**JAVA & DATABASE**

**JDBC**

This chapter provides an example of how to create a simple JDBC application. This will show you how to open a database connection, execute a SQL query, and display the results.

All the steps mentioned in this template example, would be explained in subsequent chapters of this tutorial.

Creating JDBC Application

There are following six steps involved in building a JDBC application −

* **Import the packages:** Requires that you include the packages containing the JDBC classes needed for database programming. Most often, using *import java.sql.\** will suffice.
* **Register the JDBC driver:** Requires that you initialize a driver so you can open a communication channel with the database.
* **Open a connection:** Requires using the*DriverManager.getConnection()* method to create a Connection object, which represents a physical connection with the database.
* **Execute a query:** Requires using an object of type Statement for building and submitting an SQL statement to the database.
* **Extract data from result set:** Requires that you use the appropriate*ResultSet.getXXX()* method to retrieve the data from the result set.
* **Clean up the environment:** Requires explicitly closing all database resources versus relying on the JVM's garbage collection.

Sample Code

This sample example can serve as a **template** when you need to create your own JDBC application in the future.

This sample code has been written based on the environment and database setup done in the previous chapter.

Copy and past the following example in FirstExample.java, compile and run as follows −

//STEP 1. Import required packages

import java.sql.\*;

public class FirstExample {

// JDBC driver name and database URL

static final String JDBC\_DRIVER = "com.mysql.jdbc.Driver";

static final String DB\_URL = "jdbc:mysql://localhost/EMP";

// Database credentials

static final String USER = "username";

static final String PASS = "password";

public static void main(String[] args) {

Connection conn = null;

Statement stmt = null;

try{

//STEP 2: Register JDBC driver

Class.forName("com.mysql.jdbc.Driver");

//STEP 3: Open a connection

System.out.println("Connecting to database...");

conn = DriverManager.getConnection(DB\_URL,USER,PASS);

//STEP 4: Execute a query

System.out.println("Creating statement...");

stmt = conn.createStatement();

String sql;

sql = "SELECT id, first, last, age FROM Employees";

ResultSet rs = stmt.executeQuery(sql);

//STEP 5: 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

System.out.print("ID: " + id);

System.out.print(", Age: " + age);

System.out.print(", First: " + first);

System.out.println(", Last: " + last);

}

//STEP 6: 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

System.out.println("Goodbye!");

}//end main

}//end FirstExample

Now let us compile the above example as follows −

C:\>javac FirstExample.java

C:\>

When you run **FirstExample**, it produces the following result −

C:\>java FirstExample

Connecting to database...

Creating statement...

ID: 100, Age: 18, First: Zara, Last: Ali

ID: 101, Age: 25, First: Mahnaz, Last: Fatma

ID: 102, Age: 30, First: Zaid, Last: Khan

ID: 103, Age: 28, First: Sumit, Last: Mi