SUBMITTED BY: Prachi Aggarwal

COURSE: Bsc(H) Computer Science

ROLL NO: 21570015

Examination ROLL NO: 21033570042

SUBJECT: Database Management System

SEMESTER: 4th

YEAR: 2nd

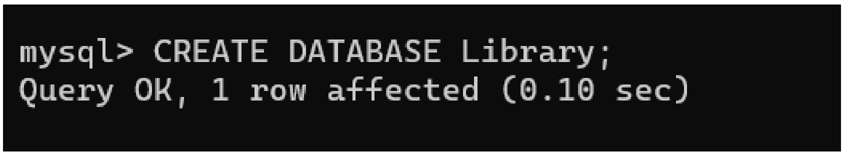
SUBMITTED TO: Deeksha Mam

TOPIC: Practical File
(Library+Univesity+Company database)

LIBRARY DATABASE

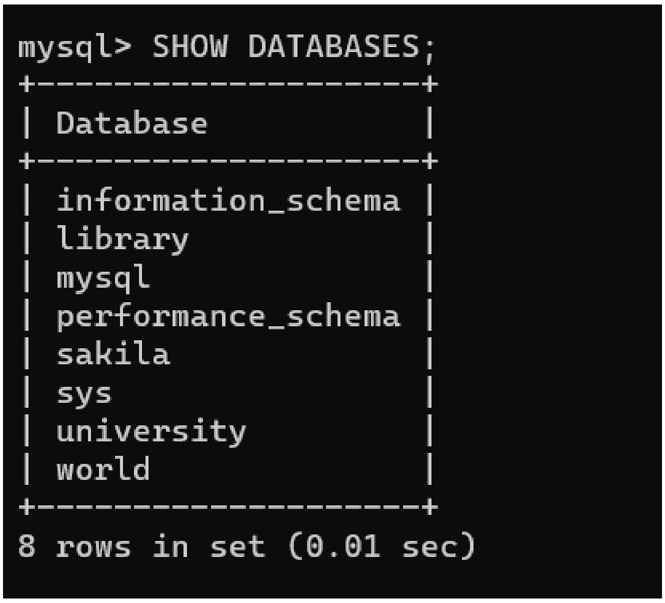
1. **Create a database having two tables with the specified fields, to computerize a library system of a Delhi University College.**

***Create Database Library**



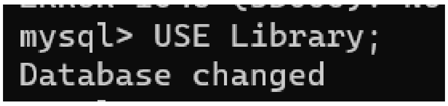
```
mysql> CREATE DATABASE Library;
Query OK, 1 row affected (0.10 sec)
```

***Show Database**



```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| library            |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| university         |
| world              |
+--------------------+
8 rows in set (0.01 sec)
```

***Use Database**

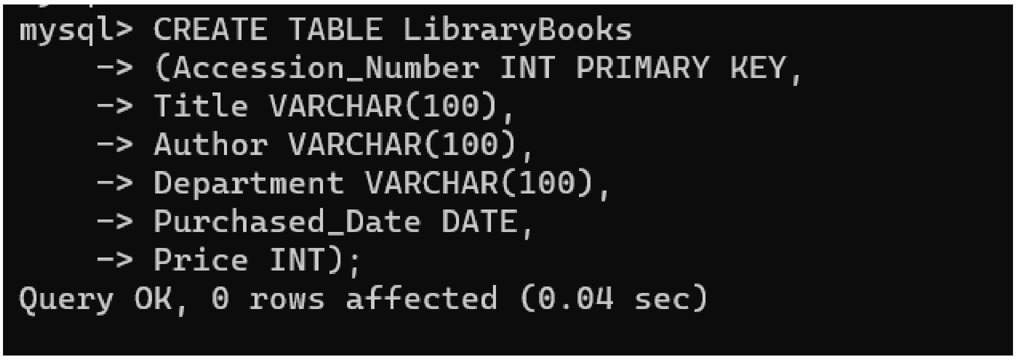


```
mysql> USE Library;
Database changed
```

(a)

***Creation of table name LibraryBooks with columns name Accession Number, Title, Author, Department, Purchase Date and Price.**

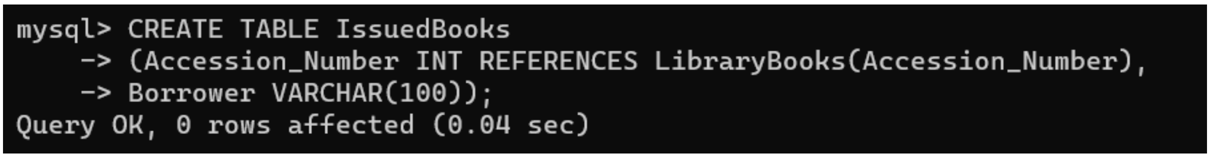
PRIMARY KEY-> Accession_Number
FOREIGN KEY-> NONE



```
mysql> CREATE TABLE LibraryBooks
    -> (Accession_Number INT PRIMARY KEY,
    -> Title VARCHAR(100),
    -> Author VARCHAR(100),
    -> Department VARCHAR(100),
    -> Purchased_Date DATE,
    -> Price INT);
Query OK, 0 rows affected (0.04 sec)
```

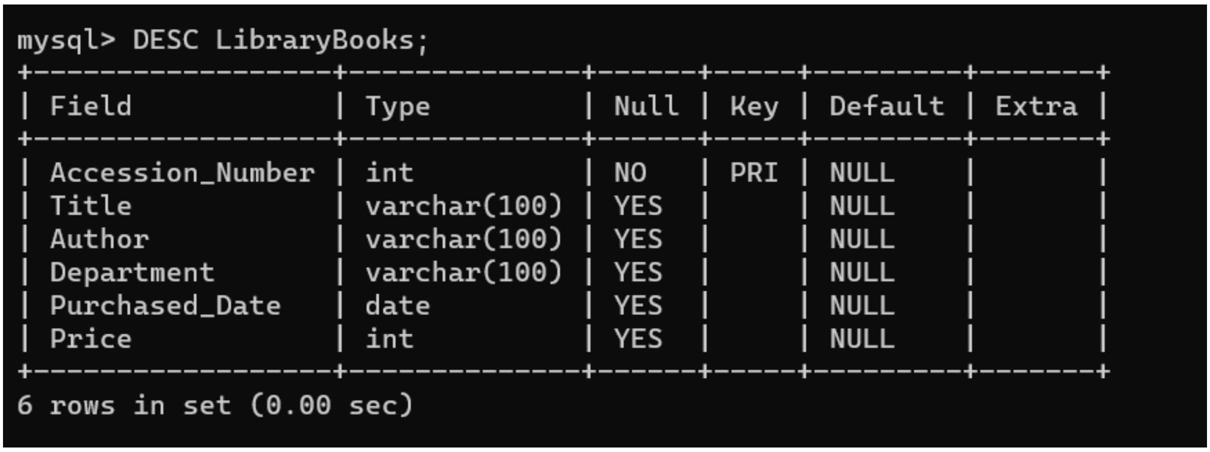
***Creation of table name IssuedBooks with columns name Accession Number and Borrower.**

<u>PRIMARY KEY</u>-> NONE
<u>FOREIGN KEY</u>-> Accession_Number



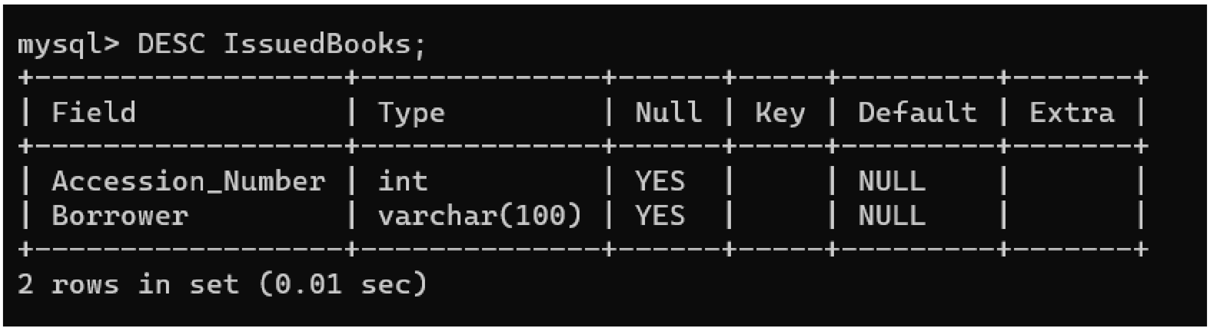
```
mysql> CREATE TABLE IssuedBooks
    -> (Accession_Number INT REFERENCES LibraryBooks(Accession_Number),
    -> Borrower VARCHAR(100));
Query OK, 0 rows affected (0.04 sec)
```

***Structure of table LibraryBooks**



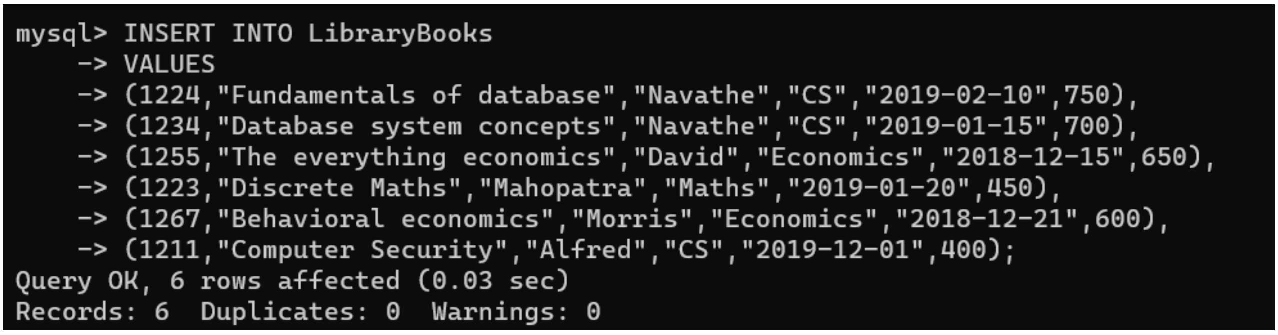
```
mysql> DESC LibraryBooks;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| Accession_Number | int          | NO   | PRI | NULL    |       |
| Title            | varchar(100) | YES  |     | NULL    |       |
| Author           | varchar(100) | YES  |     | NULL    |       |
| Department       | varchar(100) | YES  |     | NULL    |       |
| Purchased_Date   | date         | YES  |     | NULL    |       |
| Price            | int          | YES  |     | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

***Structure of table IssuedBooks**



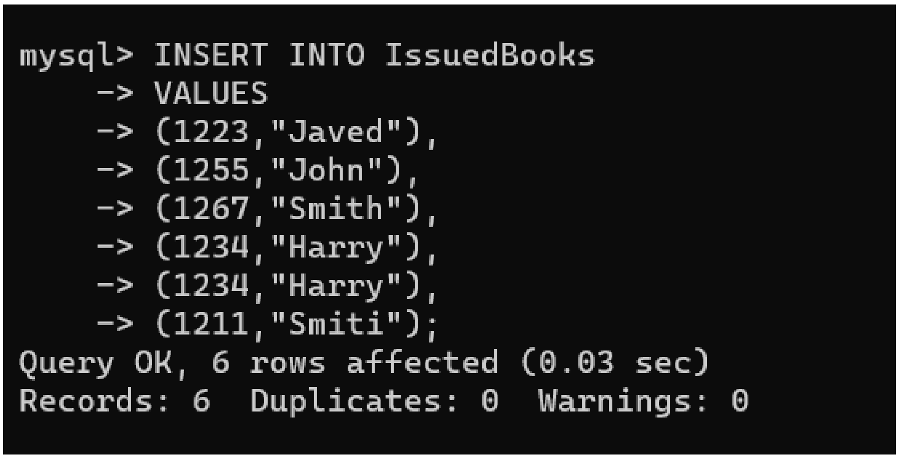
```
mysql> DESC IssuedBooks;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| Accession_Number | int          | YES  |     | NULL    |       |
| Borrower         | varchar(100) | YES  |     | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

***Insertion of records in table LibraryBooks**



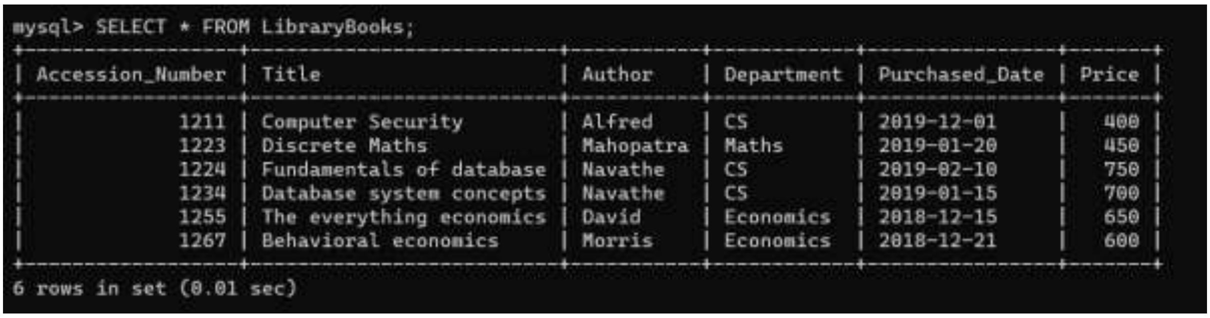
```
mysql> INSERT INTO LibraryBooks
    -> VALUES
    -> (1224,"Fundamentals of database","Navathe","CS","2019-02-10",750),
    -> (1234,"Database system concepts","Navathe","CS","2019-01-15",700),
    -> (1255,"The everything economics","David","Economics","2018-12-15",650),
    -> (1223,"Discrete Maths","Mahopatra","Maths","2019-01-20",450),
    -> (1267,"Behavioral economics","Morris","Economics","2018-12-21",600),
    -> (1211,"Computer Security","Alfred","CS","2019-12-01",400);
Query OK, 6 rows affected (0.03 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

***Insertion of records in table IssuedBooks**



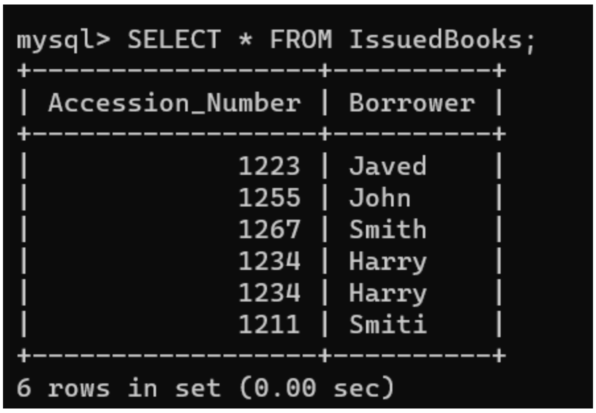
```
mysql> INSERT INTO IssuedBooks
    -> VALUES
    -> (1223,"Javed"),
    -> (1255,"John"),
    -> (1267,"Smith"),
    -> (1234,"Harry"),
    -> (1234,"Harry"),
    -> (1211,"Smiti");
Query OK, 6 rows affected (0.03 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

***Table→ LibraryBooks**



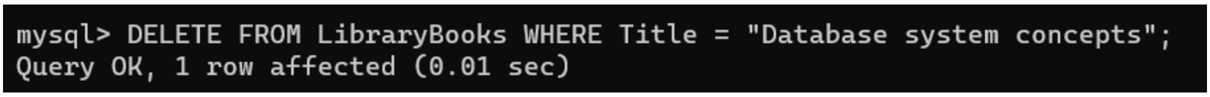
```
mysql> SELECT * FROM LibraryBooks;
+------------------+--------------------------+-----------+------------+----------------+-------+
| Accession_Number | Title                    | Author    | Department | Purchased_Date | Price |
+------------------+--------------------------+-----------+------------+----------------+-------+
|             1211 | Computer Security        | Alfred    | CS         | 2019-12-01     |   400 |
|             1223 | Discrete Maths           | Mahopatra | Maths      | 2019-01-20     |   450 |
|             1224 | Fundamentals of database | Navathe   | CS         | 2019-02-10     |   750 |
|             1234 | Database system concepts | Navathe   | CS         | 2019-01-15     |   700 |
|             1255 | The everything economics | David     | Economics  | 2018-12-15     |   650 |
|             1267 | Behavioral economics     | Morris    | Economics  | 2018-12-21     |   600 |
+------------------+--------------------------+-----------+------------+----------------+-------+
6 rows in set (0.01 sec)
```

***Table → IssuedBooks**



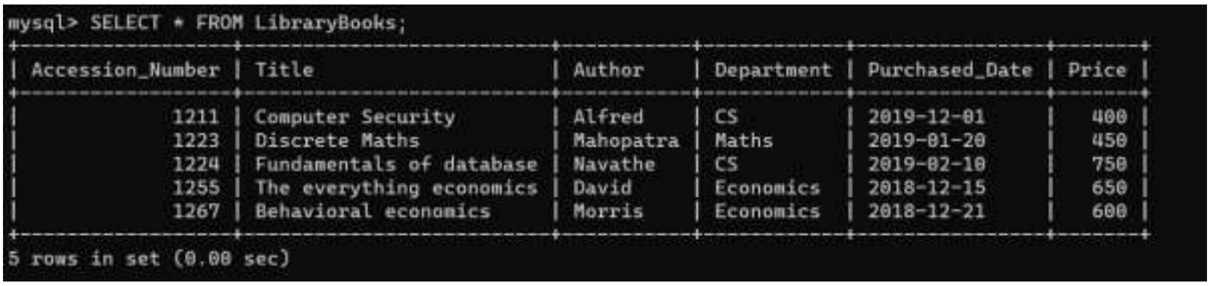
```
mysql> SELECT * FROM IssuedBooks;
+------------------+----------+
| Accession_Number | Borrower |
+------------------+----------+
|             1223 | Javed    |
|             1255 | John     |
|             1267 | Smith    |
|             1234 | Harry    |
|             1234 | Harry    |
|             1211 | Smiti    |
+------------------+----------+
6 rows in set (0.00 sec)
```

(b) Delete the record of book titled “Database System Concepts”.



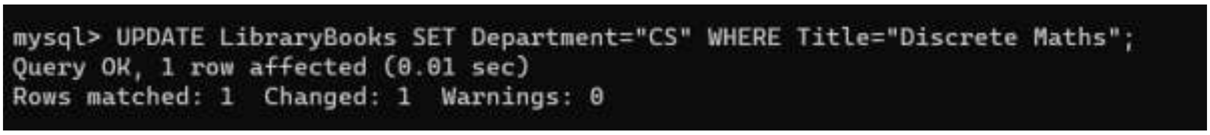
```
mysql> DELETE FROM LibraryBooks WHERE Title = "Database system concepts";
Query OK, 1 row affected (0.01 sec)
```

LibraryBooks after DELETE command



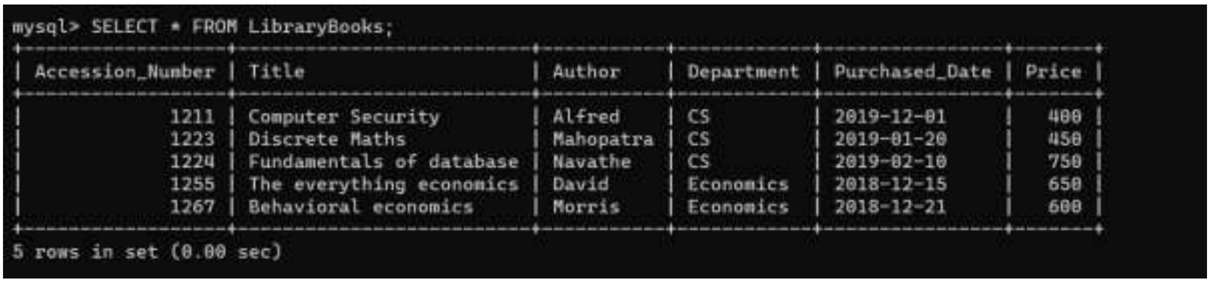
```
mysql> SELECT * FROM LibraryBooks;
+------------------+--------------------------+-----------+------------+----------------+-------+
| Accession_Number | Title                    | Author    | Department | Purchased_Date | Price |
+------------------+--------------------------+-----------+------------+----------------+-------+
|             1211 | Computer Security        | Alfred    | CS         | 2019-12-01     |   400 |
|             1223 | Discrete Maths           | Mahopatra | Maths      | 2019-01-20     |   450 |
|             1224 | Fundamentals of database | Navathe   | CS         | 2019-02-10     |   750 |
|             1255 | The everything economics | David     | Economics  | 2018-12-15     |   650 |
|             1267 | Behavioral economics     | Morris    | Economics  | 2018-12-21     |   600 |
+------------------+--------------------------+-----------+------------+----------------+-------+
5 rows in set (0.00 sec)
```

(c) Change the Department of book titled “Discrete Maths” to “CS”.



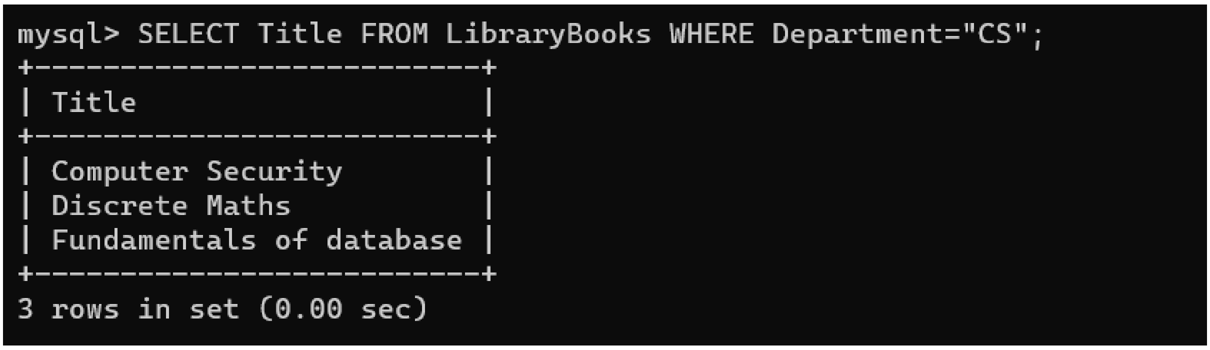
```
mysql> UPDATE LibraryBooks SET Department="CS" WHERE Title="Discrete Maths";
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

LibraryBooks after UPDATE command



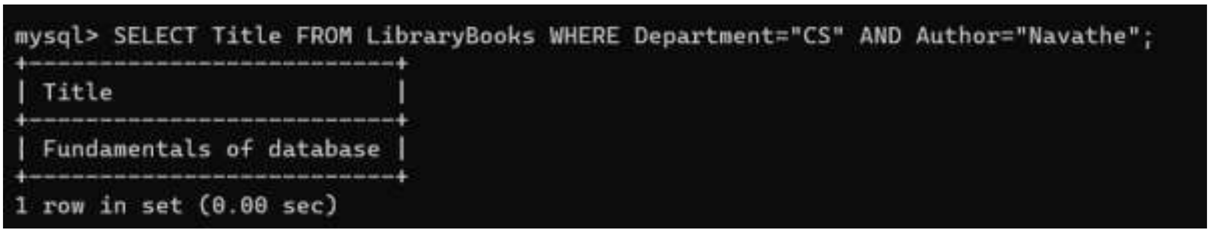
```
mysql> SELECT * FROM LibraryBooks;
+------------------+--------------------------+-----------+------------+----------------+-------+
| Accession_Number | Title                    | Author    | Department | Purchased_Date | Price |
+------------------+--------------------------+-----------+------------+----------------+-------+
|             1211 | Computer Security        | Alfred    | CS         | 2019-12-01     |   400 |
|             1223 | Discrete Maths           | Mahopatra | CS         | 2019-01-20     |   450 |
|             1224 | Fundamentals of database | Navathe   | CS         | 2019-02-10     |   750 |
|             1255 | The everything economics | David     | Economics  | 2018-12-15     |   650 |
|             1267 | Behavioral economics     | Morris    | Economics  | 2018-12-21     |   600 |
+------------------+--------------------------+-----------+------------+----------------+-------+
5 rows in set (0.00 sec)
```

(d) List all books that belong to “CS” department.



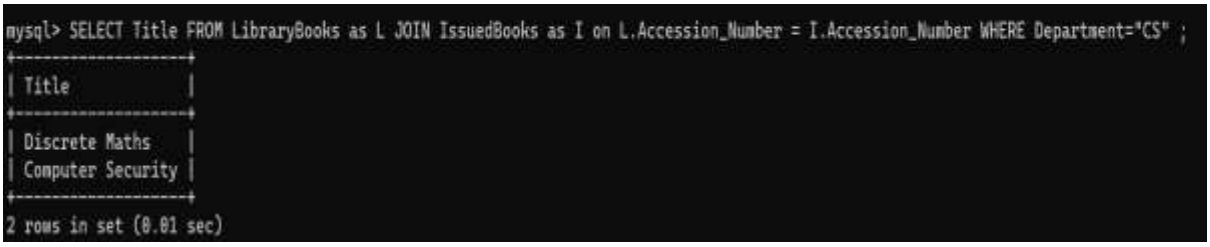
```
mysql> SELECT Title FROM LibraryBooks WHERE Department="CS";
+--------------------------+
| Title                    |
+--------------------------+
| Computer Security        |
| Discrete Maths           |
| Fundamentals of database |
+--------------------------+
3 rows in set (0.00 sec)
```

(e) List all books that belong to “CS” department and written by author “Navathe”.



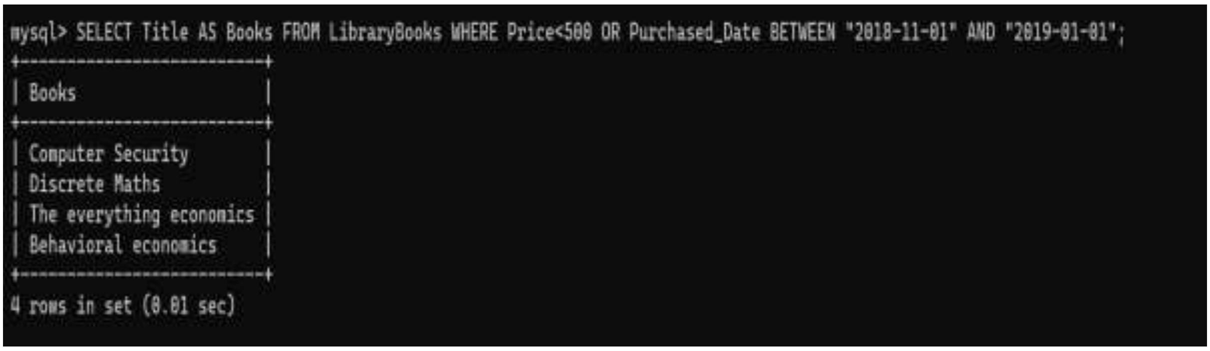
```
mysql> SELECT Title FROM LibraryBooks WHERE Department="CS" AND Author="Navathe";
+--------------------------+
| Title                    |
+--------------------------+
| Fundamentals of database |
+--------------------------+
1 row in set (0.00 sec)
```

(f) List all computer science books that have been issued.



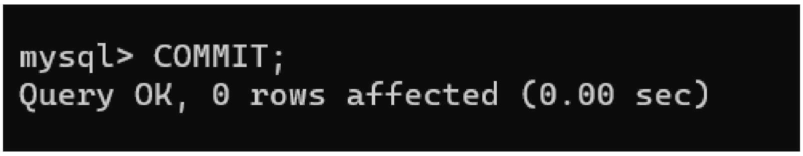
```
mysql> SELECT Title FROM LibraryBooks as L JOIN IssuedBooks as I on L.Accession_Number = I.Accession_Number WHERE Department="CS" ;
+-------------------+
| Title             |
+-------------------+
| Discrete Maths    |
| Computer Security |
+-------------------+
2 rows in set (0.01 sec)
```

(g) List all books which have a price less than 500 or purchased between “01/11/2018” and “01/01/2019”.



```
mysql> SELECT Title AS Books FROM LibraryBooks WHERE Price<500 OR Purchased_Date BETWEEN "2018-11-01" AND "2019-01-01";
+--------------------------+
| Books                    |
+--------------------------+
| Computer Security        |
| Discrete Maths           |
| The everything economics |
| Behavioral economics     |
+--------------------------+
4 rows in set (0.01 sec)
```

***Save all changes, tables and database.**



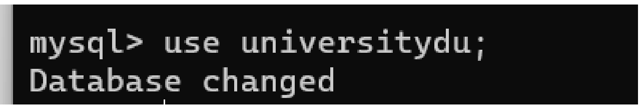
```
mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)
```

UNIVERSITY DATABASE

Create a database having three tables to sore the details of students of Computer Department in your college.

(a)Identify primary and foreign keys. Create tables and insert atleast 5 records in each table.

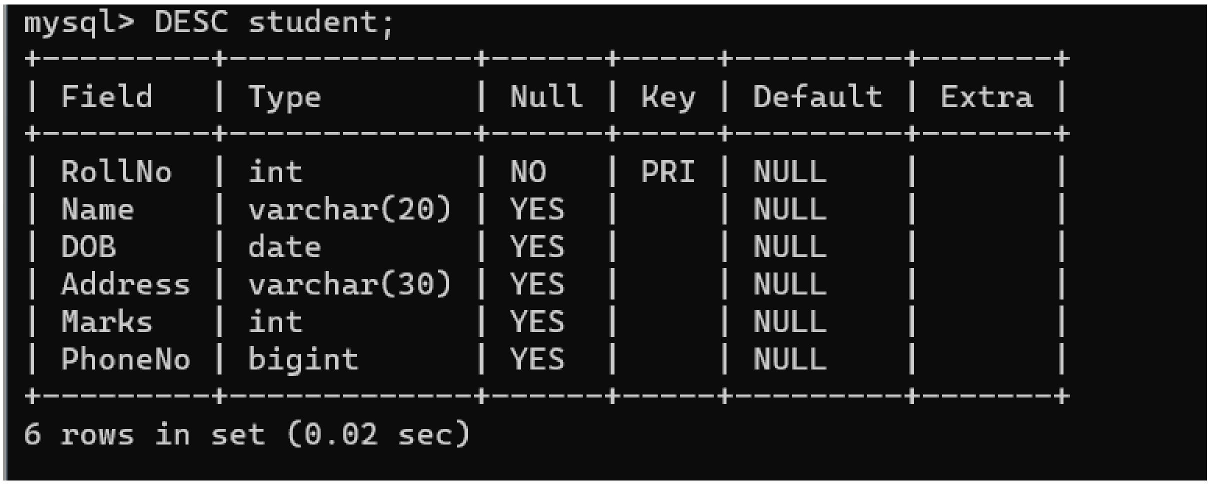
- **Create database University;**



```
mysql> use universitydu;
Database changed
```

- **Create table student→**

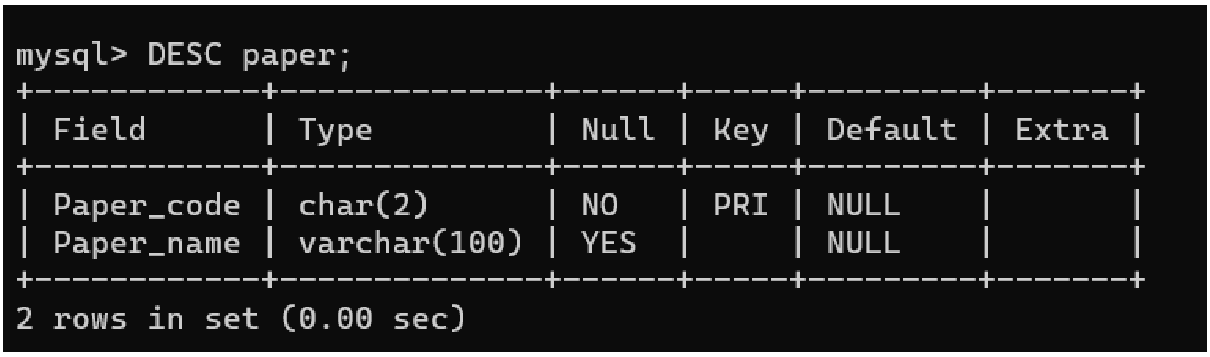
```
CREATE TABLE student
(RollNo INT PRIMARY KEY,
Name VARCHAR(20),
DOB DATE,
Address VARCHAR(40),
Marks INT,
PhoneNo BIGINT);
```



```
mysql> DESC student;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| RollNo  | int         | NO   | PRI | NULL    |       |
| Name    | varchar(20) | YES  |     | NULL    |       |
| DOB     | date        | YES  |     | NULL    |       |
| Address | varchar(30) | YES  |     | NULL    |       |
| Marks   | int         | YES  |     | NULL    |       |
| PhoneNo | bigint      | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
6 rows in set (0.02 sec)
```

- **Create table Paper**

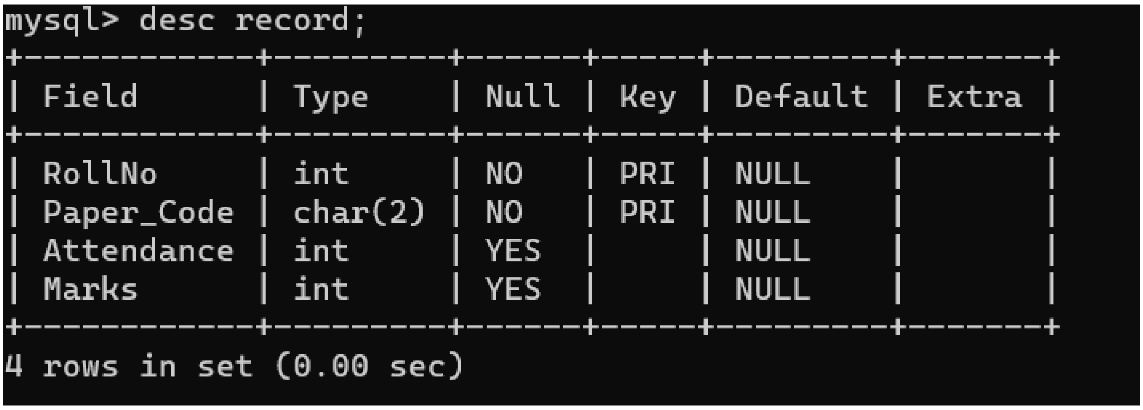
```
CREATE TABLE paper
(Paper_code CHAR(2) PRIMARY KEY,
Paper_name VARCHAR(50));
```



```
mysql> DESC paper;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| Paper_code | char(2)      | NO   | PRI | NULL    |       |
| Paper_name | varchar(100) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

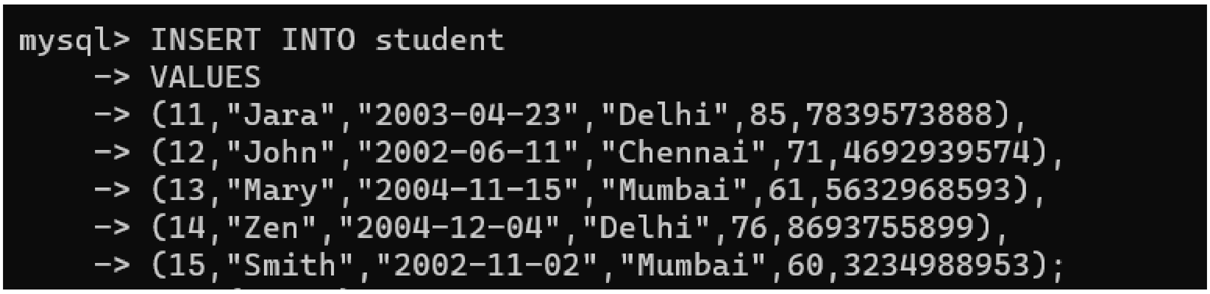
- **Create table Record**

  CREATE TABLE record
  (RollNo INT, Paper_code CHAR(2),
  Attendance INT, Marks INT,
  PRIMARY KEY(RollNo,Paper_code),
  FOREIGN KEY RollNo REFERENCES student(RollNo) ON DELETE CASCADE,
  FOREIGN KEY Paper_code REFERENCES paper(Paper_code) ON DELETE CASCADE);

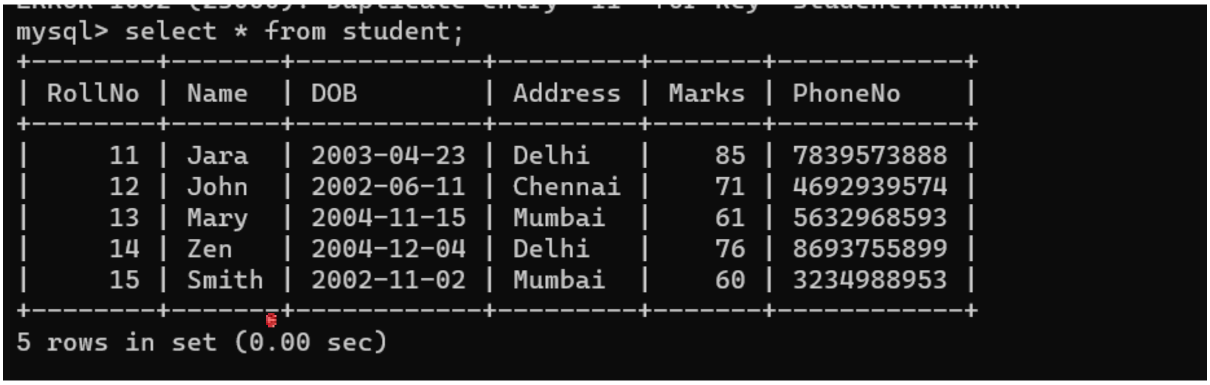


```
mysql> desc record;
+------------+---------+------+-----+---------+-------+
| Field      | Type    | Null | Key | Default | Extra |
+------------+---------+------+-----+---------+-------+
| RollNo     | int     | NO   | PRI | NULL    |       |
| Paper_Code | char(2) | NO   | PRI | NULL    |       |
| Attendance | int     | YES  |     | NULL    |       |
| Marks      | int     | YES  |     | NULL    |       |
+------------+---------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

- **Insert command in student table**

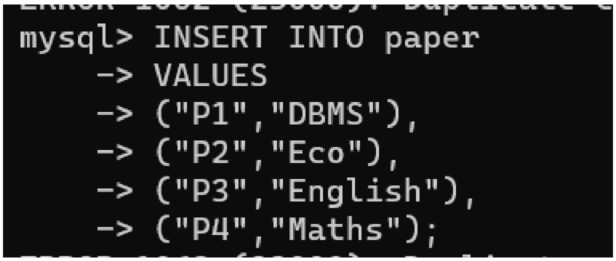


```
mysql> INSERT INTO student
    -> VALUES
    -> (11,"Jara","2003-04-23","Delhi",85,7839573888),
    -> (12,"John","2002-06-11","Chennai",71,4692939574),
    -> (13,"Mary","2004-11-15","Mumbai",61,5632968593),
    -> (14,"Zen","2004-12-04","Delhi",76,8693755899),
    -> (15,"Smith","2002-11-02","Mumbai",60,3234988953);
```

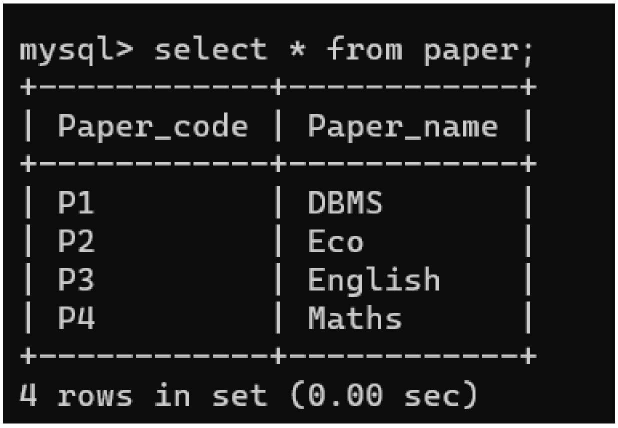


```
mysql> select * from student;
+--------+-------+------------+---------+-------+------------+
| RollNo | Name  | DOB        | Address | Marks | PhoneNo    |
+--------+-------+------------+---------+-------+------------+
|     11 | Jara  | 2003-04-23 | Delhi   |    85 | 7839573888 |
|     12 | John  | 2002-06-11 | Chennai |    71 | 4692939574 |
|     13 | Mary  | 2004-11-15 | Mumbai  |    61 | 5632968593 |
|     14 | Zen   | 2004-12-04 | Delhi   |    76 | 8693755899 |
|     15 | Smith | 2002-11-02 | Mumbai  |    60 | 3234988953 |
+--------+-------+------------+---------+-------+------------+
5 rows in set (0.00 sec)
```

- **Insert command in paper table**

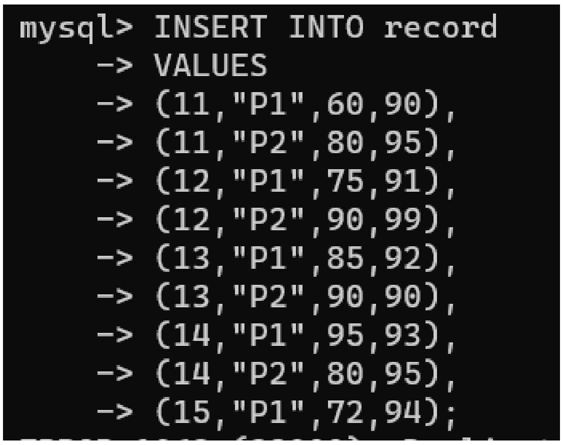


```
mysql> INSERT INTO paper
    -> VALUES
    -> ("P1","DBMS"),
    -> ("P2","Eco"),
    -> ("P3","English"),
    -> ("P4","Maths");
```

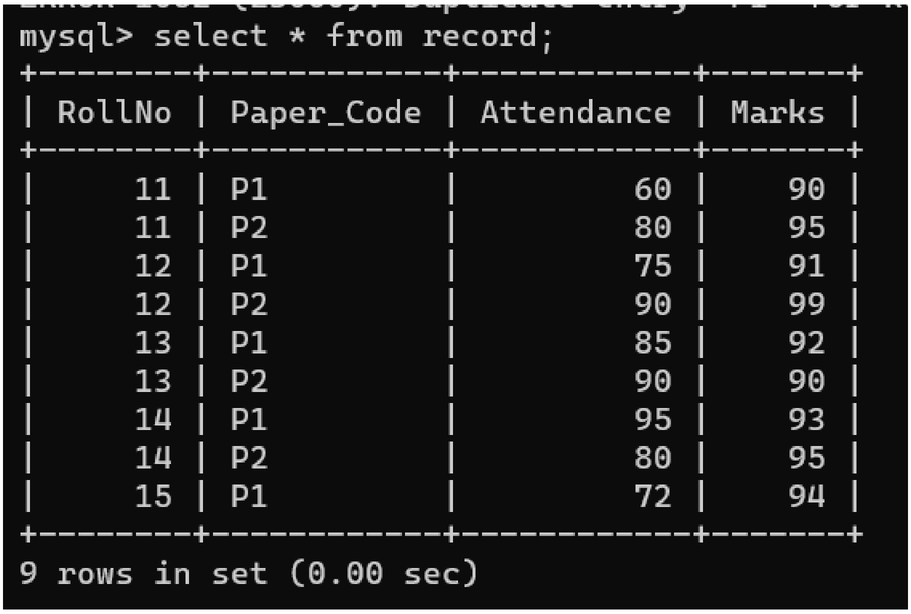


```
mysql> select * from paper;
+------------+------------+
| Paper_code | Paper_name |
+------------+------------+
| P1         | DBMS       |
| P2         | Eco        |
| P3         | English    |
| P4         | Maths      |
+------------+------------+
4 rows in set (0.00 sec)
```

- **Insert command in record table.**

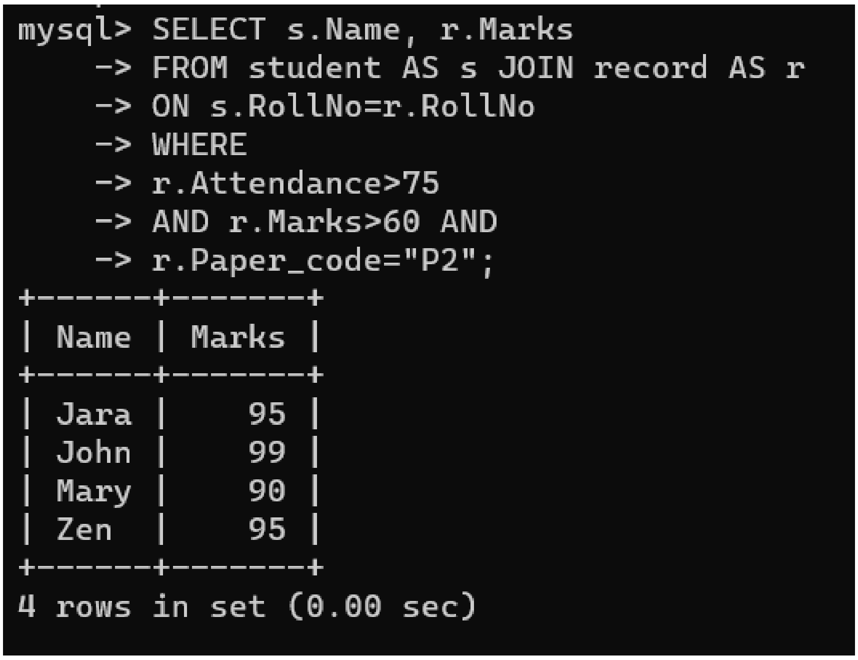


```
mysql> INSERT INTO record
    -> VALUES
    -> (11,"P1",60,90),
    -> (11,"P2",80,95),
    -> (12,"P1",75,91),
    -> (12,"P2",90,99),
    -> (13,"P1",85,92),
    -> (13,"P2",90,90),
    -> (14,"P1",95,93),
    -> (14,"P2",80,95),
    -> (15,"P1",72,94);
```



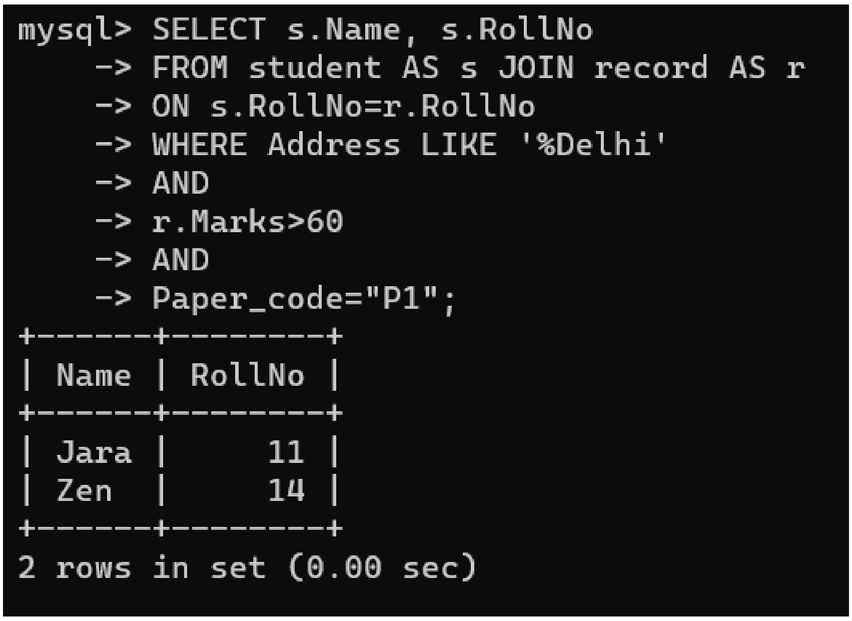
```
mysql> select * from record;
+--------+------------+------------+-------+
| RollNo | Paper_Code | Attendance | Marks |
+--------+------------+------------+-------+
|     11 | P1         |         60 |    90 |
|     11 | P2         |         80 |    95 |
|     12 | P1         |         75 |    91 |
|     12 | P2         |         90 |    99 |
|     13 | P1         |         85 |    92 |
|     13 | P2         |         90 |    90 |
|     14 | P1         |         95 |    93 |
|     14 | P2         |         80 |    95 |
|     15 | P1         |         72 |    94 |
+--------+------------+------------+-------+
9 rows in set (0.00 sec)
```

(b) Design a query that will return the records(from the second table) along with the name of students from the first table related to the students, who have more then 75% attendance and more than 60% marks in paper 2.



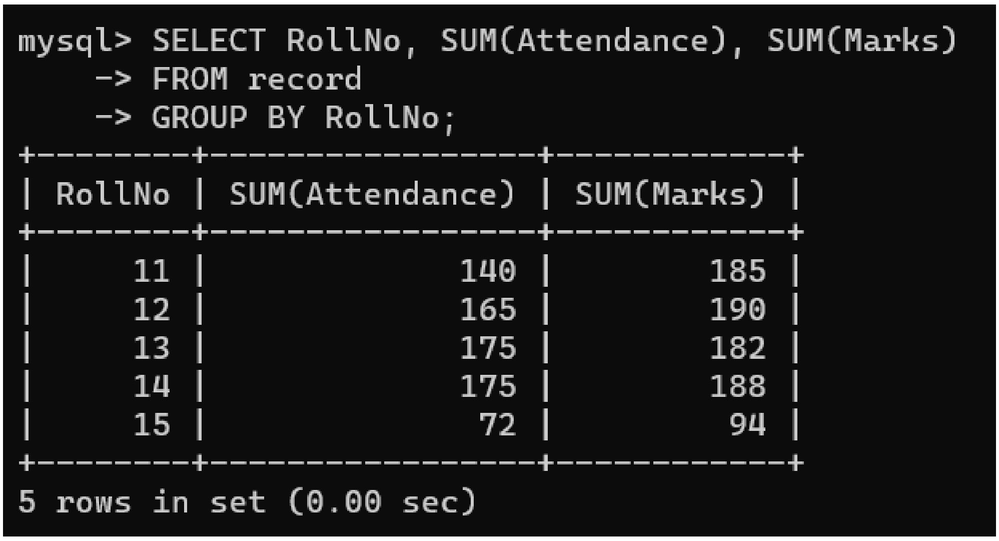
```
mysql> SELECT s.Name, r.Marks
    -> FROM student AS s JOIN record AS r
    -> ON s.RollNo=r.RollNo
    -> WHERE
    -> r.Attendance>75
    -> AND r.Marks>60 AND
    -> r.Paper_code="P2";
+------+-------+
| Name | Marks |
+------+-------+
| Jara |    95 |
| John |    99 |
| Mary |    90 |
| Zen  |    95 |
+------+-------+
4 rows in set (0.00 sec)
```

(c) List all students who live in “Delhi” and have marks greater than 60 in paper 1.



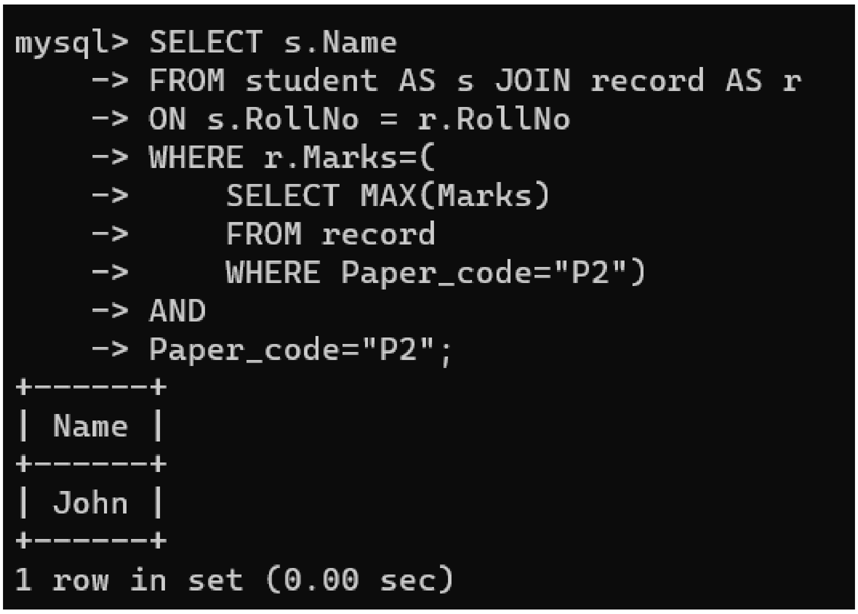
```
mysql> SELECT s.Name, s.RollNo
    -> FROM student AS s JOIN record AS r
    -> ON s.RollNo=r.RollNo
    -> WHERE Address LIKE '%Delhi'
    -> AND
    -> r.Marks>60
    -> AND
    -> Paper_code="P1";
+------+--------+
| Name | RollNo |
+------+--------+
| Jara |     11 |
| Zen  |     14 |
+------+--------+
2 rows in set (0.00 sec)
```

(d) Find total attendance and total marks attained by each student.



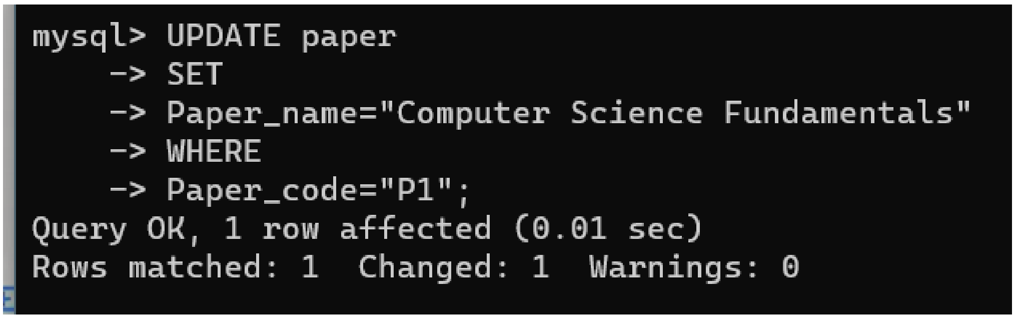
```
mysql> SELECT RollNo, SUM(Attendance), SUM(Marks)
    -> FROM record
    -> GROUP BY RollNo;
+--------+-----------------+------------+
| RollNo | SUM(Attendance) | SUM(Marks) |
+--------+-----------------+------------+
|     11 |             140 |        185 |
|     12 |             165 |        190 |
|     13 |             175 |        182 |
|     14 |             175 |        188 |
|     15 |              72 |         94 |
+--------+-----------------+------------+
5 rows in set (0.00 sec)
```

(e) List the name of student who has got the highest marks in paper 2.



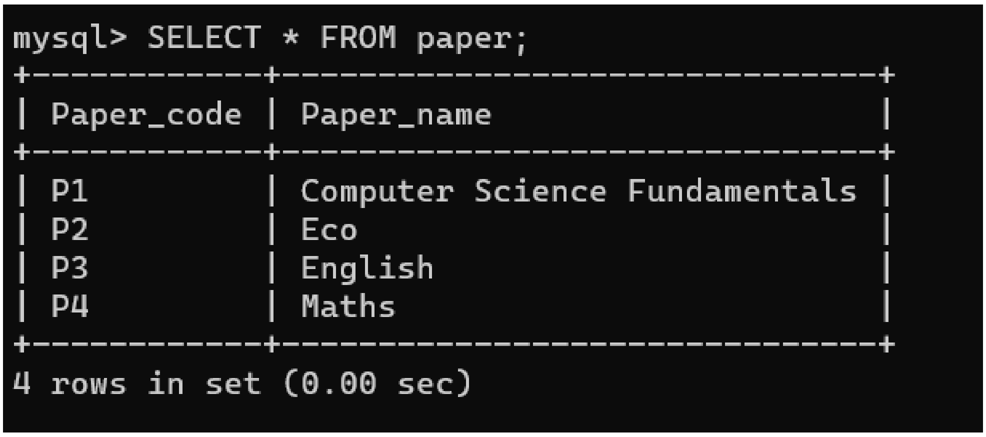
```
mysql> SELECT s.Name
    -> FROM student AS s JOIN record AS r
    -> ON s.RollNo = r.RollNo
    -> WHERE r.Marks=(
    ->     SELECT MAX(Marks)
    ->     FROM record
    ->     WHERE Paper_code="P2")
    -> AND
    -> Paper_code="P2";
+-------+
| Name  |
+-------+
| John  |
+-------+
1 row in set (0.00 sec)
```

(f)Update the name of paper with paper code P1 from DBMS to “Computer Science Fundamentals”.



```
mysql> UPDATE paper
    -> SET
    -> Paper_name="Computer Science Fundamentals"
    -> WHERE
    -> Paper_code="P1";
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

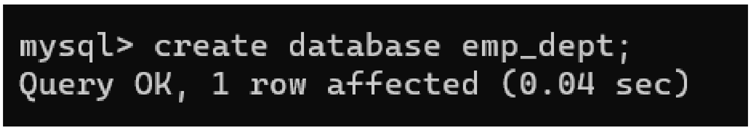
Table after updating



```
mysql> SELECT * FROM paper;
+------------+-------------------------------+
| Paper_code | Paper_name                    |
+------------+-------------------------------+
| P1         | Computer Science Fundamentals |
| P2         | Eco                           |
| P3         | English                       |
| P4         | Maths                         |
+------------+-------------------------------+
4 rows in set (0.00 sec)
```

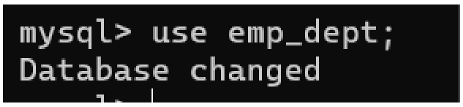
COMPANY DATABASE

Creating Database emp_dept.



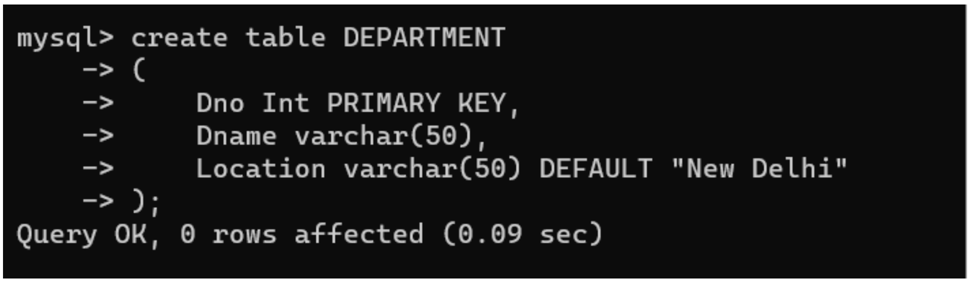
```
mysql> create database emp_dept;
Query OK, 1 row affected (0.04 sec)
```

Using Database emp_dept.



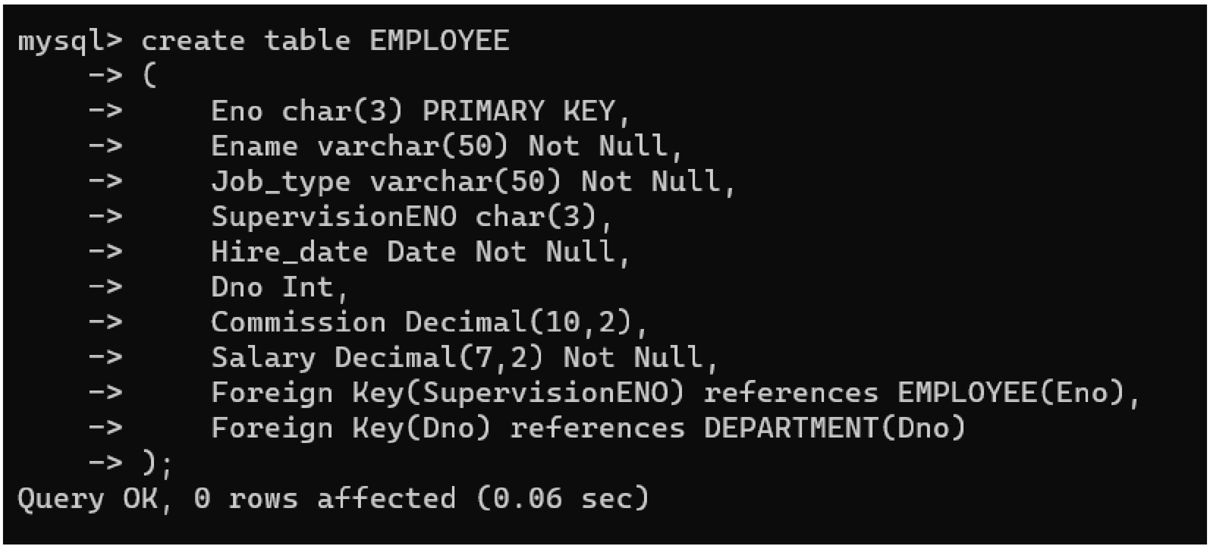
```
mysql> use emp_dept;
Database changed
```

Creation of DEPARTMENT table.



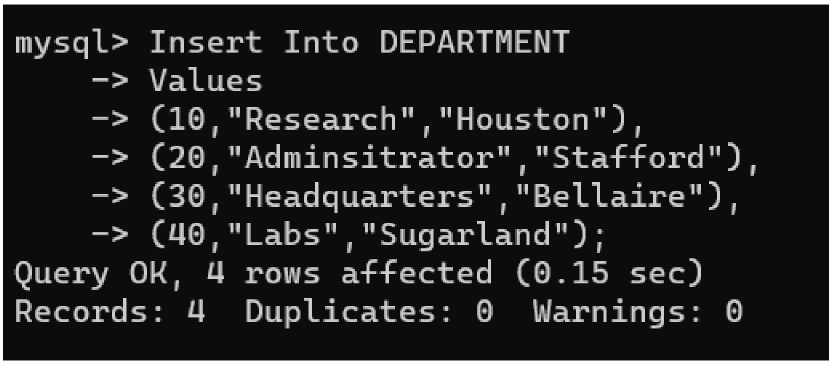
```
mysql> create table DEPARTMENT
    -> (
    ->     Dno Int PRIMARY KEY,
    ->     Dname varchar(50),
    ->     Location varchar(50) DEFAULT "New Delhi"
    -> );
Query OK, 0 rows affected (0.09 sec)
```

Creation of EMPLOYEE table.



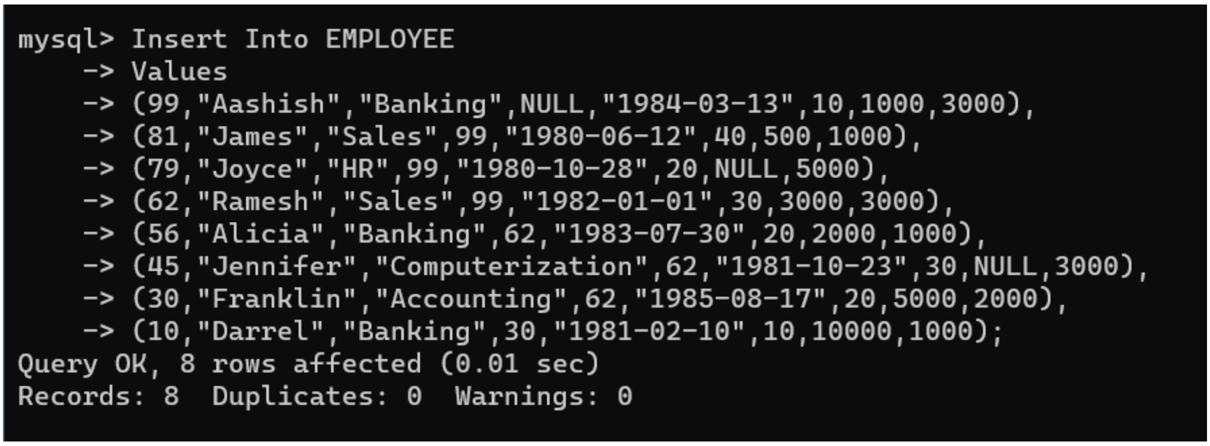
```
mysql> create table EMPLOYEE
    -> (
    ->     Eno char(3) PRIMARY KEY,
    ->     Ename varchar(50) Not Null,
    ->     Job_type varchar(50) Not Null,
    ->     SupervisionENO char(3),
    ->     Hire_date Date Not Null,
    ->     Dno Int,
    ->     Commission Decimal(10,2),
    ->     Salary Decimal(7,2) Not Null,
    ->     Foreign Key(SupervisionENO) references EMPLOYEE(Eno),
    ->     Foreign Key(Dno) references DEPARTMENT(Dno)
    -> );
Query OK, 0 rows affected (0.06 sec)
```

Inserting Data in DEPARTMENT TABLE.



```
mysql> Insert Into DEPARTMENT
    -> Values
    -> (10,"Research","Houston"),
    -> (20,"Adminsitrator","Stafford"),
    -> (30,"Headquarters","Bellaire"),
    -> (40,"Labs","Sugarland");
Query OK, 4 rows affected (0.15 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

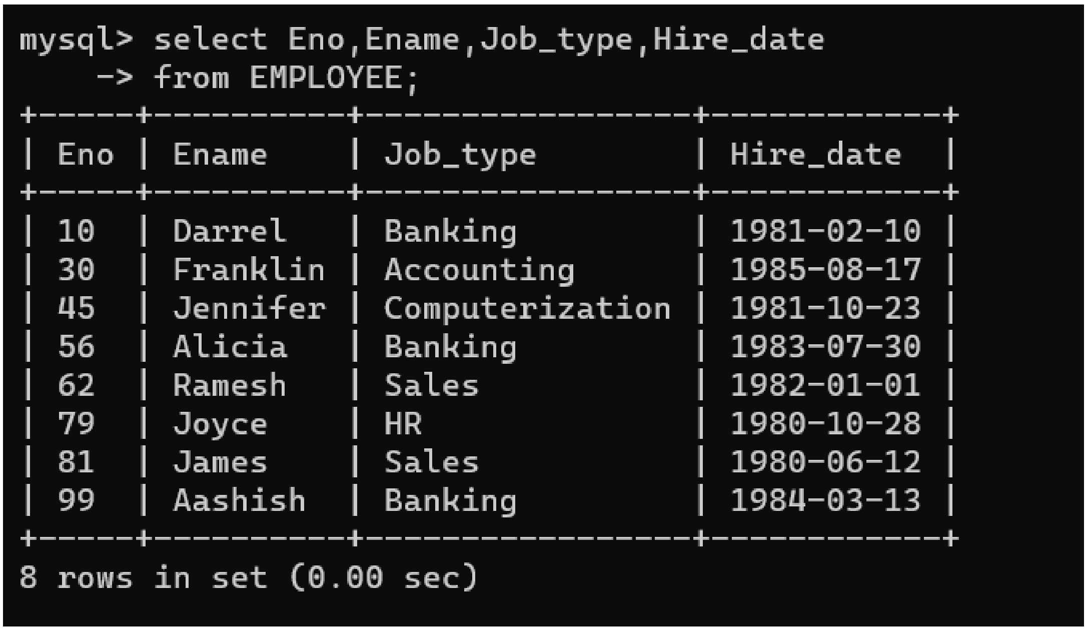
Inserting Data in EMPLOYEE TABLE.



```
mysql> Insert Into EMPLOYEE
    -> Values
    -> (99,"Aashish","Banking",NULL,"1984-03-13",10,1000,3000),
    -> (81,"James","Sales",99,"1980-06-12",40,500,1000),
    -> (79,"Joyce","HR",99,"1980-10-28",20,NULL,5000),
    -> (62,"Ramesh","Sales",99,"1982-01-01",30,3000,3000),
    -> (56,"Alicia","Banking",62,"1983-07-30",20,2000,1000),
    -> (45,"Jennifer","Computerization",62,"1981-10-23",30,NULL,3000),
    -> (30,"Franklin","Accounting",62,"1985-08-17",20,5000,2000),
    -> (10,"Darrel","Banking",30,"1981-02-10",10,10000,1000);
Query OK, 8 rows affected (0.01 sec)
Records: 8  Duplicates: 0  Warnings: 0
```

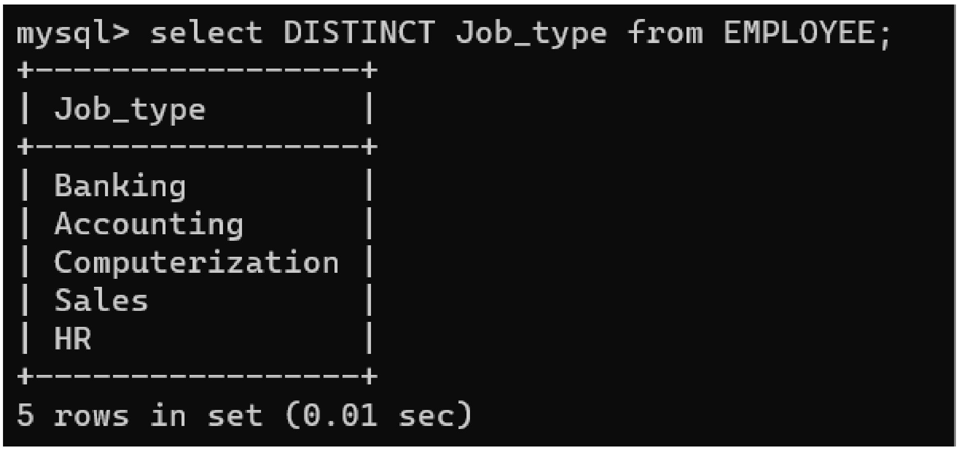
PRACTICAL QUESTIONS

1. **Query to display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first.**



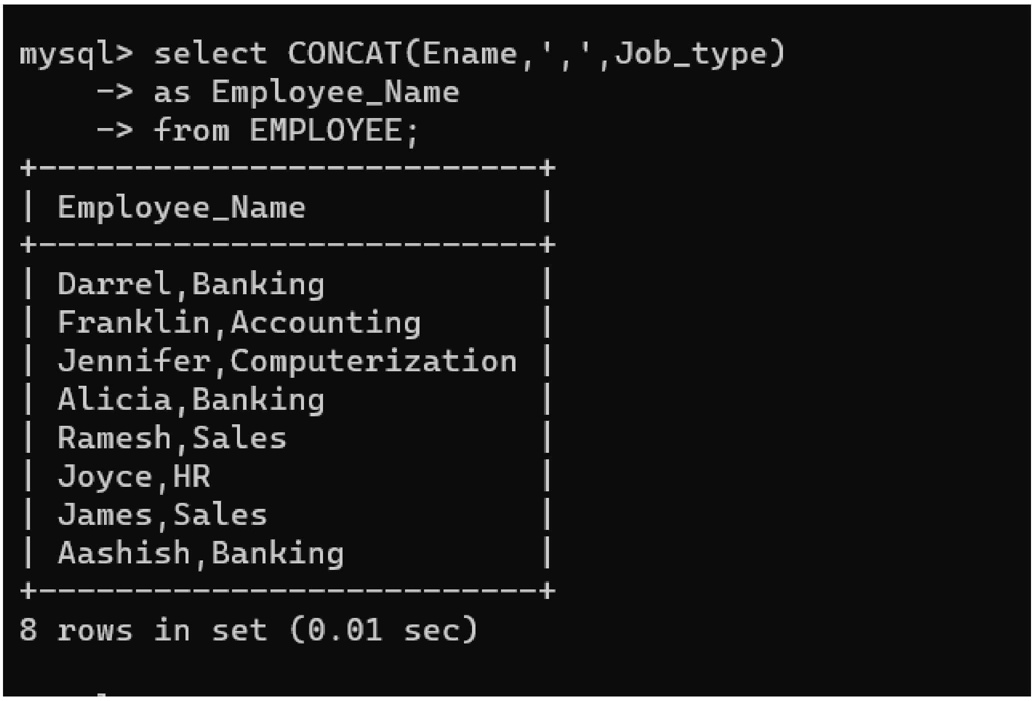
```
mysql> select Eno,Ename,Job_type,Hire_date
    -> from EMPLOYEE;
+-----+----------+-----------------+------------+
| Eno | Ename    | Job_type        | Hire_date  |
+-----+----------+-----------------+------------+
| 10  | Darrel   | Banking         | 1981-02-10 |
| 30  | Franklin | Accounting      | 1985-08-17 |
| 45  | Jennifer | Computerization | 1981-10-23 |
| 56  | Alicia   | Banking         | 1983-07-30 |
| 62  | Ramesh   | Sales           | 1982-01-01 |
| 79  | Joyce    | HR              | 1980-10-28 |
| 81  | James    | Sales           | 1980-06-12 |
| 99  | Aashish  | Banking         | 1984-03-13 |
+-----+----------+-----------------+------------+
8 rows in set (0.00 sec)
```

2. **Query to display unique Jobs from the Employee Table.**



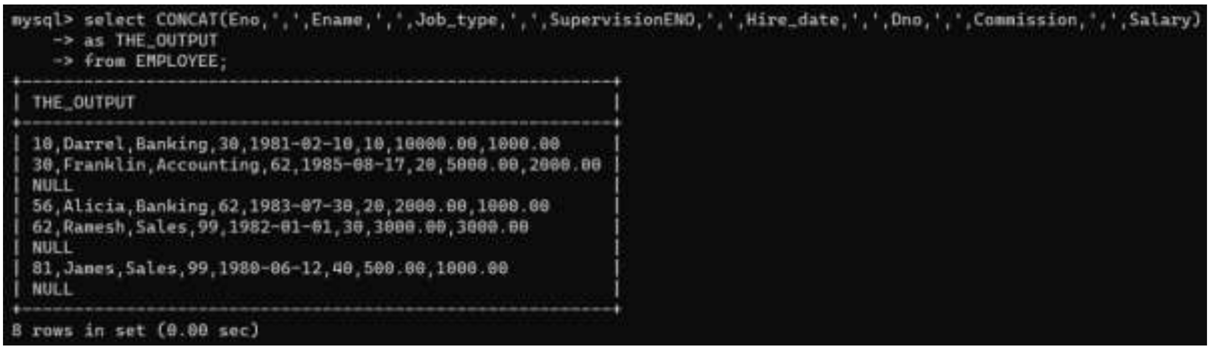
```
mysql> select DISTINCT Job_type from EMPLOYEE;
+------------------+
| Job_type         |
+------------------+
| Banking          |
| Accounting       |
| Computerization  |
| Sales            |
| HR               |
+------------------+
5 rows in set (0.01 sec)
```

3. **Query to display the Employee Name concatenated by a Job separated by a comma.**



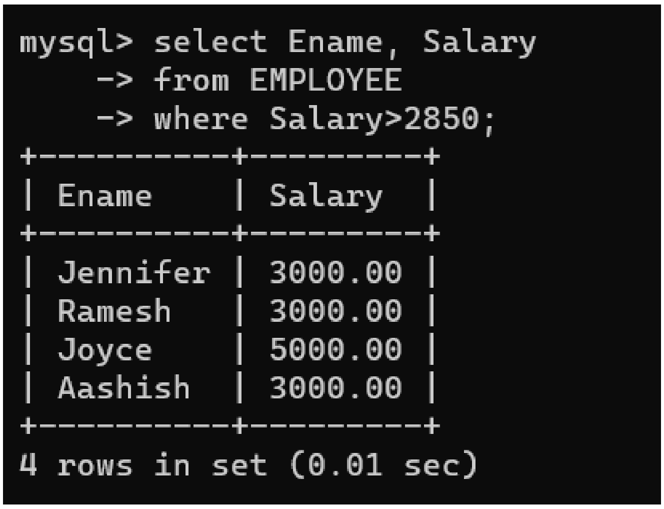
```
mysql> select CONCAT(Ename,',',Job_type)
    -> as Employee_Name
    -> from EMPLOYEE;
+--------------------------+
| Employee_Name            |
+--------------------------+
| Darrel,Banking           |
| Franklin,Accounting      |
| Jennifer,Computerization |
| Alicia,Banking           |
| Ramesh,Sales             |
| Joyce,HR                 |
| James,Sales              |
| Aashish,Banking          |
+--------------------------+
8 rows in set (0.01 sec)
```

4. **Query to display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE_OUTPUT.**



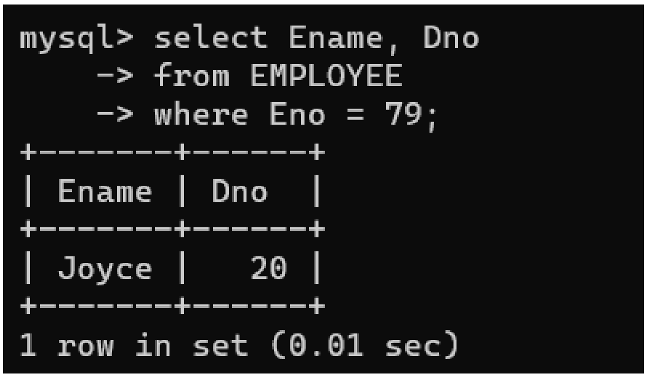
```
mysql> select CONCAT(Eno,',',Ename,',',Job_type,',',SupervisionENO,',',Hire_date,',',Dno,',',Commission,',',Salary)
    -> as THE_OUTPUT
    -> from EMPLOYEE;
+----------------------------------------------------------+
| THE_OUTPUT                                               |
+----------------------------------------------------------+
| 10,Darrel,Banking,30,1981-02-10,10,10000.00,1000.00      |
| 30,Franklin,Accounting,62,1985-08-17,20,5000.00,2000.00  |
| NULL                                                     |
| 56,Alicia,Banking,62,1983-07-30,20,2000.00,1000.00       |
| 62,Ramesh,Sales,99,1982-01-01,30,3000.00,3000.00         |
| NULL                                                     |
| 81,James,Sales,99,1980-06-12,40,500.00,1000.00           |
| NULL                                                     |
+----------------------------------------------------------+
8 rows in set (0.00 sec)
```

5. **Query to display the Employee Name and Salary of all the employees earning more than $2850.**



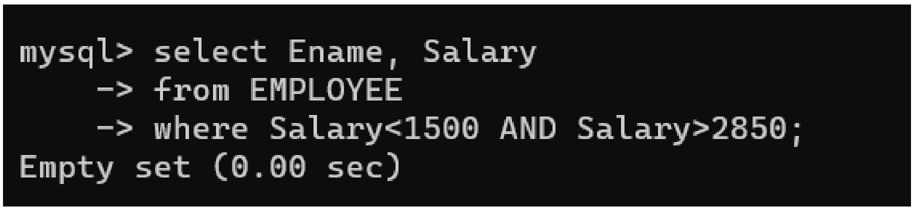
```
mysql> select Ename, Salary
    -> from EMPLOYEE
    -> where Salary>2850;
+----------+---------+
| Ename    | Salary  |
+----------+---------+
| Jennifer | 3000.00 |
| Ramesh   | 3000.00 |
| Joyce    | 5000.00 |
| Aashish  | 3000.00 |
+----------+---------+
4 rows in set (0.01 sec)
```

6. **Query to display Employee Name and Department Number for the Employee No= 7900.**



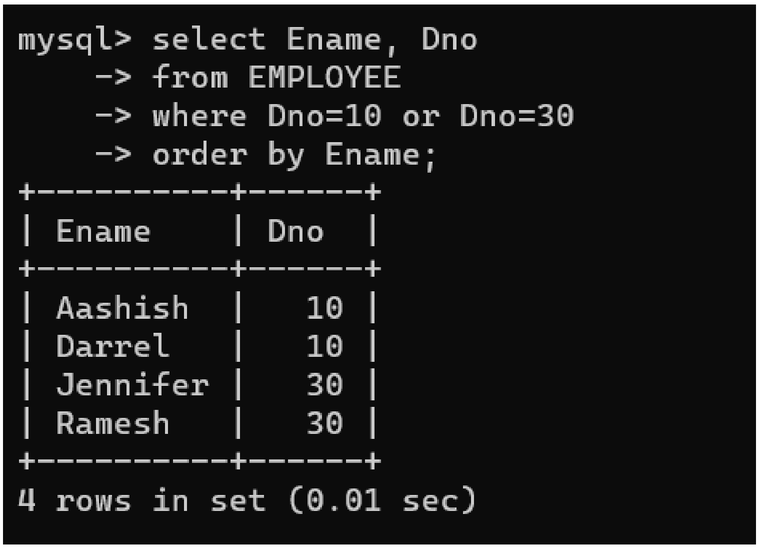
```
mysql> select Ename, Dno
    -> from EMPLOYEE
    -> where Eno = 79;
+-------+------+
| Ename | Dno  |
+-------+------+
| Joyce |   20 |
+-------+------+
1 row in set (0.01 sec)
```

7. **Query to display Employee Name and Salary for all employees whose salary is not in the range of $1500 and $2850.**



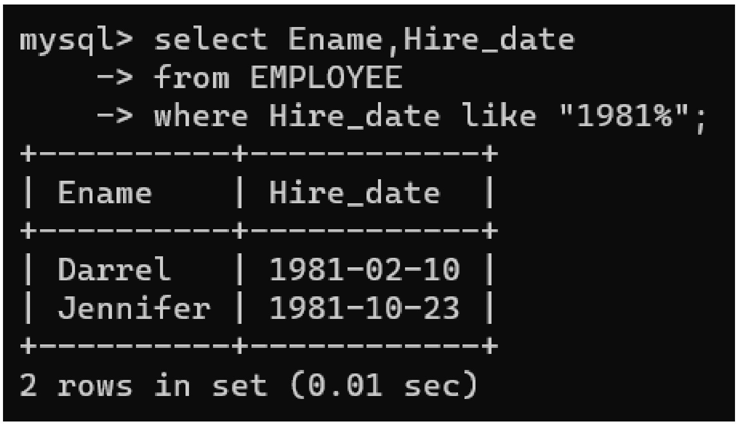
```
mysql> select Ename, Salary
    -> from EMPLOYEE
    -> where Salary<1500 AND Salary>2850;
Empty set (0.00 sec)
```

8. **Query to display Employee Name and Department No. of all the employees in Dept 10 and Dept 30 in the alphabetical order by name.**



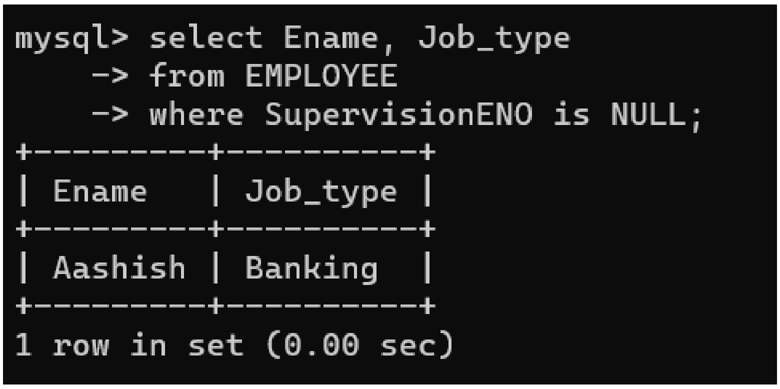
```
mysql> select Ename, Dno
    -> from EMPLOYEE
    -> where Dno=10 or Dno=30
    -> order by Ename;
+----------+------+
| Ename    | Dno  |
+----------+------+
| Aashish  |   10 |
| Darrel   |   10 |
| Jennifer |   30 |
| Ramesh   |   30 |
+----------+------+
4 rows in set (0.01 sec)
```

9. **Query to display Name and Hire Date of every Employee who was hired in 1981.**



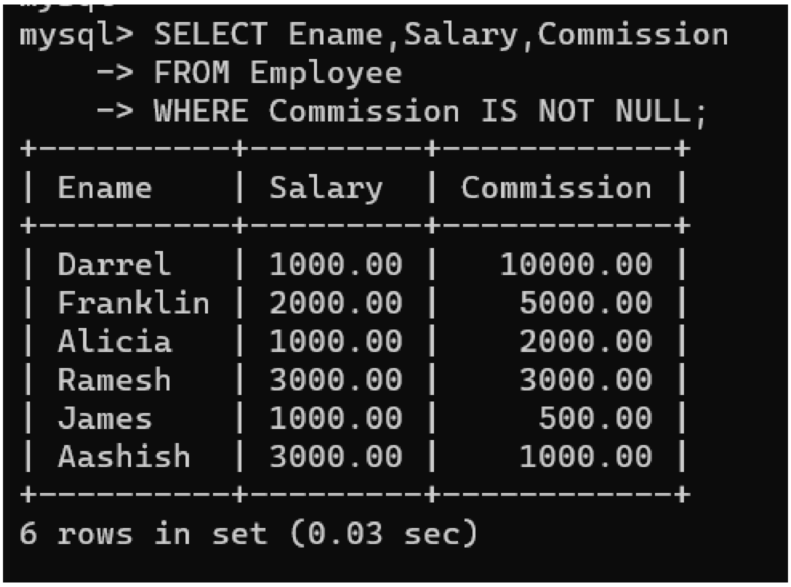
```
mysql> select Ename,Hire_date
    -> from EMPLOYEE
    -> where Hire_date like "1981%";
+----------+------------+
| Ename    | Hire_date  |
+----------+------------+
| Darrel   | 1981-02-10 |
| Jennifer | 1981-10-23 |
+----------+------------+
2 rows in set (0.01 sec)
```

10. **Query to display Name and Job of all employees who don’t have a current Manager.**



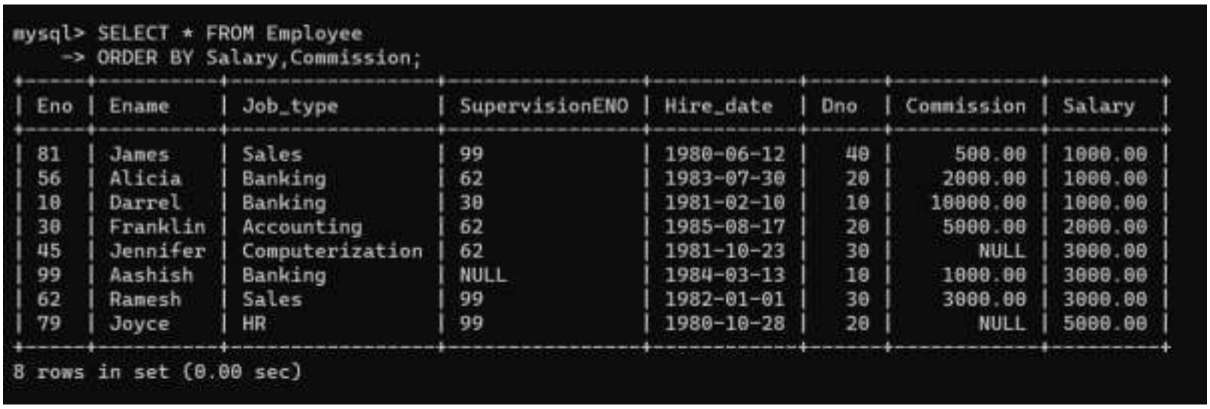
```
mysql> select Ename, Job_type
    -> from EMPLOYEE
    -> where SupervisionENO is NULL;
+---------+----------+
| Ename   | Job_type |
+---------+----------+
| Aashish | Banking  |
+---------+----------+
1 row in set (0.00 sec)
```

11. Query to display the Name, Salary and Commission for all the employees who earn commission.



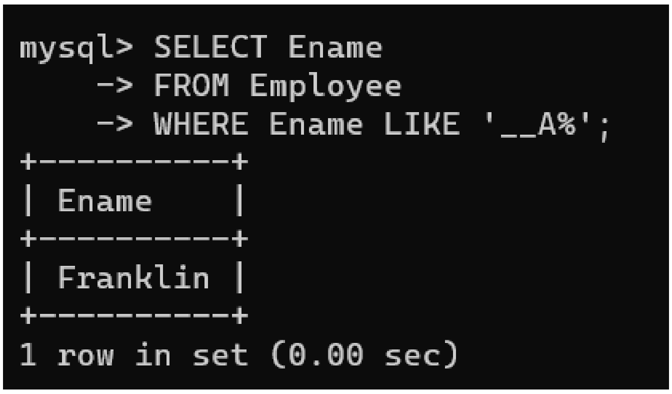
```
mysql> SELECT Ename,Salary,Commission
    -> FROM Employee
    -> WHERE Commission IS NOT NULL;
+----------+---------+------------+
| Ename    | Salary  | Commission |
+----------+---------+------------+
| Darrel   | 1000.00 |   10000.00 |
| Franklin | 2000.00 |    5000.00 |
| Alicia   | 1000.00 |    2000.00 |
| Ramesh   | 3000.00 |    3000.00 |
| James    | 1000.00 |     500.00 |
| Aashish  | 3000.00 |    1000.00 |
+----------+---------+------------+
6 rows in set (0.03 sec)
```

12. Sort the data in descending order of Salary and Commission.



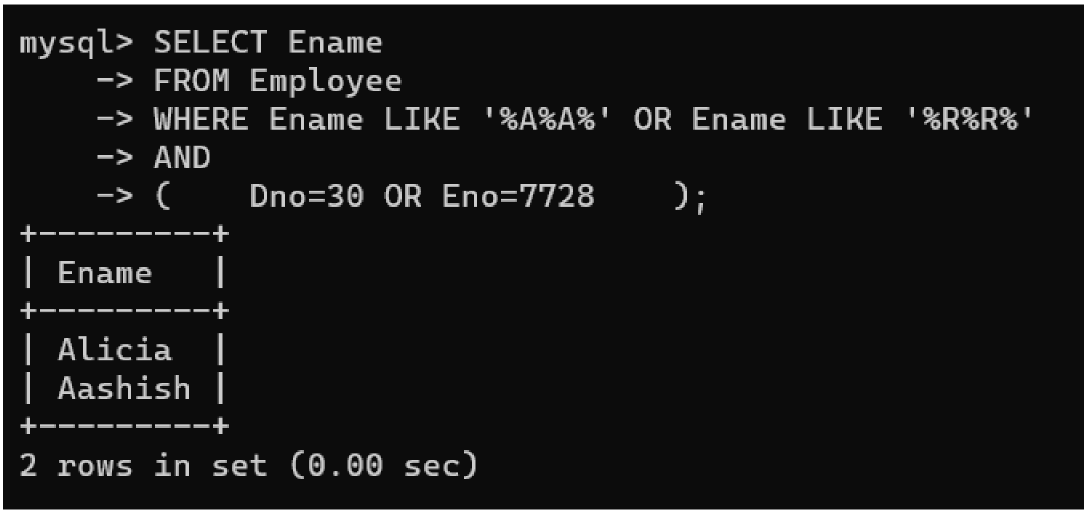
```
mysql> SELECT * FROM Employee
    -> ORDER BY Salary,Commission;
+-----+----------+------------------+---------------+------------+-----+------------+---------+
| Eno | Ename    | Job_type         | SupervisionENO | Hire_date  | Dno | Commission | Salary  |
+-----+----------+------------------+---------------+------------+-----+------------+---------+
| 81  | James    | Sales            | 99            | 1980-06-12 |  40 |     500.00 | 1000.00 |
| 56  | Alicia   | Banking          | 62            | 1983-07-30 |  20 |    2000.00 | 1000.00 |
| 10  | Darrel   | Banking          | 30            | 1981-02-10 |  10 |   10000.00 | 1000.00 |
| 30  | Franklin | Accounting       | 62            | 1985-08-17 |  20 |    5000.00 | 2000.00 |
| 45  | Jennifer | Computerization  | 62            | 1981-10-23 |  30 |       NULL | 3000.00 |
| 99  | Aashish  | Banking          | NULL          | 1984-03-13 |  10 |    1000.00 | 3000.00 |
| 62  | Ramesh   | Sales            | 99            | 1982-01-01 |  30 |    3000.00 | 3000.00 |
| 79  | Joyce    | HR               | 99            | 1980-10-28 |  20 |       NULL | 5000.00 |
+-----+----------+------------------+---------------+------------+-----+------------+---------+
8 rows in set (0.00 sec)
```

13. Query to display Name of all the employees where the third letter of their name is ‘A’.



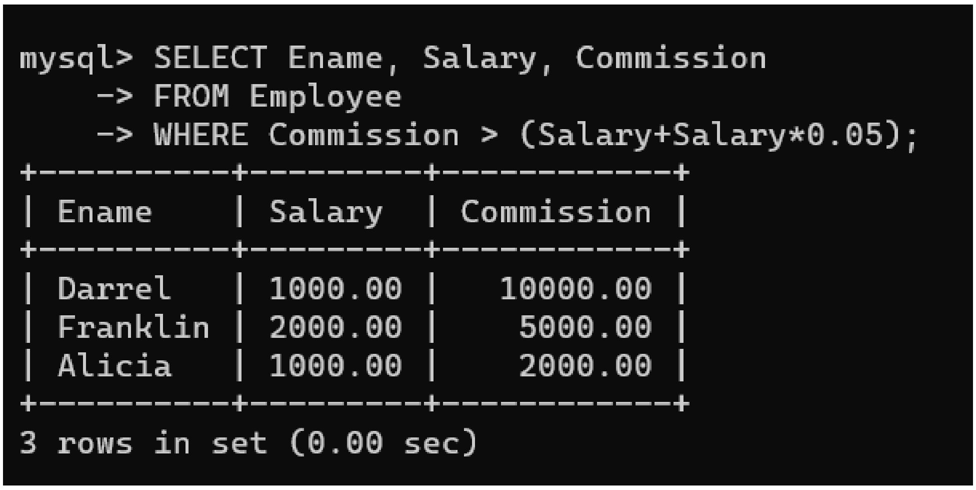
```
mysql> SELECT Ename
    -> FROM Employee
    -> WHERE Ename LIKE '__A%';
+----------+
| Ename    |
+----------+
| Franklin |
+----------+
1 row in set (0.00 sec)
```

14. Query to display Name of all employees either have two ‘R’s or have two ‘A’s in their name and are either in Dept No = 30 or their Manger’s Employee No = 7788.



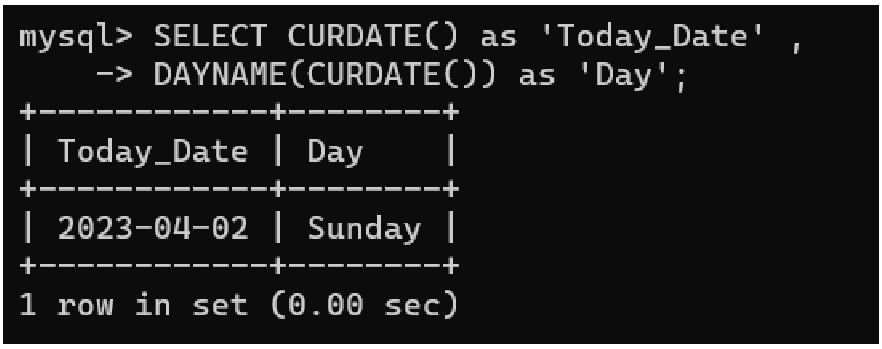
```
mysql> SELECT Ename
    -> FROM Employee
    -> WHERE Ename LIKE '%A%A%' OR Ename LIKE '%R%R%'
    -> AND
    -> (    Dno=30 OR Eno=7728    );
+----------+
| Ename    |
+----------+
| Alicia   |
| Aashish  |
+----------+
2 rows in set (0.00 sec)
```

15. Query to display Name, Salary and Commission for all employees whose Commission amount is greater than their Salary increased by 5%.



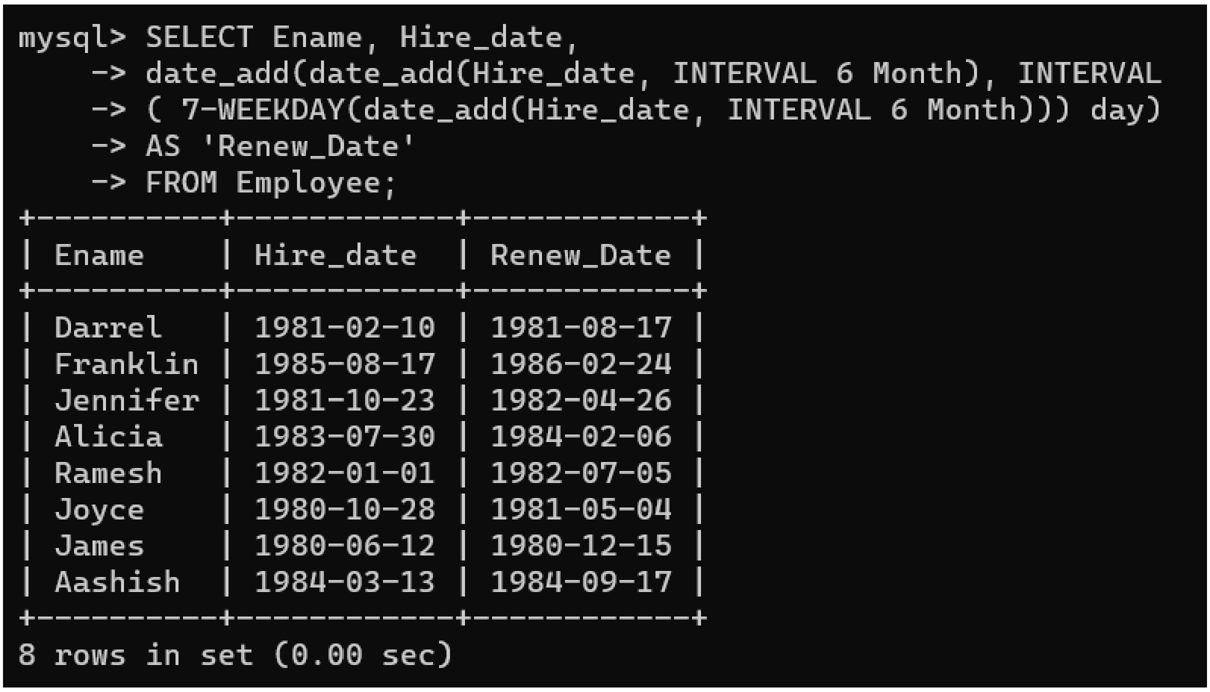
```
mysql> SELECT Ename, Salary, Commission
    -> FROM Employee
    -> WHERE Commission > (Salary+Salary*0.05);
+----------+---------+------------+
| Ename    | Salary  | Commission |
+----------+---------+------------+
| Darrel   | 1000.00 |   10000.00 |
| Franklin | 2000.00 |    5000.00 |
| Alicia   | 1000.00 |    2000.00 |
+----------+---------+------------+
3 rows in set (0.00 sec)
```

16. Query to display the Current Date along with the day name.



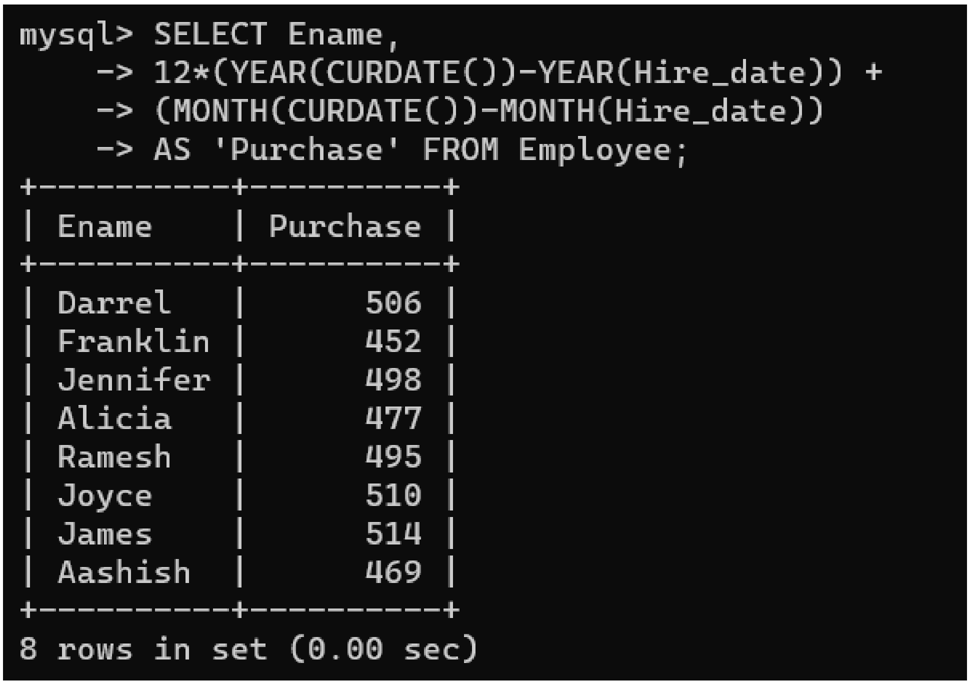
```
mysql> SELECT CURDATE() as 'Today_Date' ,
    -> DAYNAME(CURDATE()) as 'Day';
+------------+--------+
| Today_Date | Day    |
+------------+--------+
| 2023-04-02 | Sunday |
+------------+--------+
1 row in set (0.00 sec)
```

17. Query to display Name, Hire Date and Salary Review Date which is the 1st Monday after six months of employment.



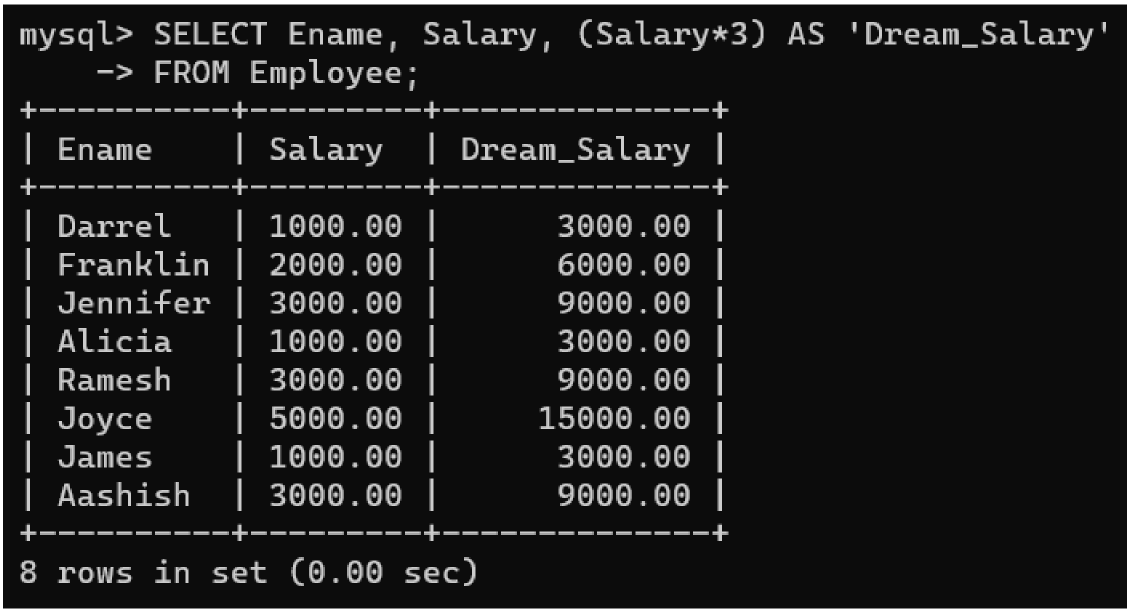
```
mysql> SELECT Ename, Hire_date,
    -> date_add(date_add(Hire_date, INTERVAL 6 Month), INTERVAL
    -> ( 7-WEEKDAY(date_add(Hire_date, INTERVAL 6 Month))) day)
    -> AS 'Renew_Date'
    -> FROM Employee;
+----------+------------+------------+
| Ename    | Hire_date  | Renew_Date |
+----------+------------+------------+
| Darrel   | 1981-02-10 | 1981-08-17 |
| Franklin | 1985-08-17 | 1986-02-24 |
| Jennifer | 1981-10-23 | 1982-04-26 |
| Alicia   | 1983-07-30 | 1984-02-06 |
| Ramesh   | 1982-01-01 | 1982-07-05 |
| Joyce    | 1980-10-28 | 1981-05-04 |
| James    | 1980-06-12 | 1980-12-15 |
| Aashish  | 1984-03-13 | 1984-09-17 |
+----------+------------+------------+
8 rows in set (0.00 sec)
```

18. Query to display Name and calculate the number of months between today and the date on which employee was hired of department 'Purchase'.



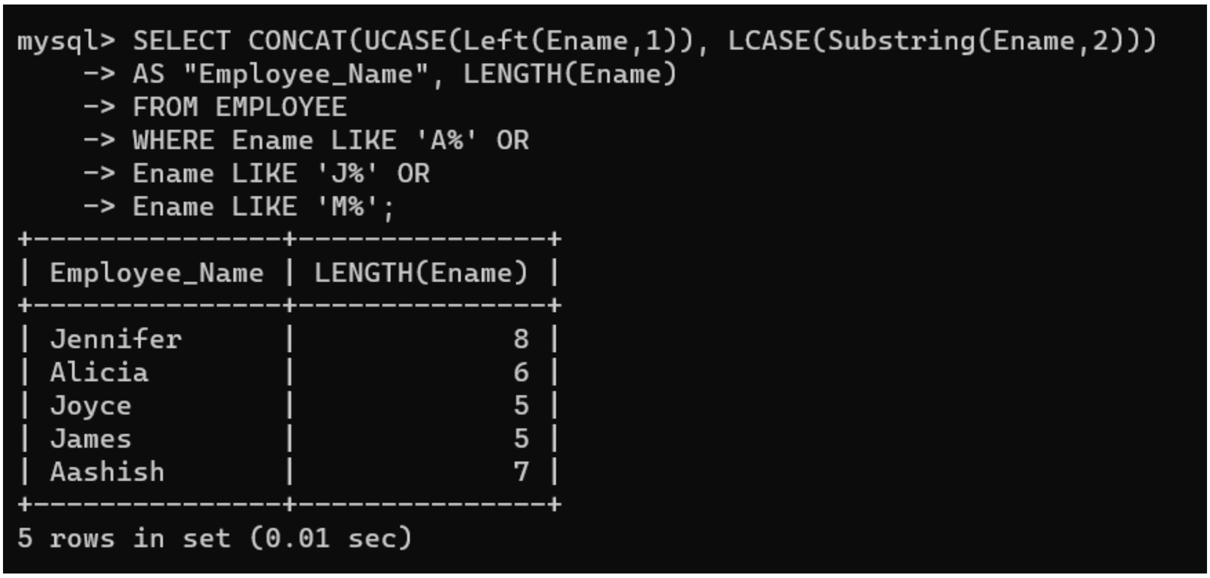
```
mysql> SELECT Ename,
    -> 12*(YEAR(CURDATE())-YEAR(Hire_date)) +
    -> (MONTH(CURDATE())-MONTH(Hire_date))
    -> AS 'Purchase' FROM Employee;
+----------+----------+
| Ename    | Purchase |
+----------+----------+
| Darrel   |      506 |
| Franklin |      452 |
| Jennifer |      498 |
| Alicia   |      477 |
| Ramesh   |      495 |
| Joyce    |      510 |
| James    |      514 |
| Aashish  |      469 |
+----------+----------+
8 rows in set (0.00 sec)
```

19. Query to display the following for each employee earns < Salary> monthly but wants < 3 * Current Salary >. Label the Column as Dream Salary.



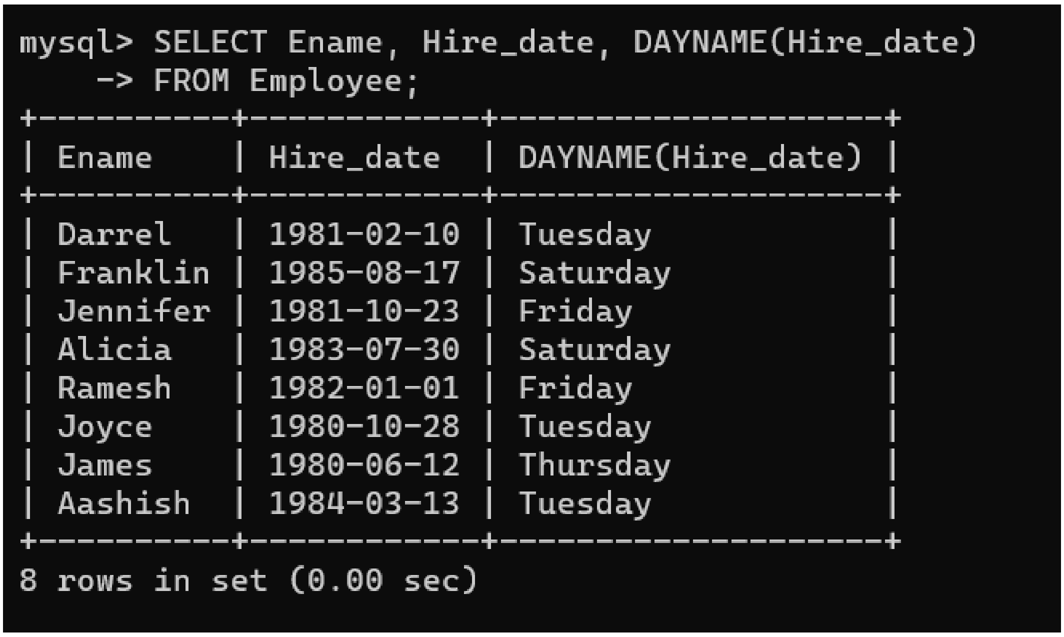
```
mysql> SELECT Ename, Salary, (Salary*3) AS 'Dream_Salary'
    -> FROM Employee;
+----------+---------+--------------+
| Ename    | Salary  | Dream_Salary |
+----------+---------+--------------+
| Darrel   | 1000.00 |      3000.00 |
| Franklin | 2000.00 |      6000.00 |
| Jennifer | 3000.00 |      9000.00 |
| Alicia   | 1000.00 |      3000.00 |
| Ramesh   | 3000.00 |      9000.00 |
| Joyce    | 5000.00 |     15000.00 |
| James    | 1000.00 |      3000.00 |
| Aashish  | 3000.00 |      9000.00 |
+----------+---------+--------------+
8 rows in set (0.00 sec)
```

20. Query to display Name with the 1st letter capitalized and all other letter lower case and length of their name of all the employees whose name starts with 'J', 'A' and 'M'.



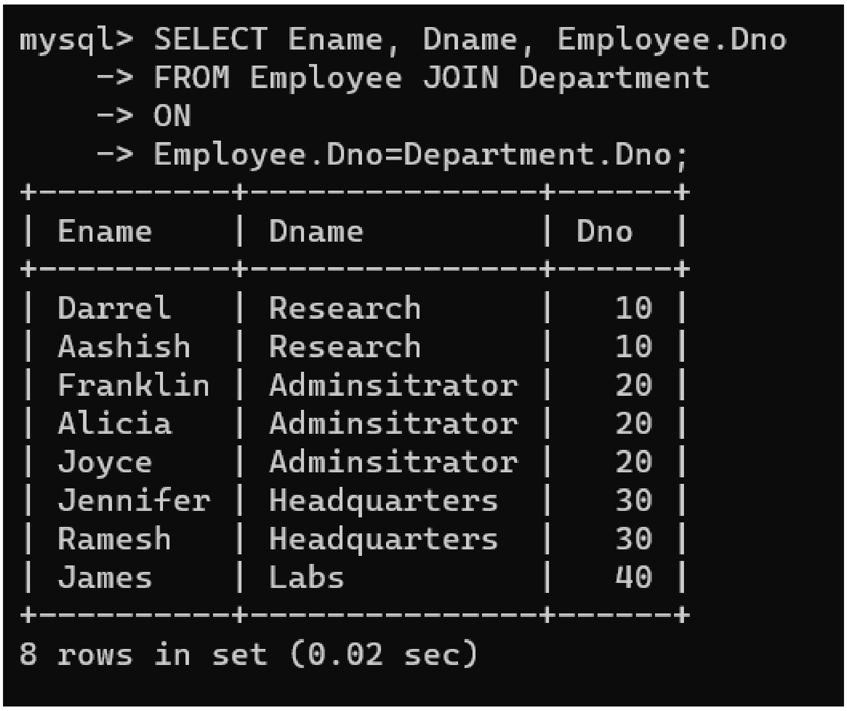
```
mysql> SELECT CONCAT(UCASE(Left(Ename,1)), LCASE(Substring(Ename,2)))
    -> AS "Employee_Name", LENGTH(Ename)
    -> FROM EMPLOYEE
    -> WHERE Ename LIKE 'A%' OR
    -> Ename LIKE 'J%' OR
    -> Ename LIKE 'M%';
+---------------+---------------+
| Employee_Name | LENGTH(Ename) |
+---------------+---------------+
| Jennifer      |             8 |
| Alicia        |             6 |
| Joyce         |             5 |
| James         |             5 |
| Aashish       |             7 |
+---------------+---------------+
5 rows in set (0.01 sec)
```

21. Query to display Name, Hire Date and Day of the week on which the employee started.



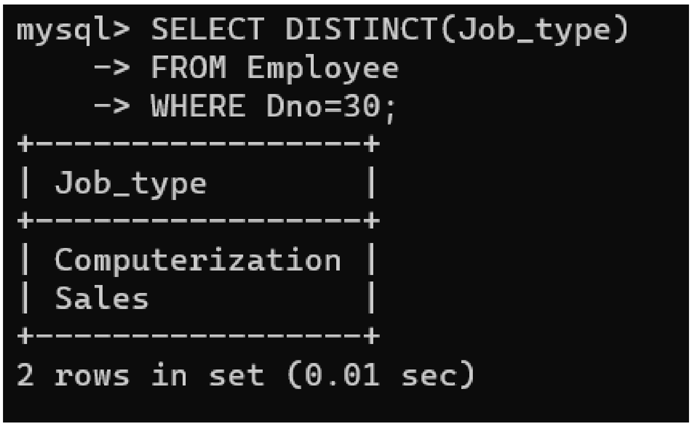
```
mysql> SELECT Ename, Hire_date, DAYNAME(Hire_date)
    -> FROM Employee;
+----------+------------+--------------------+
| Ename    | Hire_date  | DAYNAME(Hire_date) |
+----------+------------+--------------------+
| Darrel   | 1981-02-10 | Tuesday            |
| Franklin | 1985-08-17 | Saturday           |
| Jennifer | 1981-10-23 | Friday             |
| Alicia   | 1983-07-30 | Saturday           |
| Ramesh   | 1982-01-01 | Friday             |
| Joyce    | 1980-10-28 | Tuesday            |
| James    | 1980-06-12 | Thursday           |
| Aashish  | 1984-03-13 | Tuesday            |
+----------+------------+--------------------+
8 rows in set (0.00 sec)
```

22. Query to display Name, Department Name and Department No for all the employees.



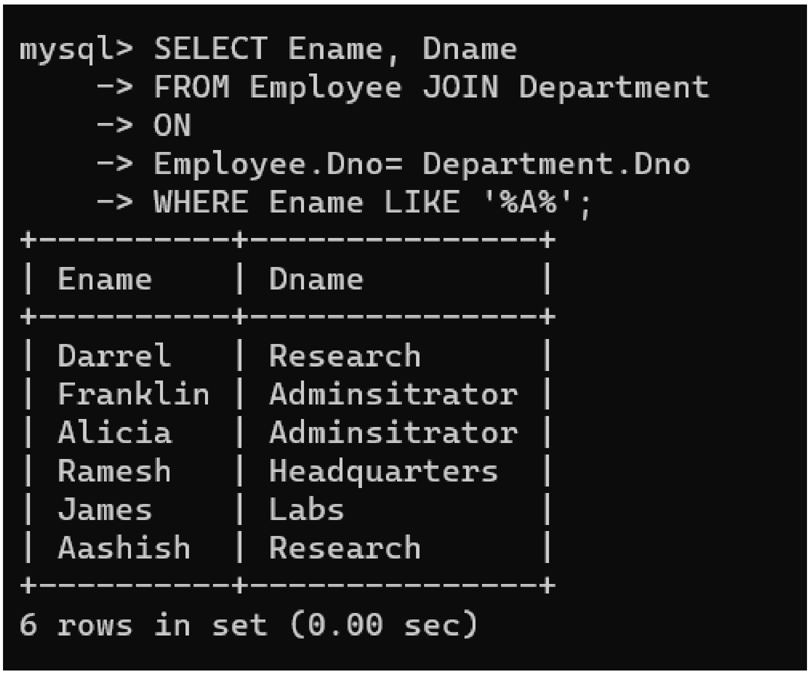
```
mysql> SELECT Ename, Dname, Employee.Dno
    -> FROM Employee JOIN Department
    -> ON
    -> Employee.Dno=Department.Dno;
+----------+---------------+------+
| Ename    | Dname         | Dno  |
+----------+---------------+------+
| Darrel   | Research      |   10 |
| Aashish  | Research      |   10 |
| Franklin | Adminsitrator |   20 |
| Alicia   | Adminsitrator |   20 |
| Joyce    | Adminsitrator |   20 |
| Jennifer | Headquarters  |   30 |
| Ramesh   | Headquarters  |   30 |
| James    | Labs          |   40 |
+----------+---------------+------+
8 rows in set (0.02 sec)
```

23. Query to display Unique Listing of all Jobs that are in Department number 30.



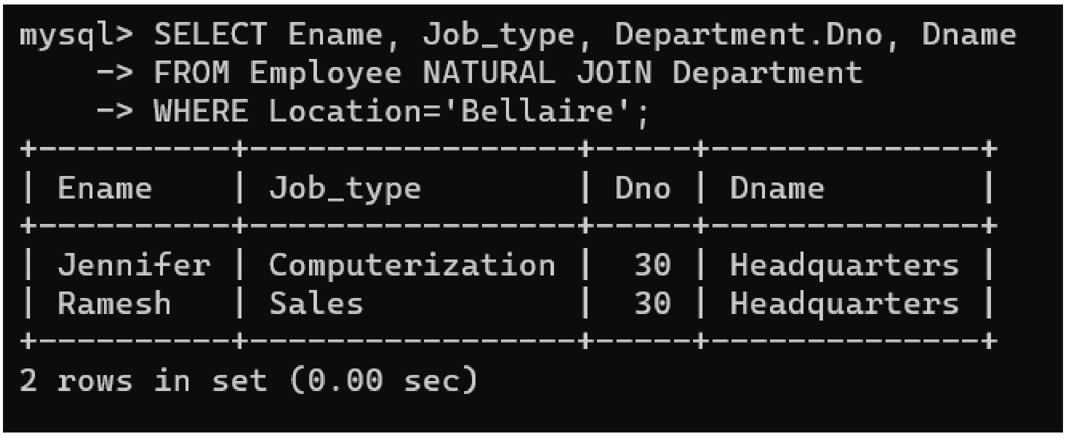
```
mysql> SELECT DISTINCT(Job_type)
    -> FROM Employee
    -> WHERE Dno=30;
+-----------------+
| Job_type        |
+-----------------+
| Computerization |
| Sales           |
+-----------------+
2 rows in set (0.01 sec)
```

24. Query to display Name, Dept Name of all employees who have an ‘A’ in their name.



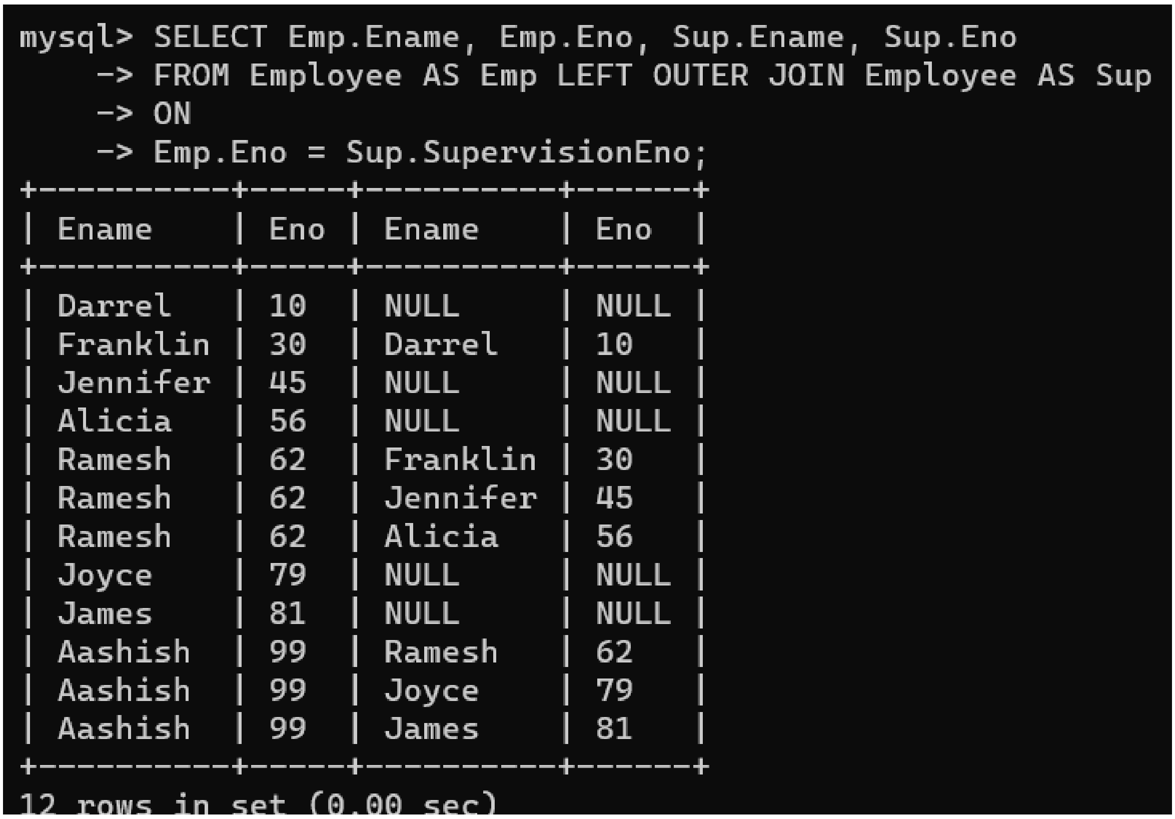
```
mysql> SELECT Ename, Dname
    -> FROM Employee JOIN Department
    -> ON
    -> Employee.Dno= Department.Dno
    -> WHERE Ename LIKE '%A%';
+----------+---------------+
| Ename    | Dname         |
+----------+---------------+
| Darrel   | Research      |
| Franklin | Adminsitrator |
| Alicia   | Adminsitrator |
| Ramesh   | Headquarters  |
| James    | Labs          |
| Aashish  | Research      |
+----------+---------------+
6 rows in set (0.00 sec)
```

25. Query to display Name, Job, Department No. And Department Name for all the employees working at the Bellaire location.



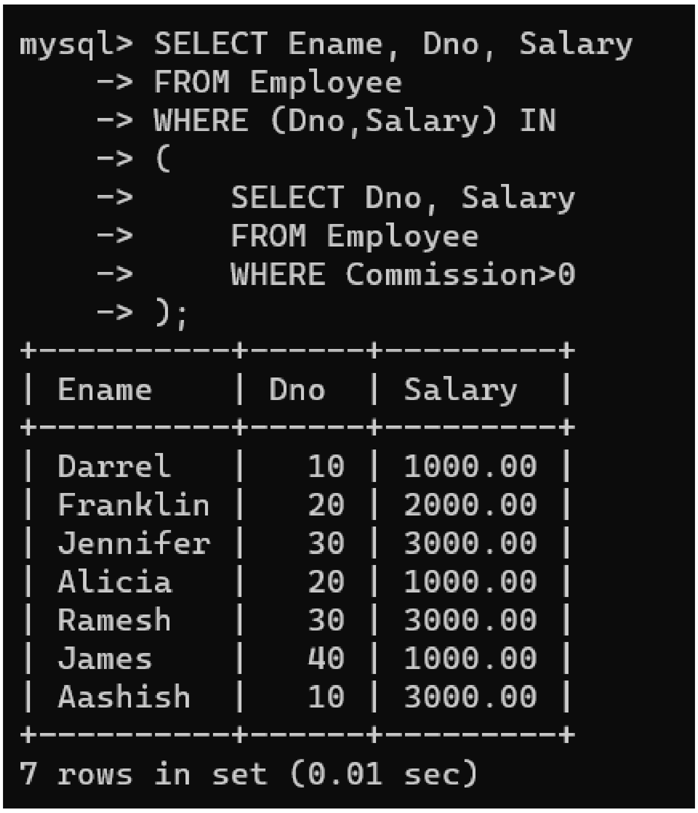
```
mysql> SELECT Ename, Job_type, Department.Dno, Dname
    -> FROM Employee NATURAL JOIN Department
    -> WHERE Location='Bellaire';
+----------+-----------------+-----+--------------+
| Ename    | Job_type        | Dno | Dname        |
+----------+-----------------+-----+--------------+
| Jennifer | Computerization |  30 | Headquarters |
| Ramesh   | Sales           |  30 | Headquarters |
+----------+-----------------+-----+--------------+
2 rows in set (0.00 sec)
```

26. Query to display Name and Employee no. Along with their supervisor's Name and the supervisor's employee no; along with the Employees' Name who do not have a supervisor.



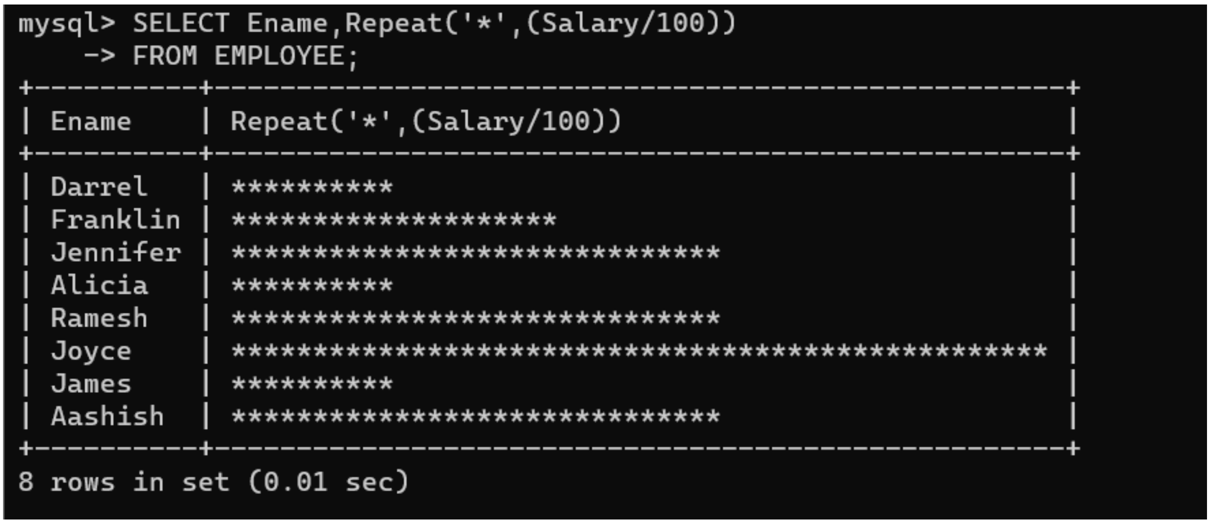
```
mysql> SELECT Emp.Ename, Emp.Eno, Sup.Ename, Sup.Eno
    -> FROM Employee AS Emp LEFT OUTER JOIN Employee AS Sup
    -> ON
    -> Emp.Eno = Sup.SupervisionEno;
+----------+-----+----------+------+
| Ename    | Eno | Ename    | Eno  |
+----------+-----+----------+------+
| Darrel   | 10  | NULL     | NULL |
| Franklin | 30  | Darrel   | 10   |
| Jennifer | 45  | NULL     | NULL |
| Alicia   | 56  | NULL     | NULL |
| Ramesh   | 62  | Franklin | 30   |
| Ramesh   | 62  | Jennifer | 45   |
| Ramesh   | 62  | Alicia   | 56   |
| Joyce    | 79  | NULL     | NULL |
| James    | 81  | NULL     | NULL |
| Aashish  | 99  | Ramesh   | 62   |
| Aashish  | 99  | Joyce    | 79   |
| Aashish  | 99  | James    | 81   |
+----------+-----+----------+------+
12 rows in set (0.00 sec)
```

27. Query to display Name, Dept No. And Salary of any employee whose department No. and salary matches both the department no. And the salary of any employee who earns a commission.



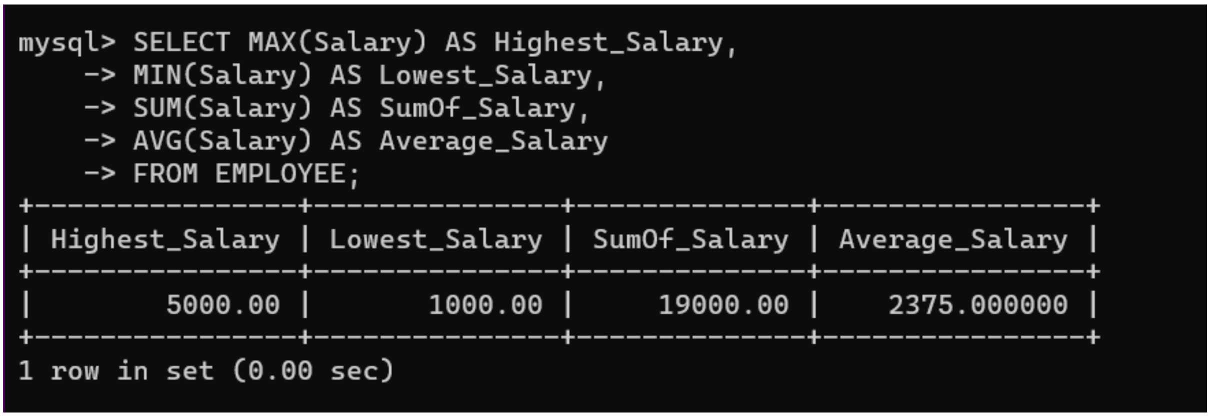
```
mysql> SELECT Ename, Dno, Salary
    -> FROM Employee
    -> WHERE (Dno,Salary) IN
    -> (
    ->     SELECT Dno, Salary
    ->     FROM Employee
    ->     WHERE Commission>0
    -> );
+----------+-------+---------+
| Ename    | Dno   | Salary  |
+----------+-------+---------+
| Darrel   |    10 | 1000.00 |
| Franklin |    20 | 2000.00 |
| Jennifer |    30 | 3000.00 |
| Alicia   |    20 | 1000.00 |
| Ramesh   |    30 | 3000.00 |
| James    |    40 | 1000.00 |
| Aashish  |    10 | 3000.00 |
+----------+-------+---------+
7 rows in set (0.01 sec)
```

28. Query to display Name and Salaries represented by asterisks, where each asterisk (*) signifies $100.



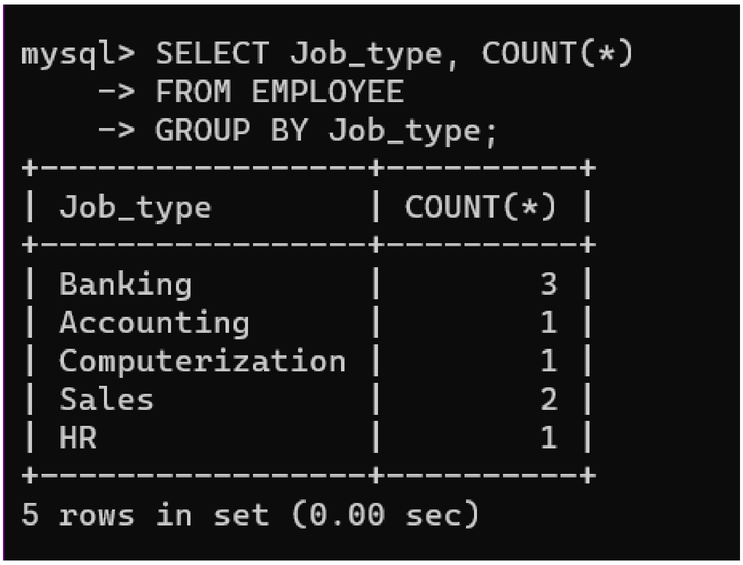
```
mysql> SELECT Ename,Repeat('*',(Salary/100))
    -> FROM EMPLOYEE;
+----------+----------------------------------------------------+
| Ename    | Repeat('*',(Salary/100))                           |
+----------+----------------------------------------------------+
| Darrel   | **********                                         |
| Franklin | ********************                               |
| Jennifer | ******************************                     |
| Alicia   | **********                                         |
| Ramesh   | ******************************                     |
| Joyce    | ************************************************** |
| James    | **********                                         |
| Aashish  | ******************************                     |
+----------+----------------------------------------------------+
8 rows in set (0.01 sec)
```

29. Query to display the Highest, Lowest, Sum and Average Salaries of all the employees.



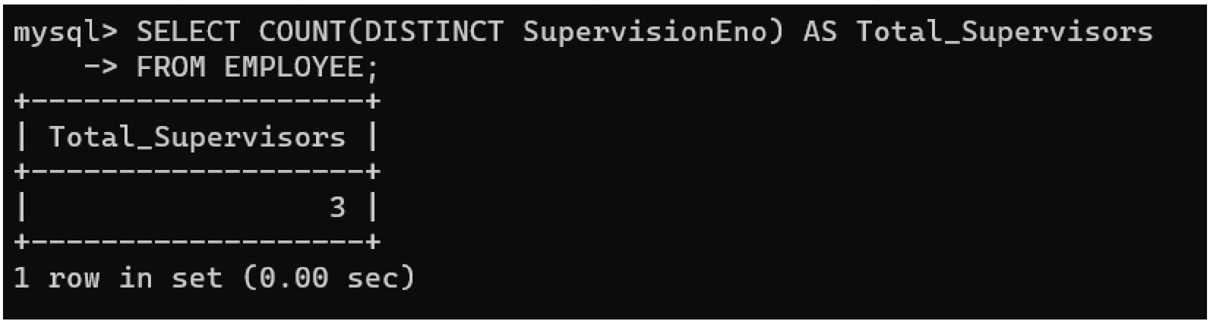
```
mysql> SELECT MAX(Salary) AS Highest_Salary,
    -> MIN(Salary) AS Lowest_Salary,
    -> SUM(Salary) AS SumOf_Salary,
    -> AVG(Salary) AS Average_Salary
    -> FROM EMPLOYEE;
+----------------+---------------+--------------+----------------+
| Highest_Salary | Lowest_Salary | SumOf_Salary | Average_Salary |
+----------------+---------------+--------------+----------------+
|        5000.00 |       1000.00 |     19000.00 |    2375.000000 |
+----------------+---------------+--------------+----------------+
1 row in set (0.00 sec)
```

30. Query to display the number of employees performing the same Job type functions.



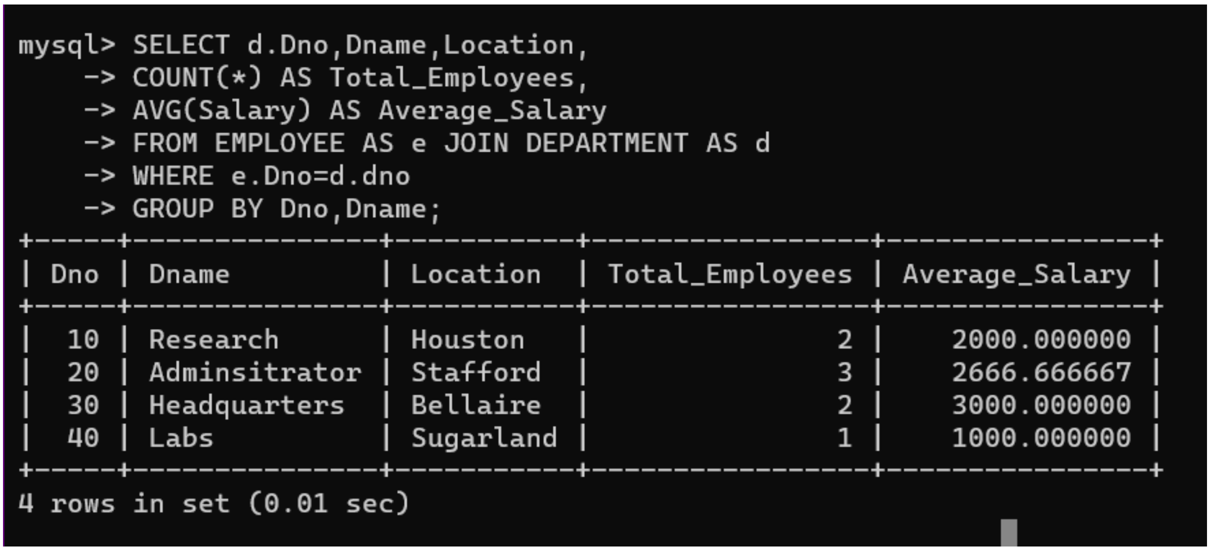
```
mysql> SELECT Job_type, COUNT(*)
    -> FROM EMPLOYEE
    -> GROUP BY Job_type;
+-----------------+----------+
| Job_type        | COUNT(*) |
+-----------------+----------+
| Banking         |        3 |
| Accounting      |        1 |
| Computerization |        1 |
| Sales           |        2 |
| HR              |        1 |
+-----------------+----------+
5 rows in set (0.00 sec)
```

31. Query to display the total number of supervisors without listing their names.



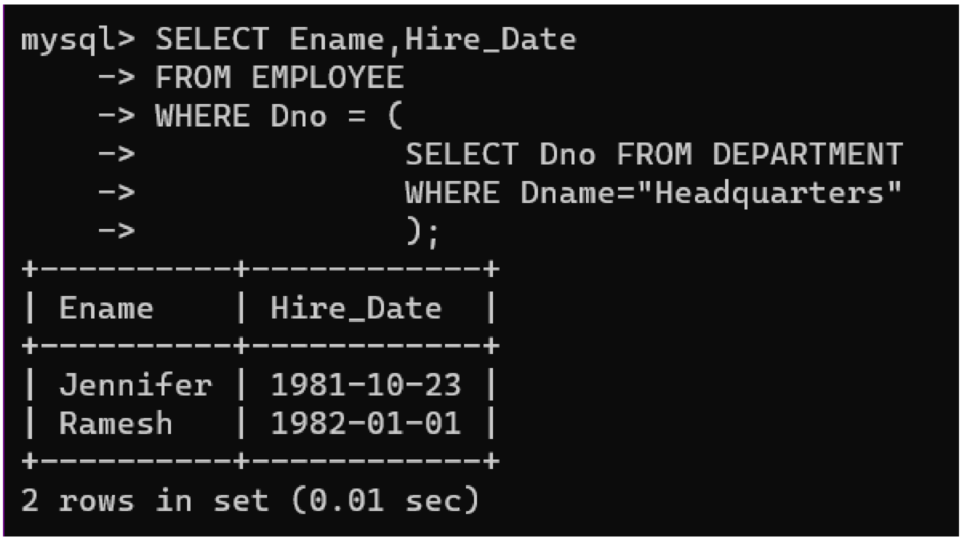
```
mysql> SELECT COUNT(DISTINCT SupervisionEno) AS Total_Supervisors
    -> FROM EMPLOYEE;
+-------------------+
| Total_Supervisors |
+-------------------+
|                 3 |
+-------------------+
1 row in set (0.00 sec)
```

32. Query to display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.



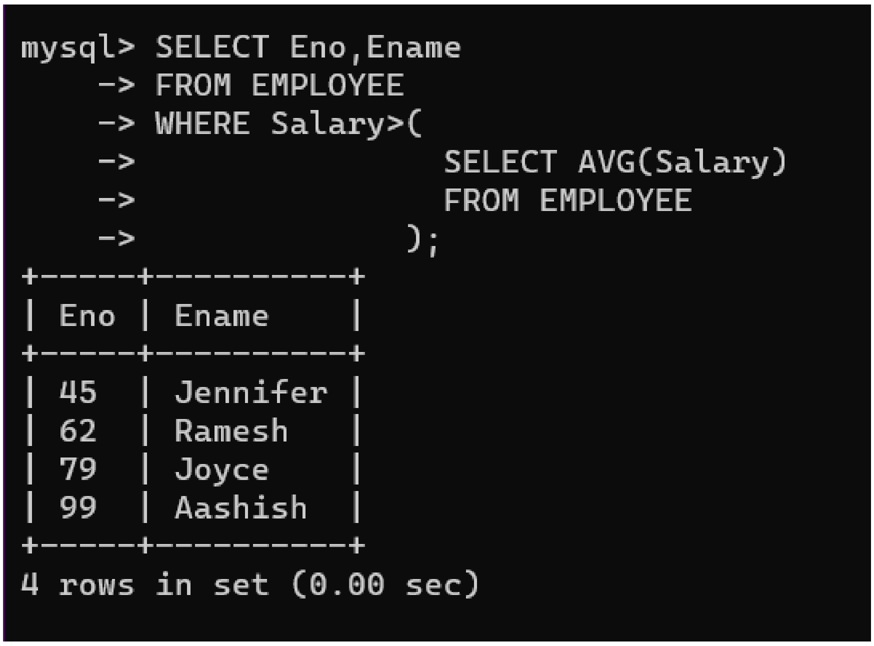
```
mysql> SELECT d.Dno,Dname,Location,
    -> COUNT(*) AS Total_Employees,
    -> AVG(Salary) AS Average_Salary
    -> FROM EMPLOYEE AS e JOIN DEPARTMENT AS d
    -> WHERE e.Dno=d.dno
    -> GROUP BY Dno,Dname;
+-----+---------------+-----------+-----------------+----------------+
| Dno | Dname         | Location  | Total_Employees | Average_Salary |
+-----+---------------+-----------+-----------------+----------------+
|  10 | Research      | Houston   |               2 |    2000.000000 |
|  20 | Adminsitrator | Stafford  |               3 |    2666.666667 |
|  30 | Headquarters  | Bellaire  |               2 |    3000.000000 |
|  40 | Labs          | Sugarland |               1 |    1000.000000 |
+-----+---------------+-----------+-----------------+----------------+
4 rows in set (0.01 sec)
```

33. Query to display Name and Hire Date for all employees in the same dept. as Headquarters.



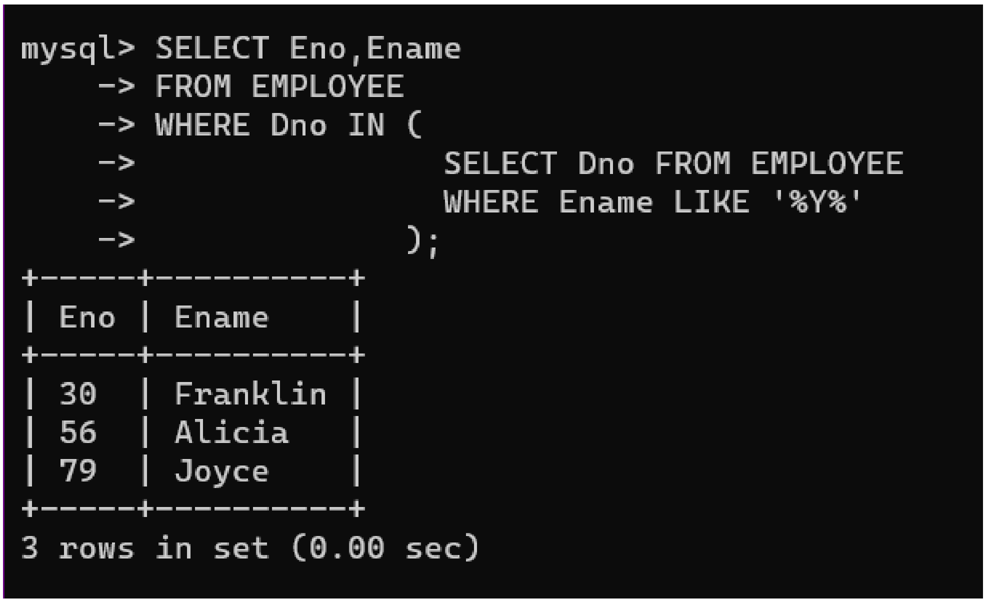
```
mysql> SELECT Ename,Hire_Date
    -> FROM EMPLOYEE
    -> WHERE Dno = (
    ->              SELECT Dno FROM DEPARTMENT
    ->              WHERE Dname="Headquarters"
    ->              );
+----------+------------+
| Ename    | Hire_Date  |
+----------+------------+
| Jennifer | 1981-10-23 |
| Ramesh   | 1982-01-01 |
+----------+------------+
2 rows in set (0.01 sec)
```

34. Query to display the Employee No. And Name for all employees who earn more than the average salary.



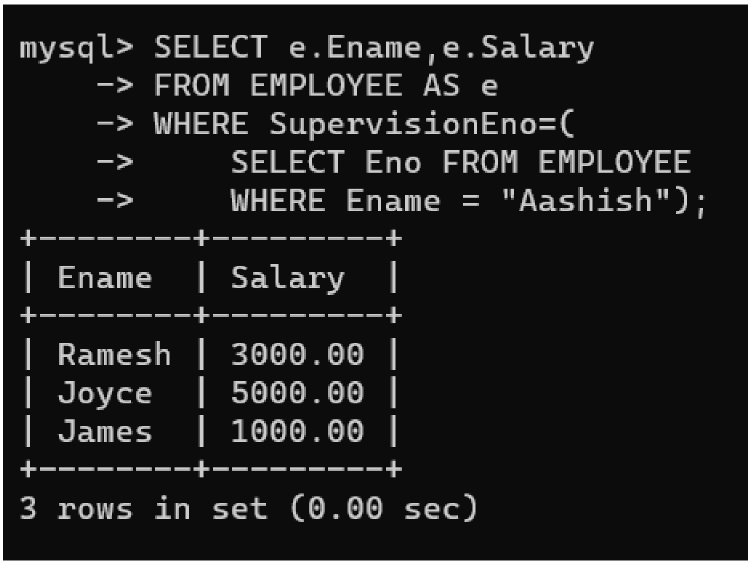
```
mysql> SELECT Eno,Ename
    -> FROM EMPLOYEE
    -> WHERE Salary>(
    ->                 SELECT AVG(Salary)
    ->                 FROM EMPLOYEE
    ->               );
+-----+----------+
| Eno | Ename    |
+-----+----------+
| 45  | Jennifer |
| 62  | Ramesh   |
| 79  | Joyce    |
| 99  | Aashish  |
+-----+----------+
4 rows in set (0.00 sec)
```

35. Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a ‘Y’.



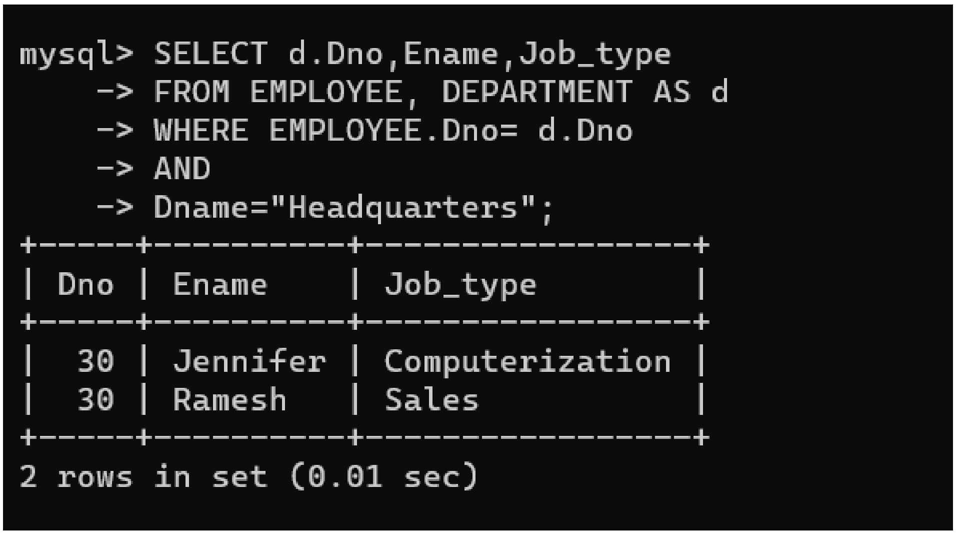
```
mysql> SELECT Eno,Ename
    -> FROM EMPLOYEE
    -> WHERE Dno IN (
    ->                 SELECT Dno FROM EMPLOYEE
    ->                 WHERE Ename LIKE '%Y%'
    ->               );
+-----+----------+
| Eno | Ename    |
+-----+----------+
| 30  | Franklin |
| 56  | Alicia   |
| 79  | Joyce    |
+-----+----------+
3 rows in set (0.00 sec)
```

36. Query to display the names and salaries of all employees who report to supervisor named ‘King’.(here we are taking “Aashish” instead of “king”).



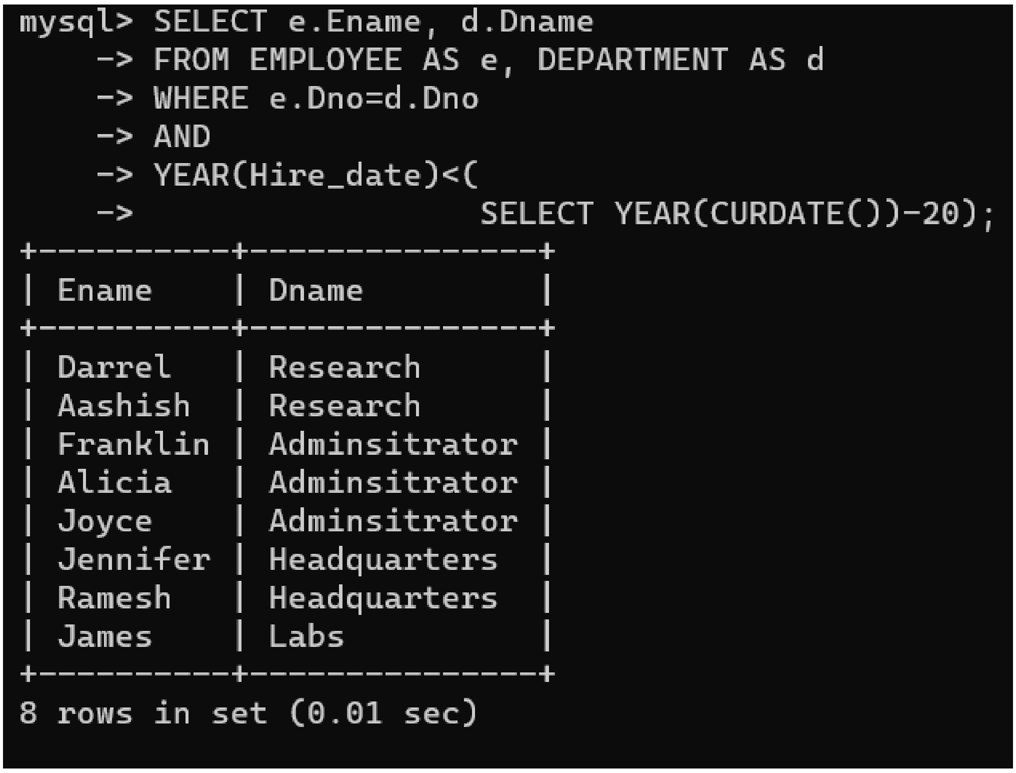
```
mysql> SELECT e.Ename,e.Salary
    -> FROM EMPLOYEE AS e
    -> WHERE SupervisionEno=(
    ->     SELECT Eno FROM EMPLOYEE
    ->     WHERE Ename = "Aashish");
+--------+---------+
| Ename  | Salary  |
+--------+---------+
| Ramesh | 3000.00 |
| Joyce  | 5000.00 |
| James  | 1000.00 |
+--------+---------+
3 rows in set (0.00 sec)
```

37. Query to display the department no, name and job for all employees in the Sales department.
(here we are using headquarters as department instead of sales).



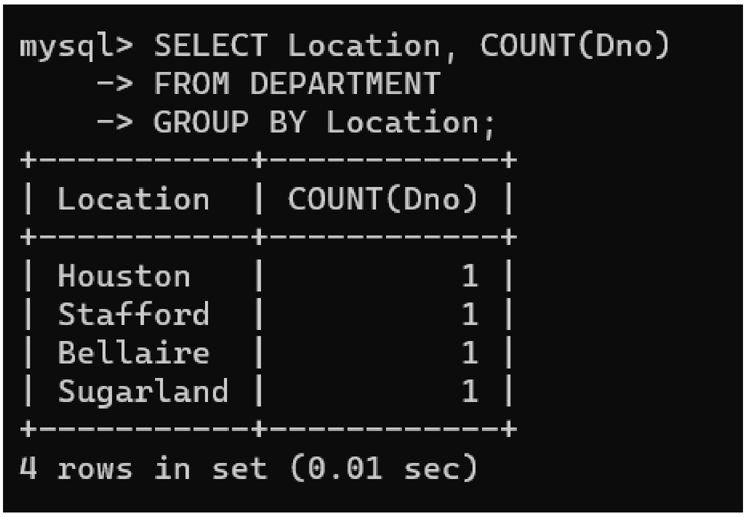
```
mysql> SELECT d.Dno,Ename,Job_type
    -> FROM EMPLOYEE, DEPARTMENT AS d
    -> WHERE EMPLOYEE.Dno= d.Dno
    -> AND
    -> Dname="Headquarters";
+-----+----------+-----------------+
| Dno | Ename    | Job_type        |
+-----+----------+-----------------+
|  30 | Jennifer | Computerization |
|  30 | Ramesh   | Sales           |
+-----+----------+-----------------+
2 rows in set (0.01 sec)
```

38. Display names of employees along with their department name who have more than 20 years’ experience.



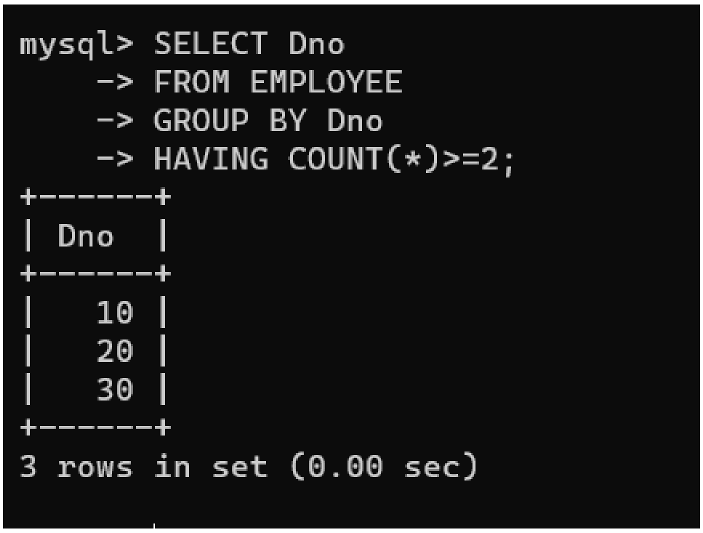
```
mysql> SELECT e.Ename, d.Dname
    -> FROM EMPLOYEE AS e, DEPARTMENT AS d
    -> WHERE e.Dno=d.Dno
    -> AND
    -> YEAR(Hire_date)<(
    ->                  SELECT YEAR(CURDATE())-20);
+----------+---------------+
| Ename    | Dname         |
+----------+---------------+
| Darrel   | Research      |
| Aashish  | Research      |
| Franklin | Adminsitrator |
| Alicia   | Adminsitrator |
| Joyce    | Adminsitrator |
| Jennifer | Headquarters  |
| Ramesh   | Headquarters  |
| James    | Labs          |
+----------+---------------+
8 rows in set (0.01 sec)
```

39. Display total number of departments at each location.



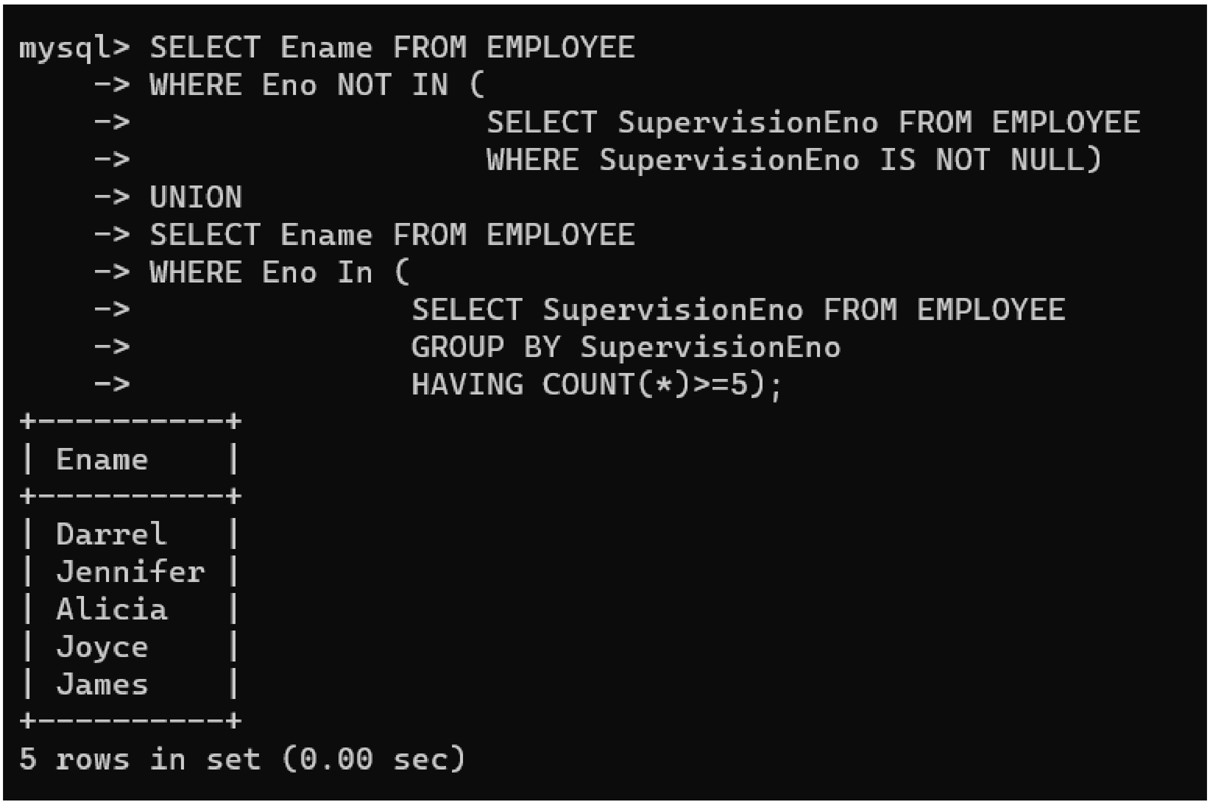
```
mysql> SELECT Location, COUNT(Dno)
    -> FROM DEPARTMENT
    -> GROUP BY Location;
+-----------+------------+
| Location  | COUNT(Dno) |
+-----------+------------+
| Houston   |          1 |
| Stafford  |          1 |
| Bellaire  |          1 |
| Sugarland |          1 |
+-----------+------------+
4 rows in set (0.01 sec)
```

40. Find the department name in which at least 20 employees work in.



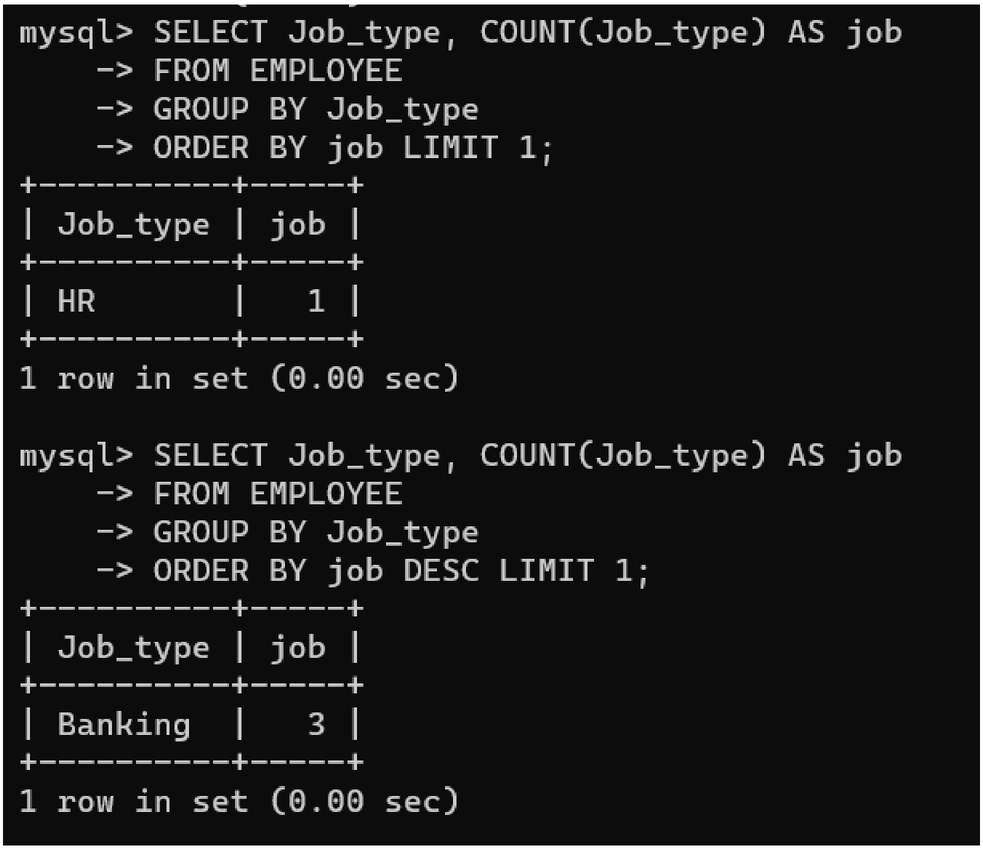
```
mysql> SELECT Dno
    -> FROM EMPLOYEE
    -> GROUP BY Dno
    -> HAVING COUNT(*)>=2;
+-------+
| Dno   |
+-------+
|    10 |
|    20 |
|    30 |
+-------+
3 rows in set (0.00 sec)
```

41. Query to find the employee' name who is not supervisor and name of supervisor supervising more than 5 employees.



```
mysql> SELECT Ename FROM EMPLOYEE
    -> WHERE Eno NOT IN (
    ->                   SELECT SupervisionEno FROM EMPLOYEE
    ->                   WHERE SupervisionEno IS NOT NULL)
    -> UNION
    -> SELECT Ename FROM EMPLOYEE
    -> WHERE Eno In (
    ->               SELECT SupervisionEno FROM EMPLOYEE
    ->               GROUP BY SupervisionEno
    ->               HAVING COUNT(*)>=5);
+----------+
| Ename    |
+----------+
| Darrel   |
| Jennifer |
| Alicia   |
| Joyce    |
| James    |
+----------+
5 rows in set (0.00 sec)
```

42. Query to display the job type with maximum and minimum employees.



```
mysql> SELECT Job_type, COUNT(Job_type) AS job
    -> FROM EMPLOYEE
    -> GROUP BY Job_type
    -> ORDER BY job LIMIT 1;
+----------+------+
| Job_type | job  |
+----------+------+
| HR       |    1 |
+----------+------+
1 row in set (0.00 sec)

mysql> SELECT Job_type, COUNT(Job_type) AS job
    -> FROM EMPLOYEE
    -> GROUP BY Job_type
    -> ORDER BY job DESC LIMIT 1;
+----------+------+
| Job_type | job  |
+----------+------+
| Banking  |    3 |
+----------+------+
1 row in set (0.00 sec)
```