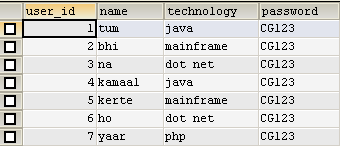
**Common Database for all the assignment of this topic**

*Create a database; with a table name USER having columns (fields) – NAME and TECHNOLOGY. Enter some data in the table.*

**Steps:**

* Create a database on MySQL (say DB name is ems),
* create a table “USER”;
* this table will be columns NAME (varchar, not null, unique) and Technology (varchar, not null), and password (varchar, not null),
* Enter some data in the table (using SQL)

**Upon executing - Select \* from user; database is showing following data in the user table**

****

**LAB 11.1**

*Write a program which fetch data from table (using JDBC) and display it on console*

**Steps:**

* Write Java class which fetches all the data of user table
* Add jar of mySql connection in the build path
* Load driver, create connection, create statement, execute query, receive response in result set
* As per result show/print the response on console

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** DBConnection {

**public** **static** **void** main(String[] args) {

**try** {

// step 1 - load the driver

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

// step 2 - create connection

Connection con = DriverManager.*getConnection*(

"jdbc:mysql://localhost:3306/ems", "root", "admin");

// step 3 - create statement

Statement st = con.createStatement();

// Creating query String

StringBuilder sb = **new** StringBuilder();

sb.append("select \* from user");

// step 4 - execute query

ResultSet rs = st.executeQuery(sb.toString());

System.*out*.println("NAME" + "\t" + "PWD" + "\t" + "TECH");

// step 5 - handing the result set;

**while** (rs.next()) {

String userName = rs.getString("name");

String password = rs.getString("password");

String technology = rs.getString("technology");

System.*out*.println(userName + "\t" + password

+ "\t" + technology);

}

} **catch** (InstantiationException e) {

e.printStackTrace();

} **catch** (IllegalAccessException e) {

e.printStackTrace();

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

**OUTPUT**

NAME PWD TECH

tum CG123 java

bhi CG123 mainframe

na CG123 dot net

kamaal CG123 java

kerte CG123 mainframe

ho CG123 dot net

yaar CG123 php

**Learning:**

* How to load DB driver, create connection, execute query, handle result

**LAB 11.2**

*Write a program- in which, end user will enter a name, and system will return the technology to which that user belongs. Use Scanner for user interaction. If user not found in DB, system should show appropriate message.*

**Steps:**

* Same as Lab 11.1; only need to add code of scanner (for user’s input) and fetch query will be change from all data to specific data

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.util.Scanner;

**public** **class** JdbcPractice {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.*in*);

System.*out*.println("Enter Name : ");

String name = sc.next();

Connection con = **null**;

**try** {

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

con = DriverManager.*getConnection*

("jdbc:mysql://localhost:3306/ems", "root", "admin");

Statement st = con.createStatement();

StringBuilder sb = **new** StringBuilder();

sb.append("select \* from user where name ='"+ name + "'");

ResultSet rs = st.executeQuery(sb.toString());

**while** (rs.next()) {

String userName = rs.getString("name");

String technology = rs.getString("technology");

System.*out*.println(userName + " belong to "

+ technology + " technology");

}

} **catch** (InstantiationException e) {

e.printStackTrace();

} **catch** (IllegalAccessException e) {

e.printStackTrace();

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}

**try** {

con.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

**OUTPUT**

Enter Name :

tum

tum belong to java technology

**Learning:**

* How to handle search query

**LAB 11.3**

*Write a program, where system will receive new data from user (which is name and technology) and add it into database. The name must be unique, so system should not allow entering duplicate data. System should show appropriate message on successful insertion or otherwise.*

**Steps:**

* Same as Lab 11.1; only before the fetch query, it will also have insert query

// Code below will come in between step 3 and step 4 of Lab 11.1 (refer to code snippet of Lab 11.1)

StringBuilder sb1 = **new** StringBuilder();

sb1.append("insert into user values(8, 'ohHo', 'java', 'pwd')");

st.executeUpdate(sb1.toString());

**Learning:**

* How to execute update query

**LAB 11.4**

*Execute the Lab 11.2 using prepared statement.*

**Steps:**

* Same as Lab 11.2; only change the statement to prepared statement to execute the fetch query; as shown in the code snippet below

/\*\*

\* Code for Lab 11.2 - using statement

\*/

/\*

Statement st = con.createStatement();

StringBuilder sb = new StringBuilder();

sb.append("select \* from user where name = '" + name + "'");

ResultSet rs = st.executeQuery(sb.toString());

\*/

/\*\*

\* Fetch query using Prepared statement

\*/

PreparedStatement ps = con.prepareStatement("select \* from user where

name = ?");

ps.setString(1, name);

ResultSet rs = ps.executeQuery();

**Learning:**

* What is prepared statement, how to create it and use it

**LAB 11.5**

*Write a program- which use to add new entry in USER table; which is combination of name and technology. Say, three are 4 insert query, and after 2 query executes, there is System.exit() to terminate program. Check the changes in DB for normal and transactional scenario.*

**Learning:**

* How to achieve transaction, and execute batch update

**LAB 11.6**

*Extension of Lab 11.5; add all 4 insert query in batch and then execute batch. System will execute (add data into database) in one go (batch update). It must be transactional (all or none)*

**Learning:**

* How to achieve transaction, and execute batch update