5.Develop a java standalone application that connects with the database (Oracle / mySql) and perform the CRUD operation on the database tables.

AIM: A java standalone application that connects with the database (Oracle / mySql) and perform the CRUD operation on the database tables.

DESCRIPTION:Eclipse IDE,Mysql are essential for performing CRUD operations.

Project Structure:1. Java file contains jdbc connectivity statements.

2. mysql contains the database and tables

3. mysql connector

Connecting to Database:

A JDBC driver is a set of Java classes that implement the JDBC interfaces, targeting a specific database. The JDBC interfaces comes with standard Java, but the mplementation of these interfaces is specific to the database we need to connect to. Such an implementation is called a JDBC driver. JDBC drivers are typically supplied by the database vendor, but may sometimes be provided by the developer community around a database.

## JDBC Driver Type List

There are four different JDBC driver types. These driver types are:

* Type 1: JDBC-ODBC bridge JDBC driver
* Type 2: Java + Native code JDBC driver
* Type 3: All Java + Middleware translation JDBC driver
* Type 4: All Java JDBC driver.

Today, most JDBC drivers are type 4 drivers. Nevertheless, I will just discuss the 4 types of JDBC drivers shortly.

## Type 1 JDBC Driver

A type 1 JDBC driver consists of a Java part that translates the JDBC interface calls to ODBC calls. An ODBC bridge then calls the ODBC driver of the given database. Type 1 drivers are (were) mostly intended to be used in the beginning, when there were no type 4 drivers (all Java drivers). Here is an illustration of how a type 1 JDBC driver is organized:

|  |
| --- |
| **[Type 1 JDBC driver.](http://tutorials.jenkov.com/images/java-jdbc/driver-type-1.png)** |
| **Type 1 JDBC driver.** |

## Type 2 JDBC Driver

A type 2 JDBC driver is like a type 1 driver, except the ODBC part is replaced with a native code part instead. The native code part is targeted at a specific database product. Here is an illustration of a type 2 JDBC driver:

|  |
| --- |
| **[Type 2 JDBC driver.](http://tutorials.jenkov.com/images/java-jdbc/driver-type-2.png)** |
| **Type 2 JDBC driver.** |

## Type 3 JDBC Driver

A type 3 JDBC driver is an all Java driver that sends the JDBC interface calls to an intermediate server. The intermediate server then connects to the database on behalf of the JDBC driver. Here is an illustration of a type 3 JDBC driver:

|  |
| --- |
| **[Type 3 JDBC driver.](http://tutorials.jenkov.com/images/java-jdbc/driver-type-3.png)** |
| **Type 3 JDBC driver.** |

## Type 4 JDBC Driver

A type 4 JDBC driver is an all Java driver which connects directly to the database. It is implemented for a specific database product. Today, most JDBC drivers are type 4 drivers. Here is an illustration of how a type 4 JDBC driver is organized:

|  |
| --- |
| **[Type 4 JDBC driver.](http://tutorials.jenkov.com/images/java-jdbc/driver-type-4.png)** |
| **Type 4 JDBC driver.** |

1.First install mysql from mysql.org

2.Install Eclipse IDE

3.download connector 

package Databasecon;

import java.sql.\*;

import java.sql.Connection;

import java.sql.DriverManager;

import java.util.\*;

public class Mysqlcon

{

public static void main(String[] args)

{

Connection cn;

Statement st;

ResultSet rs;

try

{

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

cn = DriverManager.getConnection("jdbc:mysql://localhost:3306/college", "root","root");

st = cn.createStatement();

System.out.println("Welcome To Gurunanak Institutions");

System.out.println("------MENU------");

System.out.println("1.Insert");

System.out.println("2.EDIT");

System.out.println("3.Delete");

System.out.println("4.Display");

System.out.println("5.Exit");

System.out.println("----------------");

int opt=0;

String rno = "", sname = "", mobile = "", sql ="";

Scanner sc = new Scanner(System.in);

while (opt != 5)

{

System.out.println ("Enter Your Option");

opt = sc.nextInt();

switch (opt)

{

case 1:

{

System.out.println("Enter rno");

rno = sc.next();

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

sname = sc.next();

System.out.println("Enter Mobile");

mobile = sc.next();

sql = "insert into stu values(" + "'" + rno + "'" + "," + "'" + sname + "'" + "," + "'" + mobile + "'" + ")";

if (st.executeUpdate(sql) > 0)

{

System.out.println("Record Inserted");

}

}

break;

case 2:

{

System.out.println("Enter rno to Update");

rno = sc.next();

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

sname = sc.next();

System.out.println("Enter Mobile");

mobile = sc.next();

sql = "update stu set sname=" + "'" + sname + "'" + "," + "mobile=" + "'" + mobile + "'" + " where rno='" + rno + "'";

if (st.executeUpdate(sql) > 0)

{

System.out.println("Record Updated");

}

else

{

System.out.println("Operation Failed");

}

}

break;

case 3:

{

System.out.println("Enter rno to delete");

rno = sc.next();

sql = "delete from stu where rno=" + "'" + rno + "'";

if (st.executeUpdate(sql) > 0)

{

System.out.println("Record deleted");

}

else

{

System.out.println("Operation Failed");

}

}

break;

case 4:

{

sql = "select \* from stu";

rs = st.executeQuery(sql);

System.out.println("Htno\tSname\tMobile");

while (rs.next())

{

System.out.println(rs.getString("rno") + "\t" + rs.getString("SName") + "\t" + rs.getString("mobile"));

}

rs.close();

}

break;

case 5:

{

opt = 5;

System.out.println("Thank You");

st.close();

cn.close();

}

break;

default:

{

System.out.println("Choose Option between 1 and 5 only");

}}}}

catch (Exception ex)

{

System.out.println(ex.getMessage());

}} }

**Output:-**

Welcome To Gurunanak Institutions

------MENU------

1.Insert

2.EDIT

3.Delete

4.Display

5.Exit

----------------

Enter Your Option

4

Htno Sname Mobile

100 aaa 12345

200 bbb 23456

Enter Your Option

1

Enter rno

300

Enter Name

ccc

Enter Mobile

34567

Record Inserted

Enter Your Option

4

Htno Sname Mobile

100 aaa 12345

200 bbb 23456

300 ccc 34567

Enter Your Option

3

Enter rno to delete

200

Record deleted

Enter Your Option

4

Htno Sname Mobile

100 aaa 12345

300 ccc 34567

Enter Your Option

2

Enter rno to Update

300

Enter Name

ddd

Enter Mobile

456789

Record Updated

Enter Your Option

4

Htno Sname Mobile

100 aaa 12345

300 ddd 456789

Enter Your Option