

Q.1. Write a program to show [JDBC connection](#) with MYSQL and perform the following operations:

Create table Customer with following fields:

Custno
Custame
Custaddress
Phoneno
City
Pincode
Country

- a. Insert values
- b. Delete values
- c. update city name Shimla to Shilong.
- d. Show table in the console

Program:

```
package in.jdbccon;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
```

```
import java.util.Scanner;
```

```
/*Q.1. Write a program to show JDBC connection with MYSQL and perform the following operations:
```

Create table Customer with following fields:

Custno
Custame
Custaddress
Phoneno
City
Pincode
Country*/

```
public class JdbcLabQuiz {
```

```
    static int Custno;
```

```
static String Custname;

static String Custaddress;

static String Phoneno;

static String City;

static int Pincode;

static String Country;


public static void main(String[] args) throws Exception{

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

    Scanner sc=new Scanner(System.in);

    try {

        //Registered the Driver class

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

        //Created obj for Connection

        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/my_shop","root","1234");

        System.out.println("Choose what operation want to perform : \n a. Insert values \n b. Delete values \n
c. update city name Shimla to Shilong.e \n d. Show table in the console");

        String choice = sc.next();

        // use Switch case for performing any task

        switch (choice) {

            case "insert":

                insertFunction(sc, con); // Insert method call

                break;

            case "update":

                updateFunction(sc, con); // Update method call

                break;
```

```
case "delete":

deleteFunction(sc, con); // delete method call

break;

case "display":

displayFunction(con); // Display method call

break;


default:

break;

}

}

catch (Exception e) {

e.printStackTrace();

//

}

}


// static insert method for insert the data in database table

static void insertFunction(Scanner sc, Connection con) {

// taking inputs

System.out.print("Enter Customer no.: ");

Custno=sc.nextInt();

System.out.print("Enter Customer Name: ");

Custname=sc.next();

System.out.print("Enter Customer's Address: ");

Custaddress=sc.next();

System.out.print("Enter Customer's Phone Number: ");

Phoneno=sc.next();

System.out.print("Enter Customer's City: ");

City=sc.next();
```

```

System.out.print("Enter Customer's Pincode: ");

Pincode=sc.nextInt();

System.out.print("Enter Customer's country name: ");

Country=sc.next();


// making string query

String str = "insert into Customer values(" + Custno + "," + Custname + "," + Custaddress
+ "," + Phoneno + "," + City + "," + Pincode + "," + Country + ")";


PreparedStatement ps;

try {

// pass the string into prepare statement for query preparation

ps = con.prepareStatement(str);

//execute the query

ps.executeUpdate();

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

}

catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}


// static update method for update the specific data in database table

static void updateFunction(Scanner sc, Connection con) {

//we update the data through old city to new city

System.out.print("enter the New City name : ");

String NewCity = sc.next();

System.out.print("enter the Old City name : ");

City = sc.next();


// making string query

String str = "UPDATE customer SET city=" + NewCity + " WHERE city=" + City + """;

```

```

PreparedStatement ps;

try {

ps = con.prepareStatement(str);

//execute the query

ps.executeUpdate();

System.out.println("Update Successful");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}


}


// static delete method for delete the specific data in database table

static void deleteFunction(Scanner sc, Connection con) {

System.out.print("enter the Customer id : ");

Custno = sc.nextInt();

        // we delete the data through Customer Number

String str = "delete from Customer where Custno = " + Custno + ";";

```

```

PreparedStatement ps;

try {

ps = con.prepareStatement(str);

//execute the query

ps.executeUpdate();

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

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}


}

```

```

// static Display method for showing data

static void displayFunction(Connection c) {

    // making string query

    String str = "Select * from customer";

    PreparedStatement ps;

    try {

        ps = c.prepareStatement(str);

        // Result set save the all result

        ResultSet rs = ps.executeQuery();

        // loop for display one by one

        while (rs.next()) {

            System.out.println();

            Custno = rs.getInt("Custno ");

            Custname = rs.getString("Custname ");

            Custaddress = rs.getString("Custaddress");

            Phoneno = rs.getString("Phoneno");

            City = rs.getString("City");

            Pincode = rs.getInt("Pincode");

            Country = rs.getString("Country");

            //print all the data

            System.out.println(Custno + "\t" + Custname + "\t" + Custaddress + "\t" + Phoneno + "\t"
+ City + "\t" + Pincode + "\t" + Country);

        }

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

    } catch (SQLException e) {

        e.printStackTrace();

    }

}

```

Output:

The screenshot shows a database management tool interface. On the left, a 'SCHEMAS' pane lists various databases including 'anudip', 'college', 'hellojdbc', 'jpaproject', 'my_shop', 'myshop', 'sys', and 'world'. The 'my_shop' database is selected, and its 'customer' table is highlighted. The main window displays a query result grid for the query 'SELECT * FROM my_shop.customer;'. The grid shows 5 rows of customer data. Below the grid, an 'Action Output' pane shows the execution log, indicating that the query was executed successfully and returned 5 rows.

Custno	Custname	Custaddress	Phoneno	City	Pincode	Country
1	Rishu	KamlaNagar	987654321	Agra	282004	India
3	Neha	StreetRoad	789654125	Pune	654789	India
4	Rahul	dobi	999874562	Hisar	784512	India
4	hina	Rohini	654897456	NewDelhi	2365478	India
5	Akash	JakhooRoad	698745211	Shimla	110058	India

customer 2 x Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	10:20:54	SELECT * FROM my_shop.customer LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
2	10:28:09	SELECT * FROM my_shop.customer LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

```
<terminated> JdbcLabQuiz [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (28-Sep-2023, 10:27:48 am – 10:28:01 am) [pid: 11392]
Choose what operation want to perform :
a. Insert values
b. Delete values
c. update city name Shimla to Shilong.e
d. Show table in the console
delete
enter the Customer id : 2
Delete Successful
```

Limit to 1000 rows

1 • `SELECT * FROM my_shop.customer;`

Result Grid

	Custno	Custname	Custaddress	Phoneno	City	Pincode	Country
▶	1	Rishu	KamlaNagar	987654321	Agra	282004	India
	3	Neha	StreetRoad	789654125	Pune	654789	India
	4	Rahul	dobi	999874562	Hisar	784512	India
	4	hina	Rohini	654897456	NewDelhi	2365478	India
	5	Akash	JakhooRoad	698745211	Shimla	110058	India

customer 3 × Read On

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	10:20:54	SELECT * FROM my_shop.customer LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
✓ 2	10:28:09	SELECT * FROM my_shop.customer LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
✓ 3	10:31:52	SELECT * FROM my_shop.customer LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Console ×

<terminated> JdbcLabQuiz [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (28-Sep-2023, 10:32:37 am – 10:33:14 am) [pid: 15096]

Choose what operation want to perform :

- a. Insert values
- b. Delete values
- c. update city name Shimla to Shilong.e
- d. Show table in the console

update

enter the New City name : `Shilong`

enter the Old City name : `Shimla`

Update Successful

Result Grid

	Custno	Custname	Custaddress	Phoneno	City	Pincode	Country
1	Rishu	KamlaNagar	987654321	Agra	282004	India	
3	Neha	StreetRoad	789654125	Pune	654789	India	
4	Rahul	dobi	999874562	Hisar	784512	India	
4	hina	Rohini	654897456	NewDelhi	2365478	India	
5	Akash	JakhooRoad	698745211	Shilong	110058	India	

customer 4 x Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	10:20:54	SELECT * FROM my_shop.customer LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
2	10:28:09	SELECT * FROM my_shop.customer LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
3	10:31:52	SELECT * FROM my_shop.customer LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
4	10:33:57	SELECT * FROM my_shop.customer LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

```
static String phoneno="((\\(\\d{3}\\))|\\d{3})[- ]?\\d{3}[- ]?\\d{4}";

<terminated> JdbcLabQuiz [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (28-Sep-2023, 10:36:49 am - 10:36:56 am) [pid: 5108]
Choose what operation want to perform :
a. Insert values
b. Delete values
c. update city name Shimla to Shilong.e
d. Show table in the console
display
1      Rishu   KamlaNagar      987654321      Agra   282004  India
3      Neha   StreetRoad      789654125      Pune   654789  India
4      Rahul  dobi           999874562      Hisar   784512  India
4      hina   Rohini         654897456      NewDelhi 2365478 India
5      Akash  JakhooRoad     698745211      Shilong 110058  India
Display Successful
```

Q. 2) Create below 3 tables with specified column names, datatypes and rules.

1)List the No.of students based on course wise.

List the student details which student origin Is foreign and no.of values exceeds 10?

2)List the Student,Course,Admissions details which student taken some course ?

3)List the all Student name which students grade is 'A' and "B"?

4)List the Course details which course does not have any students?

5)List the Fees details based on Student id which is more than 4000?

Program :

```
create database LabQuiz;
```

```
use LabQuiz;
```

```
create table courses(cid int primary key,cname varchar(40),shift varchar(40),fees int);
```

```
create table students(sid int primary key, sname varchar(40),origin varchar(40),type varchar(40));
```

```
create table admission(sid int, cid int,DOJ datetime,grade char);
```

```
insert into students
```

```
values (101,'Radha','local','normal');
```

```
insert into students
```

```
values (102,'Jack','Foreign','fast');
```

```
insert into students
```

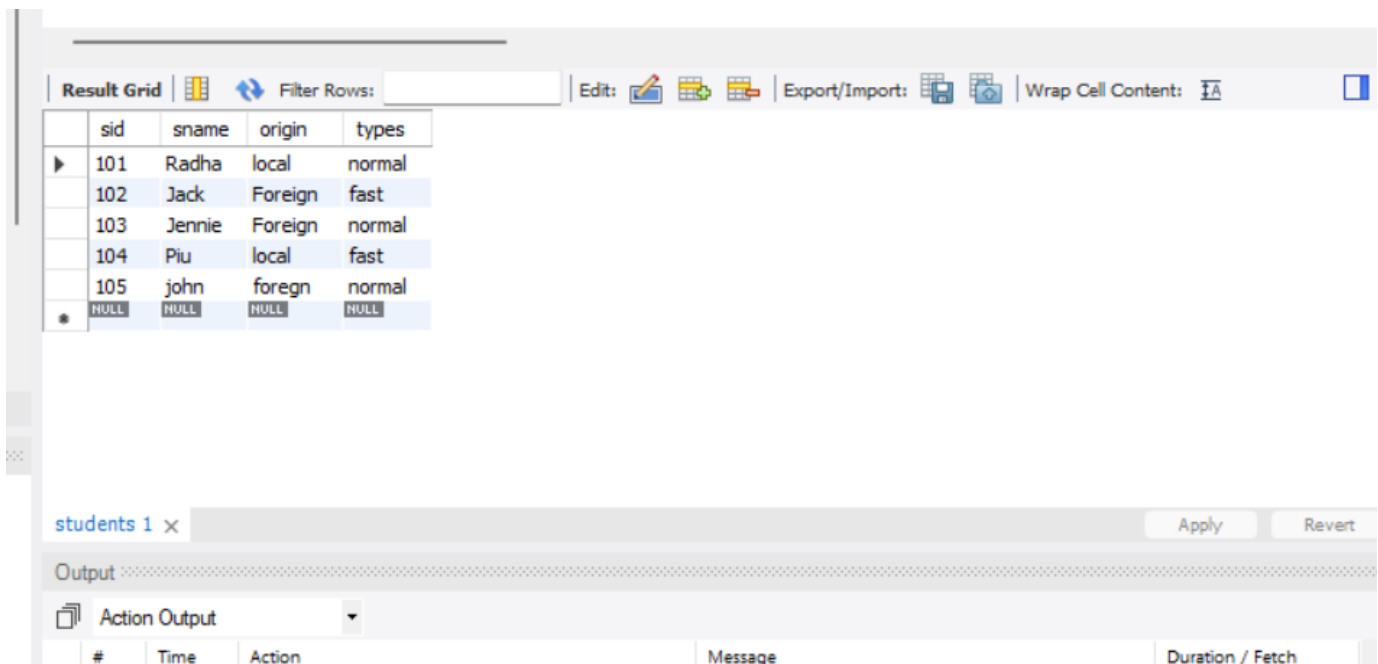
```
values (103,'Jennie','Foreign','normal');
```

```
insert into students
```

```
values (104,'Piu','local','fast');
```

```
insert into students
```

```
values (105,'john','foregn','normal');
```



The screenshot shows a database management interface. At the top, there's a toolbar with options like 'Result Grid', 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. Below the toolbar is a table with the following data:

	sid	sname	origin	types
▶	101	Radha	local	normal
	102	Jack	Foreign	fast
	103	Jennie	Foreign	normal
	104	Piu	local	fast
	105	john	foregn	normal
•	NULL	NULL	NULL	NULL

Below the table, there's a section labeled 'students 1' with 'Apply' and 'Revert' buttons. Underneath is an 'Output' section with a dropdown menu set to 'Action Output'. At the bottom, there's a table with columns: '#', 'Time', 'Action', 'Message', and 'Duration / Fetch'.

```
insert into Courses
```

```
values (111,'Java','morning',300);
```

```
insert into Courses
```

```
values (222,'Python','evening',500);
```

```
insert into Courses
```

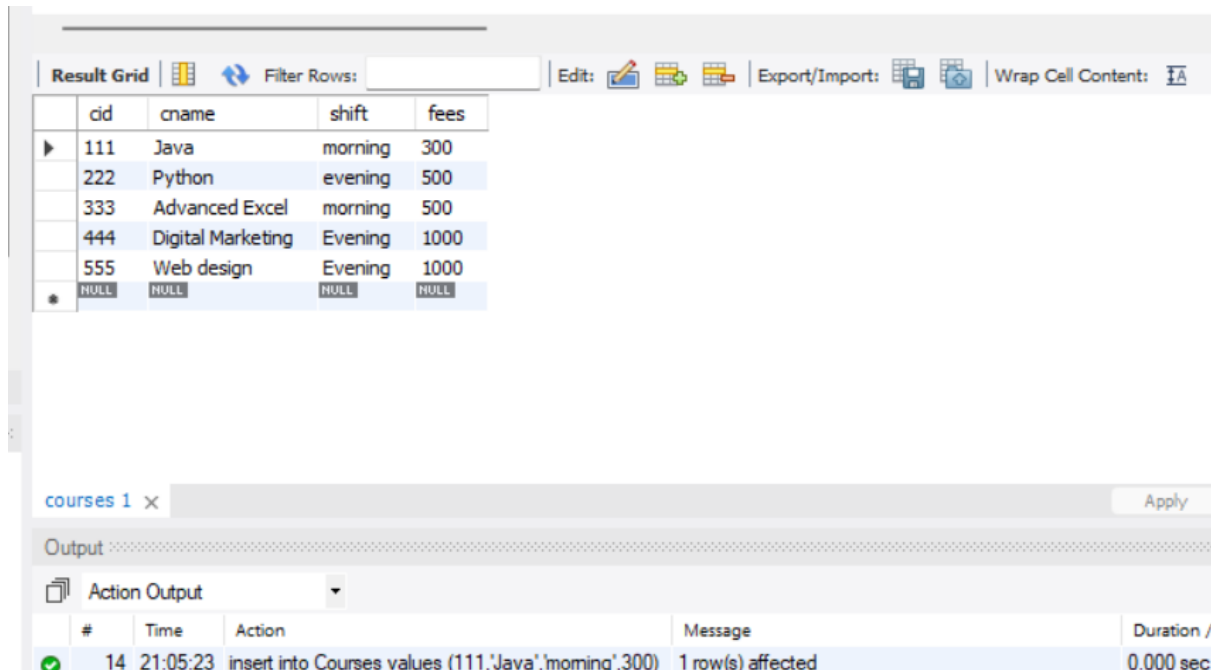
```
values (333,'Advanced Excel','morning',500);
```

insert into Courses

values (444,'Digital Marketing','Evening',1000);

insert into Courses

values (555,'Web design','Evening',1000);



The screenshot shows a database management interface. At the top, there's a 'Result Grid' with a toolbar containing icons for 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. The grid displays a table with columns: cid, cname, shift, and fees. Below the grid, there's a tab labeled 'courses 1' with an 'Apply' button. Below the tab is an 'Output' section with a dropdown menu set to 'Action Output'. The output table has columns: #, Time, Action, Message, and Duration / .

	cid	cname	shift	fees
▶	111	Java	morning	300
	222	Python	evening	500
	333	Advanced Excel	morning	500
	444	Digital Marketing	Evening	1000
	555	Web design	Evening	1000
*	NULL	NULL	NULL	NULL

#	Time	Action	Message	Duration /
✓ 14	21:05:23	insert into Courses values (111,'Java','morning',300)	1 row(s) affected	0.000 sec

insert into admission

values (101,111,'2020-01-01','A');

insert into admission

values (102,222,'2020-05-20','B');

insert into admission

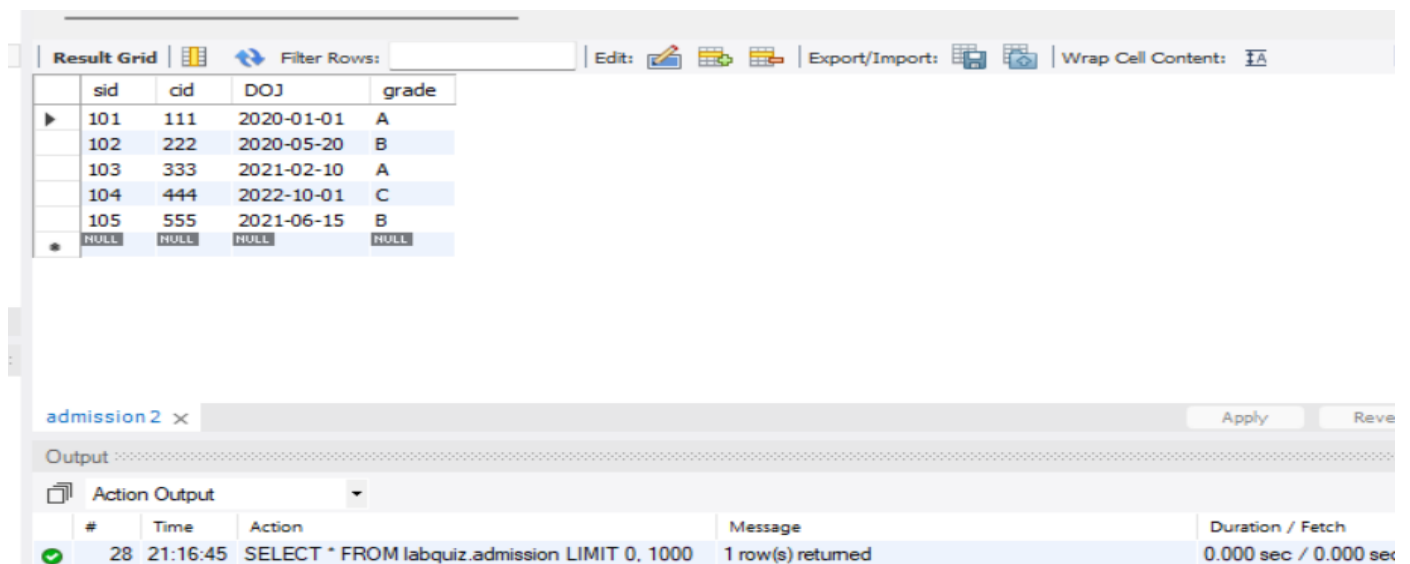
values (103,333,'2021-02-10','A');

insert into admission

values (104,444,'2022-10-01','C');

insert into admission

values (105,555,'2021-06-15','B');



The screenshot shows a database management interface. At the top, there's a 'Result Grid' with a toolbar containing icons for 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. The grid displays a table with columns: sid, cid, DOJ, and grade. Below the grid, there's a tab labeled 'admission 2' with 'Apply' and 'Reve' buttons. Below the tab is an 'Output' section with a dropdown menu set to 'Action Output'. The output table has columns: #, Time, Action, Message, and Duration / Fetch.

	sid	cid	DOJ	grade
▶	101	111	2020-01-01	A
	102	222	2020-05-20	B
	103	333	2021-02-10	A
	104	444	2022-10-01	C
	105	555	2021-06-15	B
*	NULL	NULL	NULL	NULL

#	Time	Action	Message	Duration / Fetch
✓ 28	21:16:45	SELECT * FROM labquiz.admission LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

1)List the No.of students based on course wise.

select students.sname,courses.cname from students cross join courses;

The screenshot shows a 'Result Grid' window with a toolbar at the top containing icons for 'Filter Rows', 'Export', and 'Wrap Cell Content'. The grid displays the results of a cross join between the 'students' and 'courses' tables. The columns are 'sname' and 'cname'. The data is as follows:

sname	cname
john	Java
Piu	Java
Jennie	Java
Jack	Java
Radha	Java
john	Python
Piu	Python
Jennie	Python
Jack	Python
Radha	Python
john	Advanced Excel
Piu	Advanced Excel
Jennie	Advanced Excel
Jack	Advanced Excel
Radha	Advanced Excel
john	Digital Marketing
Piu	Digital Marketing
Jennie	Digital Marketing
Jack	Digital Marketing
Radha	Digital Marketing
john	Web design

At the bottom of the window, there is a tab labeled 'Result 3' and a 'Read Only' status indicator.

The screenshot shows a 'Result Grid' window displaying a filtered result set. The columns are 'sname' and 'cname'. The data is as follows:

sname	cname
john	Web design
Piu	Web design
Jennie	Web design
Jack	Web design
Radha	Web design

At the bottom of the window, there is a tab labeled 'Result 3' and a 'Read Only' status indicator.

List the student details which student origin is foreign and no.of values exceeds 10?

select * from students where origin="foreign";

The screenshot shows a 'Result Grid' window with a toolbar at the top containing icons for 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. The grid displays the results of a query filtering for students with a foreign origin. The columns are 'sid', 'sname', 'origin', and 'types'. The data is as follows:

sid	sname	origin	types
102	Jack	foreign	fast
103	Jennie	foreign	normal
NULL	NULL	NULL	NULL

At the bottom of the window, there is a tab labeled 'students 4' and buttons for 'Apply' and 'Revert'.

select * from admission where grade="A" or "B";

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	sid	cid	DOJ	grade
▶	101	111	2020-01-01	A
	103	333	2021-02-10	A
*	NULL	NULL	NULL	NULL

admission 5

×

Apply

Revert

Output