

Smart Bridge – Modern Application Development (Java Spring Boot)

Name: P. Harivarman

Reg.No: 20BEC0640

College: VIT-Vellore

Assignment-3

1.JDBC Connection Using Eclipse java IDE:

Code:

```
import java.sql.Connection;
import java.sql.DriverManager;

//JDBC retrieve data from MYSQL workbench
public class JDBC_connection {
    public static void main(String args[]) {
        try {
            //1.register to driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            //2.connection
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/amazon", "root",
"harivarman");

            System.out.println("Successfully Connected!");
            //3.close the connection
            conn.close();
        }
        catch (Exception e) {
        }
    }
}
```

Output:



The screenshot displays the Eclipse IDE interface. The top editor shows the `JDBC_connection.java` file with the following code:

```
1 import java.sql.Connection;
2 import java.sql.DriverManager;
3
4
5 //JDBC retrieve data from MYSQL workbench
6 public class JDBC_connection {
7     public static void main(String args[]) {
8         try {
9             //1.register to driver
10            Class.forName("com.mysql.cj.jdbc.Driver");
11
12            //2.connection
13            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/amazon", "root", "harivarman");
14
15            System.out.println("Successfully Connected!");
16            //3.close the connection
17            conn.close();
18        }
19        catch (Exception e) {
20        }
21    }
22 }
```

The bottom console window shows the output of the program:

```
<terminated> JDBC_connection [Java Application] C:\Users\Harivarman\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_18.0.1.v20220515-1614\jre\bin\javaw.exe (06:
Successfully Connected!
```

2.Retrieve Data using JDBC:

Code:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

//JDBC retrieve data from MYSQL workbench
public class JDBC_Example {
    public static void main(String args[]) {
        try {
            //1.register to driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            //2.connection
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/amazon", "root",
"harivarman");

            //3.statement
            Statement stmt = conn.createStatement() ;

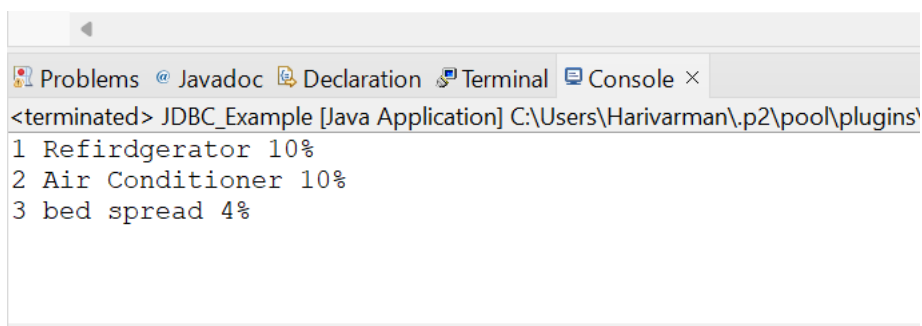
            //4.execute query
            ResultSet rs = stmt.executeQuery("select * from
festival_days");

            while (rs.next()) {
                System.out.println(rs.getInt(1) + " " +
rs.getString(2) + " "+rs.getString(3));
            }

            // 5.close the connection
            conn.close();
        } catch (Exception e) {

        }
    }
}
```

Output:

The screenshot shows an IDE's console window with tabs for Problems, Javadoc, Declaration, Terminal, and Console. The Console tab is active, displaying the output of the JDBC program. The output starts with a terminated message and then lists three rows of data: '1 Refirdgerator 10%', '2 Air Conditioner 10%', and '3 bed spread 4%'.

```
<terminated> JDBC_Example [Java Application] C:\Users\Harivarman\p2\pool\plugins'
1 Refirdgerator 10%
2 Air Conditioner 10%
3 bed spread 4%
```

3.Update Data using JDBC:

Code:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
public class JDBC_update {

    public static void main(String args[]) {
        try {
            //1.register to driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            //2.connection
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/amazon", "root",
"harivarman");

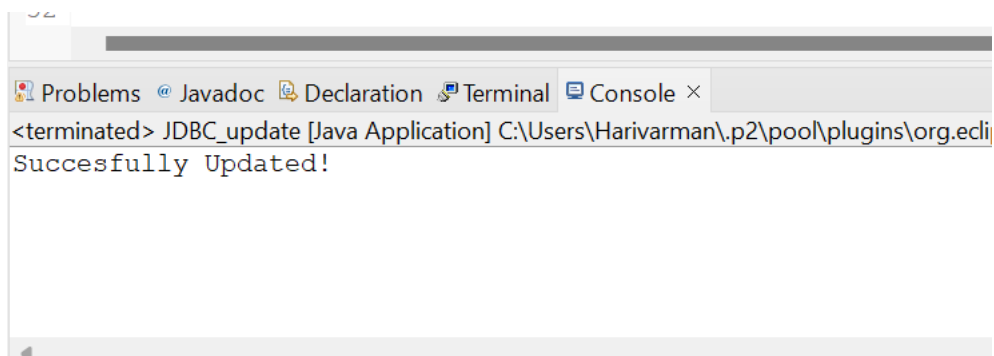
            //3.statement
            Statement stmt = conn.createStatement() ;

            //4.execute query
            int rs = stmt.executeUpdate("insert into
products(sno,productname,price,discount) values(5,'TV',50000,'20%')");

            if(rs>0) {
                System.out.println("Succesfully Updated!");
            }

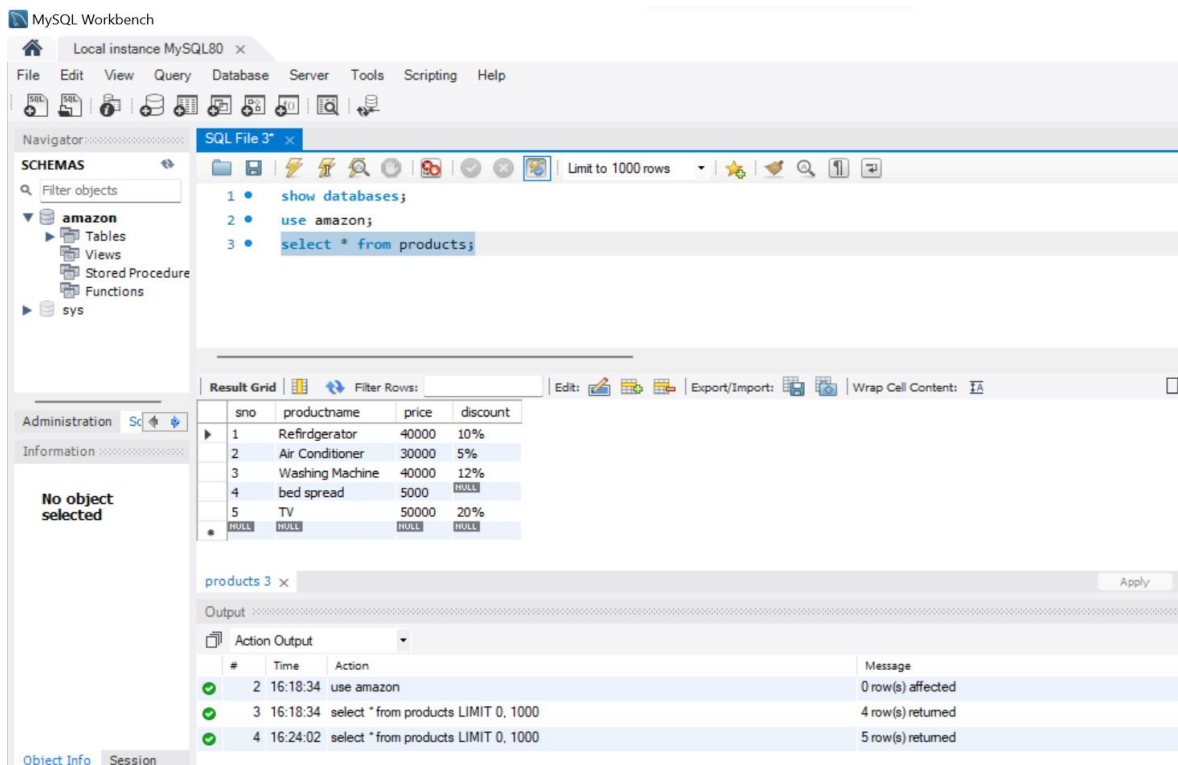
            // 5.close the connection
            conn.close();
        } catch (Exception e) {
        }
    }
}
```

IDE output:

A screenshot of an IDE's console window. The window has a title bar with icons for Problems, Javadoc, Declaration, Terminal, and Console. The console output shows the command prompt "<terminated> JDBC_update [Java Application] C:\Users\Harivarman\p2\pool\plugins\org.eclipse" followed by the message "Succesfully Updated!". The text is displayed in a monospaced font on a light background.

```
<terminated> JDBC_update [Java Application] C:\Users\Harivarman\p2\pool\plugins\org.eclipse
Succesfully Updated!
```

Before Insert the data:



MySQL Workbench interface showing the 'Before Insert' state. The SQL editor contains the following queries:

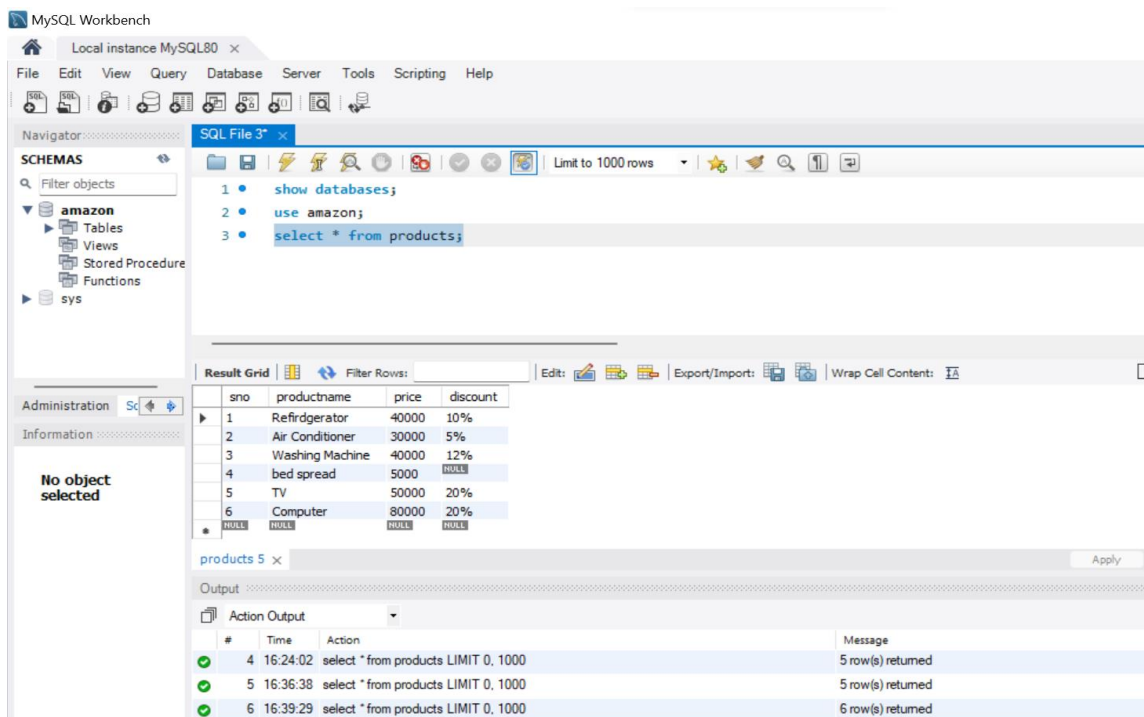
```
1 • show databases;
2 • use amazon;
3 • select * from products;
```

The Result Grid displays the following data:

sno	productname	price	discount
1	Refridgerator	40000	10%
2	Air Conditioner	30000	5%
3	Washing Machine	40000	12%
4	bed spread	5000	NULL
5	TV	50000	20%
6	Computer	80000	20%

The Output pane shows the execution of the 'select * from products;' query, returning 5 rows.

After Insert the data:



MySQL Workbench interface showing the 'After Insert' state. The SQL editor contains the same queries as before:

```
1 • show databases;
2 • use amazon;
3 • select * from products;
```

The Result Grid now displays 6 rows of product data, including the newly inserted 'Computer' row:

sno	productname	price	discount
1	Refridgerator	40000	10%
2	Air Conditioner	30000	5%
3	Washing Machine	40000	12%
4	bed spread	5000	NULL
5	TV	50000	20%
6	Computer	80000	20%

The Output pane shows the execution of the 'select * from products;' query, returning 6 rows.

Conclusion:

Using Eclipse IDE I have connected to the SQL Server and from which I have retrieved the data and also updated the data using JDBC.