ABHISHEK GAJULA

RA2111003010756

1. (Exercise on retrieving records from the table) EMPLOYEES (Employee_Id, First_Name, Last_Name, Email, Phone_Number, Hire_Date, Job_Id, Salary, Commission_Pct, Manager_Id, Department_Id)

(a) Find out the employee id, names, salaries of all the employees

ANSWER:

INSERT INTO EMPLOYEES VALUES(1, 'John', 'Doe', 'john.doe@example.com', '123456789', '2022-01-01', 'JR001', 5000.00, 0.05, 100, 10);

INSERT INTO EMPLOYEES VALUES(2, 'Jane', 'Smith', 'jane.smith@example.com', '987654321', '2022-02-15', 'JR002', 5500.00, 0.03, 100, 20);

INSERT INTO EMPLOYEES VALUES(3, 'Michael', 'Johnson', 'michael.johnson@example.com', '456123789', '2021-11-10', 'JR003', 4800.00, 0.02, 101, 30);

INSERT INTO EMPLOYEES VALUES(4, 'Emily', 'Brown', 'emily.brown@example.com', '789456123', '2022-03-20', 'JR001', 5200.00, 0.04, 101, 10);

INSERT INTO EMPLOYEES VALUES(5, 'Sarah', 'Austin', 'sarah.austin@example.com', '9876543210', '2022-04-10', 'JR002', 6000.00, 0.03, 100, 20);

SELECT * FROM EMPLOYEES;

SELECT Employee_Id, First_Name, Last_Name, Salary FROM EMPLOYEES;

queries.sql 425xmxu2v NEW MYSQL RUN

```sql
CREATE TABLE EMPLOYEES (
    Employee_Id INT PRIMARY KEY,
    First_Name VARCHAR(50),
    Last_Name VARCHAR(50),
    Email VARCHAR(100),
    Phone_Number VARCHAR(20),
    Hire_Date DATE,
    Job_Id VARCHAR(10),
    Salary DECIMAL(10, 2),
    Commission_Pct DECIMAL(5, 2),
    Manager_Id INT,
    Department_Id INT
);
INSERT INTO EMPLOYEES VALUES(1, 'John', 'Doe', 'john.doe@example.com', '123456789', '2022-01-01', 'JR001', 5
INSERT INTO EMPLOYEES VALUES(2, 'Jane', 'Smith', 'jane.smith@example.com', '987654321', '2022-02-15', 'JR002
INSERT INTO EMPLOYEES VALUES(3, 'Michael', 'Johnson', 'michael.johnson@example.com', '456123789', '2021-11-1
INSERT INTO EMPLOYEES VALUES(4, 'Emily', 'Brown', 'emily.brown@example.com', '789456123', '2022-03-20', 'JR0
INSERT INTO EMPLOYEES VALUES(5, 'Sarah', 'Austin', 'sarah.austin@example.com', '9876543210', '2022-04-10', '
SELECT * FROM EMPLOYEES;
SELECT Employee_Id, First_Name, Last_Name, Salary FROM EMPLOYEES;
SELECT Employee_Id, First_Name, Last_Name FROM EMPLOYEES WHERE Manager_Id = 100;
SELECT First_Name, Last_Name FROM EMPLOYEES WHERE Salary >= 4800;
SELECT Employee_Id, First_Name, Last_Name FROM EMPLOYEES WHERE Last_Name = 'AUSTIN';
SELECT First_Name, Last_Name FROM EMPLOYEES WHERE Department_Id IN (60, 70, 80);
SELECT DISTINCT Manager_Id FROM EMPLOYEES;
```

STDIN

Input for the program (Optional)

```
+-------------+------------+-----------+---------+
| Employee_Id | First_Name | Last_Name | Salary  |
+-------------+------------+-----------+---------+
|           1 | John       | Doe       | 5000.00 |
|           2 | Jane       | Smith     | 5500.00 |
|           3 | Michael    | Johnson   | 4800.00 |
|           4 | Emily      | Brown     | 5200.00 |
|           5 | Sarah      | Austin    | 6000.00 |
+-------------+------------+-----------+---------+
+-------------+------------+-----------+
| Employee_Id | First_Name | Last_Name |
+-------------+------------+-----------+
|           1 | John       | Doe       |
|           2 | Jane       | Smith     |
|           5 | Sarah      | Austin    |
+-------------+------------+-----------+
+------------+-----------+
| First_Name | Last_Name |
+------------+-----------+
| John       | Doe       |
| Jane       | Smith     |
| Michael    | Johnson   |
| Emily      | Brown     |
| Sarah      | Austin    |
+------------+-----------+
+-------------+------------+-----------+
| Employee_Id | First_Name | Last_Name |
+-------------+------------+-----------+
```

(b) List out the employees who works under manager 100

ANSWER:

SELECT Employee_Id, First_Name, Last_Name FROM EMPLOYEES WHERE Manager_Id = 100;

queries.sql 425xmxu2v NEW MYSQL RUN

```sql
CREATE TABLE EMPLOYEES (
    Employee_Id INT PRIMARY KEY,
    First_Name VARCHAR(50),
    Last_Name VARCHAR(50),
    Email VARCHAR(100),
    Phone_Number VARCHAR(20),
    Hire_Date DATE,
    Job_Id VARCHAR(10),
    Salary DECIMAL(10, 2),
    Commission_Pct DECIMAL(5, 2),
    Manager_Id INT,
    Department_Id INT
);
INSERT INTO EMPLOYEES VALUES(1, 'John', 'Doe', 'john.doe@example.com', '123456789', '2022-01-01', 'JR001', 5
INSERT INTO EMPLOYEES VALUES(2, 'Jane', 'Smith', 'jane.smith@example.com', '987654321', '2022-02-15', 'JR002
INSERT INTO EMPLOYEES VALUES(3, 'Michael', 'Johnson', 'michael.johnson@example.com', '456123789', '2021-11-1
INSERT INTO EMPLOYEES VALUES(4, 'Emily', 'Brown', 'emily.brown@example.com', '789456123', '2022-03-20', 'JR0
INSERT INTO EMPLOYEES VALUES(5, 'Sarah', 'Austin', 'sarah.austin@example.com', '9876543210', '2022-04-10', '
SELECT * FROM EMPLOYEES;
SELECT Employee_Id, First_Name, Last_Name, Salary FROM EMPLOYEES;
SELECT Employee_Id, First_Name, Last_Name FROM EMPLOYEES WHERE Manager_Id = 100;
SELECT First_Name, Last_Name FROM EMPLOYEES WHERE Salary >= 4800;
SELECT Employee_Id, First_Name, Last_Name FROM EMPLOYEES WHERE Last_Name = 'AUSTIN';
SELECT First_Name, Last_Name FROM EMPLOYEES WHERE Department_Id IN (60, 70, 80);
SELECT DISTINCT Manager_Id FROM EMPLOYEES;
```

STDIN

Input for the program (Optional)

```
+-------------+------------+-----------+---------+
| Employee_Id | First_Name | Last_Name | Salary  |
+-------------+------------+-----------+---------+
|           1 | John       | Doe       | 5000.00 |
|           2 | Jane       | Smith     | 5500.00 |
|           3 | Michael    | Johnson   | 4800.00 |
|           4 | Emily      | Brown     | 5200.00 |
|           5 | Sarah      | Austin    | 6000.00 |
+-------------+------------+-----------+---------+
+-------------+------------+-----------+
| Employee_Id | First_Name | Last_Name |
+-------------+------------+-----------+
|           1 | John       | Doe       |
|           2 | Jane       | Smith     |
|           5 | Sarah      | Austin    |
+-------------+------------+-----------+
+------------+-----------+
| First_Name | Last_Name |
+------------+-----------+
| John       | Doe       |
| Jane       | Smith     |
| Michael    | Johnson   |
| Emily      | Brown     |
| Sarah      | Austin    |
+------------+-----------+
+-------------+------------+-----------+
| Employee_Id | First_Name | Last_Name |
+-------------+------------+-----------+
```

(c) Find the names of the employees who have a salary greater than or equal to 4800

ANSWER:

SELECT First_Name, Last_Name FROM EMPLOYEES WHERE Salary >= 4800;

queries.sql + 425xmxu2v NEW MYSQL RUN

```sql
CREATE TABLE EMPLOYEES (
    Employee_Id INT PRIMARY KEY,
    First_Name VARCHAR(50),
    Last_Name VARCHAR(50),
    Email VARCHAR(100),
    Phone_Number VARCHAR(20),
    Hire_Date DATE,
    Job_Id VARCHAR(10),
    Salary DECIMAL(10, 2),
    Commission_Pct DECIMAL(5, 2),
    Manager_Id INT,
    Department_Id INT
);
INSERT INTO EMPLOYEES VALUES(1, 'John', 'Doe', 'john.doe@example.com', '123456789', '2022-01-01', 'JR001', 5
INSERT INTO EMPLOYEES VALUES(2, 'Jane', 'Smith', 'jane.smith@example.com', '987654321', '2022-02-15', 'JR002
INSERT INTO EMPLOYEES VALUES(3, 'Michael', 'Johnson', 'michael.johnson@example.com', '456123789', '2021-11-1
INSERT INTO EMPLOYEES VALUES(4, 'Emily', 'Brown', 'emily.brown@example.com', '789456123', '2022-03-20', 'JR0
INSERT INTO EMPLOYEES VALUES(5, 'Sarah', 'Austin', 'sarah.austin@example.com', '9876543210', '2022-04-10', '
SELECT * FROM EMPLOYEES;
SELECT Employee_Id, First_Name, Last_Name, Salary FROM EMPLOYEES;
SELECT Employee_Id, First_Name, Last_Name FROM EMPLOYEES WHERE Manager_Id = 100;
SELECT First_Name, Last_Name FROM EMPLOYEES WHERE Salary >= 4800;
SELECT Employee_Id, First_Name, Last_Name FROM EMPLOYEES WHERE Last_Name = 'AUSTIN';
SELECT First_Name, Last_Name FROM EMPLOYEES WHERE Department_Id IN (60, 70, 80);
SELECT DISTINCT Manager_Id FROM EMPLOYEES;
```

STDIN

Input for the program (Optional)

```
+-------------+------------+-----------+---------+
| Employee_Id | First_Name | Last_Name | Salary  |
+-------------+------------+-----------+---------+
|           1 | John       | Doe       | 5000.00 |
|           2 | Jane       | Smith     | 5500.00 |
|           3 | Michael    | Johnson   | 4800.00 |
|           4 | Emily      | Brown     | 5200.00 |
|           5 | Sarah      | Austin    | 6000.00 |
+-------------+------------+-----------+---------+
+-------------+------------+-----------+
| Employee_Id | First_Name | Last_Name |
+-------------+------------+-----------+
|           1 | John       | Doe       |
|           2 | Jane       | Smith     |
|           5 | Sarah      | Austin    |
+-------------+------------+-----------+
+------------+-----------+
| First_Name | Last_Name |
+------------+-----------+
| John       | Doe       |
| Jane       | Smith     |
| Michael    | Johnson   |
| Emily      | Brown     |
| Sarah      | Austin    |
+------------+-----------+
+-------------+------------+-----------+
| Employee_Id | First_Name | Last_Name |
+-------------+------------+-----------+
```

(d) List out the employees whose last name is ‘AUSTIN’

ANSWER:

SELECT Employee_Id, First_Name, Last_Name FROM EMPLOYEES WHERE Last_Name = 'AUSTIN';

queries.sql + 425xmxu2v NEW MYSQL RUN

```sql
CREATE TABLE EMPLOYEES (
    Employee_Id INT PRIMARY KEY,
    First_Name VARCHAR(50),
    Last_Name VARCHAR(50),
    Email VARCHAR(100),
    Phone_Number VARCHAR(20),
    Hire_Date DATE,
    Job_Id VARCHAR(10),
    Salary DECIMAL(10, 2),
    Commission_Pct DECIMAL(5, 2),
    Manager_Id INT,
    Department_Id INT
);
INSERT INTO EMPLOYEES VALUES(1, 'John', 'Doe', 'john.doe@example.com', '123456789', '2022-01-01', 'JR001', 5
INSERT INTO EMPLOYEES VALUES(2, 'Jane', 'Smith', 'jane.smith@example.com', '987654321', '2022-02-15', 'JR002
INSERT INTO EMPLOYEES VALUES(3, 'Michael', 'Johnson', 'michael.johnson@example.com', '456123789', '2021-11-1
INSERT INTO EMPLOYEES VALUES(4, 'Emily', 'Brown', 'emily.brown@example.com', '789456123', '2022-03-20', 'JR0
INSERT INTO EMPLOYEES VALUES(5, 'Sarah', 'Austin', 'sarah.austin@example.com', '9876543210', '2022-04-10', '
SELECT * FROM EMPLOYEES;
SELECT Employee_Id, First_Name, Last_Name, Salary FROM EMPLOYEES;
SELECT Employee_Id, First_Name, Last_Name FROM EMPLOYEES WHERE Manager_Id = 100;
SELECT First_Name, Last_Name FROM EMPLOYEES WHERE Salary >= 4800;
SELECT Employee_Id, First_Name, Last_Name FROM EMPLOYEES WHERE Last_Name = 'AUSTIN';
SELECT First_Name, Last_Name FROM EMPLOYEES WHERE Department_Id IN (60, 70, 80);
SELECT DISTINCT Manager_Id FROM EMPLOYEES;
```

STDIN

Input for the program (Optional)

```
+-------------+------------+-----------+
| Employee_Id | First_Name | Last_Name |
+-------------+------------+-----------+
|           1 | John       | Doe       |
|           2 | Jane       | Smith     |
|           5 | Sarah      | Austin    |
+-------------+------------+-----------+
+------------+-----------+
| First_Name | Last_Name |
+------------+-----------+
| John       | Doe       |
| Jane       | Smith     |
| Michael    | Johnson   |
| Emily      | Brown     |
| Sarah      | Austin    |
+------------+-----------+
+-------------+------------+-----------+
| Employee_Id | First_Name | Last_Name |
+-------------+------------+-----------+
|           5 | Sarah      | Austin    |
+-------------+------------+-----------+
+------------+
| Manager_Id |
+------------+
|        100 |
|        101 |
+------------+
```

(e) Find the names of the employees who works in departments 60,70 and 80

SELECT First_Name, Last_Name FROM EMPLOYEES WHERE Department_Id IN (60, 70, 80);

queries.sql + 425xmxu2v NEW MYSQL RUN

```sql
CREATE TABLE EMPLOYEES (
    Employee_Id INT PRIMARY KEY,
    First_Name VARCHAR(50),
    Last_Name VARCHAR(50),
    Email VARCHAR(100),
    Phone_Number VARCHAR(20),
    Hire_Date DATE,
    Job_Id VARCHAR(10),
    Salary DECIMAL(10, 2),
    Commission_Pct DECIMAL(5, 2),
    Manager_Id INT,
    Department_Id INT
);
INSERT INTO EMPLOYEES VALUES(1, 'John', 'Doe', 'john.doe@example.com', '123456789', '2022-01-01', 'JR001', 5
INSERT INTO EMPLOYEES VALUES(2, 'Jane', 'Smith', 'jane.smith@example.com', '987654321', '2022-02-15', 'JR002
INSERT INTO EMPLOYEES VALUES(3, 'Michael', 'Johnson', 'michael.johnson@example.com', '456123789', '2021-11-1
INSERT INTO EMPLOYEES VALUES(4, 'Emily', 'Brown', 'emily.brown@example.com', '789456123', '2022-03-20', 'JR0
INSERT INTO EMPLOYEES VALUES(5, 'Sarah', 'Austin', 'sarah.austin@example.com', '9876543210', '2022-04-10', '
SELECT * FROM EMPLOYEES;
SELECT Employee_Id, First_Name, Last_Name, Salary FROM EMPLOYEES;
SELECT Employee_Id, First_Name, Last_Name FROM EMPLOYEES WHERE Manager_Id = 100;
SELECT First_Name, Last_Name FROM EMPLOYEES WHERE Salary >= 4800;
SELECT Employee_Id, First_Name, Last_Name FROM EMPLOYEES WHERE Last_Name = 'AUSTIN';
SELECT First_Name, Last_Name FROM EMPLOYEES WHERE Department_Id IN (60, 70, 80);
SELECT DISTINCT Manager_Id FROM EMPLOYEES;
```

STDIN

Input for the program (Optional)

```
+-------------+------------+-----------+
| Employee_Id | First_Name | Last_Name |
+-------------+------------+-----------+
|           1 | John       | Doe       |
|           2 | Jane       | Smith     |
|           5 | Sarah      | Austin    |
+-------------+------------+-----------+
+------------+-----------+
| First_Name | Last_Name |
+------------+-----------+
| John       | Doe       |
| Jane       | Smith     |
| Michael    | Johnson   |
| Emily      | Brown     |
| Sarah      | Austin    |
+------------+-----------+
+-------------+------------+-----------+
| Employee_Id | First_Name | Last_Name |
+-------------+------------+-----------+
|           5 | Sarah      | Austin    |
+-------------+------------+-----------+
+------------+
| Manager_Id |
+------------+
|        100 |
|        101 |
+------------+
```

(f) Display the unique Manager_Id.

ANSWER:

SELECT DISTINCT Manager_Id FROM EMPLOYEES;

queries.sql + 425xmxu2v NEW MYSQL RUN

```sql
CREATE TABLE EMPLOYEES (
    Employee_Id INT PRIMARY KEY,
    First_Name VARCHAR(50),
    Last_Name VARCHAR(50),
    Email VARCHAR(100),
    Phone_Number VARCHAR(20),
    Hire_Date DATE,
    Job_Id VARCHAR(10),
    Salary DECIMAL(10, 2),
    Commission_Pct DECIMAL(5, 2),
    Manager_Id INT,
    Department_Id INT
);
INSERT INTO EMPLOYEES VALUES(1, 'John', 'Doe', 'john.doe@example.com', '123456789', '2022-01-01', 'JR001', 5
INSERT INTO EMPLOYEES VALUES(2, 'Jane', 'Smith', 'jane.smith@example.com', '987654321', '2022-02-15', 'JR002
INSERT INTO EMPLOYEES VALUES(3, 'Michael', 'Johnson', 'michael.johnson@example.com', '456123789', '2021-11-1
INSERT INTO EMPLOYEES VALUES(4, 'Emily', 'Brown', 'emily.brown@example.com', '789456123', '2022-03-20', 'JR0
INSERT INTO EMPLOYEES VALUES(5, 'Sarah', 'Austin', 'sarah.austin@example.com', '9876543210', '2022-04-10', '
SELECT * FROM EMPLOYEES;
SELECT Employee_Id, First_Name, Last_Name, Salary FROM EMPLOYEES;
SELECT Employee_Id, First_Name, Last_Name FROM EMPLOYEES WHERE Manager_Id = 100;
SELECT First_Name, Last_Name FROM EMPLOYEES WHERE Salary >= 4800;
SELECT Employee_Id, First_Name, Last_Name FROM EMPLOYEES WHERE Last_Name = 'AUSTIN';
SELECT First_Name, Last_Name FROM EMPLOYEES WHERE Department_Id IN (60, 70, 80);
SELECT DISTINCT Manager_Id FROM EMPLOYEES;
```

STDIN

Input for the program (Optional)

```
+-------------+------------+-----------+
| Employee_Id | First_Name | Last_Name |
+-------------+------------+-----------+
|           1 | John       | Doe       |
|           2 | Jane       | Smith     |
|           5 | Sarah      | Austin    |
+-------------+------------+-----------+
+------------+-----------+
| First_Name | Last_Name |
+------------+-----------+
| John       | Doe       |
| Jane       | Smith     |
| Michael    | Johnson   |
| Emily      | Brown     |
| Sarah      | Austin    |
+------------+-----------+
+-------------+------------+-----------+
| Employee_Id | First_Name | Last_Name |
+-------------+------------+-----------+
|           5 | Sarah      | Austin    |
+-------------+------------+-----------+
+------------+
| Manager_Id |
+------------+
|        100 |
|        101 |
+------------+
```

2. (Exercise on updating records in table) Create Client_master with the following fields(ClientNO, Name, Address, City, State, bal_due)

(a) Insert five records

CREATE TABLE Client_master (

ClientNO INT PRIMARY KEY,

```
    Name VARCHAR(100),
    Address VARCHAR(255),
    City VARCHAR(100),
    State VARCHAR(100),
    bal_due DECIMAL(10, 2)
);

-- Inserting sample records with Indian names into the Client_master table
INSERT INTO Client_master VALUES(1, 'Rahul Kumar', '123 Main St', 'New Delhi', 'Delhi', 1000.00);
INSERT INTO Client_master VALUES(2, 'Priya Sharma', '456 Elm St', 'Mumbai', 'Maharashtra', 2500.00);
INSERT INTO Client_master VALUES(3, 'Amit Patel', '789 Oak St', 'Ahmedabad', 'Gujarat', 1500.00);
INSERT INTO Client_master VALUES(4, 'Neha Singh', '101 Pine St', 'Bangalore', 'Karnataka', 2000.00);
INSERT INTO Client_master VALUES(5, 'Sneha Gupta', '202 Maple St', 'Kolkata', 'West Bengal', 5500.00);

SELECT * FROM Client_master;
```

queries.sql 425xp587f NEW MYSQL RUN

```sql
CREATE TABLE Client_master (
    ClientNO INT PRIMARY KEY,
    Name VARCHAR(100),
    Address VARCHAR(255),
    City VARCHAR(100),
    State VARCHAR(100),
    bal_due DECIMAL(10, 2)
);

-- Inserting sample records with Indian names into the Client_master table
INSERT INTO Client_master VALUES(1, 'Rahul Kumar', '123 Main St', 'New Delhi', 'Delhi', 1000.00);
INSERT INTO Client_master VALUES(2, 'Priya Sharma', '456 Elm St', 'Mumbai', 'Maharashtra', 2500.00);
INSERT INTO Client_master VALUES(3, 'Amit Patel', '789 Oak St', 'Ahmedabad', 'Gujarat', 1500.00);
INSERT INTO Client_master VALUES(4, 'Neha Singh', '101 Pine St', 'Bangalore', 'Karnataka', 2000.00);
INSERT INTO Client_master VALUES(5, 'Sneha Gupta', '202 Maple St', 'Kolkata', 'West Bengal', 5500.00);

SELECT * FROM Client_master;
SELECT Name FROM Client_master WHERE bal_due>5000;
UPDATE Client_master SET bal_due = 5100.00 WHERE ClientNO = 2;

SELECT * FROM Client_master;
ALTER TABLE Client_master RENAME TO Client12;
SELECT ClientNO, Name, Address, City, State, bal_due AS BALANCE FROM Client12;


DESC Client12;
```

STDIN

Input for the program (Optional)

```
+----------+--------------+--------------+----------+-------------+---------+
| ClientNO | Name         | Address      | City     | State       | bal_due |
+----------+--------------+--------------+----------+-------------+---------+
|        1 | Rahul Kumar  | 123 Main St  | New Delhi | Delhi       | 1000.00 |
|        2 | Priya Sharma | 456 Elm St   | Mumbai   | Maharashtra | 2500.00 |
|        3 | Amit Patel   | 789 Oak St   | Ahmedabad | Gujarat     | 1500.00 |
|        4 | Neha Singh   | 101 Pine St  | Bangalore | Karnataka   | 2000.00 |
|        5 | Sneha Gupta  | 202 Maple St | Kolkata  | West Bengal | 5500.00 |
+----------+--------------+--------------+----------+-------------+---------+
+-------------+
| Name        |
+-------------+
| Sneha Gupta |
+-------------+
+----------+--------------+--------------+----------+-------------+---------+
| ClientNO | Name         | Address      | City     | State       | bal_due |
+----------+--------------+--------------+----------+-------------+---------+
|        1 | Rahul Kumar  | 123 Main St  | New Delhi | Delhi       | 1000.00 |
|        2 | Priya Sharma | 456 Elm St   | Mumbai   | Maharashtra | 5100.00 |
|        3 | Amit Patel   | 789 Oak St   | Ahmedabad | Gujarat     | 1500.00 |
|        4 | Neha Singh   | 101 Pine St  | Bangalore | Karnataka   | 2000.00 |
|        5 | Sneha Gupta  | 202 Maple St | Kolkata  | West Bengal | 5500.00 |
+----------+--------------+--------------+----------+-------------+---------+
+----------+--------------+--------------+----------+-------------+---------+
| ClientNO | Name         | Address      | City     | State       | BALANCE |
+----------+--------------+--------------+----------+-------------+---------+
|        1 | Rahul Kumar  | 123 Main St  | New Delhi | Delhi       | 1000.00 |
|        2 | Priya Sharma | 456 Elm St   | Mumbai   | Maharashtra | 5100.00 |
```

(b) Find the names of clients whose bal_due> 5000 .

SELECT Name FROM Client_master WHERE bal_due>5000;

queries.sql 425xp587f NEW MYSQL RUN

```sql
CREATE TABLE Client_master (
    ClientNO INT PRIMARY KEY,
    Name VARCHAR(100),
    Address VARCHAR(255),
    City VARCHAR(100),
    State VARCHAR(100),
    bal_due DECIMAL(10, 2)
);

-- Inserting sample records with Indian names into the Client_master table
INSERT INTO Client_master VALUES(1, 'Rahul Kumar', '123 Main St', 'New Delhi', 'Delhi', 1000.00);
INSERT INTO Client_master VALUES(2, 'Priya Sharma', '456 Elm St', 'Mumbai', 'Maharashtra', 2500.00);
INSERT INTO Client_master VALUES(3, 'Amit Patel', '789 Oak St', 'Ahmedabad', 'Gujarat', 1500.00);
INSERT INTO Client_master VALUES(4, 'Neha Singh', '101 Pine St', 'Bangalore', 'Karnataka', 2000.00);
INSERT INTO Client_master VALUES(5, 'Sneha Gupta', '202 Maple St', 'Kolkata', 'West Bengal', 5500.00);

SELECT * FROM Client_master;
SELECT Name FROM Client_master WHERE bal_due>5000;
UPDATE Client_master SET bal_due = 5100.00 WHERE ClientNO = 2;

SELECT * FROM Client_master;
ALTER TABLE Client_master RENAME TO Client12;
SELECT ClientNO, Name, Address, City, State, bal_due AS BALANCE FROM Client12;


DESC Client12;
```

STDIN

Input for the program (Optional)

```
+----------+--------------+--------------+----------+-------------+---------+
| ClientNO | Name         | Address      | City     | State       | bal_due |
+----------+--------------+--------------+----------+-------------+---------+
|        1 | Rahul Kumar  | 123 Main St  | New Delhi | Delhi       | 1000.00 |
|        2 | Priya Sharma | 456 Elm St   | Mumbai   | Maharashtra | 2500.00 |
|        3 | Amit Patel   | 789 Oak St   | Ahmedabad | Gujarat     | 1500.00 |
|        4 | Neha Singh   | 101 Pine St  | Bangalore | Karnataka   | 2000.00 |
|        5 | Sneha Gupta  | 202 Maple St | Kolkata  | West Bengal | 5500.00 |
+----------+--------------+--------------+----------+-------------+---------+
+-------------+
| Name        |
+-------------+
| Sneha Gupta |
+-------------+
+----------+--------------+--------------+----------+-------------+---------+
| ClientNO | Name         | Address      | City     | State       | bal_due |
+----------+--------------+--------------+----------+-------------+---------+
|        1 | Rahul Kumar  | 123 Main St  | New Delhi | Delhi       | 1000.00 |
|        2 | Priya Sharma | 456 Elm St   | Mumbai   | Maharashtra | 5100.00 |
|        3 | Amit Patel   | 789 Oak St   | Ahmedabad | Gujarat     | 1500.00 |
|        4 | Neha Singh   | 101 Pine St  | Bangalore | Karnataka   | 2000.00 |
|        5 | Sneha Gupta  | 202 Maple St | Kolkata  | West Bengal | 5500.00 |
+----------+--------------+--------------+----------+-------------+---------+
+----------+--------------+--------------+----------+-------------+---------+
| ClientNO | Name         | Address      | City     | State       | BALANCE |
+----------+--------------+--------------+----------+-------------+---------+
|        1 | Rahul Kumar  | 123 Main St  | New Delhi | Delhi       | 1000.00 |
|        2 | Priya Sharma | 456 Elm St   | Mumbai   | Maharashtra | 5100.00 |
```


(c) Change the bal_due of ClientNO “ C123” to Rs. 5100

UPDATE Client_master SET bal_due = 5100.00 WHERE ClientNO = 2;

queries.sql 425xp587f NEW MYSQL RUN

```sql
CREATE TABLE Client_master (
    ClientNO INT PRIMARY KEY,
    Name VARCHAR(100),
    Address VARCHAR(255),
    City VARCHAR(100),
    State VARCHAR(100),
    bal_due DECIMAL(10, 2)
);

-- Inserting sample records with Indian names into the Client_master table
INSERT INTO Client_master VALUES(1, 'Rahul Kumar', '123 Main St', 'New Delhi', 'Delhi', 1000.00);
INSERT INTO Client_master VALUES(2, 'Priya Sharma', '456 Elm St', 'Mumbai', 'Maharashtra', 2500.00);
INSERT INTO Client_master VALUES(3, 'Amit Patel', '789 Oak St', 'Ahmedabad', 'Gujarat', 1500.00);
INSERT INTO Client_master VALUES(4, 'Neha Singh', '101 Pine St', 'Bangalore', 'Karnataka', 2000.00);
INSERT INTO Client_master VALUES(5, 'Sneha Gupta', '202 Maple St', 'Kolkata', 'West Bengal', 5500.00);

SELECT * FROM Client_master;
SELECT Name FROM Client_master WHERE bal_due>5000;
UPDATE Client_master SET bal_due = 5100.00 WHERE ClientNO = 2;

SELECT * FROM Client_master;
ALTER TABLE Client_master RENAME TO Client12;
SELECT ClientNO, Name, Address, City, State, bal_due AS BALANCE FROM Client12;


DESC Client12;
```

STDIN

Input for the program (Optional)

ClientNO	Name	Address	City	State	bal_due
1	Rahul Kumar	123 Main St	New Delhi	Delhi	1000.00
2	Priya Sharma	456 Elm St	Mumbai	Maharashtra	2500.00
3	Amit Patel	789 Oak St	Ahmedabad	Gujarat	1500.00
4	Neha Singh	101 Pine St	Bangalore	Karnataka	2000.00
5	Sneha Gupta	202 Maple St	Kolkata	West Bengal	5500.00

Name
Sneha Gupta

ClientNO	Name	Address	City	State	bal_due
1	Rahul Kumar	123 Main St	New Delhi	Delhi	1000.00
2	Priya Sharma	456 Elm St	Mumbai	Maharashtra	5100.00
3	Amit Patel	789 Oak St	Ahmedabad	Gujarat	1500.00
4	Neha Singh	101 Pine St	Bangalore	Karnataka	2000.00
5	Sneha Gupta	202 Maple St	Kolkata	West Bengal	5500.00

ClientNO	Name	Address	City	State	BALANCE
1	Rahul Kumar	123 Main St	New Delhi	Delhi	1000.00
2	Priya Sharma	456 Elm St	Mumbai	Maharashtra	5100.00

(d) Change the name of Client_master to Client12 .

ALTER TABLE Client_master RENAME TO Client12;

```sql
CREATE TABLE Client_master (
    ClientNO INT PRIMARY KEY,
    Name VARCHAR(100),
    Address VARCHAR(255),
    City VARCHAR(100),
    State VARCHAR(100),
    bal_due DECIMAL(10, 2)
);

-- Inserting sample records with Indian names into the Client_master table
INSERT INTO Client_master VALUES(1, 'Rahul Kumar', '123 Main St', 'New Delhi', 'Delhi', 1000.00);
INSERT INTO Client_master VALUES(2, 'Priya Sharma', '456 Elm St', 'Mumbai', 'Maharashtra', 2500.00);
INSERT INTO Client_master VALUES(3, 'Amit Patel', '789 Oak St', 'Ahmedabad', 'Gujarat', 1500.00);
INSERT INTO Client_master VALUES(4, 'Neha Singh', '101 Pine St', 'Bangalore', 'Karnataka', 2000.00);
INSERT INTO Client_master VALUES(5, 'Sneha Gupta', '202 Maple St', 'Kolkata', 'West Bengal', 5500.00);

SELECT * FROM Client_master;
SELECT Name FROM Client_master WHERE bal_due>5000;
UPDATE Client_master SET bal_due = 5100.00 WHERE ClientNO = 2;

SELECT * FROM Client_master;
ALTER TABLE Client_master RENAME TO Client12;
SELECT ClientNO, Name, Address, City, State, bal_due AS BALANCE FROM Client12;


DESC Client12;
```

STDIN

Input for the program (Optional)

ClientNO	Name	Address	City	State	bal_due
1	Rahul Kumar	123 Main St	New Delhi	Delhi	1000.00
2	Priya Sharma	456 Elm St	Mumbai	Maharashtra	5100.00
3	Amit Patel	789 Oak St	Ahmedabad	Gujarat	1500.00
4	Neha Singh	101 Pine St	Bangalore	Karnataka	2000.00
5	Sneha Gupta	202 Maple St	Kolkata	West Bengal	5500.00

ClientNO	Name	Address	City	State	BALANCE
1	Rahul Kumar	123 Main St	New Delhi	Delhi	1000.00
2	Priya Sharma	456 Elm St	Mumbai	Maharashtra	5100.00
3	Amit Patel	789 Oak St	Ahmedabad	Gujarat	1500.00
4	Neha Singh	101 Pine St	Bangalore	Karnataka	2000.00
5	Sneha Gupta	202 Maple St	Kolkata	West Bengal	5500.00

Field	Type	Null	Key	Default	Extra
ClientNO	int	NO	PRI	NULL	
Name	varchar(100)	YES		NULL	
Address	varchar(255)	YES		NULL	
City	varchar(100)	YES		NULL	
State	varchar(100)	YES		NULL	
bal_due	decimal(10,2)	YES		NULL	

(e) Display the bal_due heading as “BALANCE”.

SELECT ClientNO, Name, Address, City, State, bal_due AS BALANCE FROM Client12;

```sql
CREATE TABLE Client_master (
    ClientNO INT PRIMARY KEY,
    Name VARCHAR(100),
    Address VARCHAR(255),
    City VARCHAR(100),
    State VARCHAR(100),
    bal_due DECIMAL(10, 2)
);

-- Inserting sample records with Indian names into the Client_master table
INSERT INTO Client_master VALUES(1, 'Rahul Kumar', '123 Main St', 'New Delhi', 'Delhi', 1000.00);
INSERT INTO Client_master VALUES(2, 'Priya Sharma', '456 Elm St', 'Mumbai', 'Maharashtra', 2500.00);
INSERT INTO Client_master VALUES(3, 'Amit Patel', '789 Oak St', 'Ahmedabad', 'Gujarat', 1500.00);
INSERT INTO Client_master VALUES(4, 'Neha Singh', '101 Pine St', 'Bangalore', 'Karnataka', 2000.00);
INSERT INTO Client_master VALUES(5, 'Sneha Gupta', '202 Maple St', 'Kolkata', 'West Bengal', 5500.00);

SELECT * FROM Client_master;
SELECT Name FROM Client_master WHERE bal_due>5000;
UPDATE Client_master SET bal_due = 5100.00 WHERE ClientNO = 2;

SELECT * FROM Client_master;
ALTER TABLE Client_master RENAME TO Client12;
SELECT ClientNO, Name, Address, City, State, bal_due AS BALANCE FROM Client12;


DESC Client12;
```

STDIN

Input for the program (Optional)

ClientNO	Name	Address	City	State	bal_due
1	Rahul Kumar	123 Main St	New Delhi	Delhi	1000.00
2	Priya Sharma	456 Elm St	Mumbai	Maharashtra	5100.00
3	Amit Patel	789 Oak St	Ahmedabad	Gujarat	1500.00
4	Neha Singh	101 Pine St	Bangalore	Karnataka	2000.00
5	Sneha Gupta	202 Maple St	Kolkata	West Bengal	5500.00

ClientNO	Name	Address	City	State	BALANCE
1	Rahul Kumar	123 Main St	New Delhi	Delhi	1000.00
2	Priya Sharma	456 Elm St	Mumbai	Maharashtra	5100.00
3	Amit Patel	789 Oak St	Ahmedabad	Gujarat	1500.00
4	Neha Singh	101 Pine St	Bangalore	Karnataka	2000.00
5	Sneha Gupta	202 Maple St	Kolkata	West Bengal	5500.00

Field	Type	Null	Key	Default	Extra
ClientNO	int	NO	PRI	NULL	
Name	varchar(100)	YES		NULL	
Address	varchar(255)	YES		NULL	
City	varchar(100)	YES		NULL	
State	varchar(100)	YES		NULL	
bal_due	decimal(10,2)	YES		NULL	

3. Rollback and Commit commands Create Teacher table with the following fields(Name, DeptNo, Date of joining, DeptName, Location, Salary)

(a) Insert five records

START TRANSACTION;

-- Create the Teacher table

CREATE TABLE Teacher (

Name VARCHAR(100),

DeptNo INT,

Joining_Date DATE,

DeptName VARCHAR(100),

Location VARCHAR(100),

Salary DECIMAL(10, 2)

);

-- Insert five records

INSERT INTO Teacher (Name, DeptNo, Joining_Date, DeptName, Location, Salary)

VALUES

('John Smith', 101, '2022-01-15', 'Mathematics', 'Building A', 5000.00),

('Jane Doe', 102, '2022-02-20', 'Commerce', 'Building B', 5500.00),

('Michael Johnson', 103, '2021-12-10', 'Science', 'Building C', 4800.00),

('Emily Brown', 104, '2022-03-25', 'English', 'Building D', 5200.00),

('Sarah Austin', 101, '2022-04-30', 'Mathematics', 'Building A', 6000.00);

```sql
-- Start a transaction
START TRANSACTION;

-- Create the Teacher table
CREATE TABLE Teacher (
    Name VARCHAR(100),
    DeptNo INT,
    Joining_Date DATE,
    DeptName VARCHAR(100),
    Location VARCHAR(100),
    Salary DECIMAL(10, 2)
);

-- Insert five records
INSERT INTO Teacher (Name, DeptNo, Joining_Date, DeptName, Location, Salary)
VALUES
    ('John Smith', 101, '2022-01-15', 'Mathematics', 'Building A', 5000.00),
    ('Jane Doe', 102, '2022-02-20', 'Commerce', 'Building B', 5500.00),
    ('Michael Johnson', 103, '2021-12-10', 'Science', 'Building C', 4800.00),
    ('Emily Brown', 104, '2022-03-25', 'English', 'Building D', 5200.00),
    ('Sarah Austin', 101, '2022-04-30', 'Mathematics', 'Building A', 6000.00);

-- Increment salary by 25% for Mathematics Department
UPDATE Teacher
SET Salary = Salary * 1.25
WHERE DeptName = 'Mathematics';
SELECT * FROM Teacher;
-- Rollback the transaction
ROLLBACK;

-- Start a new transaction
START TRANSACTION;

-- Insert five records again
INSERT INTO Teacher (Name, DeptNo, Joining_Date, DeptName, Location, Salary)
VALUES
    ('John Smith', 101, '2022-01-15', 'Mathematics', 'Building A', 5000.00),
    ('Jane Doe', 102, '2022-02-20', 'Commerce', 'Building B', 5500.00),
    ('Michael Johnson', 103, '2021-12-10', 'Science', 'Building C', 4800.00),
    ('Emily Brown', 104, '2022-03-25', 'English', 'Building D', 5200.00),
    ('Sarah Austin', 101, '2022-04-30', 'Mathematics', 'Building A', 6000.00);
```

STDIN

Input for the program (Optional)

Output:

Name	DeptNo	Joining_Date	DeptName	Location	Salary
John Smith	101	2022-01-15	Mathematics	Building A	6250.00
Jane Doe	102	2022-02-20	Commerce	Building B	5500.00
Michael Johnson	103	2021-12-10	Science	Building C	4800.00
Emily Brown	104	2022-03-25	English	Building D	5200.00
Sarah Austin	101	2022-04-30	Mathematics	Building A	7500.00

Name	DeptNo	Joining_Date	DeptName	Location	Salary
John Smith	101	2022-01-15	Mathematics	Building A	6250.00
Jane Doe	102	2022-02-20	Commerce	Building B	6325.00
Michael Johnson	103	2021-12-10	Science	Building C	4800.00
Emily Brown	104	2022-03-25	English	Building D	5200.00
Sarah Austin	101	2022-04-30	Mathematics	Building A	7500.00
John Smith	101	2022-01-15	Mathematics	Building A	5000.00
Jane Doe	102	2022-02-20	Commerce	Building B	6325.00
Michael Johnson	103	2021-12-10	Science	Building C	4800.00
Emily Brown	104	2022-03-25	English	Building D	5200.00
Sarah Austin	101	2022-04-30	Mathematics	Building A	6000.00

(b) Give Increment of 25% salary for Mathematics Department .

UPDATE Teacher

```sql
-- Start a transaction
START TRANSACTION;

-- Create the Teacher table
CREATE TABLE Teacher (
    Name VARCHAR(100),
    DeptNo INT,
    Joining_Date DATE,
    DeptName VARCHAR(100),
    Location VARCHAR(100),
    Salary DECIMAL(10, 2)
);

-- Insert five records
INSERT INTO Teacher (Name, DeptNo, Joining_Date, DeptName, Location, Salary)
VALUES
    ('John Smith', 101, '2022-01-15', 'Mathematics', 'Building A', 5000.00),
    ('Jane Doe', 102, '2022-02-20', 'Commerce', 'Building B', 5500.00),
    ('Michael Johnson', 103, '2021-12-10', 'Science', 'Building C', 4800.00),
    ('Emily Brown', 104, '2022-03-25', 'English', 'Building D', 5200.00),
    ('Sarah Austin', 101, '2022-04-30', 'Mathematics', 'Building A', 6000.00);

-- Increment salary by 25% for Mathematics Department
UPDATE Teacher
SET Salary = Salary * 1.25
WHERE DeptName = 'Mathematics';
SELECT * FROM Teacher;
-- Rollback the transaction
ROLLBACK;

-- Start a new transaction
START TRANSACTION;

-- Insert five records again
INSERT INTO Teacher (Name, DeptNo, Joining_Date, DeptName, Location, Salary)
VALUES
    ('John Smith', 101, '2022-01-15', 'Mathematics', 'Building A', 5000.00),
    ('Jane Doe', 102, '2022-02-20', 'Commerce', 'Building B', 5500.00),
    ('Michael Johnson', 103, '2021-12-10', 'Science', 'Building C', 4800.00),
    ('Emily Brown', 104, '2022-03-25', 'English', 'Building D', 5200.00),
    ('Sarah Austin', 101, '2022-04-30', 'Mathematics', 'Building A', 6000.00);
```

STDIN

Input for the program (Optional)

Output:

```
+-----------------+--------+--------------+-------------+------------+---------+
| Name            | DeptNo | Joining_Date | DeptName    | Location   | Salary  |
+-----------------+--------+--------------+-------------+------------+---------+
| John Smith      |    101 | 2022-01-15   | Mathematics | Building A | 6250.00 |
| Jane Doe        |    102 | 2022-02-20   | Commerce    | Building B | 5500.00 |
| Michael Johnson |    103 | 2021-12-10   | Science     | Building C | 4800.00 |
| Emily Brown     |    104 | 2022-03-25   | English     | Building D | 5200.00 |
| Sarah Austin    |    101 | 2022-04-30   | Mathematics | Building A | 7500.00 |
+-----------------+--------+--------------+-------------+------------+---------+
+-----------------+--------+--------------+-------------+------------+---------+
| Name            | DeptNo | Joining_Date | DeptName    | Location   | Salary  |
+-----------------+--------+--------------+-------------+------------+---------+
| John Smith      |    101 | 2022-01-15   | Mathematics | Building A | 6250.00 |
| Jane Doe        |    102 | 2022-02-20   | Commerce    | Building B | 6325.00 |
| Michael Johnson |    103 | 2021-12-10   | Science     | Building C | 4800.00 |
| Emily Brown     |    104 | 2022-03-25   | English     | Building D | 5200.00 |
| Sarah Austin    |    101 | 2022-04-30   | Mathematics | Building A | 7500.00 |
| John Smith      |    101 | 2022-01-15   | Mathematics | Building A | 5000.00 |
| Jane Doe        |    102 | 2022-02-20   | Commerce    | Building B | 6325.00 |
| Michael Johnson |    103 | 2021-12-10   | Science     | Building C | 4800.00 |
| Emily Brown     |    104 | 2022-03-25   | English     | Building D | 5200.00 |
| Sarah Austin    |    101 | 2022-04-30   | Mathematics | Building A | 6000.00 |
+-----------------+--------+--------------+-------------+------------+---------+
```

SET Salary = Salary * 1.25

WHERE DeptName = 'Mathematics';

SELECT * FROM Teacher;

```sql
-- Start a transaction
START TRANSACTION;

-- Create the Teacher table
CREATE TABLE Teacher (
    Name VARCHAR(100),
    DeptNo INT,
    Joining_Date DATE,
    DeptName VARCHAR(100),
    Location VARCHAR(100),
    Salary DECIMAL(10, 2)
);

-- Insert five records
INSERT INTO Teacher (Name, DeptNo, Joining_Date, DeptName, Location, Salary)
VALUES
    ('John Smith', 101, '2022-01-15', 'Mathematics', 'Building A', 5000.00),
    ('Jane Doe', 102, '2022-02-20', 'Commerce', 'Building B', 5500.00),
    ('Michael Johnson', 103, '2021-12-10', 'Science', 'Building C', 4800.00),
    ('Emily Brown', 104, '2022-03-25', 'English', 'Building D', 5200.00),
    ('Sarah Austin', 101, '2022-04-30', 'Mathematics', 'Building A', 6000.00);

-- Increment salary by 25% for Mathematics Department
UPDATE Teacher
SET Salary = Salary * 1.25
WHERE DeptName = 'Mathematics';
SELECT * FROM Teacher;
-- Rollback the transaction
ROLLBACK;

-- Start a new transaction
START TRANSACTION;

-- Insert five records again
INSERT INTO Teacher (Name, DeptNo, Joining_Date, DeptName, Location, Salary)
VALUES
    ('John Smith', 101, '2022-01-15', 'Mathematics', 'Building A', 5000.00),
    ('Jane Doe', 102, '2022-02-20', 'Commerce', 'Building B', 5500.00),
    ('Michael Johnson', 103, '2021-12-10', 'Science', 'Building C', 4800.00),
    ('Emily Brown', 104, '2022-03-25', 'English', 'Building D', 5200.00),
    ('Sarah Austin', 101, '2022-04-30', 'Mathematics', 'Building A', 6000.00);
```

STDIN

Input for the program (Optional)

Output:

```
+-----------------+--------+--------------+-------------+------------+---------+
| Name            | DeptNo | Joining_Date | DeptName    | Location   | Salary  |
+-----------------+--------+--------------+-------------+------------+---------+
| John Smith      |    101 | 2022-01-15   | Mathematics | Building A | 6250.00 |
| Jane Doe        |    102 | 2022-02-20   | Commerce    | Building B | 5500.00 |
| Michael Johnson |    103 | 2021-12-10   | Science     | Building C | 4800.00 |
| Emily Brown     |    104 | 2022-03-25   | English     | Building D | 5200.00 |
| Sarah Austin    |    101 | 2022-04-30   | Mathematics | Building A | 7500.00 |
+-----------------+--------+--------------+-------------+------------+---------+
+-----------------+--------+--------------+-------------+------------+---------+
| Name            | DeptNo | Joining_Date | DeptName    | Location   | Salary  |
+-----------------+--------+--------------+-------------+------------+---------+
| John Smith      |    101 | 2022-01-15   | Mathematics | Building A | 6250.00 |
| Jane Doe        |    102 | 2022-02-20   | Commerce    | Building B | 6325.00 |
| Michael Johnson |    103 | 2021-12-10   | Science     | Building C | 4800.00 |
| Emily Brown     |    104 | 2022-03-25   | English     | Building D | 5200.00 |
| Sarah Austin    |    101 | 2022-04-30   | Mathematics | Building A | 7500.00 |
| John Smith      |    101 | 2022-01-15   | Mathematics | Building A | 5000.00 |
| Jane Doe        |    102 | 2022-02-20   | Commerce    | Building B | 6325.00 |
| Michael Johnson |    103 | 2021-12-10   | Science     | Building C | 4800.00 |
| Emily Brown     |    104 | 2022-03-25   | English     | Building D | 5200.00 |
| Sarah Austin    |    101 | 2022-04-30   | Mathematics | Building A | 6000.00 |
+-----------------+--------+--------------+-------------+------------+---------+
```

(c) Perform Rollback command

ROLLBACK;

```sql
-- Start a transaction
START TRANSACTION;

-- Create the Teacher table
CREATE TABLE Teacher (
    Name VARCHAR(100),
    DeptNo INT,
    Joining_Date DATE,
    DeptName VARCHAR(100),
    Location VARCHAR(100),
    Salary DECIMAL(10, 2)
);

-- Insert five records
INSERT INTO Teacher (Name, DeptNo, Joining_Date, DeptName, Location, Salary)
VALUES
    ('John Smith', 101, '2022-01-15', 'Mathematics', 'Building A', 5000.00),
    ('Jane Doe', 102, '2022-02-20', 'Commerce', 'Building B', 5500.00),
    ('Michael Johnson', 103, '2021-12-10', 'Science', 'Building C', 4800.00),
    ('Emily Brown', 104, '2022-03-25', 'English', 'Building D', 5200.00),
    ('Sarah Austin', 101, '2022-04-30', 'Mathematics', 'Building A', 6000.00);

-- Increment salary by 25% for Mathematics Department
UPDATE Teacher
SET Salary = Salary * 1.25
WHERE DeptName = 'Mathematics';
SELECT * FROM Teacher;
-- Rollback the transaction
ROLLBACK;

-- Start a new transaction
START TRANSACTION;

-- Insert five records again
INSERT INTO Teacher (Name, DeptNo, Joining_Date, DeptName, Location, Salary)
VALUES
    ('John Smith', 101, '2022-01-15', 'Mathematics', 'Building A', 5000.00),
    ('Jane Doe', 102, '2022-02-20', 'Commerce', 'Building B', 5500.00),
    ('Michael Johnson', 103, '2021-12-10', 'Science', 'Building C', 4800.00),
    ('Emily Brown', 104, '2022-03-25', 'English', 'Building D', 5200.00),
    ('Sarah Austin', 101, '2022-04-30', 'Mathematics', 'Building A', 6000.00);
```

STDIN

Input for the program (Optional)

Output:

Name	DeptNo	Joining_Date	DeptName	Location	Salary
John Smith	101	2022-01-15	Mathematics	Building A	6250.00
Jane Doe	102	2022-02-20	Commerce	Building B	5500.00
Michael Johnson	103	2021-12-10	Science	Building C	4800.00
Emily Brown	104	2022-03-25	English	Building D	5200.00
Sarah Austin	101	2022-04-30	Mathematics	Building A	7500.00

Name	DeptNo	Joining_Date	DeptName	Location	Salary
John Smith	101	2022-01-15	Mathematics	Building A	6250.00
Jane Doe	102	2022-02-20	Commerce	Building B	6325.00
Michael Johnson	103	2021-12-10	Science	Building C	4800.00
Emily Brown	104	2022-03-25	English	Building D	5200.00
Sarah Austin	101	2022-04-30	Mathematics	Building A	7500.00
John Smith	101	2022-01-15	Mathematics	Building A	5000.00
Jane Doe	102	2022-02-20	Commerce	Building B	6325.00
Michael Johnson	103	2021-12-10	Science	Building C	4800.00
Emily Brown	104	2022-03-25	English	Building D	5200.00
Sarah Austin	101	2022-04-30	Mathematics	Building A	6000.00

(d) Give Increment of 15% salary for Commerce Department

START TRANSACTION;

-- Insert five records again

INSERT INTO Teacher (Name, DeptNo, Joining_Date, DeptName, Location, Salary)

VALUES

('John Smith', 101, '2022-01-15', 'Mathematics', 'Building A', 5000.00),

('Jane Doe', 102, '2022-02-20', 'Commerce', 'Building B', 5500.00),

('Michael Johnson', 103, '2021-12-10', 'Science', 'Building C', 4800.00),

('Emily Brown', 104, '2022-03-25', 'English', 'Building D', 5200.00),

('Sarah Austin', 101, '2022-04-30', 'Mathematics', 'Building A', 6000.00);

-- Increment salary by 15% for Commerce Department

UPDATE Teacher

SET Salary = Salary * 1.15

WHERE DeptName = 'Commerce';

SELECT * FROM Teacher;

queries.sql 425xp587f NEW MYSQL RUN

```sql
    Salary DECIMAL(10, 2)
);

-- Insert five records
INSERT INTO Teacher (Name, DeptNo, Joining_Date, DeptName, Location, Salary)
VALUES
    ('John Smith', 101, '2022-01-15', 'Mathematics', 'Building A', 5000.00),
    ('Jane Doe', 102, '2022-02-20', 'Commerce', 'Building B', 5500.00),
    ('Michael Johnson', 103, '2021-12-10', 'Science', 'Building C', 4800.00),
    ('Emily Brown', 104, '2022-03-25', 'English', 'Building D', 5200.00),
    ('Sarah Austin', 101, '2022-04-30', 'Mathematics', 'Building A', 6000.00);

-- Increment salary by 25% for Mathematics Department
UPDATE Teacher
SET Salary = Salary * 1.25
WHERE DeptName = 'Mathematics';
SELECT * FROM Teacher;
-- Rollback the transaction
ROLLBACK;

-- Start a new transaction
START TRANSACTION;

-- Insert five records again
INSERT INTO Teacher (Name, DeptNo, Joining_Date, DeptName, Location, Salary)
VALUES
    ('John Smith', 101, '2022-01-15', 'Mathematics', 'Building A', 5000.00),
    ('Jane Doe', 102, '2022-02-20', 'Commerce', 'Building B', 5500.00),
    ('Michael Johnson', 103, '2021-12-10', 'Science', 'Building C', 4800.00),
    ('Emily Brown', 104, '2022-03-25', 'English', 'Building D', 5200.00),
    ('Sarah Austin', 101, '2022-04-30', 'Mathematics', 'Building A', 6000.00);

-- Increment salary by 15% for Commerce Department
UPDATE Teacher
SET Salary = Salary * 1.15
WHERE DeptName = 'Commerce';

SELECT * FROM Teacher;
-- Commit the transaction
COMMIT;
```

STDIN

Input for the program (Optional)

Output:

Name	DeptNo	Joining_Date	DeptName	Location	Salary
John Smith	101	2022-01-15	Mathematics	Building A	6250.00
Jane Doe	102	2022-02-20	Commerce	Building B	5500.00
Michael Johnson	103	2021-12-10	Science	Building C	4800.00
Emily Brown	104	2022-03-25	English	Building D	5200.00
Sarah Austin	101	2022-04-30	Mathematics	Building A	7500.00

Name	DeptNo	Joining_Date	DeptName	Location	Salary
John Smith	101	2022-01-15	Mathematics	Building A	6250.00
Jane Doe	102	2022-02-20	Commerce	Building B	6325.00
Michael Johnson	103	2021-12-10	Science	Building C	4800.00
Emily Brown	104	2022-03-25	English	Building D	5200.00
Sarah Austin	101	2022-04-30	Mathematics	Building A	7500.00
John Smith	101	2022-01-15	Mathematics	Building A	5000.00
Jane Doe	102	2022-02-20	Commerce	Building B	6325.00
Michael Johnson	103	2021-12-10	Science	Building C	4800.00
Emily Brown	104	2022-03-25	English	Building D	5200.00
Sarah Austin	101	2022-04-30	Mathematics	Building A	6000.00

(e) Perform commit command.

-- Commit the transaction

COMMIT;

queries.sql 425xp587f NEW MYSQL RUN

```sql
    Salary DECIMAL(10, 2)
);

-- Insert five records
INSERT INTO Teacher (Name, DeptNo, Joining_Date, DeptName, Location, Salary)
VALUES
    ('John Smith', 101, '2022-01-15', 'Mathematics', 'Building A', 5000.00),
    ('Jane Doe', 102, '2022-02-20', 'Commerce', 'Building B', 5500.00),
    ('Michael Johnson', 103, '2021-12-10', 'Science', 'Building C', 4800.00),
    ('Emily Brown', 104, '2022-03-25', 'English', 'Building D', 5200.00),
    ('Sarah Austin', 101, '2022-04-30', 'Mathematics', 'Building A', 6000.00);

-- Increment salary by 25% for Mathematics Department
UPDATE Teacher
SET Salary = Salary * 1.25
WHERE DeptName = 'Mathematics';
SELECT * FROM Teacher;
-- Rollback the transaction
ROLLBACK;

-- Start a new transaction
START TRANSACTION;

-- Insert five records again
INSERT INTO Teacher (Name, DeptNo, Joining_Date, DeptName, Location, Salary)
VALUES
    ('John Smith', 101, '2022-01-15', 'Mathematics', 'Building A', 5000.00),
    ('Jane Doe', 102, '2022-02-20', 'Commerce', 'Building B', 5500.00),
    ('Michael Johnson', 103, '2021-12-10', 'Science', 'Building C', 4800.00),
    ('Emily Brown', 104, '2022-03-25', 'English', 'Building D', 5200.00),
    ('Sarah Austin', 101, '2022-04-30', 'Mathematics', 'Building A', 6000.00);

-- Increment salary by 15% for Commerce Department
UPDATE Teacher
SET Salary = Salary * 1.15
WHERE DeptName = 'Commerce';

SELECT * FROM Teacher;
-- Commit the transaction
COMMIT;
```

STDIN

Input for the program (Optional)

Output:

Name	DeptNo	Joining_Date	DeptName	Location	Salary
John Smith	101	2022-01-15	Mathematics	Building A	6250.00
Jane Doe	102	2022-02-20	Commerce	Building B	5500.00
Michael Johnson	103	2021-12-10	Science	Building C	4800.00
Emily Brown	104	2022-03-25	English	Building D	5200.00
Sarah Austin	101	2022-04-30	Mathematics	Building A	7500.00

Name	DeptNo	Joining_Date	DeptName	Location	Salary
John Smith	101	2022-01-15	Mathematics	Building A	6250.00
Jane Doe	102	2022-02-20	Commerce	Building B	6325.00
Michael Johnson	103	2021-12-10	Science	Building C	4800.00
Emily Brown	104	2022-03-25	English	Building D	5200.00
Sarah Austin	101	2022-04-30	Mathematics	Building A	7500.00
John Smith	101	2022-01-15	Mathematics	Building A	5000.00
Jane Doe	102	2022-02-20	Commerce	Building B	6325.00
Michael Johnson	103	2021-12-10	Science	Building C	4800.00
Emily Brown	104	2022-03-25	English	Building D	5200.00
Sarah Austin	101	2022-04-30	Mathematics	Building A	6000.00

4. (Exercise on order by and group by clauses) Create Sales table with the following fields(Sales No, Salesname, Branch, Salesamount, DOB)

(a) Insert five records

ANSWER:

```
-- Create the Sales table

CREATE TABLE Sales (

   Sales_No INT PRIMARY KEY,

   Salesname VARCHAR(100),

   Branch VARCHAR(100),

   Salesamount DECIMAL(10, 2),

   DOB DATE

);


-- Insert five records into the Sales table

INSERT INTO Sales (Sales_No, Salesname, Branch, Salesamount, DOB) VALUES

   (1, 'John', 'Branch A', 1500.00, '1990-05-15'),

   (2, 'Alice', 'Branch B', 2000.00, '1988-08-20'),

   (3, 'Bob', 'Branch A', 1800.00, '1995-12-10'),

   (4, 'Emily', 'Branch C', 2200.00, '1992-03-25'),

   (5, 'David', 'Branch B', 1900.00, '1993-12-05');
```

queries.sql 425xp587f NEW MYSQL RUN

```sql
-- Create the Sales table
CREATE TABLE Sales (
    Sales_No INT PRIMARY KEY,
    Salesname VARCHAR(100),
    Branch VARCHAR(100),
    Salesamount DECIMAL(10, 2),
    DOB DATE
);

-- Insert five records into the Sales table
INSERT INTO Sales (Sales_No, Salesname, Branch, Salesamount, DOB) VALUES
    (1, 'John', 'Branch A', 1500.00, '1990-05-15'),
    (2, 'Alice', 'Branch B', 2000.00, '1988-08-20'),
    (3, 'Bob', 'Branch A', 1800.00, '1995-12-10'),
    (4, 'Emily', 'Branch C', 2200.00, '1992-03-25'),
    (5, 'David', 'Branch B', 1900.00, '1993-12-05');

-- Calculate total sales amount in each branch
SELECT Branch, SUM(Salesamount) AS Total_Sales_Amount
FROM Sales
GROUP BY Branch;

-- Calculate average sales amount in each branch
SELECT Branch, AVG(Salesamount) AS Average_Sales_Amount
FROM Sales
GROUP BY Branch;

-- Display salesmen and DOB born in December
SELECT Salesname, DATE_FORMAT(DOB, '%d-%b-%y') AS DOB
FROM Sales
WHERE MONTH(DOB) = 12;

-- Display name and DOB of salesmen in alphabetical order of the month
SELECT Salesname, DATE_FORMAT(DOB, '%d-%b-%y') AS DOB
FROM Sales
ORDER BY MONTH(DOB), DAY(DOB), Salesname;
```

STDIN

Input for the program (Optional)

```
+----------+--------------------+
| Branch   | Total_Sales_Amount |
+----------+--------------------+
| Branch A |            3300.00 |
| Branch B |            3900.00 |
| Branch C |            2200.00 |
+----------+--------------------+
+----------+----------------------+
| Branch   | Average_Sales_Amount |
+----------+----------------------+
| Branch A |          1650.000000 |
| Branch B |          1950.000000 |
| Branch C |          2200.000000 |
+----------+----------------------+
+-----------+-----------+
| Salesname | DOB       |
+-----------+-----------+
| Bob       | 10-Dec-95 |
| David     | 05-Dec-93 |
+-----------+-----------+
+-----------+-----------+
| Salesname | DOB       |
+-----------+-----------+
| Emily     | 25-Mar-92 |
| John      | 15-May-90 |
| Alice     | 20-Aug-88 |
| David     | 05-Dec-93 |
| Bob       | 10-Dec-95 |
+-----------+-----------+
```

(b) Calculate total salesamount in each branch

SELECT Branch, SUM(Salesamount) AS Total_Sales_Amount

FROM Sales

GROUP BY Branch;

queries.sql 425xp587f NEW MYSQL RUN

```sql
-- Create the Sales table
CREATE TABLE Sales (
    Sales_No INT PRIMARY KEY,
    Salesname VARCHAR(100),
    Branch VARCHAR(100),
    Salesamount DECIMAL(10, 2),
    DOB DATE
);

-- Insert five records into the Sales table
INSERT INTO Sales (Sales_No, Salesname, Branch, Salesamount, DOB) VALUES
    (1, 'John', 'Branch A', 1500.00, '1990-05-15'),
    (2, 'Alice', 'Branch B', 2000.00, '1988-08-20'),
    (3, 'Bob', 'Branch A', 1800.00, '1995-12-10'),
    (4, 'Emily', 'Branch C', 2200.00, '1992-03-25'),
    (5, 'David', 'Branch B', 1900.00, '1993-12-05');

-- Calculate total sales amount in each branch
SELECT Branch, SUM(Salesamount) AS Total_Sales_Amount
FROM Sales
GROUP BY Branch;

-- Calculate average sales amount in each branch
SELECT Branch, AVG(Salesamount) AS Average_Sales_Amount
FROM Sales
GROUP BY Branch;

-- Display salesmen and DOB born in December
SELECT Salesname, DATE_FORMAT(DOB, '%d-%b-%y') AS DOB
FROM Sales
WHERE MONTH(DOB) = 12;

-- Display name and DOB of salesmen in alphabetical order of the month
SELECT Salesname, DATE_FORMAT(DOB, '%d-%b-%y') AS DOB
FROM Sales
ORDER BY MONTH(DOB), DAY(DOB), Salesname;
```

STDIN

Input for the program (Optional)

```
+----------+--------------------+
| Branch   | Total_Sales_Amount |
+----------+--------------------+
| Branch A |            3300.00 |
| Branch B |            3900.00 |
| Branch C |            2200.00 |
+----------+--------------------+
+----------+----------------------+
| Branch   | Average_Sales_Amount |
+----------+----------------------+
| Branch A |          1650.000000 |
| Branch B |          1950.000000 |
| Branch C |          2200.000000 |
+----------+----------------------+
+-----------+-----------+
| Salesname | DOB       |
+-----------+-----------+
| Bob       | 10-Dec-95 |
| David     | 05-Dec-93 |
+-----------+-----------+
+-----------+-----------+
| Salesname | DOB       |
+-----------+-----------+
| Emily     | 25-Mar-92 |
| John      | 15-May-90 |
| Alice     | 20-Aug-88 |
| David     | 05-Dec-93 |
| Bob       | 10-Dec-95 |
+-----------+-----------+
```

(c) Calculate average salesamount in each branch .

SELECT Branch, AVG(Salesamount) AS Average_Sales_Amount

FROM Sales

GROUP BY Branch;

queries.sql 425xp587f NEW MYSQL RUN

```sql
-- Create the Sales table
CREATE TABLE Sales (
    Sales_No INT PRIMARY KEY,
    Salesname VARCHAR(100),
    Branch VARCHAR(100),
    Salesamount DECIMAL(10, 2),
    DOB DATE
);

-- Insert five records into the Sales table
INSERT INTO Sales (Sales_No, Salesname, Branch, Salesamount, DOB) VALUES
    (1, 'John', 'Branch A', 1500.00, '1990-05-15'),
    (2, 'Alice', 'Branch B', 2000.00, '1988-08-20'),
    (3, 'Bob', 'Branch A', 1800.00, '1995-12-10'),
    (4, 'Emily', 'Branch C', 2200.00, '1992-03-25'),
    (5, 'David', 'Branch B', 1900.00, '1993-12-05');

-- Calculate total sales amount in each branch
SELECT Branch, SUM(Salesamount) AS Total_Sales_Amount
FROM Sales
GROUP BY Branch;

-- Calculate average sales amount in each branch
SELECT Branch, AVG(Salesamount) AS Average_Sales_Amount
FROM Sales
GROUP BY Branch;

-- Display salesmen and DOB born in December
SELECT Salesname, DATE_FORMAT(DOB, '%d-%b-%y') AS DOB
FROM Sales
WHERE MONTH(DOB) = 12;

-- Display name and DOB of salesmen in alphabetical order of the month
SELECT Salesname, DATE_FORMAT(DOB, '%d-%b-%y') AS DOB
FROM Sales
ORDER BY MONTH(DOB), DAY(DOB), Salesname;
```

STDIN

Input for the program (Optional)

```
+----------+--------------------+
| Branch   | Total_Sales_Amount |
+----------+--------------------+
| Branch A |            3300.00 |
| Branch B |            3900.00 |
| Branch C |            2200.00 |
+----------+--------------------+
+----------+----------------------+
| Branch   | Average_Sales_Amount |
+----------+----------------------+
| Branch A |          1650.000000 |
| Branch B |          1950.000000 |
| Branch C |          2200.000000 |
+----------+----------------------+
+-----------+-----------+
| Salesname | DOB       |
+-----------+-----------+
| Bob       | 10-Dec-95 |
| David     | 05-Dec-93 |
+-----------+-----------+
+-----------+-----------+
| Salesname | DOB       |
+-----------+-----------+
| Emily     | 25-Mar-92 |
| John      | 15-May-90 |
| Alice     | 20-Aug-88 |
| David     | 05-Dec-93 |
| Bob       | 10-Dec-95 |
+-----------+-----------+
```

(d) Display all the salesmen, DOB who are born in the month of December as day in character format i.e. 21-Dec-09

SELECT Salesname, DATE_FORMAT(DOB, '%d-%b-%y') AS DOB

FROM Sales

WHERE MONTH(DOB) = 12;

queries.sql 425xp587f NEW MYSQL RUN

```sql
-- Create the Sales table
CREATE TABLE Sales (
    Sales_No INT PRIMARY KEY,
    Salesname VARCHAR(100),
    Branch VARCHAR(100),
    Salesamount DECIMAL(10, 2),
    DOB DATE
);

-- Insert five records into the Sales table
INSERT INTO Sales (Sales_No, Salesname, Branch, Salesamount, DOB) VALUES
    (1, 'John', 'Branch A', 1500.00, '1990-05-15'),
    (2, 'Alice', 'Branch B', 2000.00, '1988-08-20'),
    (3, 'Bob', 'Branch A', 1800.00, '1995-12-10'),
    (4, 'Emily', 'Branch C', 2200.00, '1992-03-25'),
    (5, 'David', 'Branch B', 1900.00, '1993-12-05');

-- Calculate total sales amount in each branch
SELECT Branch, SUM(Salesamount) AS Total_Sales_Amount
FROM Sales
GROUP BY Branch;

-- Calculate average sales amount in each branch
SELECT Branch, AVG(Salesamount) AS Average_Sales_Amount
FROM Sales
GROUP BY Branch;

-- Display salesmen and DOB born in December
SELECT Salesname, DATE_FORMAT(DOB, '%d-%b-%y') AS DOB
FROM Sales
WHERE MONTH(DOB) = 12;

-- Display name and DOB of salesmen in alphabetical order of the month
SELECT Salesname, DATE_FORMAT(DOB, '%d-%b-%y') AS DOB
FROM Sales
ORDER BY MONTH(DOB), DAY(DOB), Salesname;
```

STDIN

Input for the program (Optional)

```
+----------+--------------------+
| Branch   | Total_Sales_Amount |
+----------+--------------------+
| Branch A |            3300.00 |
| Branch B |            3900.00 |
| Branch C |            2200.00 |
+----------+--------------------+
+----------+----------------------+
| Branch   | Average_Sales_Amount |
+----------+----------------------+
| Branch A |          1650.000000 |
| Branch B |          1950.000000 |
| Branch C |          2200.000000 |
+----------+----------------------+
+-----------+-----------+
| Salesname | DOB       |
+-----------+-----------+
| Bob       | 10-Dec-95 |
| David     | 05-Dec-93 |
+-----------+-----------+
+-----------+-----------+
| Salesname | DOB       |
+-----------+-----------+
| Emily     | 25-Mar-92 |
| John      | 15-May-90 |
| Alice     | 20-Aug-88 |
| David     | 05-Dec-93 |
| Bob       | 10-Dec-95 |
+-----------+-----------+
```

(e) Display the name and DOB of salesman in alphabetical order of the month.

SELECT Salesname, DATE_FORMAT(DOB, '%d-%b-%y') AS DOB

FROM Sales

ORDER BY MONTH(DOB), DAY(DOB), Salesname;

queries.sql 425xp587f NEW MYSQL RUN

```sql
-- Create the Sales table
CREATE TABLE Sales (
    Sales_No INT PRIMARY KEY,
    Salesname VARCHAR(100),
    Branch VARCHAR(100),
    Salesamount DECIMAL(10, 2),
    DOB DATE
);

-- Insert five records into the Sales table
INSERT INTO Sales (Sales_No, Salesname, Branch, Salesamount, DOB) VALUES
    (1, 'John', 'Branch A', 1500.00, '1990-05-15'),
    (2, 'Alice', 'Branch B', 2000.00, '1988-08-20'),
    (3, 'Bob', 'Branch A', 1800.00, '1995-12-10'),
    (4, 'Emily', 'Branch C', 2200.00, '1992-03-25'),
    (5, 'David', 'Branch B', 1900.00, '1993-12-05');

-- Calculate total sales amount in each branch
SELECT Branch, SUM(Salesamount) AS Total_Sales_Amount
FROM Sales
GROUP BY Branch;

-- Calculate average sales amount in each branch
SELECT Branch, AVG(Salesamount) AS Average_Sales_Amount
FROM Sales
GROUP BY Branch;

-- Display salesmen and DOB born in December
SELECT Salesname, DATE_FORMAT(DOB, '%d-%b-%y') AS DOB
FROM Sales
WHERE MONTH(DOB) = 12;

-- Display name and DOB of salesmen in alphabetical order of the month
SELECT Salesname, DATE_FORMAT(DOB, '%d-%b-%y') AS DOB
FROM Sales
ORDER BY MONTH(DOB), DAY(DOB), Salesname;
```

STDIN

Input for the program (Optional)

```
+----------+--------------------+
| Branch   | Total_Sales_Amount |
+----------+--------------------+
| Branch A |            3300.00 |
| Branch B |            3900.00 |
| Branch C |            2200.00 |
+----------+--------------------+
+----------+----------------------+
| Branch   | Average_Sales_Amount |
+----------+----------------------+
| Branch A |          1650.000000 |
| Branch B |          1950.000000 |
| Branch C |          2200.000000 |
+----------+----------------------+
+-----------+-----------+
| Salesname | DOB       |
+-----------+-----------+
| Bob       | 10-Dec-95 |
| David     | 05-Dec-93 |
+-----------+-----------+
+-----------+-----------+
| Salesname | DOB       |
+-----------+-----------+
| Emily     | 25-Mar-92 |
| John      | 15-May-90 |
| Alice     | 20-Aug-88 |
| David     | 05-Dec-93 |
| Bob       | 10-Dec-95 |
+-----------+-----------+
```

5. Create an Emp table with the following fields: (EmpNo, EmpName, Job,Basic, DA, HRA,PF, GrossPay, NetPay) (Calculate DA as 30% of Basic and HRA as 40% of Basic)

(a) Insert Five Records and calculate GrossPay and NetPay.

-- Create the Emp table

CREATE TABLE Emp (

EmpNo INT PRIMARY KEY,

EmpName VARCHAR(100),

Job VARCHAR(100),

```
    Basic DECIMAL(10, 2),
    DA DECIMAL(10, 2),
    HRA DECIMAL(10, 2),
    PF DECIMAL(10, 2),
    GrossPay DECIMAL(10, 2),
    NetPay DECIMAL(10, 2)
);

-- Insert Five Records and calculate GrossPay and NetPay
INSERT INTO Emp (EmpNo, EmpName, Job, Basic, DA, HRA, PF, GrossPay, NetPay) VALUES
    (1, 'John', 'Manager', 15000.00, 4500.00, 6000.00, 1800.00, NULL, NULL),
    (2, 'Alice', 'Clerk', 10000.00, 3000.00, 4000.00, 1200.00, NULL, NULL),
    (3, 'Bob', 'Technician', 12000.00, 3600.00, 4800.00, 1440.00, NULL, NULL),
    (4, 'Emily', 'Engineer', 18000.00, 5400.00, 7200.00, 2160.00, NULL, NULL),
    (5, 'David', 'Analyst', 20000.00, 6000.00, 8000.00, 2400.00, NULL, NULL);

-- Calculate GrossPay and NetPay
UPDATE Emp SET GrossPay = Basic + DA + HRA, NetPay = Basic + DA + HRA - PF;
```

queries.sql 425xp587f NEW MYSQL RUN

```sql
-- Create the Emp table
CREATE TABLE Emp (
    EmpNo INT PRIMARY KEY,
    EmpName VARCHAR(100),
    Job VARCHAR(100),
    Basic DECIMAL(10, 2),
    DA DECIMAL(10, 2),
    HRA DECIMAL(10, 2),
    PF DECIMAL(10, 2),
    GrossPay DECIMAL(10, 2),
    NetPay DECIMAL(10, 2)
);

-- Insert Five Records and calculate GrossPay and NetPay
INSERT INTO Emp (EmpNo, EmpName, Job, Basic, DA, HRA, PF, GrossPay, NetPay) VALUES
    (1, 'John', 'Manager', 15000.00, 4500.00, 6000.00, 1800.00, NULL, NULL),
    (2, 'Alice', 'Clerk', 10000.00, 3000.00, 4000.00, 1200.00, NULL, NULL),
    (3, 'Bob', 'Technician', 12000.00, 3600.00, 4800.00, 1440.00, NULL, NULL),
    (4, 'Emily', 'Engineer', 18000.00, 5400.00, 7200.00, 2160.00, NULL, NULL),
    (5, 'David', 'Analyst', 20000.00, 6000.00, 8000.00, 2400.00, NULL, NULL);

-- Calculate GrossPay and NetPay
UPDATE Emp SET GrossPay = Basic + DA + HRA, NetPay = Basic + DA + HRA - PF;

-- Display the employees whose Basic is lowest in each department
SELECT EmpNo, EmpName, Job, Basic FROM Emp e1
WHERE Basic = (SELECT MIN(Basic) FROM Emp e2 WHERE e1.Job = e2.Job);

-- If NetPay is less than <rs 10,000 add rs.1200 as special allowances
UPDATE Emp SET NetPay = NetPay + 1200
WHERE NetPay < 10000;

-- Display the employees whose grosspay lies between 10,000 & 20,000
SELECT EmpNo, EmpName, Job, GrossPay
FROM Emp
WHERE GrossPay BETWEEN 10000 AND 20000;

-- Display all the employees who earn maximum salary
SELECT EmpNo, EmpName, Job, Basic
FROM Emp
WHERE Basic = (SELECT MAX(Basic) FROM Emp);
```

STDIN

Input for the program (Optional)

Output:

```
+-------+---------+------------+----------+
| EmpNo | EmpName | Job        | Basic    |
+-------+---------+------------+----------+
|     1 | John    | Manager    | 15000.00 |
|     2 | Alice   | Clerk      | 10000.00 |
|     3 | Bob     | Technician | 12000.00 |
|     4 | Emily   | Engineer   | 18000.00 |
|     5 | David   | Analyst    | 20000.00 |
+-------+---------+------------+----------+
+-------+---------+-------+----------+
| EmpNo | EmpName | Job   | GrossPay |
+-------+---------+-------+----------+
|     2 | Alice   | Clerk | 17000.00 |
+-------+---------+-------+----------+
+-------+---------+---------+----------+
| EmpNo | EmpName | Job     | Basic    |
+-------+---------+---------+----------+
|     5 | David   | Analyst | 20000.00 |
+-------+---------+---------+----------+
```

(b) Display the employees whose Basic is lowest in each department .

-- Display the employees whose Basic is lowest in each department

SELECT EmpNo, EmpName, Job, Basic FROM Emp e1

WHERE Basic = (SELECT MIN(Basic) FROM Emp e2 WHERE e1.Job = e2.Job);

queries.sql 425xp587f NEW MYSQL RUN

```sql
-- Create the Emp table
CREATE TABLE Emp (
    EmpNo INT PRIMARY KEY,
    EmpName VARCHAR(100),
    Job VARCHAR(100),
    Basic DECIMAL(10, 2),
    DA DECIMAL(10, 2),
    HRA DECIMAL(10, 2),
    PF DECIMAL(10, 2),
    GrossPay DECIMAL(10, 2),
    NetPay DECIMAL(10, 2)
);

-- Insert Five Records and calculate GrossPay and NetPay
INSERT INTO Emp (EmpNo, EmpName, Job, Basic, DA, HRA, PF, GrossPay, NetPay) VALUES
    (1, 'John', 'Manager', 15000.00, 4500.00, 6000.00, 1800.00, NULL, NULL),
    (2, 'Alice', 'Clerk', 10000.00, 3000.00, 4000.00, 1200.00, NULL, NULL),
    (3, 'Bob', 'Technician', 12000.00, 3600.00, 4800.00, 1440.00, NULL, NULL),
    (4, 'Emily', 'Engineer', 18000.00, 5400.00, 7200.00, 2160.00, NULL, NULL),
    (5, 'David', 'Analyst', 20000.00, 6000.00, 8000.00, 2400.00, NULL, NULL);

-- Calculate GrossPay and NetPay
UPDATE Emp SET GrossPay = Basic + DA + HRA, NetPay = Basic + DA + HRA - PF;

-- Display the employees whose Basic is lowest in each department
SELECT EmpNo, EmpName, Job, Basic FROM Emp e1
WHERE Basic = (SELECT MIN(Basic) FROM Emp e2 WHERE e1.Job = e2.Job);

-- If NetPay is less than <rs 10,000 add rs.1200 as special allowances
UPDATE Emp SET NetPay = NetPay + 1200
WHERE NetPay < 10000;

-- Display the employees whose grosspay lies between 10,000 & 20,000
SELECT EmpNo, EmpName, Job, GrossPay
FROM Emp
WHERE GrossPay BETWEEN 10000 AND 20000;

-- Display all the employees who earn maximum salary
SELECT EmpNo, EmpName, Job, Basic
FROM Emp
WHERE Basic = (SELECT MAX(Basic) FROM Emp);
```

STDIN

Input for the program (Optional)

Output:

```
+-------+---------+------------+----------+
| EmpNo | EmpName | Job        | Basic    |
+-------+---------+------------+----------+
|     1 | John    | Manager    | 15000.00 |
|     2 | Alice   | Clerk      | 10000.00 |
|     3 | Bob     | Technician | 12000.00 |
|     4 | Emily   | Engineer   | 18000.00 |
|     5 | David   | Analyst    | 20000.00 |
+-------+---------+------------+----------+
+-------+---------+-------+----------+
| EmpNo | EmpName | Job   | GrossPay |
+-------+---------+-------+----------+
|     2 | Alice   | Clerk | 17000.00 |
+-------+---------+-------+----------+
+-------+---------+---------+----------+
| EmpNo | EmpName | Job     | Basic    |
+-------+---------+---------+----------+
|     5 | David   | Analyst | 20000.00 |
+-------+---------+---------+----------+
```

(c) If NetPay is less than <rs 10,000 add rs.1200 as special allowances.

-- If NetPay is less than <rs 10,000 add rs.1200 as special allowances

UPDATE Emp SET NetPay = NetPay + 1200

WHERE NetPay < 10000;

queries.sql 425xp587f NEW MYSQL RUN

```sql
-- Create the Emp table
CREATE TABLE Emp (
    EmpNo INT PRIMARY KEY,
    EmpName VARCHAR(100),
    Job VARCHAR(100),
    Basic DECIMAL(10, 2),
    DA DECIMAL(10, 2),
    HRA DECIMAL(10, 2),
    PF DECIMAL(10, 2),
    GrossPay DECIMAL(10, 2),
    NetPay DECIMAL(10, 2)
);

-- Insert Five Records and calculate GrossPay and NetPay
INSERT INTO Emp (EmpNo, EmpName, Job, Basic, DA, HRA, PF, GrossPay, NetPay) VALUES
    (1, 'John', 'Manager', 15000.00, 4500.00, 6000.00, 1800.00, NULL, NULL),
    (2, 'Alice', 'Clerk', 10000.00, 3000.00, 4000.00, 1200.00, NULL, NULL),
    (3, 'Bob', 'Technician', 12000.00, 3600.00, 4800.00, 1440.00, NULL, NULL),
    (4, 'Emily', 'Engineer', 18000.00, 5400.00, 7200.00, 2160.00, NULL, NULL),
    (5, 'David', 'Analyst', 20000.00, 6000.00, 8000.00, 2400.00, NULL, NULL);

-- Calculate GrossPay and NetPay
UPDATE Emp SET GrossPay = Basic + DA + HRA, NetPay = Basic + DA + HRA - PF;

-- Display the employees whose Basic is lowest in each department
SELECT EmpNo, EmpName, Job, Basic FROM Emp e1
WHERE Basic = (SELECT MIN(Basic) FROM Emp e2 WHERE e1.Job = e2.Job);

-- If NetPay is less than <rs 10,000 add rs.1200 as special allowances
UPDATE Emp SET NetPay = NetPay + 1200
WHERE NetPay < 10000;

-- Display the employees whose grosspay lies between 10,000 & 20,000
SELECT EmpNo, EmpName, Job, GrossPay
FROM Emp
WHERE GrossPay BETWEEN 10000 AND 20000;

-- Display all the employees who earn maximum salary
SELECT EmpNo, EmpName, Job, Basic
FROM Emp
WHERE Basic = (SELECT MAX(Basic) FROM Emp);
```

STDIN

Input for the program (Optional)

Output:

```
+-------+---------+------------+----------+
| EmpNo | EmpName | Job        | Basic    |
+-------+---------+------------+----------+
|     1 | John    | Manager    | 15000.00 |
|     2 | Alice   | Clerk      | 10000.00 |
|     3 | Bob     | Technician | 12000.00 |
|     4 | Emily   | Engineer   | 18000.00 |
|     5 | David   | Analyst    | 20000.00 |
+-------+---------+------------+----------+
+-------+---------+-------+----------+
| EmpNo | EmpName | Job   | GrossPay |
+-------+---------+-------+----------+
|     2 | Alice   | Clerk | 17000.00 |
+-------+---------+-------+----------+
+-------+---------+---------+----------+
| EmpNo | EmpName | Job     | Basic    |
+-------+---------+---------+----------+
|     5 | David   | Analyst | 20000.00 |
+-------+---------+---------+----------+
```

(d) Display the employees whose grosspay lies between 10,000 & 20,000.

-- Display the employees whose grosspay lies between 10,000 & 20,000

SELECT EmpNo, EmpName, Job, GrossPay

FROM Emp

WHERE GrossPay BETWEEN 10000 AND 20000;

queries.sql 425xp587f NEW MYSQL RUN

```sql
-- Create the Emp table
CREATE TABLE Emp (
    EmpNo INT PRIMARY KEY,
    EmpName VARCHAR(100),
    Job VARCHAR(100),
    Basic DECIMAL(10, 2),
    DA DECIMAL(10, 2),
    HRA DECIMAL(10, 2),
    PF DECIMAL(10, 2),
    GrossPay DECIMAL(10, 2),
    NetPay DECIMAL(10, 2)
);

-- Insert Five Records and calculate GrossPay and NetPay
INSERT INTO Emp (EmpNo, EmpName, Job, Basic, DA, HRA, PF, GrossPay, NetPay) VALUES
    (1, 'John', 'Manager', 15000.00, 4500.00, 6000.00, 1800.00, NULL, NULL),
    (2, 'Alice', 'Clerk', 10000.00, 3000.00, 4000.00, 1200.00, NULL, NULL),
    (3, 'Bob', 'Technician', 12000.00, 3600.00, 4800.00, 1440.00, NULL, NULL),
    (4, 'Emily', 'Engineer', 18000.00, 5400.00, 7200.00, 2160.00, NULL, NULL),
    (5, 'David', 'Analyst', 20000.00, 6000.00, 8000.00, 2400.00, NULL, NULL);

-- Calculate GrossPay and NetPay
UPDATE Emp SET GrossPay = Basic + DA + HRA, NetPay = Basic + DA + HRA - PF;

-- Display the employees whose Basic is lowest in each department
SELECT EmpNo, EmpName, Job, Basic FROM Emp e1
WHERE Basic = (SELECT MIN(Basic) FROM Emp e2 WHERE e1.Job = e2.Job);

-- If NetPay is less than <rs 10,000 add rs.1200 as special allowances
UPDATE Emp SET NetPay = NetPay + 1200
WHERE NetPay < 10000;

-- Display the employees whose grosspay lies between 10,000 & 20,000
SELECT EmpNo, EmpName, Job, GrossPay
FROM Emp
WHERE GrossPay BETWEEN 10000 AND 20000;

-- Display all the employees who earn maximum salary
SELECT EmpNo, EmpName, Job, Basic
FROM Emp
WHERE Basic = (SELECT MAX(Basic) FROM Emp);
```

STDIN

Input for the program (Optional)

Output:

```
+-------+---------+------------+----------+
| EmpNo | EmpName | Job        | Basic    |
+-------+---------+------------+----------+
|     1 | John    | Manager    | 15000.00 |
|     2 | Alice   | Clerk      | 10000.00 |
|     3 | Bob     | Technician | 12000.00 |
|     4 | Emily   | Engineer   | 18000.00 |
|     5 | David   | Analyst    | 20000.00 |
+-------+---------+------------+----------+
+-------+---------+-------+----------+
| EmpNo | EmpName | Job   | GrossPay |
+-------+---------+-------+----------+
|     2 | Alice   | Clerk | 17000.00 |
+-------+---------+-------+----------+
+-------+---------+---------+----------+
| EmpNo | EmpName | Job     | Basic    |
+-------+---------+---------+----------+
|     5 | David   | Analyst | 20000.00 |
+-------+---------+---------+----------+
```

(e) Display all the employees who earn maximum salary.

-- Display all the employees who earn maximum salary

SELECT EmpNo, EmpName, Job, Basic

FROM Emp

WHERE Basic = (SELECT MAX(Basic) FROM Emp);

queries.sql 425xp587f NEW MYSQL RUN

```sql
-- Create the Emp table
CREATE TABLE Emp (
    EmpNo INT PRIMARY KEY,
    EmpName VARCHAR(100),
    Job VARCHAR(100),
    Basic DECIMAL(10, 2),
    DA DECIMAL(10, 2),
    HRA DECIMAL(10, 2),
    PF DECIMAL(10, 2),
    GrossPay DECIMAL(10, 2),
    NetPay DECIMAL(10, 2)
);

-- Insert Five Records and calculate GrossPay and NetPay
INSERT INTO Emp (EmpNo, EmpName, Job, Basic, DA, HRA, PF, GrossPay, NetPay) VALUES
    (1, 'John', 'Manager', 15000.00, 4500.00, 6000.00, 1800.00, NULL, NULL),
    (2, 'Alice', 'Clerk', 10000.00, 3000.00, 4000.00, 1200.00, NULL, NULL),
    (3, 'Bob', 'Technician', 12000.00, 3600.00, 4800.00, 1440.00, NULL, NULL),
    (4, 'Emily', 'Engineer', 18000.00, 5400.00, 7200.00, 2160.00, NULL, NULL),
    (5, 'David', 'Analyst', 20000.00, 6000.00, 8000.00, 2400.00, NULL, NULL);

-- Calculate GrossPay and NetPay
UPDATE Emp SET GrossPay = Basic + DA + HRA, NetPay = Basic + DA + HRA - PF;

-- Display the employees whose Basic is lowest in each department
SELECT EmpNo, EmpName, Job, Basic FROM Emp e1
WHERE Basic = (SELECT MIN(Basic) FROM Emp e2 WHERE e1.Job = e2.Job);

-- If NetPay is less than <rs 10,000 add rs.1200 as special allowances
UPDATE Emp SET NetPay = NetPay + 1200
WHERE NetPay < 10000;

-- Display the employees whose grosspay lies between 10,000 & 20,000
SELECT EmpNo, EmpName, Job, GrossPay
FROM Emp
WHERE GrossPay BETWEEN 10000 AND 20000;

-- Display all the employees who earn maximum salary
SELECT EmpNo, EmpName, Job, Basic
FROM Emp
WHERE Basic = (SELECT MAX(Basic) FROM Emp);
```

STDIN

Input for the program (Optional)

Output:

```
+-------+---------+------------+----------+
| EmpNo | EmpName | Job        | Basic    |
+-------+---------+------------+----------+
|     1 | John    | Manager    | 15000.00 |
|     2 | Alice   | Clerk      | 10000.00 |
|     3 | Bob     | Technician | 12000.00 |
|     4 | Emily   | Engineer   | 18000.00 |
|     5 | David   | Analyst    | 20000.00 |
+-------+---------+------------+----------+
+-------+---------+-------+----------+
| EmpNo | EmpName | Job   | GrossPay |
+-------+---------+-------+----------+
|     2 | Alice   | Clerk | 17000.00 |
+-------+---------+-------+----------+
+-------+---------+---------+----------+
| EmpNo | EmpName | Job     | Basic    |
+-------+---------+---------+----------+
|     5 | David   | Analyst | 20000.00 |
+-------+---------+---------+----------+
```