JDBC CRUD

Step 1: wap to create Database?

Step 2: wap to create table inside database.

Step 3: wap to inside the data inside table?

Step 4: Wap to Update the data inside table

Step 5: wap to Alter the table by adding column

Step 6: wap to Alter the table\_by deleteing column

Step 7: Table has been Drop

Step 8: fetch the value

Step 9: Delete the value row by row

Step 10: wap to fetch the values which are description inside table

Step 11: wap to update the data by using prepared statement

Step 12 : wap to insert data using dynamic

Step 13: wap to delete the data by using prepared statement dynamic input

**Eclipse ->** <https://www.eclipse.org/downloads/packages/release>

**2023-03**

<https://mvnrepository.com/>

mysql connector

[8.0.28](https://mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.28)

<https://mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.28>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

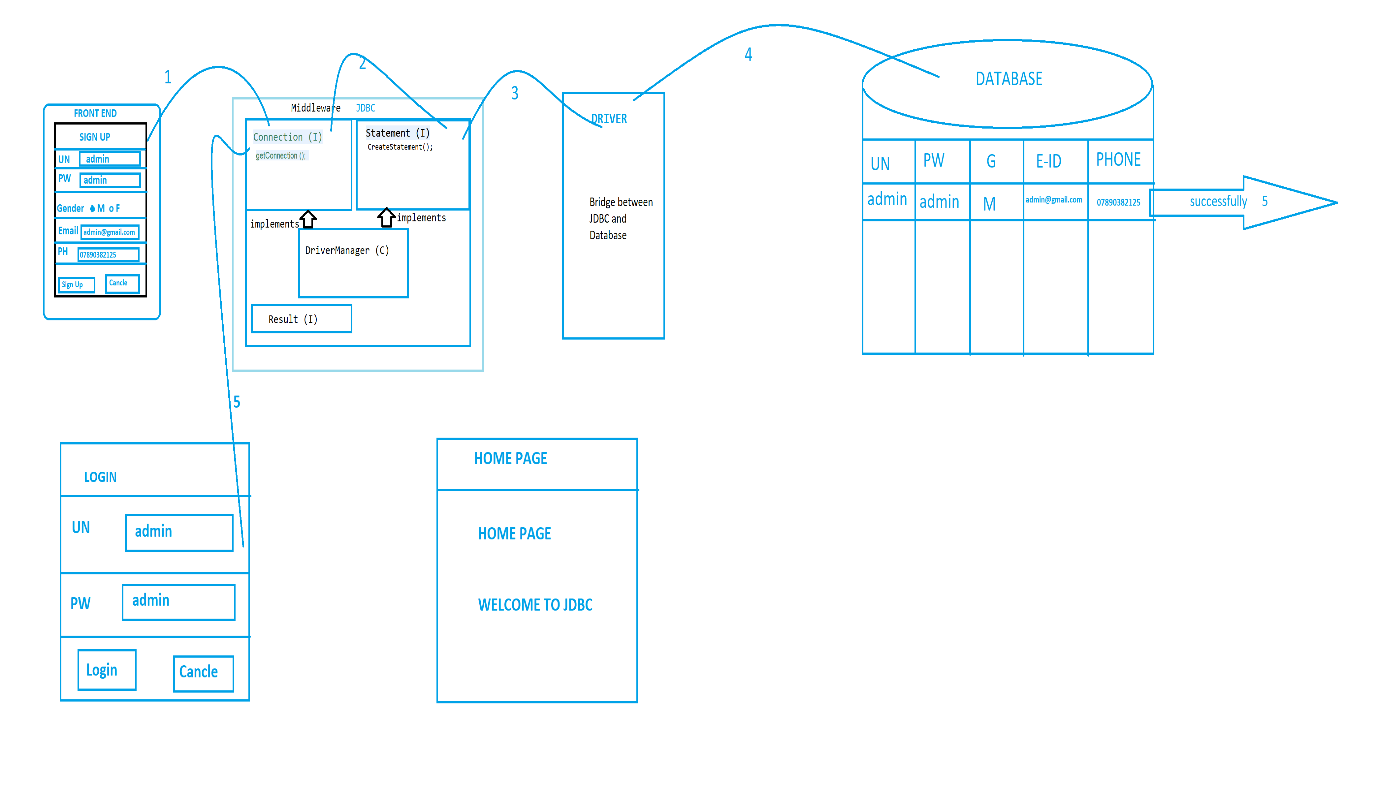
<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.28</version>

</dependency>

**JDBC** 

STEP 1: - If user enter any data in the front-end layer the data with flow to the middle way.

**STEP 2: -** In the middleware we are have the technology called code Java, Jdbc, Hibernate, sulet, spring as now we are learning one of the middleware technologies called jdbc.

**STEP 3: -** In the jdvc we are having sub components called connection interface statement interface result interface, driver manager class sub between connection interface it is one of the inbuild interface which present in side jdvc Technology. inside interface we are having one abstract method called get connection which is used to stable the connection between jdbc and database.

**SUB: - STATEMENT INTERFACE: -** It is also one of the inbuilt interface which is present Inside jdbc Technology. inside the statement there are server abstade method manual create statement, execute, execute query , execute update are present among this method create statement is the matter which is used to carry the date of form middleware to backend.

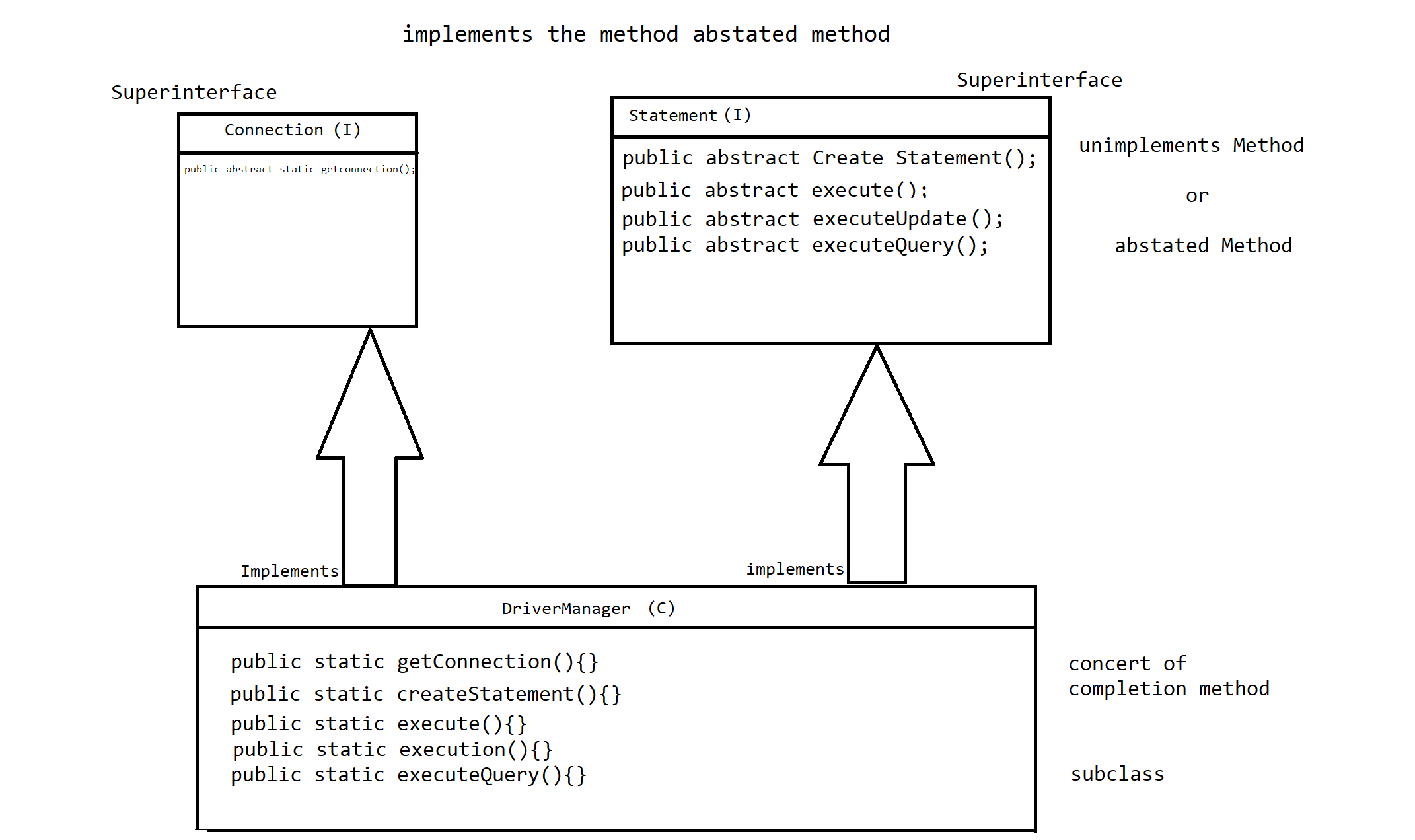
**RESULT INTERFACE:-** It is the also in present Inside jdbc Technology. it is used to face are to complete the data inside the database.

**STEP 4: - DRIVER: -** Driver is one of the complete it will completed Bridge between jdbc and backend. this driver side by from the respective data vendors

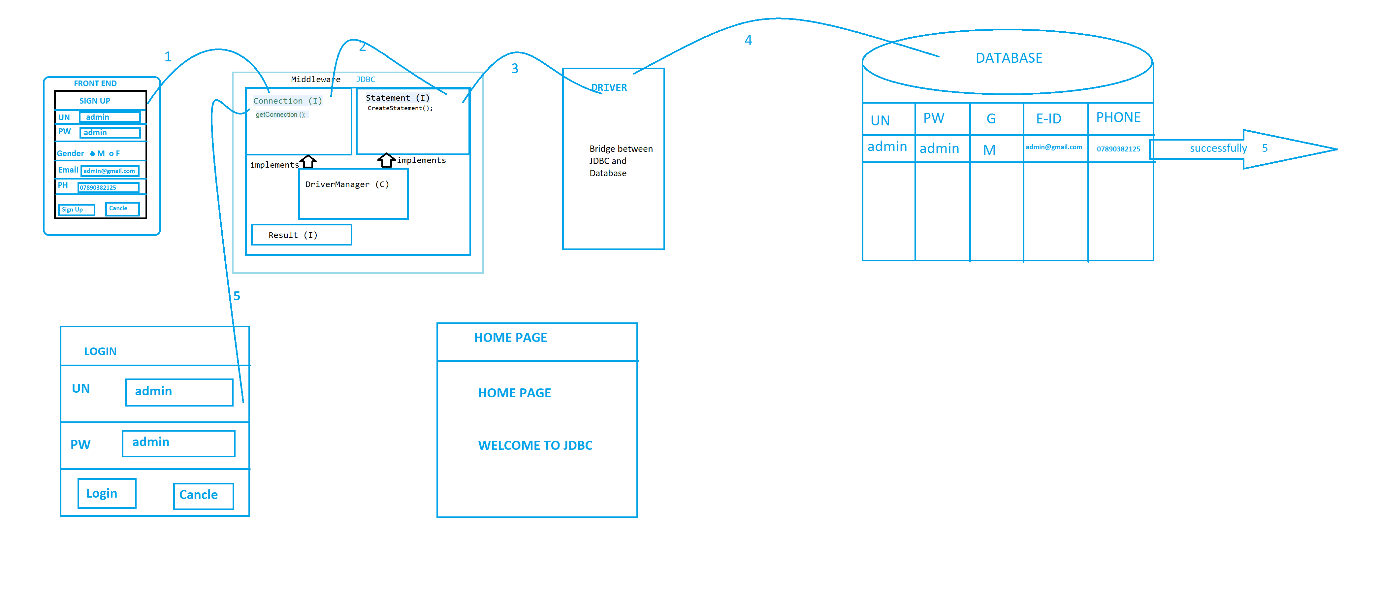
**For Example**

If I using mysql components database then drive shoud also be taken from my sql company only.

implement the method abstract method



Q. Write all the six connection staps in JDBC.



Import the package.

1. Load the driver.
2. Register the driver.
3. Establish the connection.
4. Create the statement.
5. Execute the query.
6. Close the connection.

**Steps to Connect with Database, explained in Detail (Java Database Connectivity) JDBC: -**

1. Load the driver. “Path”
2. Register the driver

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

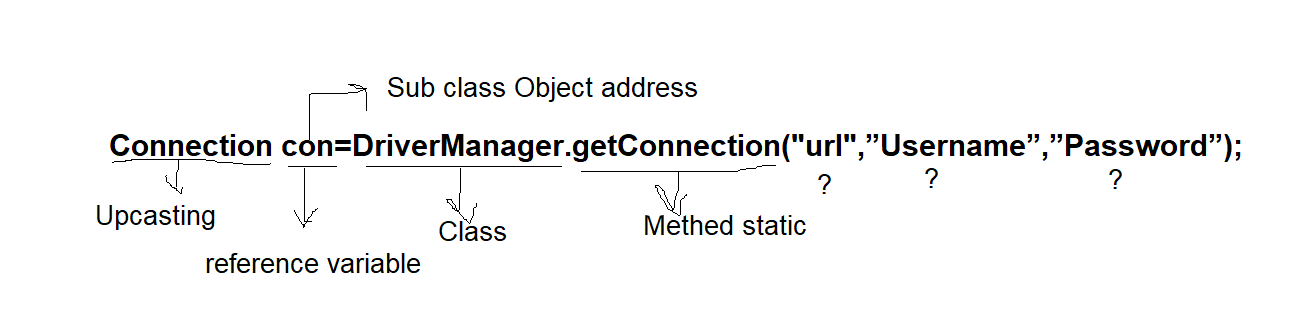
Class Name

Method() static method.

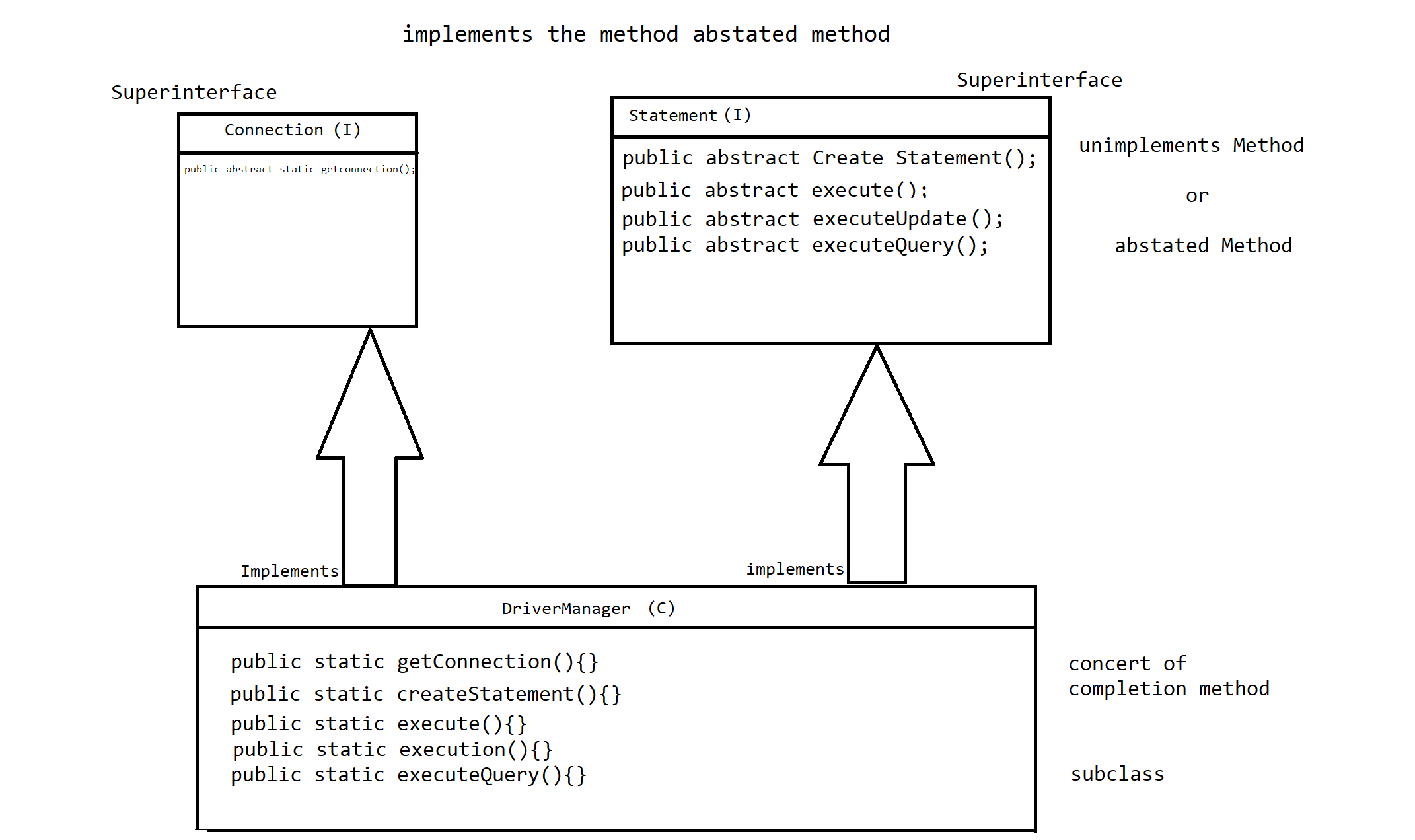
standard syntax given by the respective database vendors to register the driver.

1. Establish the connection.

Connection con=DriverManager.getConnection("url",”Username”,”Password”);



1. Create the statement.



1. Execute the query.
2. Close the connection.

Q. what do you mean by abstract method overriding.?

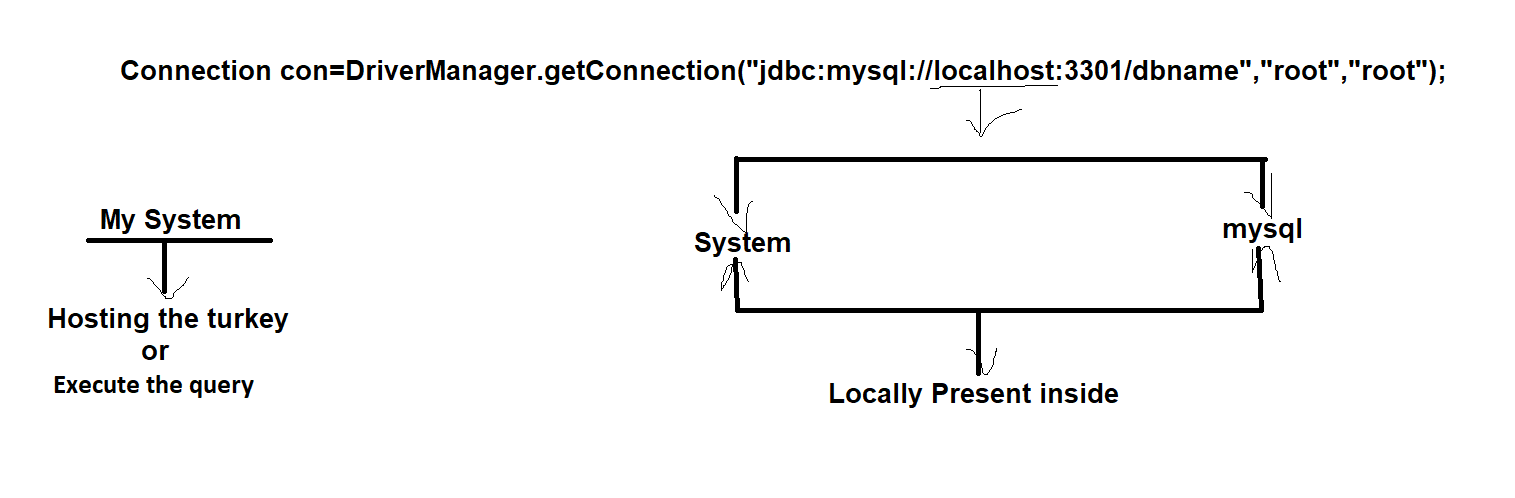
converting the abstract method into connected complete method by using implements class is called as abstract method.

Q. when ClassNotFoundException in occurred?

where Java vowels musense JVM faces any default to find the execute location the class. inside Java Platform this class not found Exception will occurs.

This classnotfoundexception is one of the Cast exceptions we can handle this exception by using try and catch or throws.

1. Load the driver.
2. Register the driver.
3. Establish the connection.



Pom.xml

Import

<dependencies>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.28</version>

</dependency>

</dependencies>

Q.1 wap to create Database?

// wap to create Database ?

**package** Start\_JDBC;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Createdb {

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException, SQLException {

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

Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306", "root", "root");

Statement statement = connection.createStatement();

statement.execute("create Database JDBC\_curd");

System.***out***.println("create Database");

}

}

Q. 2 wap to create table inside database.

// wap to create table inside database

**package** Start\_JDBC;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Create\_Table\_inside\_db {

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException, SQLException {

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

Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/JDBC\_curd", "root", "root");

Statement statement = connection.createStatement();

statement.execute("create table Teacher (t\_id integer primary key, t\_name varchar (30) not null, t\_eid varchar (70) not null unique, t\_mobile integer not null unique)");

System.***out***.println("Table has been created");

}

}

Q. 3 wap to inside the data inside table?

// wap to inside the data inside table?

**package** Start\_JDBC;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Inside\_the\_data\_inside\_table {

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException, SQLException {

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

Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/JDBC\_curd", "root", "root");

Statement statement = connection.createStatement();

statement.execute("insert into Teacher values (1,'Sachin','sachin@gmail.com',7890382125)");

System.***out***.println("Inside the data inside table");

}

}

Q. 4. Wap to Update the data inside table

// 4. Wap to Update the data inside table

**package** Start\_JDBC;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Update\_the\_data\_inside\_table {

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException, SQLException {

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

Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/JDBC\_curd", "root", "root");

Statement statement = connection.createStatement();

statement.execute("update Teacher set t\_name='Sachin Kumar',t\_eid='Sachink@gmail.com', t\_mobile='07890382125'");

System.***out***.println("Data has been Update");

}

}

Q. 5. wap to Alter the table by adding column

// 5. wap to Alter the table by adding column

**package** Start\_JDBC;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Alter\_the\_table\_by\_adding\_column {

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException, SQLException {

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

Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/JDBC\_curd", "root", "root");

Statement statement = connection.createStatement();

statement.execute("alter table Teacher add column t\_salary bigint not null");

System.***out***.println("Table has been alter Teacher");

}

}

Q.6 wap to Alter the table\_by deleteing column

// 6. wap to Alter the table\_by deleteing column

**package** Start\_JDBC;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Alter\_the\_table\_by\_deleteing\_column {

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException, SQLException {

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

Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/JDBC\_curd", "root", "root");

Statement statement = connection.createStatement();

statement.execute("alter table Teacher drop column t\_mobile");

System.***out***.println("Column has been Drop");

}

}

Q.7 Table has been Drop

// Table has been Drop

**package** Start\_JDBC;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Delete\_table {

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException, SQLException {

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

Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/JDBC\_curd", "root", "root");

Statement statement = connection.createStatement();

statement.execute(" drop table Teacher ");

System.***out***.println("Table has been Drop");

}

}

Q.8 fetch the value

//8 fetch the value

**package** Start\_JDBC;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Fetch\_the\_value {

**public** **static** **void** main(String[] args) **throws** SQLException, ClassNotFoundException {

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

Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/JDBC\_curd", "root", "root");

Statement statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery("select\*from teacher where t\_id=1");

**while**(resultSet.next())

{

System.***out***.println("t\_id"+" "+resultSet.getInt(1)+"\n"+"t\_name"+" "+resultSet.getString(2)+"\n"+"t\_eid"+" "+resultSet.getString(3)+"\n"+"t\_mobile"+" "+resultSet.getBigDecimal(4)+"\n"+"t\_salary"+" "+resultSet.getBigDecimal(5));

}

}

}

Q. 9 Delete the value row by row

// 9 Delete the value row by row

**package** Start\_JDBC;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Delete\_the\_value\_row\_by\_row {

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException, SQLException {

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

Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/JDBC\_curd", "root", "root");

Statement statement = connection.createStatement();

statement.execute("delete from Teacher where t\_id =2");

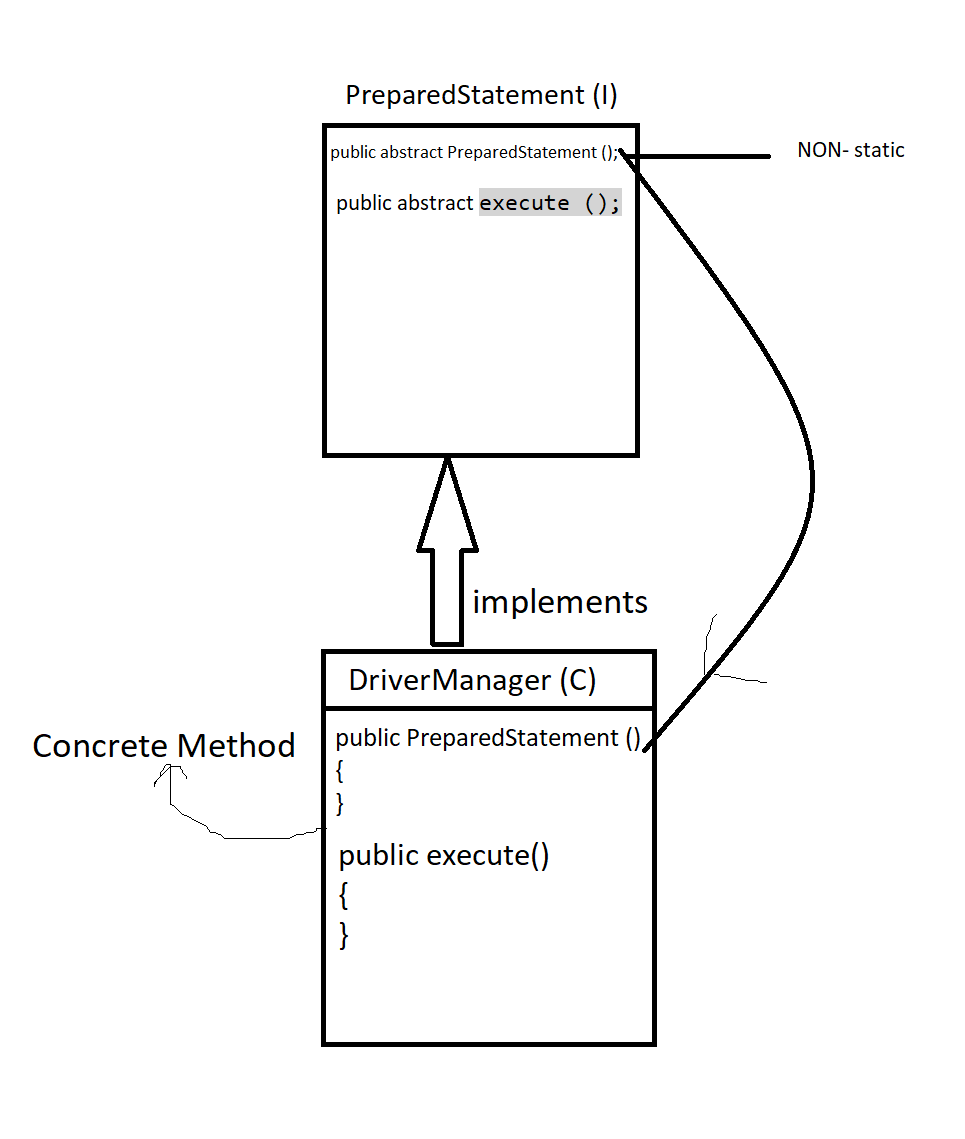
System.***out***.println("1st row data has been deleted Successfully");

}

}

PreparedStatement interface (I)

PreparedStatement one is the interface which belongs to java sql package. by using the PreparedStatement we are going to insert the data in dynamic manner by using scanner class.



JDBC Driver

There are 4 types of JDBC drivers

1. JDBC-ODBC bridge driver (open database connect)
2. Native-API driver (partially java driver)
3. Network Protocol driver (fully java driver)
4. Thin driver (fully java driver)

Q.10 wap to update the data by using prepared statement.

// 10 wap to update the data by using prepared statement

**package** Start\_JDBC;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.SQLException;

**import** java.util.Scanner;

**public** **class** update\_the\_data\_by\_using\_prepared\_statement {

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException, SQLException {

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

Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/JDBC\_curd", "root", "root");

PreparedStatement preparedStatement = connection.prepareStatement("update Teacher set t\_name=?,t\_eid=?,t\_mobile=?,t\_salary=? where t\_id=?");

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

System.***out***.println("Enter the Teacher Name to Update");

String t\_name = scanner.next();

System.***out***.println("Enter the Teacher Email Id to Update");

String t\_email = scanner.next();

System.***out***.println("Enter the Teacher Mobile Number to Update");

**long** t\_phone = scanner.nextLong();

System.***out***.println("Enter the Teacher Salary to Update");

**long** t\_salary = scanner.nextLong();

**int** t\_id = scanner.nextInt();

preparedStatement.setString(1, t\_name);

preparedStatement.setString(2, t\_email);

preparedStatement.setLong(3, t\_phone);

preparedStatement.setLong(4, t\_salary);

preparedStatement.setInt(5, t\_id);

preparedStatement.execute();

System.***out***.println("Data has been update");

}

}

Q.11 wap to delete the data by using prepared statement dynamic input

// 11 wap to delete the data by using prepared statement dynamic input

**package** Start\_JDBC;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.SQLException;

**import** java.util.Scanner;

**public** **class** Delete\_the\_data\_by\_using\_prepared\_statement {

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException, SQLException {

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

Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/JDBC\_curd", "root", "root");

PreparedStatement preparedStatement = connection.prepareStatement("delete from Teacher where t\_id=?");

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

System.***out***.println("Enter the Teacher ID to Delete the Data");

**int** t\_id = scanner.nextInt();

preparedStatement.setInt(1, t\_id);

preparedStatement.execute();

System.***out***.println("Data has been Deleted");

}

}

Q12 wap to fetch the values which are description inside table

//12 wap to fetch the values which are description inside table

**package** Start\_JDBC;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Fetch\_the\_values\_which\_are\_description\_inside\_table {

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException, SQLException {

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

Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/JDBC\_curd", "root", "root");

Statement statement =connection.createStatement();

ResultSet resultSet = statement.executeQuery("select \* from Teacher");

**while**(resultSet.next())

{

System.***out***.println("Teacher ID "+resultSet.getInt(1));

System.***out***.println("Teacher Name "+resultSet.getString(2));

System.***out***.println("Teacher Email id "+resultSet.getString(3));

System.***out***.println("Teacher Mobile Number "+resultSet.getLong(4));

System.***out***.println("Teacher Salary "+resultSet.getLong(5));

}

}

}

Q 13 wap to insert data using dynamic

// 13 wap to insert data using dynamic

**package** Start\_JDBC;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.SQLException;

**import** java.util.Scanner;

**public** **class** Insert\_data\_using\_dynamic {

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException, SQLException {

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

Connection connection = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/JDBC\_curd", "root", "root");

PreparedStatement preparedStatement= connection.prepareStatement("Insert into Teacher values (?,?,?,?,?)");

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

System.***out***.println("Enter the Teacher Id to Insert");

**int** t\_id = scanner.nextInt();

System.***out***.println("Enter the Teacher Name to Insert");

String t\_name = scanner.next();

System.***out***.println("Enter the Teacher Email Id to Insert");

String t\_email = scanner.next();

System.***out***.println("Enter the Teacher Mobile Number to Insert");

**long** t\_phone = scanner.nextLong();

System.***out***.println("Enter the Teacher Salary to Insert");

**long** t\_salary = scanner.nextLong();

preparedStatement.setInt(1, t\_id);

preparedStatement.setString(2, t\_name);

preparedStatement.setString(3, t\_email);

preparedStatement.setLong(4, t\_phone);

preparedStatement.setLong(5, t\_salary);

preparedStatement.execute();

System.***out***.println("Data Insert");

}

}