Servlet Database Connectivity

Servlet – Database Connectivity

- A Java Database Connectivity (JDBC) driver is a software component **enabling a Java application to interact with a database**
- The JDBC library includes APIs for each of the tasks mentioned below that are commonly associated with database usage.
  - Making a connection to a database.
  - Creating SQL or MySQL statements.
  - Executing SQL or MySQL queries in the database.
  - Viewing & Modifying the resulting records.

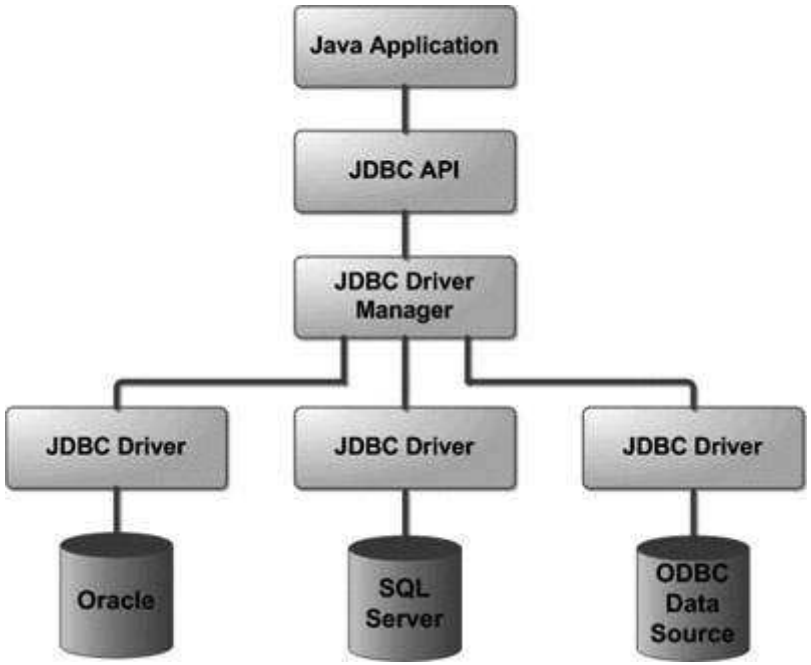
JDBC Architecture

- The JDBC API supports both two-tier and three-tier processing models for database access but in general, JDBC Architecture consists of two layers
  - **JDBC API:** This provides the application-to-JDBC Manager connection.
  - **JDBC Driver API:** This supports the JDBC Manager-to-Driver Connection.
- The JDBC API uses a driver manager and database-specific drivers to provide transparent connectivity to heterogeneous databases.

- The JDBC driver manager ensures that the correct driver is used to access each data source. The driver manager is capable of supporting multiple concurrent drivers connected to multiple heterogeneous databases.

JDBC Architecture(cont..)

- Following is the architectural diagram, which shows the location of the driver manager with respect to the JDBC drivers and the Java application –



Common JDBC Components

The JDBC API provides the following interfaces and classes −

- **DriverManager:** This class manages a list of database drivers. Matches connection requests from the java application with the proper database driver using communication sub protocol. The first driver that recognizes a certain subprotocol under JDBC will be used to establish a database Connection.

- **Driver:** This interface handles the communications with the database server. You will interact directly with Driver objects very rarely. Instead, you use DriverManager objects, which manages objects of this type. It also abstracts the details associated with working with Driver objects.

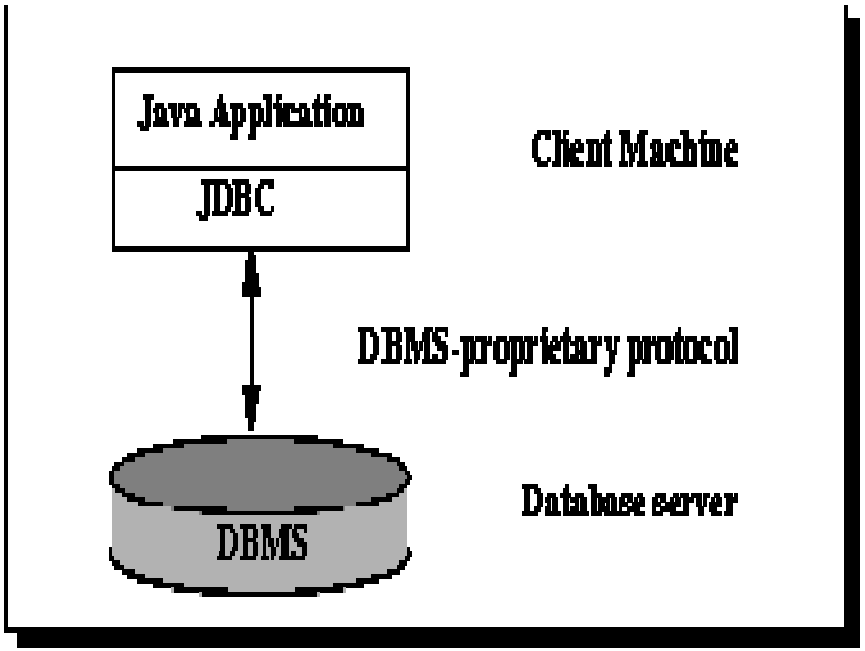
- **Connection:** This interface with all methods for contacting a database. The connection object represents communication context, i.e., all communication with database is through connection object only.

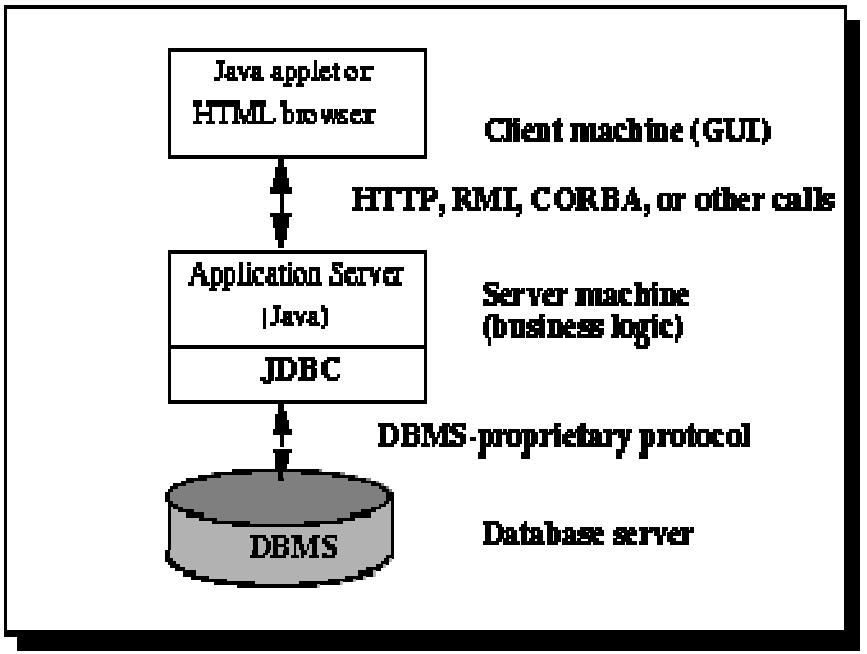
- **Statement:** You use objects created from this interface to submit the SQL statements to the database. Some derived interfaces accept parameters in addition to executing stored procedures.

- **ResultSet:** These objects hold data retrieved from a database after you execute an SQL query using Statement objects. It acts as an iterator to allow you to move through its data.

- **SQLException:** This class handles any errors that occur in a database application.

Database Access Models





MySql Installation

- Use **mysql-5.5.30-win32** to install mysql
- While installing, provide user name and password
- Copy the **mysql-connector-java-3.1.14-bin** to "C:\Program Files\Apache Software Foundation\Tomcat 7.0\lib"
- To run mysql go to Start→ All Program→ MySqL→ MySql5.5 Command Line Client and double click it

Creation of table

- Provide password “ ”
- Use the default database as follows

```
mysql> use TEST;
mysql> create table Employees
(
        id int not null,
        age int not null,
        first varchar (255),
        last varchar (255)
);
```

- Insert new rows as follows
  - INSERT INTO Employees VALUES (100, 18, 'Zara', 'Ali');

Java code snippet to access database

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class DatabaseAccess extends HttpServlet{

  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
            throws ServletException, IOException
  {
     // JDBC driver name and database URL
     final String JDBC_DRIVER="com.mysql.jdbc.Driver";
     final String DB_URL="jdbc:mysql://localhost:3306/TEST";

     //  Database credentials
     final String USER = "root";
     final String PASS = "angel";

        Statement stmt=null;
        Connection conn =null;
```

...Contd

```
// Set response content type
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String title = "Database Result";
    String docType =
      "<!doctype html public \"-//w3c//dtd html 4.0 " +
       "transitional//en\">\n";
       out.println(docType +
       "<html>\n" +
       "<head><title>" + title + "</title></head>\n" +
       "<body bgcolor=\"#f0f0f0\">\n" +
       "<h1 align=\"center\">" + title + "</h1>\n");
```

...Contd

```
try{
      // Register JDBC driver
      Class.forName("com.mysql.jdbc.Driver");

      // Open a connection
      conn = DriverManager.getConnection(DB_URL,
   USER, PASS);

      // Execute SQL query
      stmt = conn.createStatement();
      String sql;
      sql = "SELECT id, first, last, age FROM Employees";
      ResultSet rs = stmt.executeQuery(sql);
```

...Contd

```
// Extract data from result set
        while(rs.next()){
          //Retrieve by column name
          int id  = rs.getInt("id");
          int age = rs.getInt("age");
          String first = rs.getString("first");
          String last = rs.getString("last");

          //Display values
          out.println("ID: " + id + "<br>");
          out.println(", Age: " + age + "<br>");
          out.println(", First: " + first + "<br>");
          out.println(", Last: " + last + "<br>");
        }
        out.println("</body></html>");
```

...Contd

```
// Clean-up environment
        rs.close();
        stmt.close();
        conn.close();
     }catch(SQLException se){
        //Handle errors for JDBC
        se.printStackTrace();
     }catch(Exception e){
        //Handle errors for Class.forName
        e.printStackTrace();
     }finally{
        //finally block used to close resources
        try{
           if(stmt!=null)
              stmt.close();
        }catch(SQLException se2){
        }// nothing we can do
        try{
           if(conn!=null)
           conn.close();
        }catch(SQLException se){
           se.printStackTrace();
        }//end finally try
     } //end try
  }
}
```

Output

- Navigate to http://localhost:8080