

1) Create a table name STUDENT with following structure.

SL.NO	Column Name	Description	Data Type
1	RegNo	Registration Number	NUMBER(3)
2	Name	Student Name	VARCHAR(15)
3	Gender	Gender of the student	CHAR(1)
4	DOB	Date of Birth	DATE
5	MobileNo	Mobile Number	NUMBER(10)
6	City	Location of stay	VARCHAR(15)

```
mysql> use student;
Database changed
mysql> create table students( Regno INT(15) PRIMARY KEY, Name VARCHAR(15), Gender char(25), dateofbirth DATE, mobile int(10), city varchar(15));
Query OK, 0 rows affected, 2 warnings (0.06 sec)

mysql> desc students;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Regno      | int           | NO   | PRI | NULL    |       |
| Name       | varchar(15)   | YES  |     | NULL    |       |
| Gender     | char(25)      | YES  |     | NULL    |       |
| dateofbirth | date          | YES  |     | NULL    |       |
| mobile     | int           | YES  |     | NULL    |       |
| city       | varchar(15)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.08 sec)
```

Create a table name FACULTY with following structure.

SL. NO	Column Name	Description	Data Type
1	FacNo	Faculty Identifier	VARCHAR(4)
2	FacName	Faculty Name	VARCHAR(15)
3	Gender	Gender of faculty	CHAR(1)
4	DOB	Date of Birth	DATE
5	DOJ	Date of Join	DATE
6	MobileNo	Mobile Number	NUMBER(10)

```
mysql> create database faculty;
Query OK, 1 row affected (0.03 sec)

mysql> use faculty;
Database changed
```

```
mysql> CREATE TABLE FACULTY (
->     FacNo VARCHAR(4) PRIMARY KEY,
->     FacName VARCHAR(15),
->     Gender CHAR(1),
->     DOB DATE,
->     DOJ DATE,
->     MobileNo INT(10));
Query OK, 0 rows affected, 1 warning (0.07 sec)
```

```
mysql> desc faculty;
```

Field	Type	Null	Key	Default	Extra
FacNo	varchar(4)	NO	PRI	NULL	
FacName	varchar(15)	YES		NULL	
Gender	char(1)	YES		NULL	
DOB	date	YES		NULL	
DOJ	date	YES		NULL	
MobileNo	int	YES		NULL	

```
6 rows in set (0.01 sec)
```

2. Create a table name DEPARTMENT with following structure.

SL. NO	Column Name	Description	Data Type
1	DeptNo	Department Identifier	VARCHAR(4)
2	DeptName	Department Name	VARCHAR(15)
3	DeptHead	Department Head	VARCHAR(4)

```
mysql> create database department;
Query OK, 1 row affected (0.01 sec)

mysql> use department;
Database changed
mysql> CREATE TABLE DEPARTMENT (
    ->     DeptNo VARCHAR(4) PRIMARY KEY,
    ->     DeptName VARCHAR(15),
    ->     DeptHead VARCHAR(4)
    -> );
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> desc department;
```

Field	Type	Null	Key	Default	Extra
DeptNo	varchar(4)	NO	PRI	NULL	
DeptName	varchar(15)	YES		NULL	
DeptHead	varchar(4)	YES		NULL	

```
3 rows in set (0.02 sec)
```

Create a table name COURSE with following structure.

SL.NO	Column Name	Description	Data Type
1	CourseNo	Course Identifier	VARCHAR(3)
2	CourseDesc	Course Description	VARCHAR(14)
3	CourseType	Course Type	CHAR(1)
4	SemNo	Semester Number	CHAR(1)
5	HallNo	Hall Number	VARCHAR(4)
6	FacNo	Faculty Identifier	VARCHAR(4)

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

```
mysql> use course;
Database changed
```

```
mysql> CREATE TABLE COURSE (
    ->     CourseNo VARCHAR(3) PRIMARY KEY,
    ->     CourseDesc VARCHAR(14),
    ->     CourseType CHAR(1),
    ->     SemNo CHAR(1),
    ->     HallNo VARCHAR(4),
    ->     FacNo VARCHAR(4)
    -> );
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> desc course;
```

Field	Type	Null	Key	Default	Extra
CourseNo	varchar(3)	NO	PRI	NULL	
CourseDesc	varchar(14)	YES		NULL	
CourseType	char(1)	YES		NULL	
SemNo	char(1)	YES		NULL	
HallNo	varchar(4)	YES		NULL	
FacNo	varchar(4)	YES		NULL	

6 rows in set (0.01 sec)

3. Create a table EMPLOYEE with following schema: (Emp\_no, E\_name, E\_address, E\_ph\_no, Dept\_no, Dept\_name, Job\_id, Salary)

Write SQL queries for following question:

1. Insert atleast 5 rows in the table.
2. Display all the information of EMP table.
3. Display the record of each employee who works in department D10.
4. Update the city of Emp\_no-12 with current city as Nagpur.
5. Display the details of Employee who works in department MECH.
6. Display the complete record of employees working in SALES Department.
7. Delete the email\_id of employee James.

```

mysql> create database employee;
Query OK, 1 row affected (0.02 sec)

mysql> use employee;
Database changed
mysql> CREATE TABLE EMPLOYEE (
  ->     Emp_no INT PRIMARY KEY,
  ->     E_name VARCHAR(50),
  ->     E_address VARCHAR(100),
  ->     E_ph_no VARCHAR(15),
  ->     Dept_no VARCHAR(10),
  ->     Dept_name VARCHAR(50),
  ->     Job_id VARCHAR(10),
  ->     Salary DECIMAL(10, 2)
  -> );
Query OK, 0 rows affected (0.13 sec)

mysql> INSERT INTO EMPLOYEE (Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name, Job_id, Salary)
  -> VALUES
  -> (10, 'Alice', 'Mumbai', '9876543210', 'D10', 'HR', 'HR01', 60000),
  -> (11, 'Bob', 'Chennai', '9765432101', 'MECH', 'Mechanical', 'ME01', 55000),
  -> (12, 'James', 'Delhi', '9654321012', 'SALES', 'Sales', 'SA01', 45000),
  -> (13, 'John', 'Kolkata', '9543210123', 'D10', 'HR', 'HR02', 62000),
  -> (14, 'Sophia', 'Bangalore', '9432101234', 'MECH', 'Mechanical', 'ME02', 58000);
Query OK, 5 rows affected (0.04 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM EMPLOYEE;

```

Emp_no	E_name	E_address	E_ph_no	Dept_no	Dept_name	Job_id	Salary
10	Alice	Mumbai	9876543210	D10	HR	HR01	60000.00
11	Bob	Chennai	9765432101	MECH	Mechanical	ME01	55000.00
12	James	Delhi	9654321012	SALES	Sales	SA01	45000.00
13	John	Kolkata	9543210123	D10	HR	HR02	62000.00
14	Sophia	Bangalore	9432101234	MECH	Mechanical	ME02	58000.00

```
mysql> SELECT *
-> FROM EMPLOYEE
-> WHERE Dept_no = 'D10';
```

Emp_no	E_name	E_address	E_ph_no	Dept_no	Dept_name	Job_id	Salary
10	Alice	Mumbai	9876543210	D10	HR	HR01	60000.00
13	John	Kolkata	9543210123	D10	HR	HR02	62000.00

2 rows in set (0.01 sec)

```
mysql> UPDATE EMPLOYEE
-> SET E_address = 'Nagpur'
-> WHERE Emp_no = 12;
```

Query OK, 1 row affected (0.02 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> SELECT *
-> FROM EMPLOYEE
-> WHERE Dept_no = 'MECH';
```

Emp_no	E_name	E_address	E_ph_no	Dept_no	Dept_name	Job_id	Salary
11	Bob	Chennai	9765432101	MECH	Mechanical	ME01	55000.00
14	Sophia	Bangalore	9432101234	MECH	Mechanical	ME02	58000.00

2 rows in set (0.00 sec)

```
mysql> SELECT *
-> FROM EMPLOYEE
-> WHERE Dept_name = 'Sales';
```

Emp_no	E_name	E_address	E_ph_no	Dept_no	Dept_name	Job_id	Salary
12	James	Nagpur	9654321012	SALES	Sales	SA01	45000.00

1 row in set (0.00 sec)

```
mysql> UPDATE EMPLOYEE
-> SET E_ph_no = NULL
-> WHERE E_name = 'James';
```

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> SELECT * FROM EMPLOYEE;
```

Emp_no	E_name	E_address	E_ph_no	Dept_no	Dept_name	Job_id	Salary
10	Alice	Mumbai	9876543210	D10	HR	HR01	60000.00
11	Bob	Chennai	9765432101	MECH	Mechanical	ME01	55000.00
12	James	Nagpur	NULL	SALES	Sales	SA01	45000.00
13	John	Kolkata	9543210123	D10	HR	HR02	62000.00
14	Sophia	Bangalore	9432101234	MECH	Mechanical	ME02	58000.00

5 rows in set (0.01 sec)

4. A database is being constructed for storing sales information system. A product can be described with a unique product number, product name, selling price, manufacturer name. The product can sale to a particular client and each client have it own unique client number, client name, client addresses, city, pin code, state and total balance to be required to paid. Each client order to buy product from the salesman. In the order, it has unique sales order number, sales order date, client number, salesman number (unique), billed whole payment by the party or not and its delivery date. The salesman have the name, addresses, city, pin code, state, salary of the sales man, delivery date, total quantity ordered, product rate.

Write the SQL queries for the following –

- (a) Retrieve the list of names and the cities of all the clients.
- (b) List the various products available.
- (c) Find the names of all clients having 'a' as the second letter in their names.
- (d) List all the clients who are located in TEZPUR.
- (e) Find the products whose selling price is greater than 2000 and less than or equal to 5000
- (f) Add a new column NEW\_PRICE into the product\_master table.
- (g) Rename the column product\_rate of Sales\_Order\_Details to new\_product\_rate.
- (h) List the products in sorted order of their description.
- (i) Display the order number and date on which the clients placed their order.

```
mysql> CREATE DATABASE SalesInformationSystem;  
Query OK, 1 row affected (0.02 sec)
```

```
mysql> USE SalesInformationSystem;  
Database changed
```

```
mysql> CREATE TABLE PRODUCT_MASTER (  
->     Product_No INT PRIMARY KEY,  
->     Product_Name VARCHAR(50),  
->     Selling_Price DECIMAL(10, 2),  
->     Manufacturer_Name VARCHAR(50)  
-> );
```

```
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> CREATE TABLE CLIENT (  
->     Client_No INT PRIMARY KEY,  
->     Client_Name VARCHAR(50),  
->     Client_Address VARCHAR(100),  
->     City VARCHAR(50),  
->     Pin_Code VARCHAR(10),  
->     State VARCHAR(50),  
->     Total_Balance DECIMAL(10, 2)  
-> );
```

```
Query OK, 0 rows affected (0.21 sec)
```

```
mysql> CREATE TABLE SALES_ORDER (  
->     Order_No INT PRIMARY KEY,  
->     Order_Date DATE,  
->     Client_No INT,  
->     Salesman_No INT,  
->     Billed CHAR(1), -- 'Y' for Yes, 'N' for No  
->     Delivery_Date DATE,  
->     FOREIGN KEY (Client_No) REFERENCES CLIENT(Client_No)  
-> );
```

```
Query OK, 0 rows affected (0.94 sec)
```



```

mysql> CREATE TABLE SALESMAN (
  ->     Salesman_No INT PRIMARY KEY,
  ->     Name VARCHAR(50),
  ->     Address VARCHAR(100),
  ->     City VARCHAR(50),
  ->     Pin_Code VARCHAR(10),
  ->     State VARCHAR(50),
  ->     Salary DECIMAL(10, 2),
  ->     Delivery_Date DATE,
  ->     Total_Quantity_Ordered INT,
  ->     Product_Rate DECIMAL(10, 2)
  -> );
Query OK, 0 rows affected (0.08 sec)

mysql> INSERT INTO PRODUCT_MASTER (Product_No, Product_Name, Selling_Price, Manufacturer_Name)
  -> VALUES
  -> (101, 'Laptop', 45000, 'HP'),
  -> (102, 'Printer', 5000, 'Canon'),
  -> (103, 'Monitor', 12000, 'Dell'),
  -> (104, 'Mouse', 700, 'Logitech'),
  -> (105, 'Keyboard', 1200, 'Logitech');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO CLIENT (Client_No, Client_Name, Client_Address, City, Pin_Code, State, Total_Balance)
  -> VALUES
  -> (201, 'Alice', '123 Street A', 'Mumbai', '400001', 'Maharashtra', 25000),
  -> (202, 'Bob', '456 Street B', 'Tezpur', '784001', 'Assam', 15000),

  -> (203, 'Charlie', '789 Street C', 'Chennai', '600001', 'Tamil Nadu', 30000),
  -> (204, 'David', '101 Street D', 'Tezpur', '784002', 'Assam', 20000),
  -> (205, 'Eve', '202 Street E', 'Delhi', '110001', 'Delhi', 10000);
Query OK, 5 rows affected (0.06 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO SALES_ORDER (Order_No, Order_Date, Client_No, Salesman_No, Billed, Delivery_Date)
  -> VALUES
  -> (301, '2024-11-01', 201, 401, 'Y', '2024-11-05'),
  -> (302, '2024-11-02', 202, 402, 'N', '2024-11-07'),
  -> (303, '2024-11-03', 203, 403, 'Y', '2024-11-06'),
  -> (304, '2024-11-04', 204, 404, 'N', '2024-11-08'),
  -> (305, '2024-11-05', 205, 401, 'Y', '2024-11-10');
Query OK, 5 rows affected (0.05 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO SALESMAN (Salesman_No, Name, Address, City, Pin_Code, State, Salary, Delivery_Date, Total_Quantity_Ordered, Product_Rate)
  -> VALUES
  -> (401, 'John', '12 Sales St.', 'Mumbai', '400001', 'Maharashtra', 40000, '2024-11-10', 5, 1000),
  -> (402, 'Paul', '34 Sales Ave.', 'Tezpur', '784001', 'Assam', 35000, '2024-11-07', 8, 500),
  -> (403, 'Mike', '56 Sales Blvd.', 'Chennai', '600001', 'Tamil Nadu', 45000, '2024-11-06', 3, 2000),
  -> (404, 'Sarah', '78 Sales Ln.', 'Tezpur', '784002', 'Assam', 30000, '2024-11-08', 6, 1200),
  -> (405, 'Emma', '90 Sales Rd.', 'Delhi', '110001', 'Delhi', 50000, '2024-11-10', 10, 1500);
Query OK, 5 rows affected (0.02 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> select*from PRODUCT_MASTER;
+-----+-----+-----+-----+
| Product_No | Product_Name | Selling_Price | Manufacturer_Name |
+-----+-----+-----+-----+
| 101 | Laptop | 45000.00 | HP |
| 102 | Printer | 5000.00 | Canon |
| 103 | Monitor | 12000.00 | Dell |
| 104 | Mouse | 700.00 | Logitech |
| 105 | Keyboard | 1200.00 | Logitech |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

```
mysql> select*from CLIENT;
```

Client_No	Client_Name	Client_Address	City	Pin_Code	State	Total_Balance
201	Alice	123 Street A	Mumbai	400001	Maharashtra	25000.00
202	Bob	456 Street B	Tezpur	784001	Assam	15000.00
203	Charlie	789 Street C	Chennai	600001	Tamil Nadu	30000.00
204	David	101 Street D	Tezpur	784002	Assam	20000.00
205	Eve	202 Street E	Delhi	110001	Delhi	10000.00

```
5 rows in set (0.00 sec)
```

```
mysql> select*from SALES_ORDER;
```

Order_No	Order_Date	Client_No	Salesman_No	Billed	Delivery_Date
301	2024-11-01	201	401	Y	2024-11-05
302	2024-11-02	202	402	N	2024-11-07
303	2024-11-03	203	403	Y	2024-11-06
304	2024-11-04	204	404	N	2024-11-08
305	2024-11-05	205	401	Y	2024-11-10

```
5 rows in set (0.00 sec)
```

```
mysql> select*from SALESMAN;
```

Salesman_No	Name	Address	City	Pin_Code	State	Salary	Delivery_Date	Total_Quantity_Ordered	Product_Rate
401	John	12 Sales St.	Mumbai	400001	Maharashtra	40000.00	2024-11-10	5	1000.00
402	Paul	34 Sales Ave.	Tezpur	784001	Assam	35000.00	2024-11-07	8	500.00
403	Mike	56 Sales Blvd.	Chennai	600001	Tamil Nadu	45000.00	2024-11-06	3	2000.00
404	Sarah	78 Sales Ln.	Tezpur	784002	Assam	30000.00	2024-11-08	6	1200.00
405	Emma	90 Sales Rd.	Delhi	110001	Delhi	50000.00	2024-11-10	10	1500.00

```
5 rows in set (0.01 sec)
```

```
mysql> SELECT Client_Name, City
-> FROM CLIENT;
```

Client_Name	City
Alice	Mumbai
Bob	Tezpur
Charlie	Chennai
David	Tezpur
Eve	Delhi

5 rows in set (0.00 sec)

```
mysql> SELECT Product_Name
-> FROM PRODUCT_MASTER;
```

Product_Name
Laptop
Printer
Monitor
Mouse
Keyboard

5 rows in set (0.00 sec)

```
mysql> SELECT Client_Name
-> FROM CLIENT
-> WHERE Client_Name LIKE '_a%';
```

Client_Name
David

1 row in set (0.01 sec)

```
mysql> SELECT *
-> FROM PRODUCT_MASTER
-> ORDER BY Product_Name ASC;
```

Product_No	Product_Name	Selling_Price	Manufacturer_Name	NEW_PRICE
105	Keyboard	1200.00	Logitech	NULL
101	Laptop	45000.00	HP	NULL
103	Monitor	12000.00	Dell	NULL
104	Mouse	700.00	Logitech	NULL
102	Printer	5000.00	Canon	NULL

5 rows in set (0.00 sec)

```
mysql> SELECT Order_No, Order_Date
-> FROM SALES_ORDER;
```

Order_No	Order_Date
301	2024-11-01
302	2024-11-02
303	2024-11-03
304	2024-11-04
305	2024-11-05

5 rows in set (0.00 sec)

```
mysql> SELECT *
  -> FROM CLIENT
  -> WHERE City = 'Tezpur';
```

Client_No	Client_Name	Client_Address	City	Pin_Code	State	Total_Balance
202	Bob	456 Street B	Tezpur	784001	Assam	15000.00
204	David	101 Street D	Tezpur	784002	Assam	20000.00

```
2 rows in set (0.00 sec)
```

  

```
mysql> SELECT Product_Name, Selling_Price
  -> FROM PRODUCT_MASTER
  -> WHERE Selling_Price > 2000 AND Selling_Price <= 5000;
```

Product_Name	Selling_Price
Printer	5000.00

```
1 row in set (0.01 sec)
```

  

```
mysql> SELECT Product_Name, Selling_Price
  -> FROM PRODUCT_MASTER
  -> WHERE Selling_Price > 2000 AND Selling_Price <= 5000;
```

Product_Name	Selling_Price
Printer	5000.00

```
1 row in set (0.00 sec)
```

  

```
mysql> ALTER TABLE PRODUCT_MASTER
  -> ADD NEW_PRICE DECIMAL(10, 2);
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

9. Consider the schema for College Database:

STUDENT(USN, SName, Address, Phone, Gender)

SEMSEC(SSID, Sem, Sec)

CLASS(USN, SSID)

COURSE(Subcode, Title, Sem, Credits)

IAMARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)

Write SQL queries to

1. List all the student details studying in fourth semester 'C' section.
  2. Compute the total number of male and female students in each semester and in each section.
  3. Create a view of Test1 marks of student USN '1BI15CS101' in all Courses.
  4. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.
  5. Categorize students based on the following criterion:  
 If FinalIA = 17 to 20 then CAT = 'Outstanding'  
 If FinalIA = 12 to 16 then CAT = 'Average'  
 If FinalIA < 12 then CAT = 'Weak'
- Give these details only for 8th semester A, B, and C section students.

```
mysql> use student;
Database changed
mysql> CREATE TABLE STUDENTT (
  ->   USN VARCHAR(10) PRIMARY KEY,
  ->   SName VARCHAR(50),
  ->   Address VARCHAR(100),
  ->   Phone VARCHAR(15),
  ->   Gender CHAR(1));
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE SEMSEC (
  ->   SSID INT PRIMARY KEY,
  ->   Sem INT,
  ->   Sec CHAR(1)
  -> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE CLASS (
  ->   USN VARCHAR(10),
  ->   SSID INT,
  ->   PRIMARY KEY (USN, SSID),
  ->   FOREIGN KEY (USN) REFERENCES STUDENTT(USN),
  ->   FOREIGN KEY (SSID) REFERENCES SEMSEC(SSID)
  -> );
Query OK, 0 rows affected (0.07 sec)

mysql> CREATE TABLE COURSE (
  ->   Subcode VARCHAR(10) PRIMARY KEY,
  ->   Title VARCHAR(100),
  ->   Sem INT,
  ->   Credits INT
  -> );
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> CREATE TABLE IAMARKS (  
->     USN VARCHAR(10),  
->     Subcode VARCHAR(10),  
->     SSID INT,  
->     Test1 INT,  
->     Test2 INT,  
->     Test3 INT,  
->     FinalIA DECIMAL(5, 2),  
->     PRIMARY KEY (USN, Subcode, SSID),  
->     FOREIGN KEY (USN) REFERENCES STUDENTT(USN),  
->     FOREIGN KEY (Subcode) REFERENCES COURSE(Subcode),  
->     FOREIGN KEY (SSID) REFERENCES SEMSEC(SSID)  
-> );
```

Query OK, 0 rows affected (0.06 sec)

```
mysql> SELECT S.*  
-> FROM STUDENTT S  
-> JOIN CLASS C ON S.USN = C.USN  
-> JOIN SEMSEC SS ON C.SSID = SS.SSID  
-> WHERE SS.Sem = 4 AND SS.Sec = 'C';
```

Empty set (0.01 sec)

```
mysql> SELECT SS.Sem, SS.Sec, S.Gender, COUNT(*) AS Total_Students  
-> FROM STUDENTT S  
-> JOIN CLASS C ON S.USN = C.USN  
-> JOIN SEMSEC SS ON C.SSID = SS.SSID  
-> GROUP BY SS.Sem, SS.Sec, S.Gender;
```

Empty set (0.01 sec)

```
mysql> CREATE VIEW Test1Marks AS  
-> SELECT IAM.Subcode, C.Title, IAM.Test1  
-> FROM IAMARKS IAM  
-> JOIN COURSE C ON IAM.Subcode = C.Subcode  
-> WHERE IAM.USN = '1BI15CS101';
```

Query OK, 0 rows affected (0.02 sec)

```
mysql> UPDATE IAMARKS
    -> SET FinalIA = (Test1 + Test2 + Test3 - LEAST(Test1, Test2, Test3)) / 2;
Query OK, 0 rows affected (0.01 sec)
Rows matched: 0  Changed: 0  Warnings: 0

mysql> SELECT S.USN, S.SName, SS.Sem, SS.Sec, IAM.FinalIA,
    -> CASE
    ->     WHEN IAM.FinalIA BETWEEN 17 AND 20 THEN 'Outstanding'
    ->     WHEN IAM.FinalIA BETWEEN 12 AND 16 THEN 'Average'
    ->     WHEN IAM.FinalIA < 12 THEN 'Weak'
    ->     ELSE 'Undefined'
    -> END AS CAT
    -> FROM STUDENTT S
    -> JOIN CLASS C ON S.USN = C.USN
    -> JOIN SEMSEC SS ON C.SSID = SS.SSID
    -> JOIN IAMARKS IAM ON S.USN = IAM.USN
    -> WHERE SS.Sem = 8 AND SS.Sec IN ('A', 'B', 'C');
Empty set (0.00 sec)

mysql> desc studentt;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| USN    | varchar(10)   | NO   | PRI | NULL    |       |
| SName  | varchar(50)   | YES  |     | NULL    |       |
| Address | varchar(100)  | YES  |     | NULL    |       |
| Phone  | varchar(15)   | YES  |     | NULL    |       |
| Gender | char(1)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

10. Consider the schema for Company Database:EMPLOYEE (SSN, Name, Address, Sex, Salary, SuperSSN, DNo), DEPARTMENT (DNo, DName, MgrSSN, MgrStartDate), DLOCATION (DNo,DLoc) PROJECT (PNo, PName, PLocation, DNo),WORKS\_ON (SSN, PNo, Hours)

Write SQL queries to

1. Make a list of all project numbers for projects that involve an employee whose last name is 'Scott',either as a worker or as a manager of the department that controls the project.
2. Show the resulting salaries if every employee working on the 'IoT' project is given a 10 percent raise.
3. Find the sum of the salaries of all employees of the 'Accounts' department, as well as the maximum salary, the minimum salary, and the average salary in this department.
4. Retrieve the name of each employee who works on all the projects controlled by department number 5.
5. For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6,00,000.

Database changed

```
mysql> CREATE TABLE EMPLOYEE (  
->     SSN VARCHAR(9) PRIMARY KEY,  
->     Name VARCHAR(100),  
->     Address VARCHAR(255),  
->     Sex CHAR(1),  
->     Salary DECIMAL(10, 2),  
->     SuperSSN VARCHAR(9),  
->     DNo INT);
```

Query OK, 0 rows affected (0.08 sec)

```
mysql> CREATE TABLE DEPARTMENT (  
->     DNo INT PRIMARY KEY,  
->     DName VARCHAR(100),  
->     MgrSSN VARCHAR(9),  
->     MgrStartDate DATE);
```

Query OK, 0 rows affected (0.06 sec)

```
mysql> CREATE TABLE DLOCATION (  
->     DNo INT,  
->     DLoc VARCHAR(100),  
->     PRIMARY KEY (DNo, DLoc));
```

Query OK, 0 rows affected (0.06 sec)

```
mysql> CREATE TABLE PROJECT (  
->     PNo INT PRIMARY KEY,  
->     PName VARCHAR(100),  
->     PLocation VARCHAR(100),  
->     DNo INT);
```

Query OK, 0 rows affected (0.04 sec)



```

mysql> CREATE TABLE WORKS_ON (
    ->     SSN VARCHAR(9),                -- Employee SSN
    ->     PNo INT,                       -- Project Number
    ->     Hours DECIMAL(5, 2),           -- Hours worked on the project
    ->     PRIMARY KEY (SSN, PNo),        -- Composite Primary Key (SSN and P
    ->     FOREIGN KEY (SSN) REFERENCES EMPLOYEE(SSN), -- Foreign Key to EMPLOYEE
    ->     FOREIGN KEY (PNo) REFERENCES PROJECT(PNo));
Query OK, 0 rows affected (0.06 sec)

mysql> -- Inserting data into DEPARTMENT
mysql> INSERT INTO DEPARTMENT (DNo, DName, MgrSSN, MgrStartDate) VALUES
    -> (1, 'HR', '123456789', '2015-01-01'),
    -> (2, 'Engineering', '987654321', '2018-06-15');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql>
mysql> -- Inserting data into EMPLOYEE
mysql> INSERT INTO EMPLOYEE (SSN, Name, Address, Sex, Salary, SuperSSN, DNo) VALUES
    -> ('123456789', 'John Doe', '123 Main St', 'M', 60000, NULL, 1),
    -> ('987654321', 'Jane Smith', '456 Oak St', 'F', 75000, '123456789', 2);
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql>
mysql> -- Inserting data into DLOCATION
mysql> INSERT INTO DLOCATION (DNo, DLoc) VALUES
    -> (1, 'New York'),
    -> (2, 'San Francisco');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

```

```

mysql>
mysql> -- Inserting data into PROJECT
mysql> INSERT INTO PROJECT (PNo, PName, PLocation, DNo) VALUES
    -> (101, 'Project A', 'New York', 1),
    -> (102, 'Project B', 'San Francisco', 2);
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql>
mysql> -- Inserting data into WORKS_ON
mysql> INSERT INTO WORKS_ON (SSN, PNo, Hours) VALUES
    -> ('123456789', 101, 40),
    -> ('987654321', 102, 35);
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> -- Query to find project numbers for projects involving an employee whose last name is 'Scott'
mysql> SELECT DISTINCT W.PNo
    -> FROM WORKS_ON W
    -> JOIN EMPLOYEE E ON W.SSN = E.SSN
    -> WHERE E.Name LIKE '%Scott%'
    -> UNION
    -> SELECT DISTINCT P.PNo
    -> FROM PROJECT P
    -> JOIN DEPARTMENT D ON P.DNo = D.DNo
    -> JOIN EMPLOYEE E ON D.MgrSSN = E.SSN
    -> WHERE E.Name LIKE '%Scott%';
Empty set (0.00 sec)

mysql> SELECT E.Name, E.Salary * 1.10 AS NewSalary
    -> FROM EMPLOYEE E
    -> JOIN WORKS_ON W ON E.SSN = W.SSN
    -> JOIN PROJECT P ON W.PNo = P.PNo
    -> WHERE P.PName = 'IoT';
Empty set (0.00 sec)

```

```

-> WHERE D.DName = 'ACCOUNTS' ,
+-----+-----+-----+-----+
| TotalSalaries | MaxSalary | MinSalary | AvgSalary |
+-----+-----+-----+-----+
|          NULL |        NULL |        NULL |        NULL |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

```

mysql> SELECT E.Name
-> FROM EMPLOYEE E
-> JOIN WORKS_ON W ON E.SSN = W.SSN
-> JOIN PROJECT P ON W.PNo = P.PNo
-> WHERE P.DNo = 5
-> GROUP BY E.Name
-> HAVING COUNT(DISTINCT P.PNo) = (SELECT COUNT(*) FROM PROJECT WHERE DNo = 5);
Empty set (0.01 sec)

```

```

mysql> SELECT E.DNo, COUNT(*) AS HighSalaryEmployees
-> FROM EMPLOYEE E
-> WHERE E.Salary > 600000
-> GROUP BY E.DNo
-> HAVING COUNT(E.SSN) > 5;
Empty set (0.00 sec)

```

```

mysql> desc employee;

```

```

+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| SSN   | varchar(9) | NO | PRI | NULL | |
| Name  | varchar(100) | YES | | NULL | |
| Address | varchar(255) | YES | | NULL | |
| Sex   | char(1) | YES | | NULL | |
| Salary | decimal(10,2) | YES | | NULL | |
| SuperSSN | varchar(9) | YES | | NULL | |
| DNo   | int | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

```

```
mysql> select*from department;
```

DNo	DName	MgrSSN	MgrStartDate
1	HR	123456789	2015-01-01
2	Engineering	987654321	2018-06-15

2 rows in set (0.00 sec)

```
mysql> select*from employee;
```

SSN	Name	Address	Sex	Salary	SuperSSN	DNo
123456789	John Doe	123 Main St	M	60000.00	NULL	1
987654321	Jane Smith	456 Oak St	F	75000.00	123456789	2

2 rows in set (0.00 sec)

```
mysql> select*from dlocation;
```

DNo	DLoc
1	New York
2	San Francisco

2 rows in set (0.00 sec)

```
mysql> select*from project;
```

PNo	PName	PLocation	DNo
101	Project A	New York	1
102	Project B	San Francisco	2

2 rows in set (0.00 sec)