

Supplement IV.B: Tutorial for MySQL

For Introduction to Java Programming By Y. Daniel Liang

This supplement covers the following topics:

- Starting and Stopping MySQL Server
- Connecting to MySQL
- Creating User Accounts
- Using MySQL
- MySQL Administration
- Accessing MySQL using Java

NOTE: Please use MySQL 4.1 or high with this tutorial.

0 Introduction

MySQL is a popular database with more than 4 million users. It is one of the fastest relational databases in the market. Many companies are using it to support their websites, data warehouses, and business applications. MySQL was developed by a Swedish company named MySQL AB. The product is distributed under GNU General Public License (GPL). It can support multiple users concurrently on the network. Students can connect to a MySQL database server standalone on their own computer or from the network. You can download it free from <http://dev.mysql.com/downloads/mysql/4.1.html>. MySQL runs on Windows, Linux and Solaris. This tutorial demonstrates using MySQL from the Windows operating system.

1 Starting and Stopping MySQL Server

If your database server is not started, you can start it by typing "net start mysql" from the MySQL bin directory as shown in Figure 1.1. You can stop it by typing the command net stop mysql.

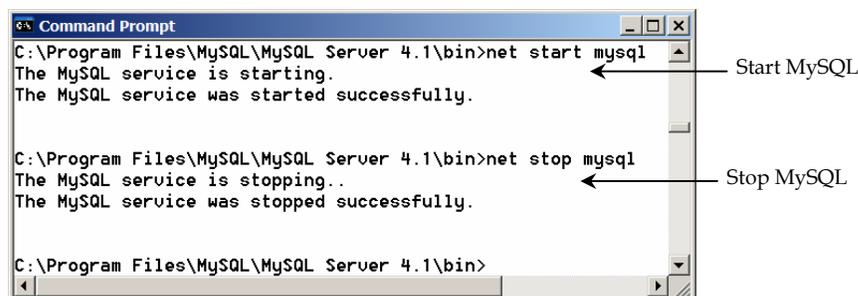


Figure 1.1

You need to start MySQL only once after the installation.

NOTE

<side remark: MySQL running?>

Your system may be set up to start MySQL automatically. In this case, you don't have to restart it.

NOTE

<side remark: MySQL bin directory>

By default, MySQL 4.1 is installed under c:\Program Files\MySQL directory. If it is installed elsewhere on your system, please use an appropriate path to locate the bin directory.

2 Connecting to MySQL

Assume that you have installed MySQL with the default configuration; you can connect MySQL from the DOS command prompt using the command `mysql` from the MySQL bin directory, as shown in Figure 1.2.

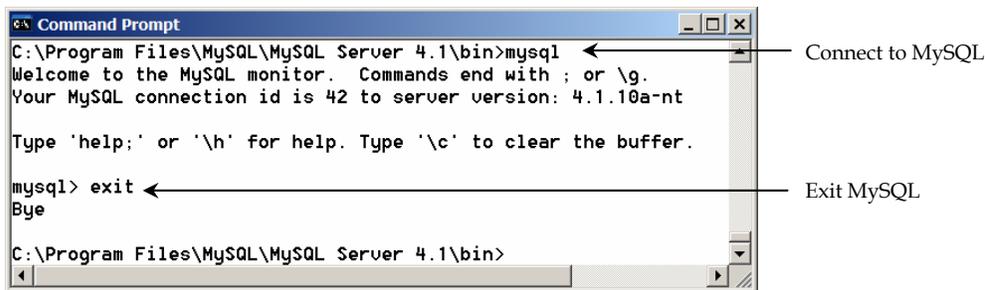


Figure 1.2

You can connect a MySQL database server from the command window.

If you are assigned a username and password, use the following command to login to the database.

```
mysql -u username -p
```

You will be prompted to enter the password.

To connect a remote MySQL database server, use

```
mysql -h hostname -u username -p
```

To disconnect from MySQL, type `exit`, as shown in Figure 1.2.

3 Creating User Accounts

If you are the owner of the database, you can create accounts for other user to access the database. When you install the database, you have the option to create an anonymous account. The user can login to the database using this account without a username and password. You may use the following command in the mysql command prompt to create a user.

```
grant all privileges on *.* to 'scott'@'localhost'  
identified by 'tiger';
```

Now you can connect to the database using the following command from the DOS command prompt:

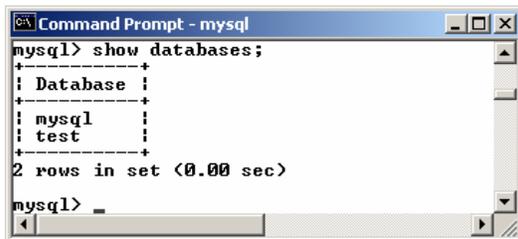
```
mysql -u scott -p
```

To allow the account to be connected from a remote machine, you need to issue another grant command from the mysql command prompt as follows:

```
grant all privileges on *.* to 'scott'@'%'  
identified by 'tiger';
```

4 Using MySQL

By default, the server contains two databases named mysql and test. You can see these two databases displayed in Figure 1.3 using the command show databases.



```
Command Prompt - mysql
mysql> show databases;
+-----+
| Database |
+-----+
| mysql    |
| test     |
+-----+
2 rows in set (0.00 sec)

mysql> _
```

Figure 1.3

The show databases command displays all available databases in the MySQL database server.

<side remark: create database>

The mysql database contains the tables that store the information about the server and its users. This database is intended for the server administrator to use. For example, the administrator can use it to create users and grant or revoke user privileges. Since you are the owner of the server installed on your system, you have full access to the mysql database. However, you should not create user tables in the mysql database. You can use the test database to

store data or create new databases. You can also create a new database using the command create database *databasename* or drop an existing database using the command drop database *databasename*.

<side remark: use database>

To select a database for use, type the use *databasename* command. Since the test database is created by default in every MySQL database, let us use it to demonstrate SQL commands. As shown in Figure 1.4, the test database is selected.



Figure 1.4

The use test command selects the test database.

Enter the following SQL statements from the MySQL command prompt, as shown in Figure 1.5. The create table statement creates a table named State. The insert statement inserts the values into the table. The select statement displays the contents from the table.

```
create table State(  
  name varchar(15) not null,  
  capital varchar(25),  
  population integer);  
  
insert into State values ('Georgia', 'Atlanta', 8383915);  
insert into State values ('New York', 'Albany', 19011378);  
  
select * from State;
```

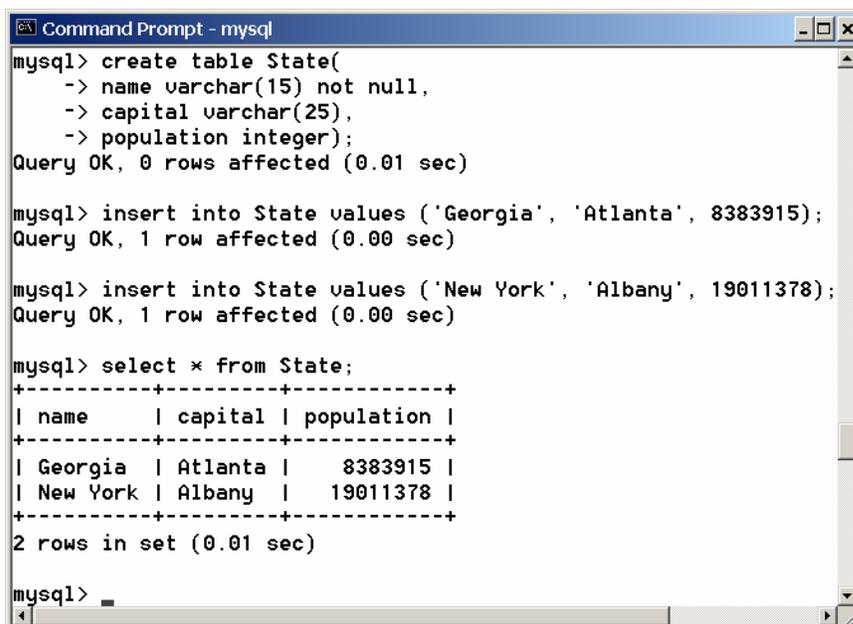
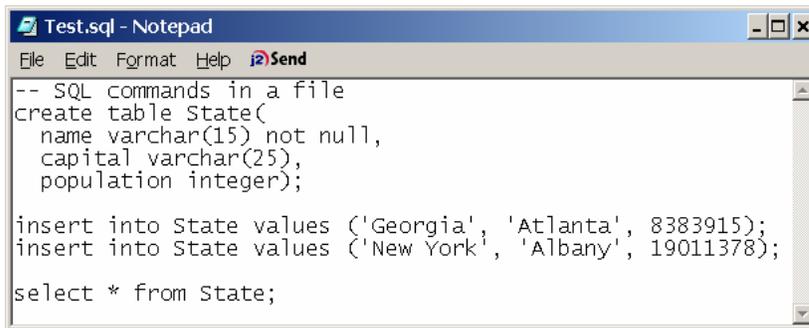


Figure 1.5

The execution result of the SQL statements is displayed in the MySQL command tool.

If you have typing errors, you have to retype the whole command. To avoid retyping the whole command, you can save the command in a file, and then run the command from the file. To do so, create a text file, e.g., named `test.sql`, which contains the commands. You can create the text file using any text editor, e.g., Notepad, as shown in Figure 1.6. To comment a line, precede with two dashes. You can now run the script file by typing `source test.sql` from the SQL command prompt, as shown in Figure 1.7.

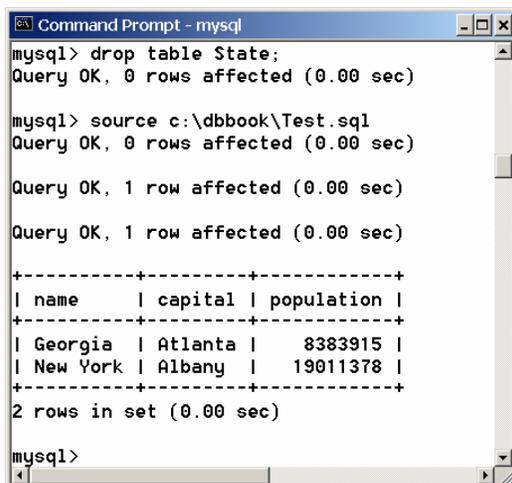


```
Test.sql - Notepad
File Edit Format Help Send
-- SQL commands in a file
create table State(
  name varchar(15) not null,
  capital varchar(25),
  population integer);

insert into State values ('Georgia', 'Atlanta', 8383915);
insert into State values ('New York', 'Albany', 19011378);
select * from State;
```

Figure 1.6

You can use Notepad to create a text file for SQL commands.



```
Command Prompt - mysql
mysql> drop table State;
Query OK, 0 rows affected (0.00 sec)

mysql> source c:\dbbook\Test.sql
Query OK, 0 rows affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

+-----+-----+-----+
| name   | capital | population |
+-----+-----+-----+
| Georgia | Atlanta | 8383915   |
| New York | Albany  | 19011378  |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Figure 1.7

You can run the SQL commands in a script file from MySQL.

5 MySQL Administrations

You can download MySQL Control Center from www.mysql.com. MySQL Control Center is a valuable utility for administering MySQL databases. You can add and drop databases, create and remove users and tables, grant privileges from this easy-to-use GUI utility.

6 Accessing MySQL Using Java

NOTE: you are not familiar with SQL and JDBC, please Chapter 30 on basics of SQL and how to write Java database programs.

The JDBC driver for MySQL is `com.mysql.jdbc.Driver` contained in `c:\book\mysqljdbc.jar`. To run a Java program that accesses a MySQL database, you must put `mysqljdbc.jar` in the classpath.

The database URL for MySQL is `jdbc:mysql://hostname:port/dbname`. By default, the port number is 3306. For example, if the database is named test on the localhost, the URL is `jdbc:mysql://localhost/test`.

The following is a sample program

```
import java.sql.*;

public class SimpleJdbc {
    public static void main(String[] args)
        throws SQLException, ClassNotFoundException {
        // Load the JDBC driver
        Class.forName("com.mysql.jdbc.Driver");
        System.out.println("Driver loaded");

        // Establish a connection
        Connection connection = DriverManager.getConnection
            ("jdbc:mysql://localhost/test");
        System.out.println("Database connected");

        // Create a statement
        Statement statement = connection.createStatement();

        // Execute a statement
        ResultSet resultSet = statement.executeQuery
            ("select firstName, mi, lastName from Student where lastName "
            + " = 'Smith'");
```

```
    // Iterate through the result and print the student names
    while (resultSet.next())
        System.out.println(resultSet.getString(1) + "\t" +
            resultSet.getString(2) + "\t" + resultSet.getString(3));

    // Close the connection
    connection.close();
}
}
```