**JDBC Exercises**

* **JDBC - Environment Setup**
* To start developing with JDBC, you should setup your JDBC environment by following the steps shown below.
* We assume that you are working on a Windows platform.
* **Install Java**
* Install J2SE Development Kit 5.0 (JDK 5.0) from Java Official Site.
* Make sure following environment variables are set as described below −
* JAVA\_HOME: This environment variable should point to the directory where you installed the JDK, e.g. C:\Program Files\Java\jdk1.5.0.
* CLASSPATH: This environment variable should have appropriate paths set, e.g. C:\Program Files\Java\jdk1.8.0\_20\jre\lib.
* PATH: This environment variable should point to appropriate JRE bin, e.g. C:\Program Files\Java\jre1.5.0\_20\bin.
* It is possible you have these variable set already, but just to make sure here's how to check.
* Go to the control panel and double-click on System. If you are a Windows, it is possible you have to open Performance and Maintenance before you will see the System icon.
* Go to the Advanced tab and click on the Environment Variables.
* Now check if all the above mentioned variables are set properly.
* You automatically get both JDBC packages java.sql and javax.sql, when you install J2SE Development Kit 8.0 (JDK 8.0).
* **Install Database**
* The most important thing you will need, of course is an actual running database with a table that you can query and modify.
* Install a database that is most suitable for you. You can have plenty of choices and most common are −
* **MySQL** DB: MySQL is an open source database. You can download it from MySQL Official Site. We recommend downloading the full Windows installation.
* In addition, download and install MySQL Administrator as well as MySQL Query Browser. These are GUI based tools that will make your development much easier.
* Finally, download and unzip MySQL Connector/J (the MySQL JDBC driver) in a convenient directory. For the purpose of this session we will assume that you have installed the driver at C:\Program Files\MySQL\mysql-connector-java-5.1.8.
* Accordingly, set CLASSPATH variable to C:\Program Files\MySQL\mysql-connector-java-5.1.8\mysql-connector-java-5.1.8-bin.jar. Your driver version may vary based on your installation.
* PostgreSQL DB: PostgreSQL is an open source database. You can download it from PostgreSQL Official Site.
* The Postgres installation contains a GUI based administrative tool called pgAdmin III. JDBC drivers are also included as part of the installation.
* Oracle DB: Oracle DB is a commercial database sold by Oracle. We assume that you have the necessary distribution media to install it.
* Oracle installation includes a GUI based administrative tool called Enterprise Manager. JDBC drivers are also included as a part of the installation.
* **Install Database Drivers**
* The latest JDK includes a JDBC-ODBC Bridge driver that makes most Open Database Connectivity (ODBC) drivers available to programmers using the JDBC API.
* Now a day, most of the Database vendors are supplying appropriate JDBC drivers along with Database installation. So, you should not worry about this part.
* **Set Database Credential**
* For this tutorial we are going to use MySQL database. When you install any of the above database, its administrator ID is set to root and gives provision to set a password of your choice.
* Using root ID and password you can either create another user ID and password, or you can use root ID and password for your JDBC application.
* There are various database operations like database creation and deletion, which would need administrator ID and password.
* For rest of the JDBC tutorial, we would use MySQL Database with username as ID and password as password.
* If you do not have sufficient privilege to create new users, then you can ask your Database Administrator (DBA) to create a user ID and password for you.
* **Create Database**
* To create the EMP database, use the following steps −
* Open a Command Prompt and change to the installation directory as follows −
* C:\>
* C:\>cd Program Files\MySQL\bin
* C:\Program Files\MySQL\bin>
* Note: The path to mysqld.exe may vary depending on the install location of MySQL on your system. You can also check documentation on how to start and stop your database server.
* Start the database server by executing the following command, if it is already not running.
* C:\Program Files\MySQL\bin>mysqld
* C:\Program Files\MySQL\bin>
* Login to the database as follows −
* C:\Program Files\MySQL\bin>mysql -u root -p
* Enter password: \*\*\*\*\*\*\*\*
* mysql>
* Create Database
* The CREATE DATABASE statement is used for creating a new database. The syntax is -
* SQL> CREATE DATABASE DATABASE\_NAME;
* Example
* The following SQL statement creates a Database named EMP -
* SQL> CREATE DATABASE EMP;
* Drop Database
* The DROP DATABASE statement is used for deleting an existing database. The syntax is -
* SQL> DROP DATABASE DATABASE\_NAME;
* Note: To create or drop a database you should have administrator privilege on your database server.
* Be careful, deleting a database would loss all the data stored in the database.
* Create Table
* The CREATE TABLE statement is used for creating a new table. The syntax is -
* SQL> CREATE TABLE table\_name (
* column\_name column\_data\_type,
* column\_name column\_data\_type,
* column\_name column\_data\_type
* ...
* );
* Example
* The following SQL statement creates a table named Employees with four columns -
* SQL> CREATE TABLE Employees (
* id INT NOT NULL,
* age INT NOT NULL,
* first VARCHAR(255),
* last VARCHAR(255),
* PRIMARY KEY ( id )
* );

create table Employees(id int not null, age int not null, first varchar (255), last varchar (255));

* Drop Table
* The DROP TABLE statement is used for deleting an existing table. The syntax is -
* SQL> DROP TABLE table\_name;
* Example
* The following SQL statement deletes a table named Employees -
* SQL> DROP TABLE Employees;
* INSERT Data
* The syntax for INSERT, looks similar to the following, where column1, column2, and so on represents the new data to appear in the respective columns -
* SQL> INSERT INTO table\_name VALUES (column1, column2, ...);
* Example
* The following SQL INSERT statement inserts a new row in the Employees database created earlier -
* mysql>
* INSERT INTO Employees VALUES (100, 18, 'Zara', 'Ali');
* INSERT INTO Employees VALUES (101, 25, 'Mahnaz', 'Fatma');
* INSERT INTO Employees VALUES (102, 30, 'Zaid', 'Khan');
* INSERT INTO Employees VALUES (103, 28, 'Sumit', 'Mittal');
* INSERT INTO Employees VALUES (104, 18, 'Gopinathan', 'Munappy');
* SELECT Data
* The SELECT statement is used to retrieve data from a database. The syntax for SELECT is -
* SQL> SELECT column\_name, column\_name, ...
* FROM table\_name
* WHERE conditions;
* The WHERE clause can use the comparison operators such as =, !=, <, >, <=,and >=, as well as the BETWEEN and LIKE operators.
* Example
* The following SQL statement selects the age, first and last columns from the Employees table, where id column is 100 -
* SQL> SELECT first, last, age
* FROM Employees
* WHERE id = 100;
* The following SQL statement selects the age, first and last columns from the Employees table where first column contains Zara -
* SQL> SELECT first, last, age
* FROM Employees
* WHERE first LIKE '%Zara%';
* UPDATE Data
* The UPDATE statement is used to update data. The syntax for UPDATE is -
* SQL> UPDATE table\_name
* SET column\_name = value, column\_name = value, ...
* WHERE conditions;
* The WHERE clause can use the comparison operators such as =, !=, <, >, <=,and >=, as well as the BETWEEN and LIKE operators.
* Example
* The following SQL UPDATE statement changes the age column of the employee whose id is 100 -
* SQL> UPDATE Employees SET age=20 WHERE id=100;
* DELETE Data
* The DELETE statement is used to delete data from tables. The syntax for DELETE is -
* SQL> DELETE FROM table\_name WHERE conditions;
* The WHERE clause can use the comparison operators such as =, !=, <, >, <=,and >=, as well as the BETWEEN and LIKE operators.
* Example
* The following SQL DELETE statement deletes the record of the employee whose id is 100 -
* SQL> DELETE FROM Employees WHERE id=100;