Q. 1. Perform the following tasks:

1. Create Student table with following attributes (STUDENT\_ID , FIRST\_NAME, LAST\_NAME, PHONE\_NUMBER, MARKS, COURSE\_ID).

CODE:

CREATE TABLE Student (

STUDENT\_ID INT PRIMARY KEY,

FIRST\_NAME VARCHAR(50),

LAST\_NAME VARCHAR(50),

PHONE\_NUMBER VARCHAR(15),

MARKS INT,

COURSE\_ID INT

FOREIGN KEY (COURSE\_ID) REFERECES Course(Course\_ID)

);

OUTPUT:

A black screen with white text

Description automatically generated

1. Create Course table with following attributes (COURSE\_ID, COURSE\_NAME).

CODE:

CREATE TABLE Course (

COURSE\_ID INT PRIMARY KEY,

COURSE\_NAME VARCHAR(100)

);

OUTPUT:

A black screen with white text

Description automatically generated

1. Write a SQL statement to insert 5 records with your own value into the tables.

CODE:

INSERT INTO Course (COURSE\_ID, COURSE\_NAME) VALUES

(1, 'Mathematics),

(2, 'Physics),

(3, 'Chemistry),

(4, 'Biology'),

(5, 'Computer Science');

OUTPUT:

A black screen with white text

Description automatically generated

d. Write a query to get the number of students with the same course.

CODE:

SELECT c.COURSE\_NAME, COUNT(s.STUDENT\_ID) AS NumberOfStudents

FROM Student s

JOIN Course c ON s.COURSE\_ID = c.COURSE\_ID

GROUP BY c.COURSE\_NAME;FROM Student

GROUP BY COURSE\_ID;

OUTPUT:

A black screen with white text

Description automatically generated

f. Write a query to get the student name, course name and marks of the students.

CODE:

SELECT s.FIRST\_NAME, s.LAST\_NAME, c.COURSE\_NAME, s.MARKS

FROM Student s

JOIN Course c ON s.COURSE\_ID = c.COURSE\_ID;

OUTPUT:

A black screen with white text

Description automatically generated

g. Write a query to get the Average marks of students course wise.

CODE:

SELECT c.COURSE\_NAME, AVG(s.MARKS) AS AverageMarks

FROM Student s

JOIN Course c ON s.COURSE\_ID = c.COURSE\_ID

GROUP BY c.COURSE\_NAME;

OUTPUT:

A black screen with white text

Description automatically generated

Q. 2.  Create database for hospital management system & Perform the following tasks:

a. Create HEALTH CARE WORKERS table with following attributes (EMPLOYEE\_ID , FIRST\_NAME, LAST\_NAME,EMAIL, PHONE\_NUMBER, HIRE\_DATE, SALARY, DESIGNATION).

CODE:

CREATE TABLE HEALTH\_CARE\_WORKERS (

EMPLOYEE\_ID INT PRIMARY KEY,

FIRST\_NAME VARCHAR(50),

LAST\_NAME VARCHAR(50),

EMAIL VARCHAR(100),

PHONE\_NUMBER VARCHAR(15),

HIRE\_DATE DATE,

SALARY DECIMAL(10, 2),

DESIGNATION VARCHAR(50)

);

OUTPUT:

A screenshot of a computer program

Description automatically generated

b. Create PATIENT table with following attributes (PATIENT\_ID,NAME, PHONE\_NUMBER).

CODE:

CREATE TABLE PATIENT (

PATIENT\_ID INT PRIMARY KEY,

NAME VARCHAR(100),

PHONE\_NUMBER VARCHAR(15)

);

OUTPUT:

A black screen with white text

Description automatically generated

c. Write a SQL statement to insert 10 records with your own value into the tables.

CODE:

INSERT INTO HEALTH\_CARE\_WORKERS (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, SALARY, DESIGNATION) VALUES

(1, 'Alice', 'Smith', 'alice.smith@example.com', '5551234567', '2020-01-15', 75000.00, 'Doctor'),

(2, 'Bob', 'Johnson', 'bob.johnson@example.com', '5552345678', '2019-03-22', 68000.00, 'Nurse'),

(3, 'Carol', 'Williams', 'carol.williams@example.com', '5553456789', '2021-06-10', 72000.00, 'Pharmacist'),

(4, 'David', 'Brown', 'david.brown@example.com', '5554567890', '2018-11-05', 80000.00, 'Surgeon'),

(5, 'Emma', 'Jones', 'emma.jones@example.com', '5555678901', '2022-07-25', 69000.00, 'Radiologist'),

(6, 'Frank', 'Garcia', 'frank.garcia@example.com', '5556789012', '2020-02-20', 73000.00, 'General Practitioner'),

(7, 'Grace', 'Martinez', 'grace.martinez@example.com', '5557890123', '2019-08-15', 71000.00, 'Pediatrician'),

(8, 'Henry', 'Rodriguez', 'henry.rodriguez@example.com', '5558901234', '2018-12-30', 77000.00, 'Orthopedic Surgeon'),

(9, 'Ivy', 'Wilson', 'ivy.wilson@example.com', '5559012345', '2021-04-10', 74000.00, 'Anesthesiologist'),

(10, 'Jack', 'Lee', 'jack.lee@example.com', '5550123456', '2019-09-25', 70000.00, 'Dermatologist');

OUTPUT:

A black screen with white text

Description automatically generated

d. Write a query to get the names (first\_name, last\_name),Designation, salary.

CODE:

SELECT FIRST\_NAME, LAST\_NAME, DESIGNATION, SALARY

FROM HEALTH\_CARE\_WORKERS;

OUTPUT:

A screenshot of a computer

Description automatically generated

e. Write a query to get the number of employees with the same Designation

CODE:

SELECT DESIGNATION, COUNT(\*) AS NumberOfEmployees

FROM HEALTH\_CARE\_WORKERS

GROUP BY DESIGNATION;

OUTPUT:

A black screen with white text

Description automatically generated

f. Write a query to get employee name who are getting salary more than 25000.

CODE:

SELECT FIRST\_NAME, LAST\_NAME

FROM HEALTH\_CARE\_WORKERS

WHERE SALARY > 25000;

OUTPUT:



g. Fetch HEALTH CARE WORKERS name using their employee id.

CODE:

SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME

FROM HEALTH\_CARE\_WORKERS

WHERE EMPLOYEE\_ID = 1;

OUTPUT:

A black screen with white text

Description automatically generated

Q. 3 Consider two tables, customers and orders, with the following structures:

Customers Table: customer\_id (Primary Key) first\_name Last\_name

Orders Table: order\_id (Primary Key) customer\_id (Foreign Key) order\_date Total\_amount

Write an SQL query to retrieve the first and last names of customers along with the order date and total amount of their orders.

A computer screen with white text

Description automatically generated

A computer screen with white text

Description automatically generated

Use an INNER JOIN to connect the two tables.

A screenshot of a computer

Description automatically generated

4.Consider two tables, departments and employees, with the following structures:

Departments Table: department\_id (Primary Key) department\_name

Employees Table: employee\_id (Primary Key) first\_name last\_name department\_id (Foreign Key)

A screen shot of a computer

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

Write an SQL query to retrieve a list of all departments and the names of employees who belong to each department. Use a LEFT JOIN to include departments that have no employees.

A screenshot of a computer program

Description automatically generated

4. 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

Insert 5 records in Customer table.

a.     Insert values

b.    Delete values

c.     update city name Shimla to Shilong.

d.    Show table in the console

CODE:

package DataStructure;

import java.sql.\*;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

public class Create {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/school ";

String user = "root";

String password = "123456789";

//Table creation is here..

try {

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

Connection con = DriverManager.getConnection(url, user, password);

Statement stmt = con.createStatement();

//Table values insertion is here...

String insert = "INSERT INTO customer\_detail VALUES ('CUST001', 'John Doe', '123 Main St', '+1234567890', 'New York', '10001', 'USA'),('CUST002', 'Jane Smith', '456 Elm St', '+1987654321', 'Los Angeles', '90001', 'USA'),('CUST003', 'Alice Johnson', '789 Oak Ave', '+4455667788', 'London', 'SW1A 1AA', 'UK'),('CUST004', 'Juan Martinez', '234 Calle Principal', '+1122334455', 'Madrid', '28001', 'Spain'),('CUST005', 'Hiroshi Tanaka', '567 Ginza Street', '+81987654321', 'Tokyo', '100-0001', 'Japan')";

stmt.addBatch(insert);

stmt.executeBatch();

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

//Table Record Deletion is here...

/\*

String delete = "DELETE customer\_detail WHERE country = 'UK'";

stmt.addBatch(delete);

stmt.executeBatch();

System.out.println("Record Deleted Successfully.");

/\*String update ="UPDATE customer\_detail SET city = 'Shilong' WHERE city = 'Shimla'";

stmt.addBatch(update);

stmt.executeBatch();

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

\*/

String sql = "SELECT cust\_no, cust\_name, cust\_address, phoneno, city, pincode, country FROM customer\_detail";

ResultSet rs = stmt.executeQuery(sql);

// 4. Printing the table header

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

System.out.println("| Cust No | Cust Name | Cust Address | Phone No | City | Pincode | Country |");

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

// 5. Iterating through the result set and printing each row

while (rs.next()) {

String custNo = rs.getString("cust\_no");

String custName = rs.getString("cust\_name");

String custAddress = rs.getString("cust\_address");

String phoneNo = rs.getString("phoneno");

String city = rs.getString("city");

String pincode = rs.getString("pincode");

String country = rs.getString("country");

// Print each row of data

System.out.printf("| %-9s | %-15s | %-20s | %-11s | %-13s | %-10s | %-12s |\n",

custNo, custName, custAddress, phoneNo, city, pincode, country);

}

// 6. Printing the table footer

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

rs.close();

stmt.close();

con.close();

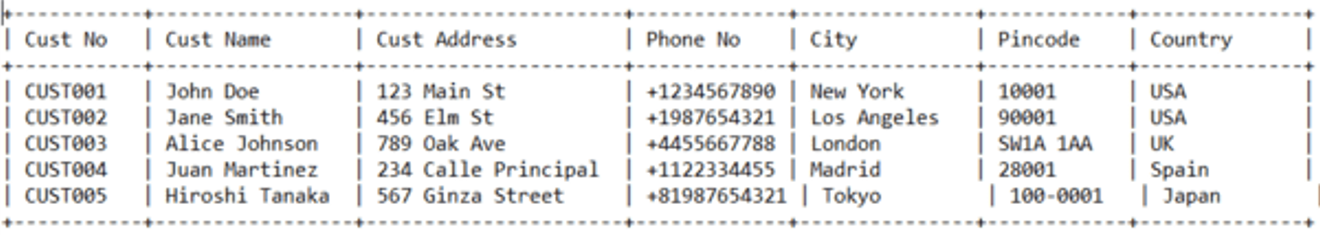
}catch(Exception e) {

System.err.println(e);

}

}

}



1. Insert values

CODE:

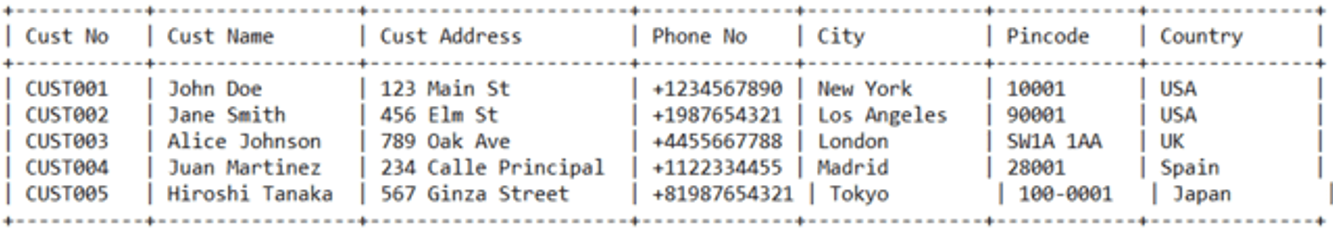
String insert = "INSERT INTO customer\_detail VALUES ('CUST001', 'John Doe', '123 Main St', '+1234567890', 'New York', '10001', 'USA'),('CUST002', 'Jane Smith', '456 Elm St', '+1987654321', 'Los Angeles', '90001', 'USA'),('CUST003', 'Alice Johnson', '789 Oak Ave', '+4455667788', 'London', 'SW1A 1AA', 'UK'),('CUST004', 'Juan Martinez', '234 Calle Principal', '+1122334455', 'Madrid', '28001', 'Spain'),('CUST005', 'Hiroshi Tanaka', '567 Ginza Street', '+81987654321', 'Tokyo', '100-0001', 'Japan')";

stmt.addBatch(insert);

stmt.executeBatch();

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

OUTPUT:



1. Delete values

CODE:

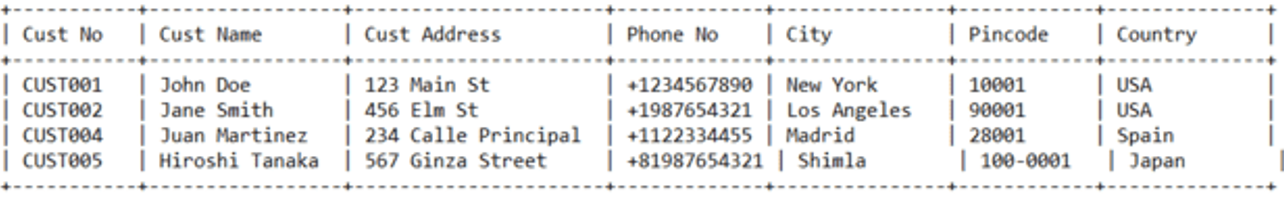
String delete = "DELETE customer\_detail WHERE country = 'UK'";

stmt.addBatch(delete);

stmt.executeBatch();

System.out.println("Record Deleted Successfully.");

OUTPUT:



1. update city name Shimla to Shilong.

CODE:

String update ="UPDATE customer\_detail SET city = 'Shilong' WHERE city = 'Shimla'";

stmt.addBatch(update);

stmt.executeBatch();

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

OUTPUT:

A close-up of a number

Description automatically generated

1. Show table in the console

OUTPUT:

A close-up of a number

Description automatically generated