DBMS Assignment C2

Program:

```
package com.company;

import java.sql.*;

public class Main {

  public static void main(String[] args) {

        // write your code here

    try{

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

      Connection con=DriverManager.getConnection(

          "jdbc:mysql://localhost:3306/today","root","dhruvil283");

      Statement stmt = con.createStatement();

     stmt.executeUpdate("create table a(ID int,Name varchar(20),Age int(3));");

     stmt.executeUpdate("insert into a values (1,'Dhruvil',20)");

     stmt.executeUpdate("insert into a values (2,'Soham',20)");

      ResultSet rs = stmt.executeQuery("select * from a");

      while (rs.next()) {

        System.out.println(rs.getInt(1) + "  " + rs.getString(2) + "  " + rs.getString(3));

      }

      con.close();


    } catch (Exception e) {

      System.out.println(e);

    }

  }

}
```

****

Output:

```
1   Dhruvil   20

2   Soham   20
```

****

MES College of Engineering Pune-01

Department of Computer Engineering

Name of Student: Dhruvil Shah	**Class: TE Comp 1**
Semester/Year: 5th/2020	**Roll No: 047**
Date of Performance:	**Date of Submission:**
Examined By:	**Experiment No: Group C-02**

PART: C ASSIGNMENT NO: 02

AIM: To implement MYSQL/Oracle database connectivity with PHP/ python/Java Implement Database navigation operations (add, delete, edit,) using ODBC/JDBC.

OBJECTIVES:

- To connect to MySQL database with the help of Java program.
- Implement database operations on connected database.

PRE-REQUISITES:

- Basics of relational database-MySQL.
- Mysql 5.7 or later.
- mysql-connector-java_8.0.12
- JDK 1.6
- Eclipse 4.2

THEORY:

MySQL provides connectivity for client applications developed in the Java programming language with MySQL Connector/J, a driver that implements the Java Database Connectivity (JDBC) API.

MySQL Connector/J is a JDBC Type 4 driver. Different versions are available that are compatible with the JDBC 3.0 and JDBC 4.x specifications. The Type 4 designation means that the driver is a pure Java implementation of the MySQL protocol and does not rely on the MySQL client libraries.

The interface for accessing relational databases from Java is *Java Database Connectivity (JDBC)*. Via JDBC you create a connection to the database, issue database queries and update as well as receive the results.

JDBC provides an interface which allows you to perform SQL operations independently of the instance of the used database. To use JDBC, you require the database specific implementation of the JDBC driver. For large-scale programs that use common design patterns of data access, consider using one of the popular persistence frameworks such as Hibernate, Spring's JDBC templates or MyBatis SQL Maps to reduce the amount of JDBC code for you to debug, tune, secure, and maintain.

When you are using JDBC outside of an application server, the DriverManager class manages the establishment of connections. Specify to the DriverManager which JDBC drivers to try to make Connections with. The easiest way to do this is to use Class.forName() on the class that implements the java.sql.Driver interface. With MySQL Connector/J, the name of this class is com.mysql.jdbc.Driver. With this method, you could use an external configuration file to supply the driver class name and driver parameters to use when connecting to a database.

In order to connect to MySQL database you need three things database URL, username, and password and we are using default user root here, which is created during MySQL installation.

The following section of Java code shows how you might register MySQL Connector/J from the main() method of your application. Make sure you have connector installed correctly and the CLASSPATH set up. Also, ensure that MySQL is configured to accept external TCP/IP connections.

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

public class LoadDriver {
   public static void main(String[] args) {
      try {
         // The newInstance() call is a work around for some
         // broken Java implementations
```

```
            Class.forName("com.mysql.jdbc.Driver").newInstance();
        } catch (Exception ex) {
            // handle the error
        }
    }
}
```

After the driver has been registered with the DriverManager, you can obtain a Connection instance that is connected to a particular database by calling DriverManager.getConnection():

When host is not specified, the default value of localhost is used. port is a standard port number, i.e., an integer between 1 and 65535. The default port number for an ordinary MySQL connection is **3306**, and it is **33060** for a connection using the X Protocol. If port is not specified, the corresponding default is used.

```
Import java.sql.*;

            class MysqlCon{
            public static void main(String args[]){
            try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection con=DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/dbname","root","root");
            //here dbname is database name, root is username and password
            Statement stmt=con.createStatement();
            ResultSetrs=stmt.executeQuery("select * from emp");
            while(rs.next())
            System.out.println(rs.getInt(1)+""+rs.getString(2)+""+rs.getString(3));
            con.close()
            }catch(Exceptione){System.out.println(e);}
            }
            }
```

Statement objects allow you to execute basic SQL queries and retrieve the results through the ResultSet class, which is described later. To create a Statement instance, you call the createStatement() method on the Connection object you have retrieved using one of the DriverManager.getConnection() or DataSource.getConnection() methods described earlier.

Once you have a Statement instance, you can execute a SELECT query by calling the executeQuery(String) method with the SQL you want to use. To update data in the database, use the executeUpdate(String SQL) method. This method returns the number of rows matched by the update statement, not the number of rows that were modified.

If you do not know ahead of time whether the SQL statement will be a SELECT or an UPDATE/INSERT, then you can use the execute(String SQL) method. This method will return true if the SQL query was a SELECT, or false if it was an UPDATE, INSERT, or DELETE statement. If the statement was a SELECT query, you can retrieve the results by calling the getResultSet() method. If the statement was an UPDATE, INSERT, or DELETE statement, you can retrieve the affected rows count by calling getUpdateCount() on the Statement instance.

CONCLUSION:

QUESTIONS:

1. Enlist applications where the MySQL Java connectivity might be used. Give examples of other database connectivities.
2. Write installation steps for Mysql on Ubuntu.
3. What are different Packages needed to connect Java with MySQL?
4. What is relational database? Give examples.
5. The architecture used in this assignment is three tier or two tier? Justify your answer.

Q1 Enlist applications where the MySQL, Java connectivity might be used. Give examples of other database connectivities.

Ans Applications where MySQL, Java connectivity is used:

1] SQL Formatter

2] WWW SQL Design.

Other database connectivities:

1] Mongo DB

2] Oracle

3] SQLite

4] Ingres.

Q2 Write installation steps for MySQL on Ubuntu

Ans Step 1: Installing MySQL : sudo apt install mysql-server

Step 2: Configuring MySQL

Step 3: Adjusting User authentication and privileges

Step 4: Testing MySQL

Q3 What are different packages needed to connect java with MySQL

Ans The different packages needed to connect java to with MySQL are:

```
import java.sql.*;
```

Also add the mysqlconnector.jar file.

Q4 What is relational database ? Give examples.

Ans A relational database is a type of database that stores and provides access to data points that are related to one another. Relational databases are based on the relational model, an intutive, straightforward way of representing data in tables eg. MySQL, Microsoft SQL Server, IBM D2.

Q5 The architecture used in this assignment is three tier or two tier ? Justify your answer.

Ans The architecture used in this assignment is two tier architecture. Since we are using only two layers ie client and database. Using a client server architecture in the form of MySQL database.