# MATLAB

## Database Toolbox

# *What Is the Database Toolbox?*

- The Database Toolbox is one of an extensive collection of toolboxes for use with MATLAB . The Database Toolbox enables you to move data (both importing and exporting) between MATLAB and popular relational databases .
- With the Database Toolbox, you can bring data from an existing database into MATLAB, use any of MATLAB’s computational and analytic tools, and store the results back in the database or in another database. You read from the database, importing the data into the MATLAB workspace.
- For example, a financial analyst working on a mutual fund could import a company’s financial data into MATLAB, run selected analyses, and store the results for future tracking. The analyst could then export the saved results to a database.

# How Databases Connect to MATLAB?

- The Database Toolbox connects MATLAB to a database using MATLAB functions. Data is retrieved from the database as a string, parsed into the correct data types, and stored in a MATLAB cell array. At that point, you use MATLAB’s extensive set of tools to work with the data. You can include Database Toolbox functions in MATLAB M-files. To export the data from MATLAB to a database, you use MATLAB functions.
- The Database Toolbox also comes with the Visual Query Builder (VQB), an easy-to-use graphical user interface for retrieving data from your database. With the VQB, you build queries to retrieve data by selecting information from lists rather than by entering MATLAB functions. The VQB retrieves the data into a MATLAB cell array so you then can process the data using MATLAB’s suite of functions. With the VQB, you can display the retrieved information in relational tables, reports, and charts.

# Features of the Database Toolbox

- Data types are automatically preserved in MATLAB – No data massaging or manipulation is required. The data is stored in cell arrays, which support mixed data types.
- Different databases can be used in a single session – Import data from one database, perform calculations, and export the modified or unmodified data to another database. Multiple databases can be open during a session.
- Database connections remain open until explicitly closed – Once connection to a database has been established, it remains open during the entire MATLAB session until you explicitly close it. This improves access and reduces the number of functions necessary to import/export data.
- Retrieval of large data sets or partial data sets – You can retrieve large data sets from a database in a single fetch or in discrete amounts using multiple fetches.

# *Features of the Database Toolbox*

- Retrieval of database metadata – You do not need to know the table names, field names, and properties of the database structure to access the database, but can retrieve that information using Database Toolbox functions.
- Visual Query Builder – If you are unfamiliar with SQL, you can retrieve information from databases via this easy-to-use graphical interface.

# *Installing the Database Toolbox*

- **Setting Up a Data Source :**

Before you can connect from the Database Toolbox to a database, you need to set up a data source, such as driver, directory, server, or network names. You assign a name to each data source.

- **The instructions for setting up a data source differ slightly depending on your configuration. Use one of these sets of instructions:**

- For MATLAB PC platforms whose database resides on that PC, use “Setting Up a Local Data for ODBC Drivers” on page 1-6.
- For MATLAB PC platforms whose database resides on another system to which the PC is networked, use “Setting Up a Remote Data Source for ODBC Drivers” on page 1-8.
- For MATLAB platforms that connect to a database via a JDBC driver, use “Setting Up a Data for JDBC Drivers” on page 1-12.

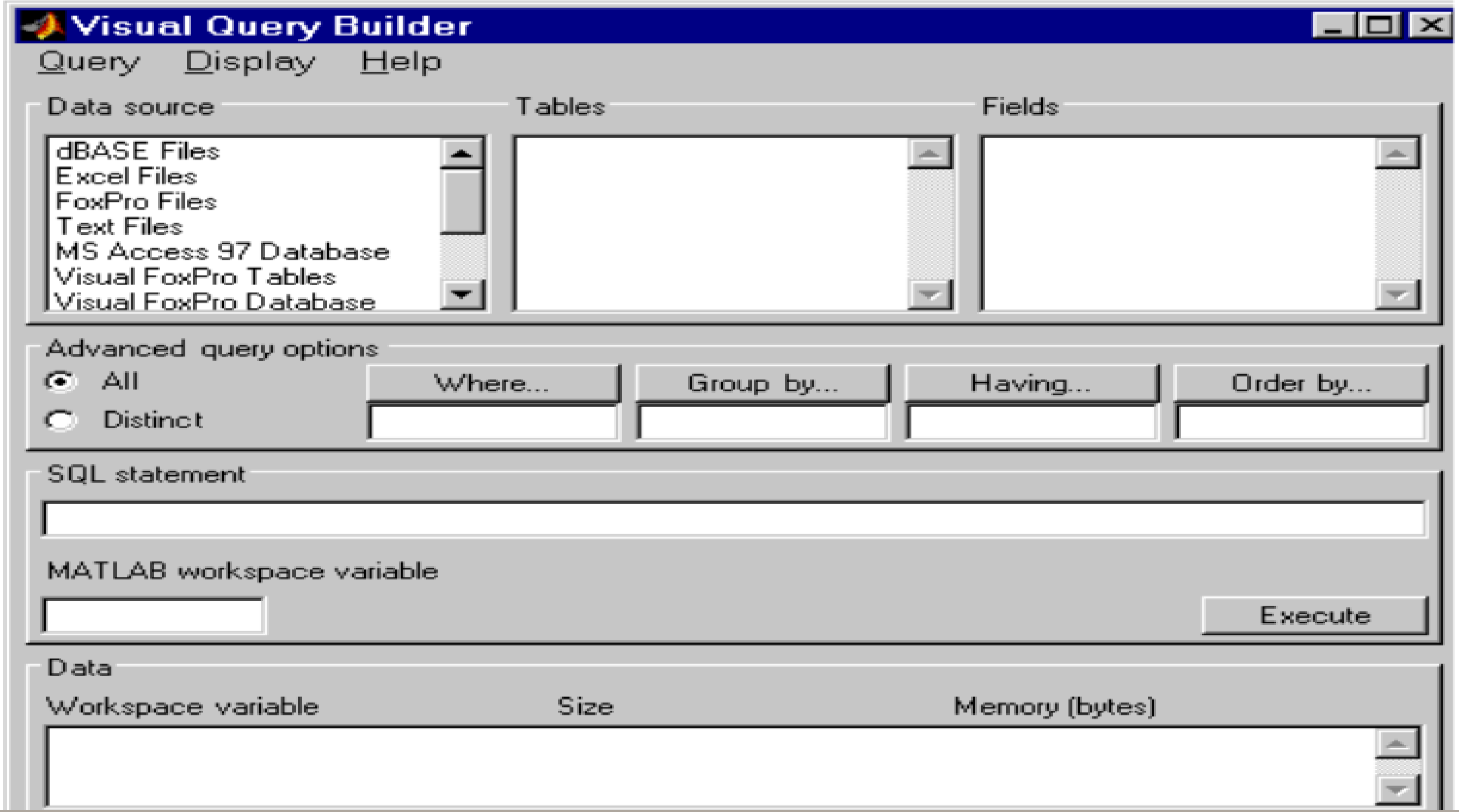
# *Starting the Database Toolbox*

- To use the Database Toolbox functions, just type the function you want to use. For more information, see “Tutorial for Functions” on page 3-1.

- To start the Visual Query Builder, type query builder . For more information, see “Visual Query Builder Tutorial” on page 2-1.

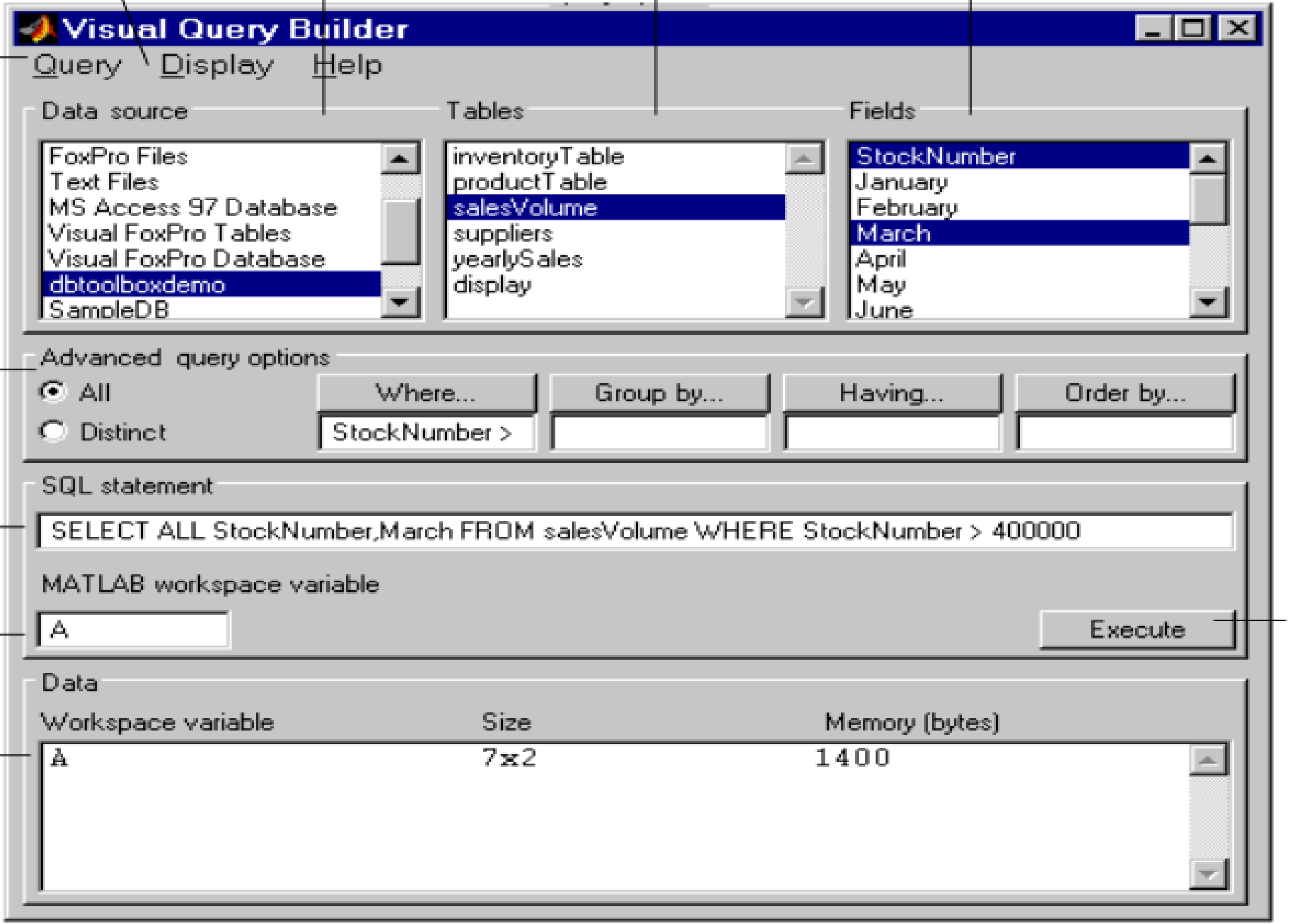
# Starting the Visual Query Builder

The **Visual Query Builder** dialog box appears.



To Quit : from Query ,Select **EXIT**

# Visual Query Builder Interface Tutorial



Data source:
- FoxPro Files
- Text Files
- MS Access 97 Database
- Visual FoxPro Tables
- Visual FoxPro Database
- dbtoolboxdemo (selected)
- SampleDB

Tables:
- inventoryTable
- productTable
- salesVolume (selected)
- suppliers
- yearlySales
- display

Fields:
- StockNumber (selected)
- January
- February
- March (selected)
- April
- May
- June

Advanced query options

● All

○ Distinct

Where...: StockNumber >

SQL statement

SELECT ALL StockNumber,March FROM salesVolume WHERE StockNumber > 400000

MATLAB workspace variable: A

Data

| Workspace variable | Size | Memory (bytes) |
|---|---|---|
| A | 7x2 | 1400 |

1 Select the data source.

2 Select the tables.

3 Select the fields you want to retrieve.

4 Refine the query, if needed.

5 View the SQL statement.

6 Assign a variable for the results.

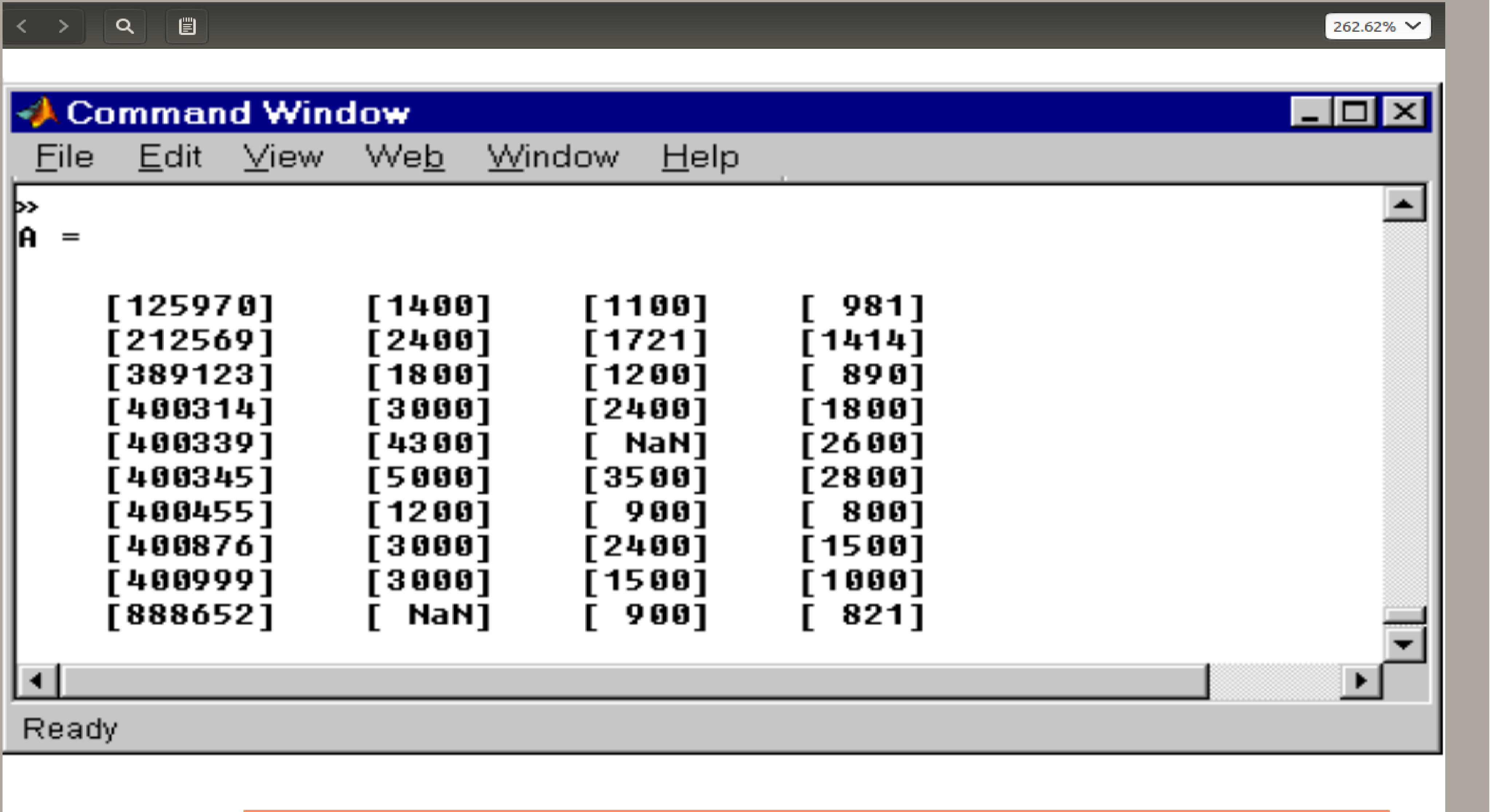
7 Run the query.

8 Double-click to view query results in the MATLAB command window.

9 View query results in table, chart, and report formats.

10 Save, load, and run queries.

# Visual Query Builder Interface Tutorial



```
>>
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]
```

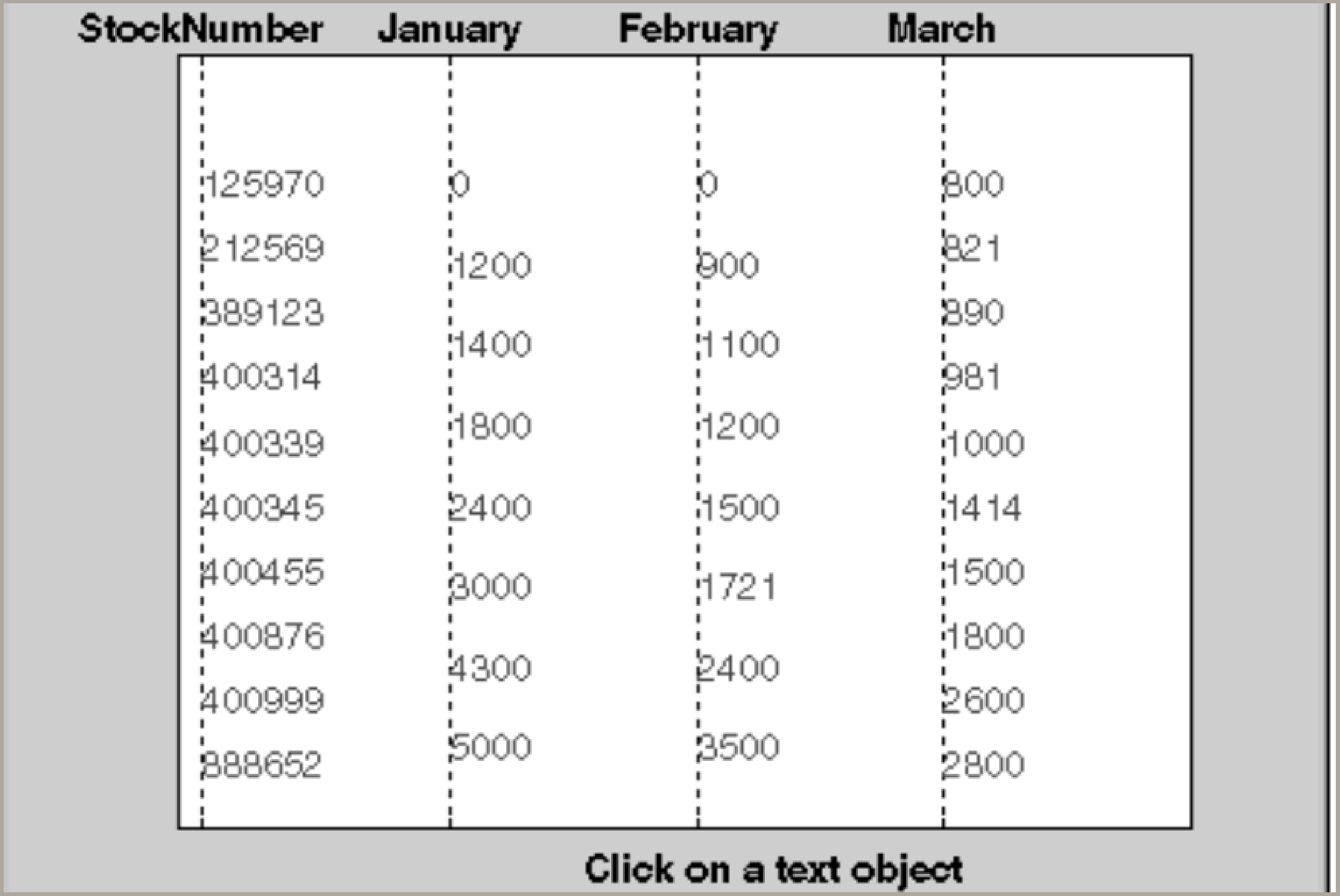
# *Viewing Query Results*

- After running a query in the Visual Query Builder, you can view :
- The retrieved data in the MATLAB command window, as described “Building, Running, and Saving a Query” .
- A “Relational Display of Data” on page 2-13.
- A “Chart Display of Results” on page 2-16; for example, a pie chart.
- A “Report Display of Results in a Table” on page 2-19.
- A “Display of Results in the Report Generator” on page 2-20

# Viewing Query Results

- **1- Relational Display of Data :**

***1 After executing a query, select Data from the Display menu. The query results appear in a figure window.***



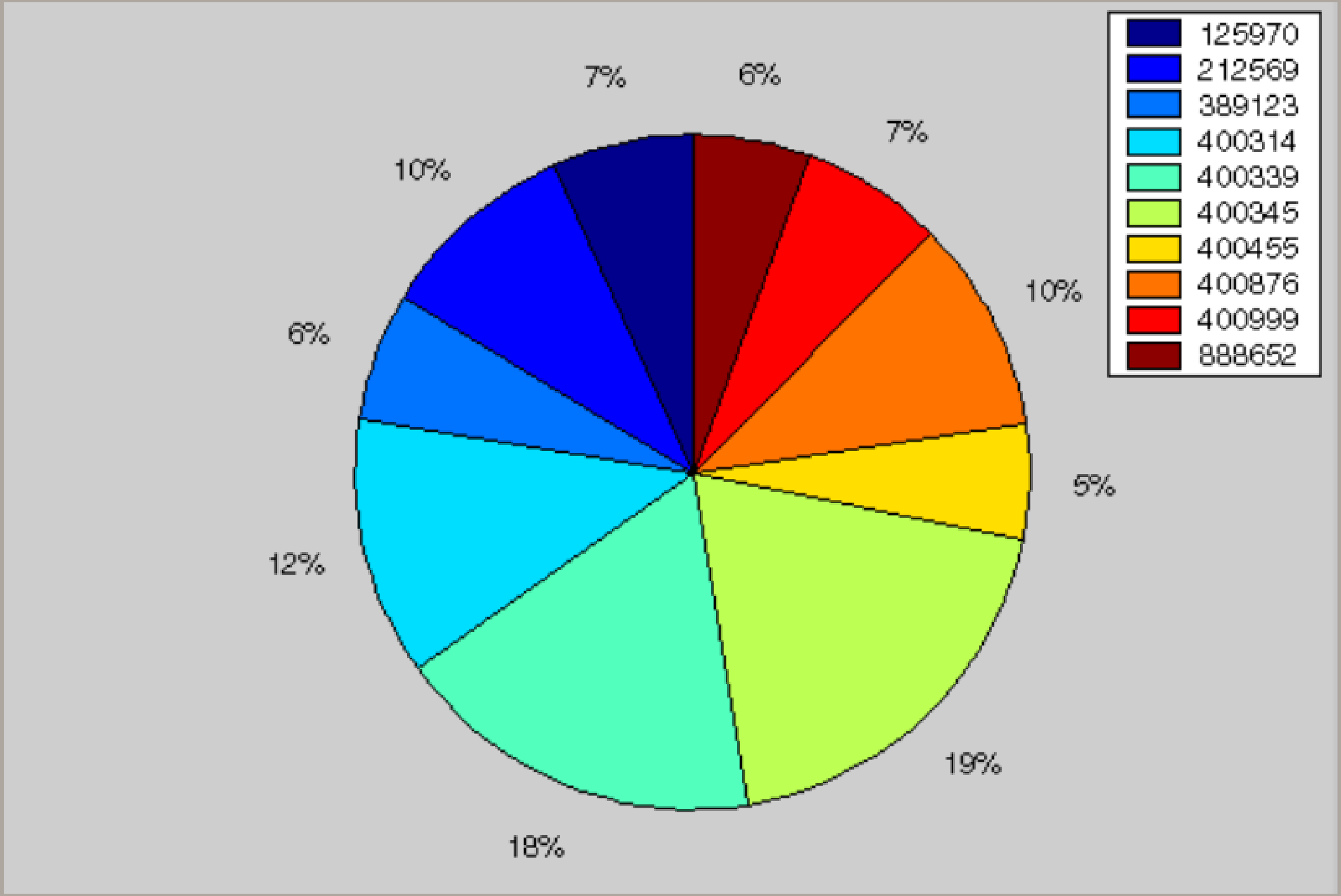
| StockNumber | January | February | March |
|---|---|---|---|
| 125970 | 0 | 0 | 800 |
| 212569 | 1200 | 900 | 821 |
| 389123 | 1400 | 1100 | 890 |
| 400314 | 1800 | 1200 | 981 |
| 400339 | 2400 | 1500 | 1000 |
| 400345 | 3000 | 1721 | 1414 |
| 400455 | 4300 | 2400 | 1500 |
| 400876 | 5000 | 3500 | 1800 |
| 400999 | | | 2600 |
| 888652 | | | 2800 |

Click on a text object

# Viewing Query Results

- **2- Chart Display of Results :**

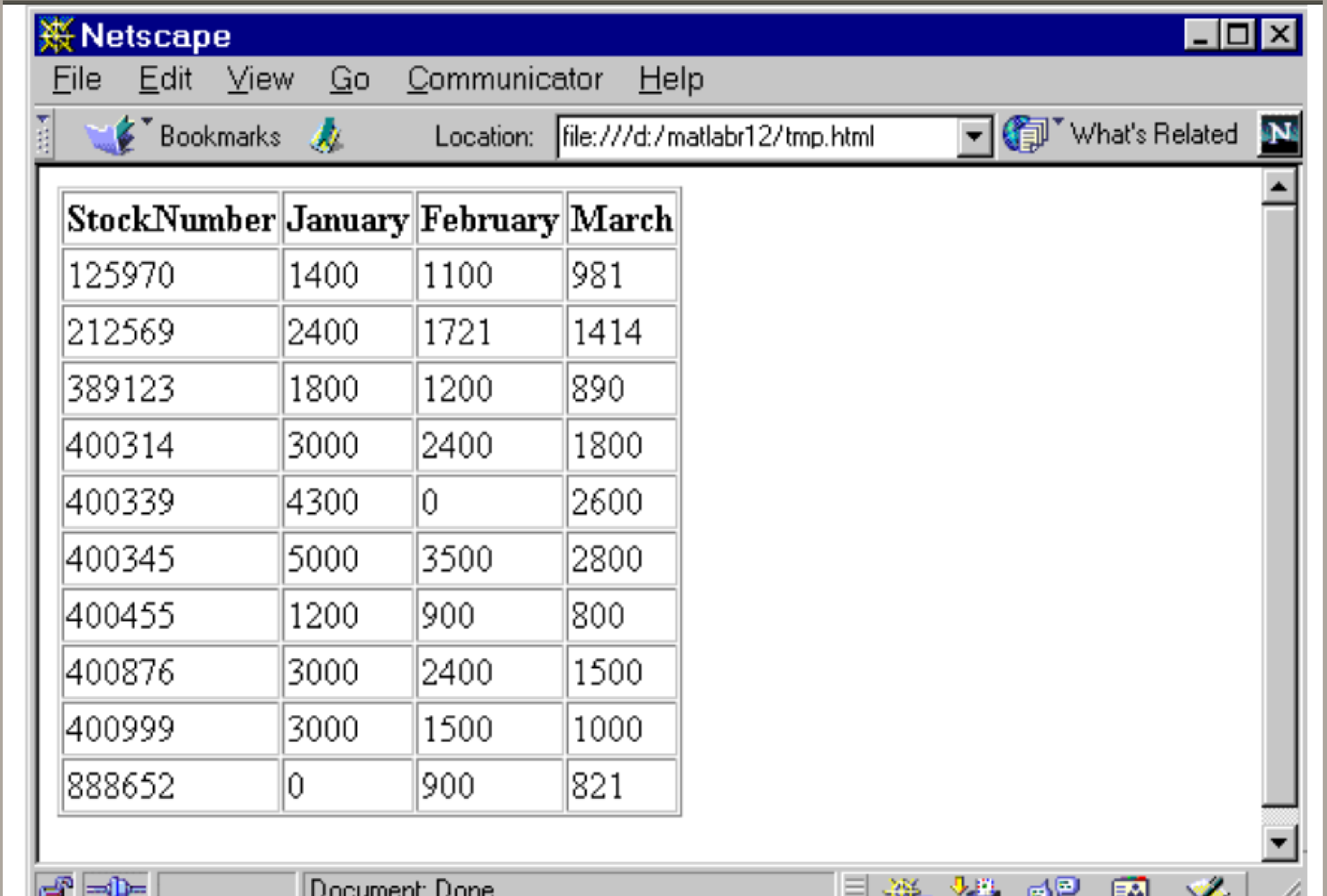
***After executing a query, select Chart from the Display menu. The Charting dialog box appears.***



# Viewing Query Results

- **3-Report Display of Results in a Table :**

***The report display presents the results in your system's default Web browser.***



Netscape
File Edit View Go Communicator Help
Bookmarks
Location: file:///d:/matlabr12/tmp.html
What's Related
Document: Done

| StockNumber | January | February | March |
|---|---|---|---|
| 125970 | 1400 | 1100 | 981 |
| 212569 | 2400 | 1721 | 1414 |
| 389123 | 1800 | 1200 | 890 |
| 400314 | 3000 | 2400 | 1800 |
| 400339 | 4300 | 0 | 2600 |
| 400345 | 5000 | 3500 | 2800 |
| 400455 | 1200 | 900 | 800 |
| 400876 | 3000 | 2400 | 1500 |
| 400999 | 3000 | 1500 | 1000 |
| 888652 | 0 | 900 | 821 |

# DataBase ToolBox Functions

- **About Objects and Methods for the Database Toolbox :**

  - Cursor .
  - Database .
  - Database metadata .
  - Driver .
  - Drivermanager .
  - Resultset .
  - Resultset metadata .

# Importing Data into MATLAB from a Database

- **database : conn = database('SampleDB', ' ' , ' ')** ► *you define a MATLAB variable, conn , to be the returned connection object. This connection stays open until you close it with the close function. For the database function, you provide the name of the database, which is the data source SampleDB for this example. The other two arguments for the database function are username and password . For this example they are empty strings because the SampleDB database does not username or password.*

- **exec : curs = exec(conn, 'select country from customers')** ► *Open a cursor and execute an SQL statement – type ,In the exec function, conn is the name of the connection object. The second argument, select country from customers , is a valid SQL statement that selects the country column of data from the customers table. The exec command returns a cursor object. In this example, you assign the MATLAB variable curs to the returned cursor object. curs = Attributes: [] , Data: 0 , DatabaseObject: [1x1 database] ,RowLimit: 0 ,SQLQuery: 'select country from customers' ,Message: [] , Type: 'Database Cursor Object' , ResultSet: [1x1 sun.jdbc.odbc.JdbcOdbcResultSet] , Cursor: [1x1 com.mathworks.toolbox.database.sqlExec] ,Statement: [1x1 sun.jdbc.odbc.JdbcOdbcStatement] , Fetch: 0 .*

- **logintimeout : logintimeout(5)** ► *set the maximum time, in seconds, you want to allow the MATLAB session to try to connect to a database. This prevents the MATLAB session from hanging up if a database connection fails.*

- **ping : ping(conn)** ► *Check the connection status – type , MATLAB returns status information about the connection, indicating that the connection was successful. Database Product Name: 'ACCESS' ,Database Product Version: '3.50.0000' , JDBC Driver Name: 'JDBC-ODBC Bridge (odbcjt32.dll)' ,  JDBCDriver Version: '1.1001 (04.00.4202)' , Max Database Connections: 64 , Current UserName: 'admin' , DatabaseURL: 'jdbc:odbc:SampleDB' .*

# Importing Data into MATLAB from a Database

- Fetch : curs = fetch(curs, 10) → *Import data into MATLAB – type , fetch is the function that imports data. It has the following two arguments*

** in this example:*

*– curs : the cursor object returned by exec ,*

*– 10 : the maximum number of rows you want to be returned by fetch . The RowLimit argument is optional. If RowLimit is omitted, MATLAB imports all remaining rows.*

- *close(curs)*
- *close(conn)*

- *conn = database('SampleDB', ' ', ' ');*
- *curs = exec(conn, 'select country from customers');*
- *curs = fetch(curs, 10);*

# *Exporting Data from MATLAB to a New Record in a Database*

- insert(conn, 'Avg_Freight_Cost', colnames , exdata) :

  -conn is the connection object .

  -Avg_Freight_Cost is the name of the table .

# *Exporting Data from MATLAB, Replacing Existing Data in a Database*

- update(conn, 'Avg_Freight_Cost', colnames, exdata, whereclause)

# Examples

## *Example 1 – set rowLimit of Cursor*

-
  ```
  conn=database('orcl','scott','tiger','oracle.jdbc.driver...OracleDriver','jdbc:oracle:thin:@144.212.33.228:1521:') ;

  curs=exec(conn, 'select * from EMP') ;

  set(curs, 'RowLimit' , 5) ;

  curs=fetch(curs) ;
  ```