that catalog. The **Schema** field updates to reflect your selections.
 - Alternatively, you can select a schema without specifying a catalog; that is, when the **Catalog** field set to <default>. The **Tables** field updates to reflect the schema you selected.
-

4 In the **Tables** list, select `salesVolume` as the table that contains the data you want to import.

The set of **Fields** (column names) in the table appears.

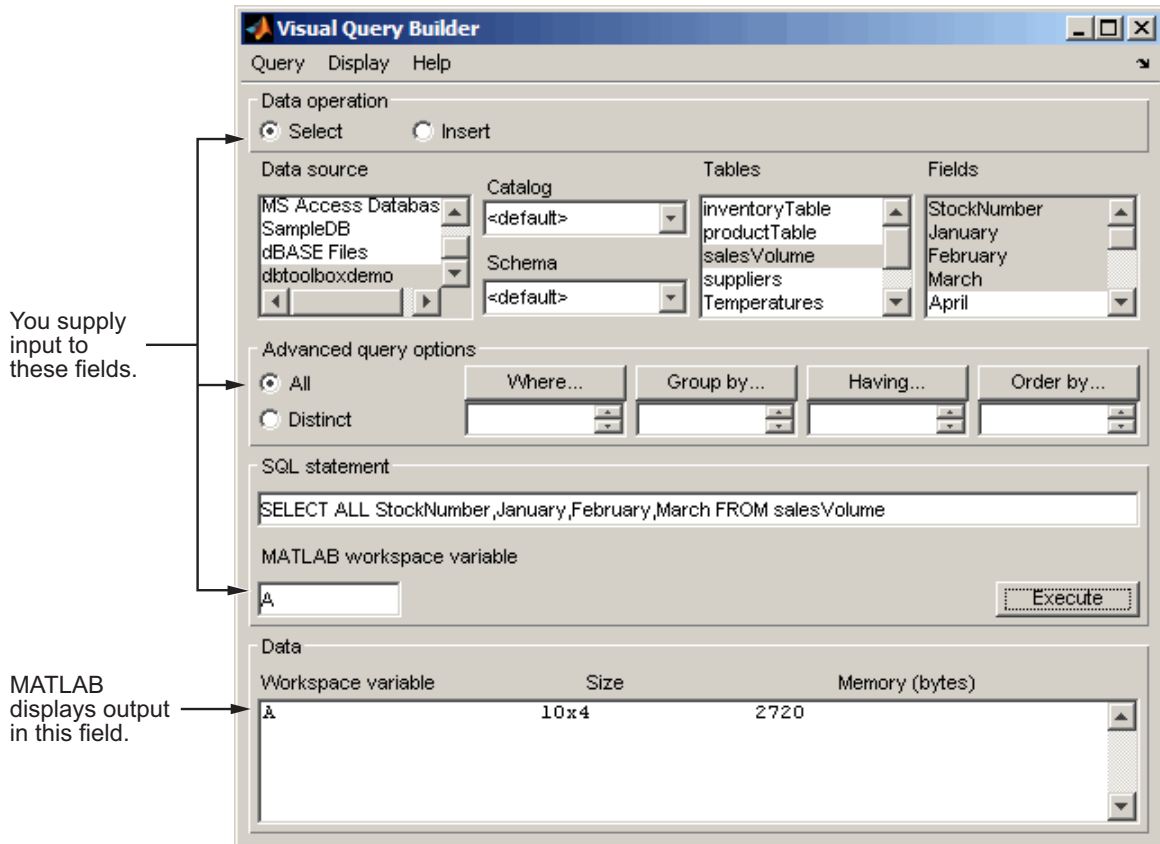
5 In the **Fields** list, select `StockNumber`, `January`, `February`, and `March` as the fields that contain the data you want to import.

Tip To select more than one field, hold down the **Ctrl** or **Shift** key while selecting multiple fields. To clear an entry, use **Ctrl+click**.

VQB adds each field you select to the query in the **SQL statement** field.

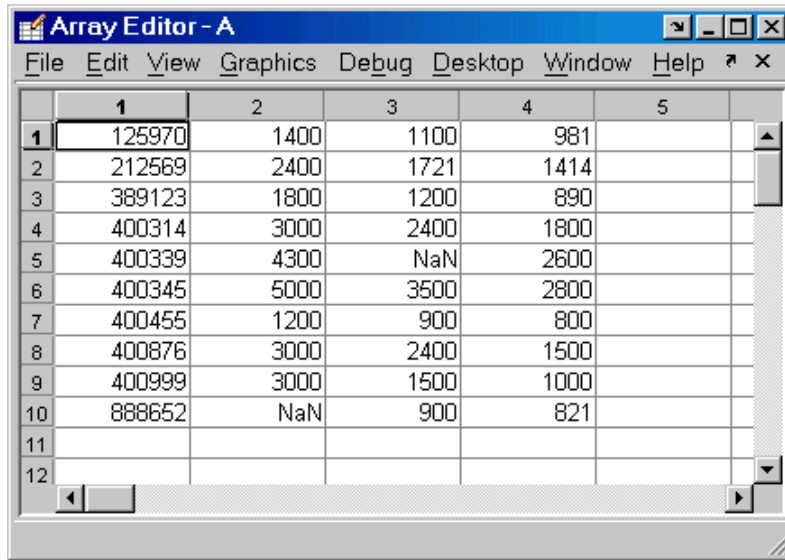
6 Enter the name `A` in the **MATLAB workspace variable** field. `A` is a cell array that stores the data that the query returns.

7 Click **Execute** to run the query and import the data. The **Data** field displays information about the query result.



- 8 Double-click A in the **Data** area. The contents of A appear in the Variable Editor, where you can view and edit the data. In this example, sales for item 400876 are 3000 in January, 2400 in February, and 1500 in March.

For more information about using the Variable Editor, see “Viewing and Editing Workspace Variables with the Variable Editor” in the MATLAB documentation.



The screenshot shows the MATLAB Array Editor window titled "Array Editor - A". The window contains a table with 12 rows and 5 columns. The columns are labeled 1 through 5, and the rows are labeled 1 through 12. The data in the table is as follows:

	1	2	3	4	5
1	125970	1400	1100	981	
2	212569	2400	1721	1414	
3	389123	1800	1200	890	
4	400314	3000	2400	1800	
5	400339	4300	NaN	2600	
6	400345	5000	3500	2800	
7	400455	1200	900	800	
8	400876	3000	2400	1500	
9	400999	3000	1500	1000	
10	888652	NaN	900	821	
11					
12					

Alternatively, you can view the contents of A by entering A in the MATLAB Command Window.

```

>> A

A =

    [125970]    [1400]    [1100]    [ 981]
    [212569]    [2400]    [1721]    [1414]
    [389123]    [1800]    [1200]    [ 890]
    [400314]    [3000]    [2400]    [1800]
    [400339]    [4300]    [ NaN]    [2600]
    [400345]    [5000]    [3500]    [2800]
    [400455]    [1200]    [ 900]    [ 800]
    [400876]    [3000]    [2400]    [1500]
    [400999]    [3000]    [1500]    [1000]
    [888652]    [ NaN]    [ 900]    [ 821]

>>

```

Saving Queries

- 1 Select **Query > Save**. The Save SQL Statement dialog box appears.
- 2 Enter a name (without spaces) for the query into the **File name** field and click **Save**. Save the query as `basic.qry`.

Note When you save a **Select** query (a query that imports data), MATLAB does not save your specified preferences or the workspace variable that contains the query results. This prevents you from inadvertently overwriting an existing variable in the MATLAB workspace when you run a saved query.

When you save an **Insert** query (a query that exports data), MATLAB saves the workspace variable whose data you exported, but does not save your preferences.

Running Saved Queries

- 1 Select **Query > Load**. The Load SQL Statement dialog box appears.
- 2 Select the name of the query you want to load and click **Open**. The VQB fields reflect the values for the saved query.
- 3 Run a **Select** query to import data into the MATLAB workspace, or an **Insert** query to export data from the MATLAB workspace.
 - To run a **Select** query, use the **MATLAB workspace variable** field to assign a variable to the data and click **Execute**.
 - For an **Insert** query, the saved query may include a workspace variable, which appears as part of the **MATLAB command** field. Type that variable name or enter a new name in the **MATLAB workspace variable** field. Press **Return** or **Enter** to see the updated **MATLAB command**.
- 4 Click **Execute** to run the query.

Tip You can generate a file that runs the query from the MATLAB Command Window in the future. For more information, see “Saving Queries in Files” in the Database Toolbox User’s Guide documentation.

Editing Queries

Edit a query using one of the following options:

- Changing your selections
- Editing the **SQL statement** field
- Editing the **MATLAB command** field

Clearing Variables from the VQB Data Area

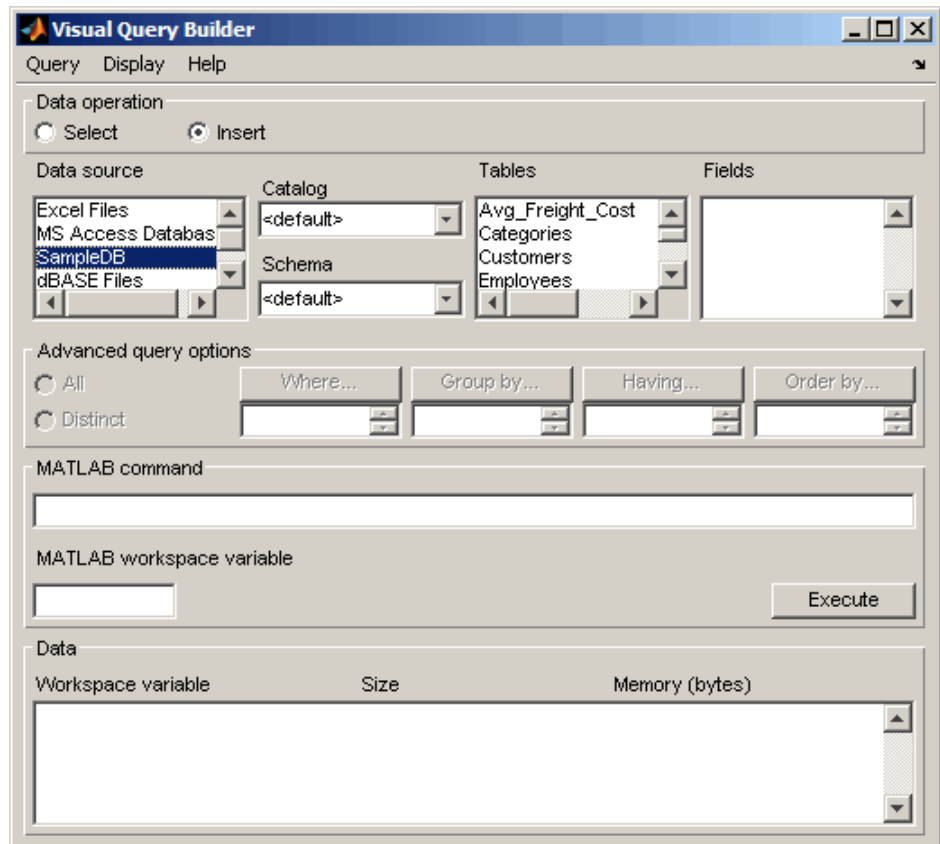
Variables in the **Data** area include those you assigned in the Command Window and those that contain query results. The variables do not appear in the **Data** area until you execute a query. They then remain in the **Data**

area until you clear them. To clear the variables, run the `clear` function in the Command Window.

Using Queries to Export MATLAB Workspace Data to a Database

To build, run, and save a query that exports data from the MATLAB workspace into new rows in a database:

- 1 Select **Data Operation** > **Insert** to select data to export.
- 2 Select **SampleDB** as the data source to which to export data from the **Data source** list box. The **Catalog**, **Schema**, and **Tables** fields for **SampleDB** appear.



3 Do not specify values for **Catalog** and **Schema**.

4 In the **Tables** list box, select `Avg_Freight_Cost` as the table to which you want to export data from the MATLAB software.

The set of **Fields** (column names) in your selected table appears.

5 In the **Fields** list box, select `Calc_Date` and `Avg_Cost` as the fields to which you want to export data from the MATLAB software.

VQB adds each field you select to the query in the **MATLAB command** field.

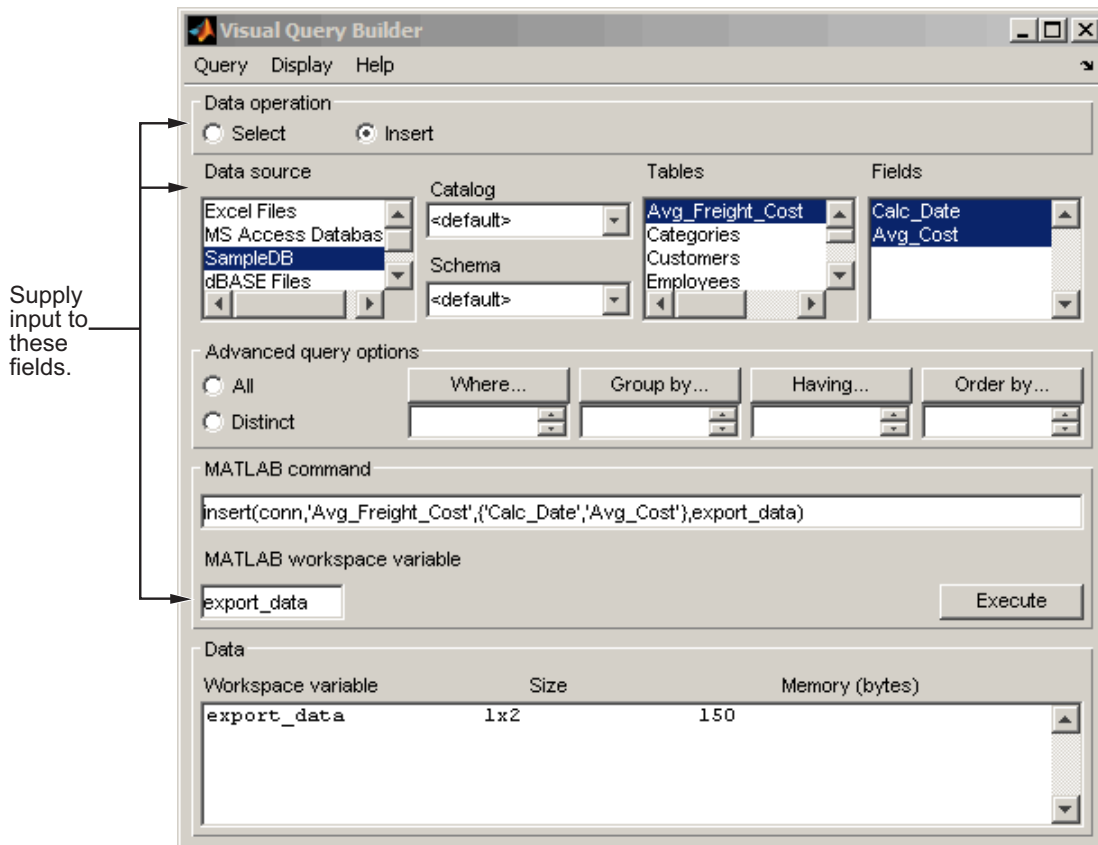
- 6** In the MATLAB workspace, assign the data you want to export to a cell array, `export_data`.

```
export_data = {'07-Aug-2003',50.44};
```

- 7** In the **MATLAB workspace variable** field, enter the name of the variable containing data to export, `export_data`. Press **Enter** or **Return** to view the **MATLAB command** that exports the data.

- 8** Click **Execute** to run the query to export the data.

Information about the exported data appears in the **Data** area.



- 9 View the Avg_Freight_Cost table in the Microsoft Access database to verify the query results.



	Calc_Date	Avg_Cost
	07-Aug-2003	50
▶		0

Record: 2 of

The Avg_Cost value was rounded to a whole number to match the properties of that field in the Access software.

- 10 To save this query, select **Query > Save** and name it `export.qry`.

Exiting Visual Query Builder

To exit Visual Query Builder, select **Query > Exit**.

Learning More

In this section...
“Next Steps” on page 1-31
“Product Help” on page 1-31
“MathWorks Online” on page 1-32

Next Steps

To learn more, see the following topics in the *Database Toolbox User’s Guide*.

To...	See...
Learn about Visual Query Builder	“Using Visual Query Builder”
Learn about Database Toolbox functions	“Using Database Toolbox Functions”
Get information about system requirements, including supported platforms and databases	“Working with Databases”
Develop advanced queries	“Additional Advanced Query Options”

Product Help

More information is available with your product installation. In the MATLAB toolbar, select **Help > Product Help** for help, and then click **Database Toolbox** in the **Contents** pane.

For...	See...
List of functions	“Function Reference”
Advanced tutorials	Examples
Product demonstrations	Database Toolbox Demos
What’s new in the Database Toolbox	Database Toolbox Release Notes

MathWorks Online

For additional information and support, go to the MathWorks® Database Toolbox web site at <http://www.mathworks.com/products/database/>.

C

clearing variables from Data area 1-26

D

Data area in VQB

clearing 1-26

example 1-22

data sources

about 1-4

defining

local ODBC 1-7

JDBC

defining 1-12

selecting for VQB 1-20

Database Toolbox

starting 1-16

drivers

about 1-4

E

editing queries 1-26

examples

setting up 1-5

executing queries

VQB 1-22

exiting

Visual Query Builder 1-30

exporting data

using VQB 1-27

F

feature 1-16

fields

selecting for VQB 1-22

I

importing data

using VQB 1-20

J

JDBC

data sources

defining for VQB 1-13

L

loading saved queries 1-26

M

MATLAB

workspace variables in VQB 1-22

matlabroot 1-5

multiple entries, selecting 1-22

O

ODBC

data sources, setting up 1-7

Q

queries

creating with VQB 1-20

editing 1-26

executing 1-22

executing for export 1-29

exporting with VQB 1-27

loading saved queries 1-26

results

VQB 1-22

saving 1-25

R

results

from query 1-22

- viewing 1-23
- retrieving data
 - from database 1-20
- running queries 1-22

S

- saving queries 1-25
- selecting multiple entries in VQB 1-22
- speed
 - database access 1-5
- starting
 - Database Toolbox 1-16

T

- tables

- selecting for VQB 1-22

V

- Visual Query Builder
 - exiting 1-30
 - JDBC data sources, defining 1-13
 - limitations 1-18
 - ODBC data sources, defining 1-7
 - starting 1-16

W

- workspace variables in VQB 1-22
 - clearing from Data area 1-26