Sri Venkateswara College of Engineering

Department of Information technology

**Online Java Certification Programme**

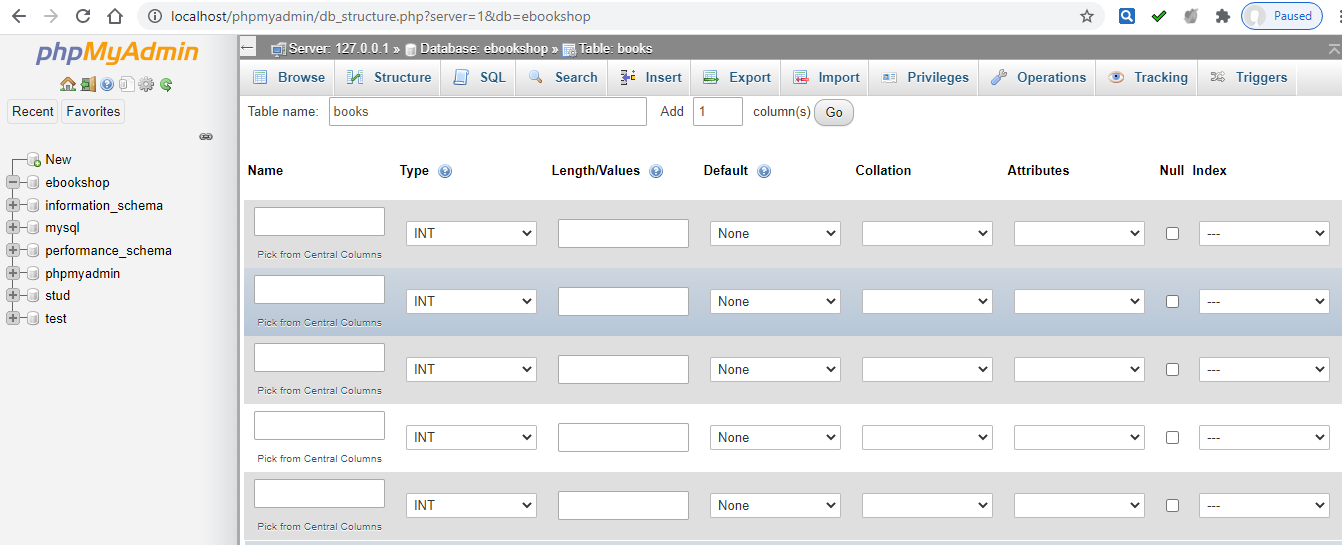
**JDBC Connectivity Hands-on exercise with solution**

1. Create a database “ebookshop” which contains a table called “books” with 5 columns. Establish JDBC Connectivity in java using Netbeans

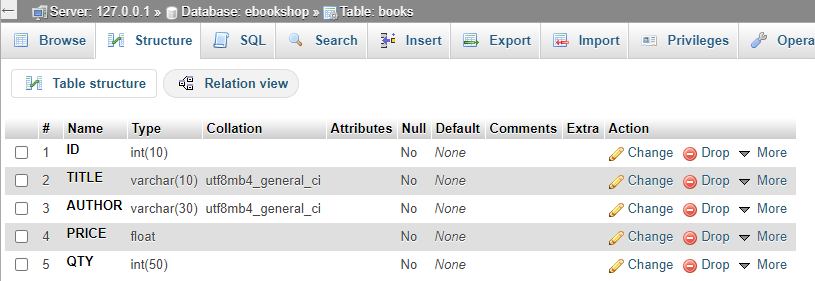
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **TITLE** | **AUTHOR** | **PRICE** | **QTY** |
| 1001 | Java Programming | Mohammed Ali | 150.00 | 5 |
| 1002 | Web Programming | Kevin Jones | 230.12 | 2 |
| 1003 | DBMS | Russel | 375.00 | 8 |
| 1004 | Data Structures | Schiller | 275.25 | 4 |
| 1005 | Mobile Computing | Jeffrey Jackson | 500.00 | 3 |

**Solution**

Step 1: Create a database and table

****

Step 2: Create structure of the table



Step 3: Insert the values for each row in the table

**Program Code**

package project1;

import java.sql.\*;

/\*\*

\*

\* @author DELL 3543

\*/

public class Project1 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

Connection con;

Statement st;

ResultSet rs;

try

{

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

con=DriverManager.getConnection(

"jdbc:mysql://localhost:3306/ebookshop","root","");

st=con.createStatement();

rs=st.executeQuery("select \* from books");

while(rs.next())

{

System.out.println("\nID : "+rs.getInt(1));

System.out.println("TITLE : "+rs.getString(2));

System.out.println("AUTHOR : "+rs.getString(3));

System.out.println("PRICE : "+rs.getFloat(4));

System.out.println("QTY : "+rs.getInt(5));

}

}

catch(Exception e)

{

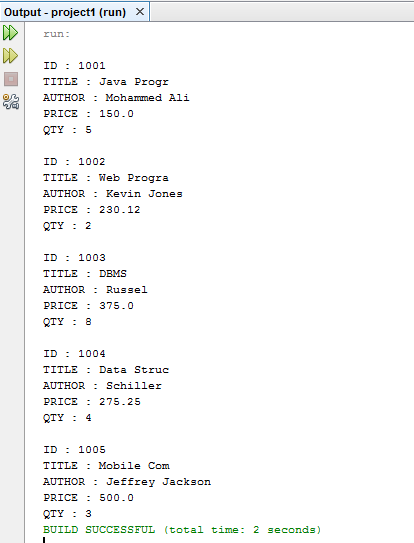
System.out.println("Error : " + e);

}

}

}

**Output**

****

1. Create a database “ebookshop” which contains a table called “books” with 5 columns. Establish JDBC Connectivity in java using Netbeans

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **TITLE** | **AUTHOR** | **PRICE** | **QTY** |
| 1001 | Java Programming | Mohammed Ali | 150.00 | 5 |
| 1002 | Web Programming | Kevin Jones | 230.12 | 2 |
| 1003 | DBMS | Russel | 375.00 | 8 |
| 1004 | Data Structures | Schiller | 275.25 | 4 |
| 1005 | Mobile Computing | Jeffrey Jackson | 500.00 | 3 |

1. **Update the table with the following values**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **TITLE** | **AUTHOR** | **PRICE** | **QTY** |
| 1001 | Java Programming | Mohammed Ali | 150.00 | 8 |
| 1002 | Web Programming | Kevin Jones | 230.12 | 2 |
| 1003 | DBMS | Russel | 375.00 | 8 |
| 1004 | Data Structures | David Thomas | 275.25 | 4 |
| 1005 | Mobile Computing | Jeffrey Jackson | 500.00 | 3 |

1. **Delete one row from the table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **TITLE** | **AUTHOR** | **PRICE** | **QTY** |
| 1001 | Java Programming | Mohammed Ali | 150.00 | 8 |
| 1003 | DBMS | Russel | 375.00 | 8 |
| 1004 | Data Structures | David Thomas | 275.25 | 4 |
| 1005 | Mobile Computing | Jeffrey Jackson | 500.00 | 3 |

1. **Sort the table by ascending order (By TITLE)**
2. **Sort the table by descending order (By AUTHOR)**

**Solution**

**Program**

package project1;

import java.sql.\*;

/\*\*

\*

\* @author DELL 3543

\*/

public class Project1 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

Connection con;

Statement st;

ResultSet rs;

try

{

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

con=DriverManager.getConnection(

"jdbc:mysql://localhost:3306/ebookshop","root","");

st=con.createStatement();

String sql = "UPDATE books " +

"SET AUTHOR = 'DavidThomas' WHERE ID in (1004)";

st.executeUpdate(sql);

String sql1 = "UPDATE books " +

"SET QTY = 8 WHERE ID in (1001)";

st.executeUpdate(sql1);

rs=st.executeQuery("select \* from books");

while(rs.next())

{

System.out.println("\nID : "+rs.getInt(1));

System.out.println("TITLE : "+rs.getString(2));

System.out.println("AUTHOR : "+rs.getString(3));

System.out.println("PRICE : "+rs.getFloat(4));

System.out.println("QTY : "+rs.getInt(5));

}

String sql2 = "DELETE FROM books " +

"WHERE id = 1002";

st.executeUpdate(sql2);

rs=st.executeQuery("select \* from books");

while(rs.next())

{

System.out.println("\nID : "+rs.getInt(1));

System.out.println("TITLE : "+rs.getString(2));

System.out.println("AUTHOR : "+rs.getString(3));

System.out.println("PRICE : "+rs.getFloat(4));

System.out.println("QTY : "+rs.getInt(5));

}

// Extract records in ascending order by first name.

System.out.println("Fetching records in ascending order...");

String sql3 = "SELECT ID,TITLE,AUTHOR,PRICE,QTY FROM books" +

" ORDER BY TITLE ASC";

rs = st.executeQuery(sql3);

while(rs.next())

{

System.out.println("\nID : "+rs.getInt(1));

System.out.println("TITLE : "+rs.getString(2));

System.out.println("AUTHOR : "+rs.getString(3));

System.out.println("PRICE : "+rs.getFloat(4));

System.out.println("QTY : "+rs.getInt(5));

}

System.out.println("Fetching records in descending order...");

String sql4 = "SELECT ID,TITLE,AUTHOR,PRICE,QTY FROM books" +

" ORDER BY AUTHOR DESC";

rs = st.executeQuery(sql4);

while(rs.next())

{

System.out.println("\nID : "+rs.getInt(1));

System.out.println("TITLE : "+rs.getString(2));

System.out.println("AUTHOR : "+rs.getString(3));

System.out.println("PRICE : "+rs.getFloat(4));

System.out.println("QTY : "+rs.getInt(5));

}

}

catch(Exception e)

{

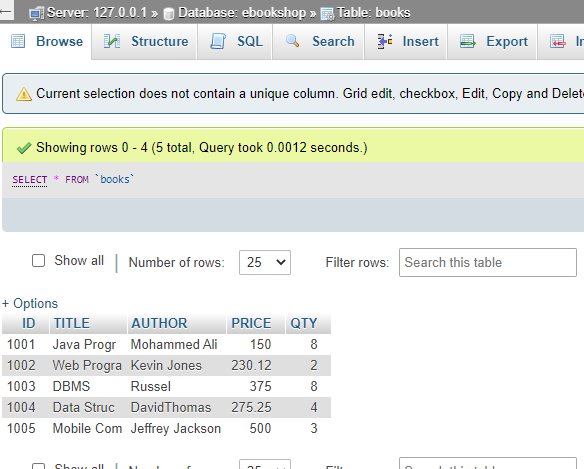
System.out.println("Error : " + e);

}

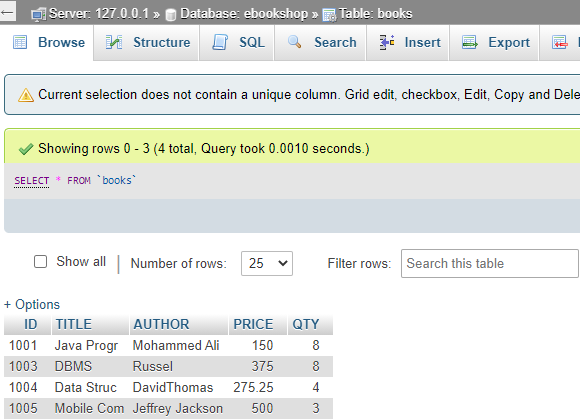
}

}

**Updation**

****

**Deletion**

****