```java
/**
* Q 1) Write a program to perform CRUD operations using JDBC.
*/

import java.sql.*;
import java.util.Scanner;

public class assign2 {


public static void main(String[] args) {
Connection con = null;
Scanner sc = new Scanner(System.in);

try {

/**
* Need to execute first export CLASSPATH=$CLASSPATH:mysql-connector-java-8.0.21.jar
*/

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/kaustubh?
allowPublicKeyRetrieval=true&useSSL=false", "kaustubh",
"kaustubh");

// connection TEST
System.out.println("Connection is successful !!!!!");

// Create table if does not exists
String CREATE_TABLE_SQL = "CREATE TABLE IF NOT EXISTS kaustubh.student (" + "id INT,"
+ "sname VARCHAR(30), " + "sclass VARCHAR(30)," + "PRIMARY KEY (id))";

// end create table if does not exists

Statement stmt = con.createStatement();

stmt.executeUpdate(CREATE_TABLE_SQL);

System.out.println("Table created");

// Some initial fixed insertion queries

insert(con, 1, "Kaustubh", "MCA II");
insert(con, 2, "Angad", "MCA II");
insert(con, 3, "Sagar", "MCA III");
insert(con, 4, "Rajesh", "MCA II");

// Some initial fixed insertion queries

selection(con, sc, stmt);
```

```
} catch (Exception e) {
e.printStackTrace();
}
}

static void selection(Connection con, Scanner sc, Statement stmt) {
System.out
.println("Select Operation to Perform :\n1.Insert\n2.Update\n3.Delete\n4.Select\n5.Drop Table\
n6.Exit");

int choice = sc.nextInt();

switch (choice) {
case 1:
System.out.println("Enter Student id ,Student Name & Student Class");
int id=sc.nextInt();
sc.nextLine();

String sname=sc.nextLine();
String sclass=sc.nextLine();

insert(con, id, sname, sclass);
selection(con, sc, stmt);
break;

case 2:

System.out.println("Enter ID to Update & new Class");
int sid=sc.nextInt();
sc.nextLine();
String ssclass=sc.nextLine();
updateData(con, sid, ssclass);
selection(con, sc, stmt);
break;


case 3:
System.out.println("Enter ID to Delete");
delete(con, sc.nextInt());
selection(con, sc, stmt);
break;


case 4:
showData(con);
selection(con, sc, stmt);
break;


case 5:
try {
```

```
stmt.executeUpdate("DELETE from student");
System.out.println("Student Table DROPPED . Program needs to restart to fire initial queries");
sc.close();
con.close();

} catch (SQLException e) {
e.printStackTrace();
}
break;



case 6: System.exit(0);
break;



default:System.out.println("The option you selected was invalid\nPlease try again?");
selection(con, sc,stmt);
break;


}


}

static void insert(Connection con,Integer sid,String name, String ssclass){

try {
String sql = "INSERT INTO student (id, sname, sclass) VALUES (?, ?, ?)";
PreparedStatement statement = con.prepareStatement(sql);
statement.setInt(1, sid);
statement.setString(2, name);
statement.setString(3, ssclass);
int rowsInserted = statement.executeUpdate();
if (rowsInserted > 0) {
System.out.println("A new student was inserted successfully!");
}

} catch (Exception e) {
e.printStackTrace();

}

}

static void showData(Connection con) {
try {

String selectSql = "SELECT * FROM student";
```

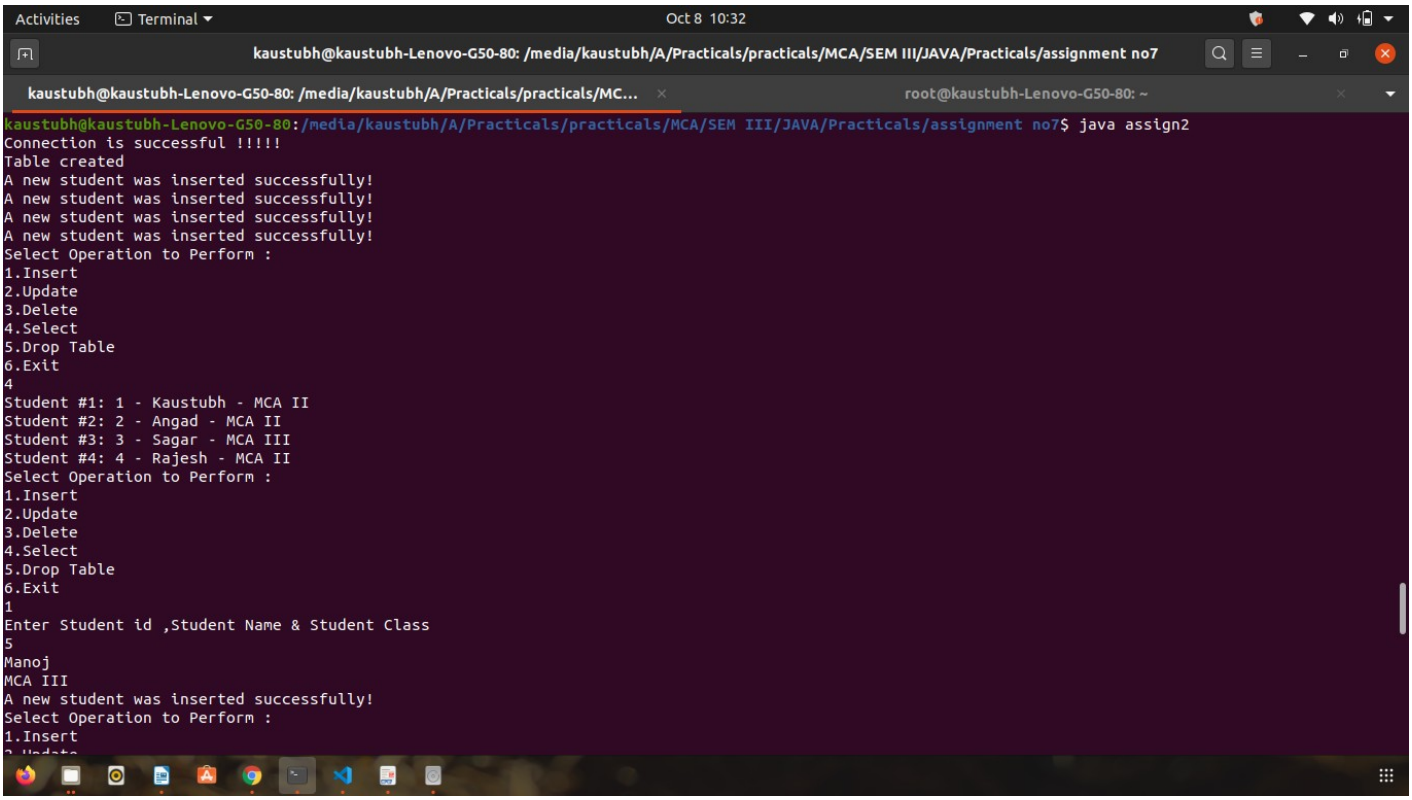
```
Statement selectStatement = con.createStatement();
ResultSet result = selectStatement.executeQuery(selectSql);
int count = 0;
while (result.next()) {
String id = result.getString("id");
String sname = result.getString("sname");
String sclass = result.getString("sclass");
String output = "Student #%d: %s - %s - %s";
System.out.println(String.format(output, ++count, id, sname, sclass));
}
} catch (Exception e) {

e.printStackTrace();
}
}

static void updateData(Connection con,Integer id,String sclass) {
try {
String updateSql = "UPDATE student SET sclass=? WHERE id=?";
PreparedStatement updateStatement = con.prepareStatement(updateSql);
updateStatement.setString(1, "MCA I");
updateStatement.setInt(2, 4);
int rowsUpdated = updateStatement.executeUpdate();
if (rowsUpdated > 0) {
System.out.println("An existing student was updated successfully!");
}

} catch (Exception e) {
e.printStackTrace();
}
}


static void delete(Connection con,Integer sid)
{
try {
String deleteSql = "DELETE FROM student WHERE id=?";
PreparedStatement deletestatement = con.prepareStatement(deleteSql);
deletestatement.setInt(1, sid);
int rowsDeleted = deletestatement.executeUpdate();
if (rowsDeleted > 0) {
System.out.println("A Student was deleted successfully!");
}
} catch (Exception e) {
e.printStackTrace();
}
}


}
```

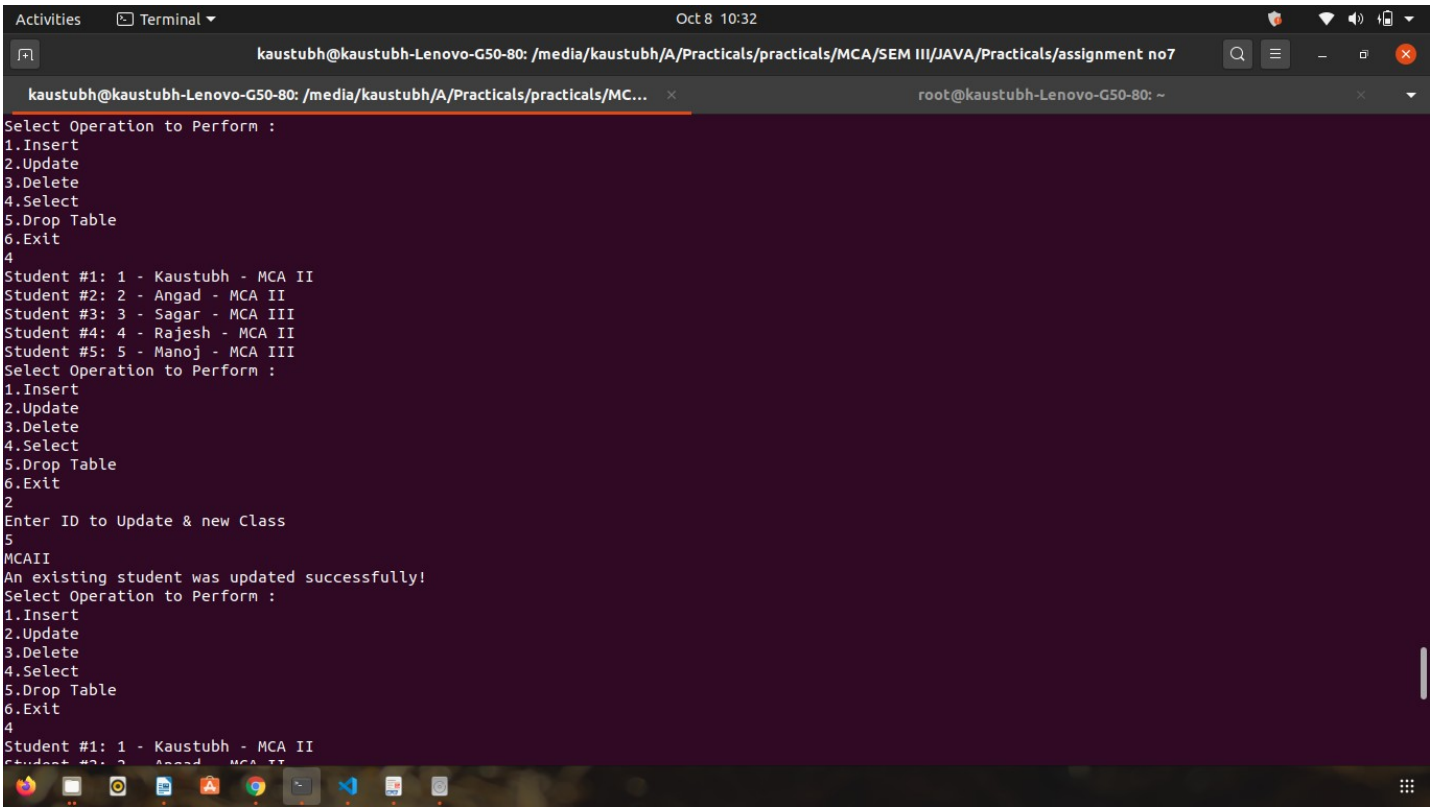
Output:

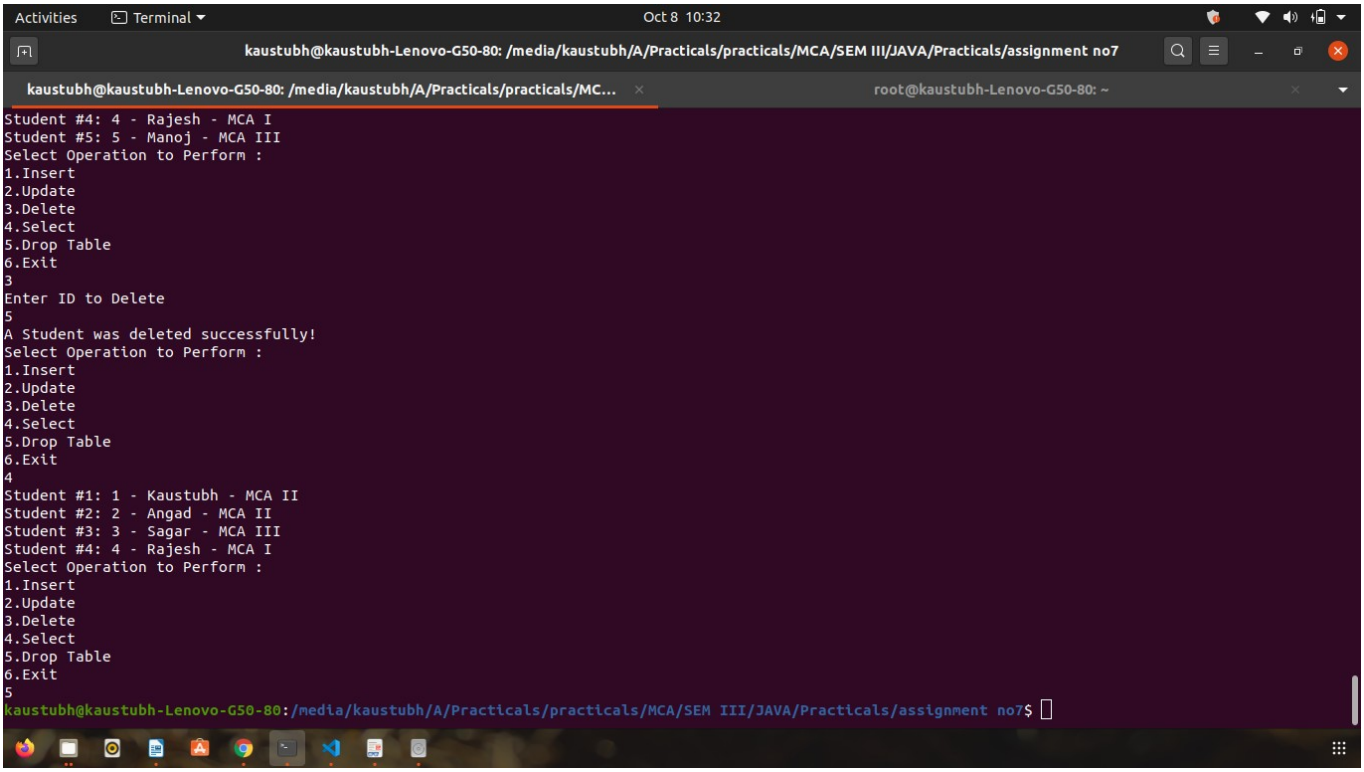
```
kaustubh@kaustubh-Lenovo-G50-80:/media/kaustubh/A/Practicals/
practicals/MCA/SEM III/JAVA/Practicals/assignment no7$ javac
assign2.java
kaustubh@kaustubh-Lenovo-G50-80:/media/kaustubh/A/Practicals/
practicals/MCA/SEM III/JAVA/Practicals/assignment no7$ java
assign2
Connection is successful !!!!!
Table created
A new student was inserted successfully!
A new student was inserted successfully!
A new student was inserted successfully!
A new student was inserted successfully!
Select Operation to Perform :
1.Insert
2.Update
3.Delete
4.Select
5.Drop Table
6.Exit
4
Student #1: 1 - Kaustubh - MCA II
Student #2: 2 - Angad - MCA II
Student #3: 3 - Sagar - MCA III
Student #4: 4 - Rajesh - MCA II
Select Operation to Perform :
1.Insert
2.Update
3.Delete
4.Select
5.Drop Table
6.Exit
1
Enter Student id ,Student Name & Student Class
5
Manoj
MCA III
A new student was inserted successfully!
Select Operation to Perform :
1.Insert
2.Update
3.Delete
4.Select
5.Drop Table
6.Exit
4
Student #1: 1 - Kaustubh - MCA II
Student #2: 2 - Angad - MCA II
Student #3: 3 - Sagar - MCA III
Student #4: 4 - Rajesh - MCA II
```

```
Student #5: 5 - Manoj - MCA III
Select Operation to Perform :
1.Insert
2.Update
3.Delete
4.Select
5.Drop Table
6.Exit
2
Enter ID to Update & new Class
5
MCAII
An existing student was updated successfully!
Select Operation to Perform :
1.Insert
2.Update
3.Delete
4.Select
5.Drop Table
6.Exit
4
Student #1: 1 - Kaustubh - MCA II
Student #2: 2 - Angad - MCA II
Student #3: 3 - Sagar - MCA III
Student #4: 4 - Rajesh - MCA I
Student #5: 5 - Manoj - MCA III
Select Operation to Perform :
1.Insert
2.Update
3.Delete
4.Select
5.Drop Table
6.Exit
3
Enter ID to Delete
5
A Student was deleted successfully!
Select Operation to Perform :
1.Insert
2.Update
3.Delete
4.Select
5.Drop Table
6.Exit
4
Student #1: 1 - Kaustubh - MCA II
Student #2: 2 - Angad - MCA II
Student #3: 3 - Sagar - MCA III
Student #4: 4 - Rajesh - MCA I
Select Operation to Perform :
1.Insert
2.Update
```

```
3.Delete
4.Select
5.Drop Table
6.Exit
5
Student Table DROPPED . Program needs to restart to fire initial
queries
```






Activities
Terminal
Oct 8 10:32
kaustubh@kaustubh-Lenovo-G50-80: /media/kaustubh/A/Practicals/practicals/MCA/SEM III/JAVA/Practicals/assignment no7
kaustubh@kaustubh-Lenovo-G50-80: /media/kaustubh/A/Practicals/practicals/MC...
root@kaustubh-Lenovo-G50-80: ~
Student #4: 4 - Rajesh - MCA I
Student #5: 5 - Manoj - MCA III
Select Operation to Perform :
1.Insert
2.Update
3.Delete
4.Select
5.Drop Table
6.Exit
3
Enter ID to Delete
5
A Student was deleted successfully!
Select Operation to Perform :
1.Insert
2.Update
3.Delete
4.Select
5.Drop Table
6.Exit
4
Student #1: 1 - Kaustubh - MCA II
Student #2: 2 - Angad - MCA II
Student #3: 3 - Sagar - MCA III
Student #4: 4 - Rajesh - MCA I
Select Operation to Perform :
1.Insert
2.Update
3.Delete
4.Select
5.Drop Table
6.Exit
5
kaustubh@kaustubh-Lenovo-G50-80:/media/kaustubh/A/Practicals/practicals/MCA/SEM III/JAVA/Practicals/assignment no7$