MODULE 4 :- Introduction to Database

SQL Queries

1. Create Table Name : Student and Exam

Primary Key

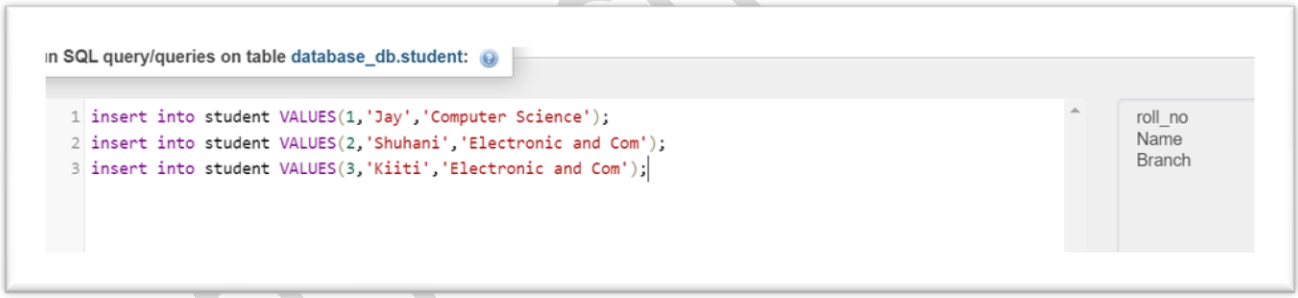
Student

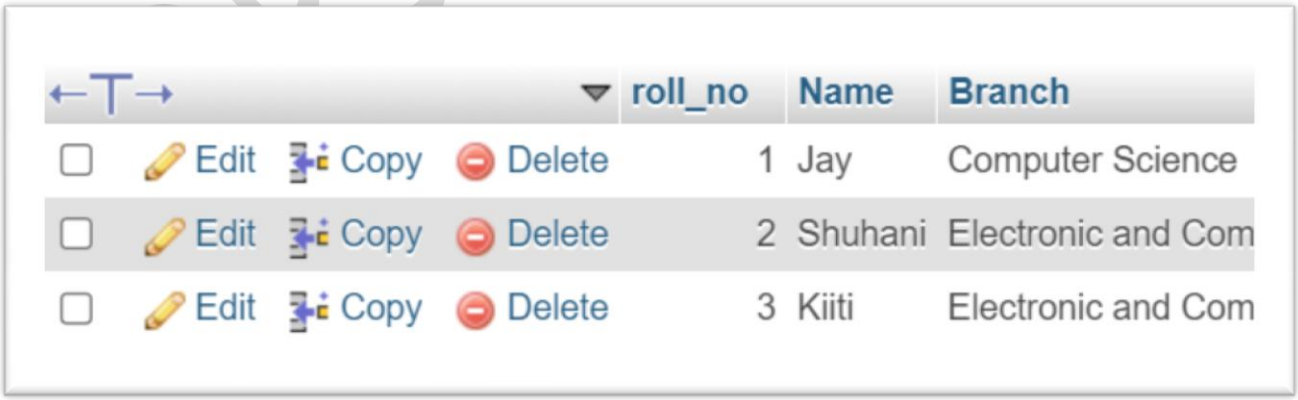
Rollno	Name	Branch
1	Jay	Computer Science
2	Suhani	Electronic and Com
3	Kriti	Electronic and Com

Foreign Key

Exam

Rollno	S_code	Marks	P_code
1	CS11	50	CS
1	CS12	60	CS
2	EC101	66	EC
2	EC102	70	EC
3	EC101	45	EC
3	EC102	50	EC





Run SQL query/queries on database database_db:

```
1 create TABLE Exam
2 (
3     Roll_no int,
4     S_code varchar(20),
5     Marks int,
6     P_code varchar(20),
7     FOREIGN KEY(roll_no) REFERENCES student(roll_no
8     );
```

```
1 insert into exam VALUES(1,'CS11',50,'CS');
2 insert into exam VALUES(1,'CS12',60,'CS');
3 insert into exam VALUES(2,'EC101',66,'EC');
4 insert into exam VALUES(2,'EC102',70,'EC');
5 insert into exam VALUES(3,'EC101',45,'EC');
6 insert into exam VALUES(3,'EC102',50,'EC');
```

Roll_no
S_code
Marks
P_code

Extra options

Roll_no	S_code	Marks	P_code
1	CS11	50	CS
1	CS12	60	CS
2	EC101	66	EC
2	EC102	70	EC
3	EC101	45	EC
3	EC102	50	EC

2. Create table given below: Employee and IncentiveTable

Employee_id	First_name	Last_name	Salary	Joining_date	Department
1	John	Abraham	1000000	01-JAN-13 12.00.00 AM	Banking
2	Michael	Clarke	800000	01-JAN-13 12.00.00 AM	Insurance
3	Roy	Thomas	700000	01-FEB-13 12.00.00 AM	Banking
4	Tom	Jose	600000 .	01-FEB-13 12.00.00 AM	Insurance
5	Jerry	Pinto	650000	01-FEB-13 12.00.00 AM	Insurance
6	Philip	Mathew	750000	01-JAN-13 12.00.00 AM	Services
7	TestName1	123	650000	01-JAN-13 12.00.00 AM	Services
8	TestName2	Lname%	600000	01-FEB-13 12.00.00 AM	Insurance

Name: Employee

Table Name: Incentive

Employee_ref_id	Incentive_date	Incentive_amount
1	01-FEB-13	5000
2	01-FEB-13	3000
3	01-FEB-13	4000
1	01-JAN-13	4500
2	01-JAN-13	3500

Run SQL query/queries on database database_db:

```
1 create TABLE Employee
2 (
3     Emp_id int PRIMARY KEY,
4     First_Name varchar(20),
5     Last_Name varchar(20),
6     Salary int,
7     Joining_date varchar(30),
8     Department varchar(20)
9     );
```

Run SQL query/queries on database database_db:

```
insert into employee VALUES(1,'John','Abraham',1000000,'01-JAN-13 12.00.00 AM','Banking');
insert into employee VALUES(2,'Michael','Clerke',800000,'01-JAN-13 12.00.00 AM','Insurance');
insert into employee VALUES(3,'Roy','Thomas',700000,'01-FEB-13 12.00.00 AM','Banking');
insert into employee VALUES(4,'Tom','Jose',600000,'01-FEB-13 12.00.00 AM','Insurance');
insert into employee VALUES(5,'Jerry','Pinto',6500000,'01-FEB-13 12.00.00 AM','Insurance');
insert into employee VALUES(6,'JPhilip','Mathew',750000,'01-JAN-13 12.00.00 AM','Service');
insert into employee VALUES(7,'TestName1','123',650000,'01-JAN-13 12.00.00 AM','Service');
insert into employee VALUES(8,'TestName2','LName%',600000,'01-FEB-13 12.00.00 AM','Insurance');
```

mp_id	First_Name	Last_Name	Salary	Joining_date	Department
1	John	Abraham	1000000	01-JAN-13 12.00.00 AM	Banking
2	Michael	Clerke	800000	01-JAN-13 12.00.00 AM	Insurance
3	Roy	Thomas	700000	01-FEB-13 12.00.00 AM	Banking
4	Tom	Jose	600000	01-FEB-13 12.00.00 AM	Insurance
5	Jerry	Pinto	6500000	01-FEB-13 12.00.00 AM	Insurance
6	JPhilip	Mathew	750000	01-JAN-13 12.00.00 AM	Service
7	TestName1	123	650000	01-JAN-13 12.00.00 AM	Service
8	TestName2	LName%	600000	01-FEB-13 12.00.00 AM	Insurance

Run SQL query/queries on database database_db:

```
1 create TABLE incentive
2 (
3     Emp_id int,
4     Incentive_date varchar(20),
5     Incentive_amount int,
6     FOREIGN KEY(emp_id) REFERENCES employee(emp_id)
7     );
```

Run SQL query/queries on database database_db:

```
1 INSERT into incentive VALUES(1,'01-FEB-13',5000);
2 INSERT into incentive VALUES(2,'01-FEB-13',3000);
3 INSERT into incentive VALUES(3,'01-FEB-13',4000);
4 INSERT into incentive VALUES(1,'01-JAN-13',4500);
5 INSERT into incentive VALUES(2,'01-JAN-13',3500);
```

Extra options

Emp_id	Incentive_date	Incentive_amount
1	01-FEB-13	5000
2	01-FEB-13	3000
3	01-FEB-13	4000
1	01-JAN-13	4500
2	01-JAN-13	3500

3. Get First_Name from employee table using Tom name “Employee Name”.

Answer : `SELECT First_Name from employee where first_name='Tom';`

First_Name
Tom

4. Get FIRST_NAME, Joining Date, and Salary from employee table.

Answer : `SELECT first_name,Joining_date,salary from employee;`

first_name	Joining_date	salary
John	01-JAN-13 12.00.00 AM	1000000
Michael	01-JAN-13 12.00.00 AM	800000
Roy	01-FEB-13 12.00.00 AM	700000
Tom	01-FEB-13 12.00.00 AM	600000
Jerry	01-FEB-13 12.00.00 AM	6500000
JPhilip	01-JAN-13 12.00.00 AM	750000
TestName1	01-JAN-13 12.00.00 AM	650000
TestName2	01-FEB-13 12.00.00 AM	600000

5. Get all employee details from the employee table order by First_Name Ascending and Salary descending?

Answer : `SELECT * FROM `employee` ORDER BY First_Name,Salary DESC;`

Emp_id	First_Name ▲ 1	Last_Name	Salary ▼ 2	Joining_date	Department
5	Jerry	Pinto	6500000	01-FEB-13 12.00.00 AM	Insurance
1	John	Abraham	1000000	01-JAN-13 12.00.00 AM	Banking
6	JPhilip	Mathew	750000	01-JAN-13 12.00.00 AM	Service
2	Michael	Clerke	800000	01-JAN-13 12.00.00 AM	Insurance
3	Roy	Thomas	700000	01-FEB-13 12.00.00 AM	Banking
7	TestName1	123	650000	01-JAN-13 12.00.00 AM	Service
8	TestName2	LName%	600000	01-FEB-13 12.00.00 AM	Insurance
4	Tom	Jose	600000	01-FEB-13 12.00.00 AM	Insurance

6. Get employee details fromemployee table whose first name contains ‘J’.

Answer :
```
SELECT * FROM `employee` where First_Name like 'j%';
```

Emp_id	First_Name	Last_Name	Salary	Joining_date	Department
1	John	Abraham	1000000	01-JAN-13 12.00.00 AM	Banking
5	Jerry	Pinto	6500000	01-FEB-13 12.00.00 AM	Insurance
6	JPhilip	Mathew	750000	01-JAN-13 12.00.00 AM	Service

7. Get department wise maximum salary from employee table order by salary ascending.

Answer :
```
SELECT department,MAX(Salary) as 'Maximum
Salary' from employee GROUP BY Department ORDER BY Salary;
```

department	Maximum Salary
Service	750000
Insurance	6500000
Banking	1000000

9. Select first_name, incentive amount from employee and incentives table forthose employees who have incentives and incentive amount greater than 3000.

Answer :
```
SELECT e.first_name, i.incentive_amount from employee e INNER JOIN inc
entive i on e.emp_id=i.Emp_id WHERE i.Incentive_amount > 3000;
```

Extra options

first_name	incentive_amount
John	5000
Roy	4000
John	4500
Michael	3500

10. Create After Insert trigger on Employee table which insert records in view table.

Answer :-

```
create table viewtable
(
    id int,
    First_name varchar(30),
    Last_name varchar(30),
    salary int,
    Joining_date varchar(30),
    depart varchar(30),
    date_time timestamp,
    task text

    );
```

```
Create trigger tri_employee after insert on employee
For EACH ROW
BEGIN
     Insert into viewtable (id, first_name, last_name, salary, joining_date, depart, task)
     VALUES (new.emp_id, new.first_name, new.last_name,
new.salary,new.joining_date,new.department,'Inserted Successfully');
END
```

Extra options

id	First_name	Last_name	salary	Joining_date	depart	date_time	task
10	Amanhusain	Ansari	30000	01-FEB_13	CS	2024-10-14 17:11:23	inserted Successfully

11. Create table given below: Salesperson and Customer

TABLE-1

TABLE NAME- SALSEPERSON

(PK)SNo	SNAME	CITY	COMM
1001	Peel	London	.12
1002	Serres	San Jose	.13
1004	Motika	London	.11
1007	Rafkin	Barcelona	.15
1003	Axelrod	New York	.1

TABLE-2

TABLE NAME- CUSTOMER

(PK)CNM.	CNAME	CITY	RATING	(FK)SNo
201	Hoffman	London	100	1001
202	Giovanne	Roe	200	1003
203	Liu	San Jose	300	1002
204	Grass	Barcelona	100	1002
206	Clemens	London	300	1007
207	Pereira	Roe	100	1004

```
CREATE TABLE salesperson
(
    SNo int PRIMARY KEY,
    Sname varchar(20),
    City varchar(20),
    COMM float

    );
```

```
INSERT INTO salesperson VALUES(1001,'Peel','London',12);
INSERT INTO salesperson VALUES(1002,'Serres','San Jose',13);
INSERT INTO salesperson VALUES(1004'Motika','London',11);
INSERT INTO salesperson VALUES(1007,'Rafkin','Barcelona',15);
INSERT INTO salesperson VALUES(1003,'Axelrod','New York',1);
```

SNO	SNAME	CITY	COMM
1001	Peel	London	12
1002	Serres	San Jose	13
1003	Axelord	New York	1
1004	Motika	London	11
1007	Rafkin	Barcelona	15

```sql
create table Customer
(
    CNM int PRIMARY KEY,
    CNAME varchar(20),
    CITY varchar(20),
    RATING int,
    SNO int,
    FOREIGN KEY(SNO) REFERENCES salesperson(SNO)
    );
```

```sql
insert into customer VALUES(201,'Hoffman','London',100,1001);
insert into customer VALUES(202,'Giovanne','Roe',200,1003);
insert into customer VALUES(203,'Liu','San Jose',300,1002);
insert into customer VALUES(204,'Grass','Barcelona',100,1002);
insert into customer VALUES(206,'Ciemens','London',300,1007);
insert into customer VALUES(207,'Pereira','Roe',100,1004);
```

CNM	CNAME	CITY	RATING	SNO
201	Hoffman	London	100	1001
202	Giovanne	Roe	200	1003
203	Liu	San Jose	300	1002
204	Grass	Barcelona	100	1002
206	Ciemens	London	300	1007
207	Pereira	Roe	100	1004

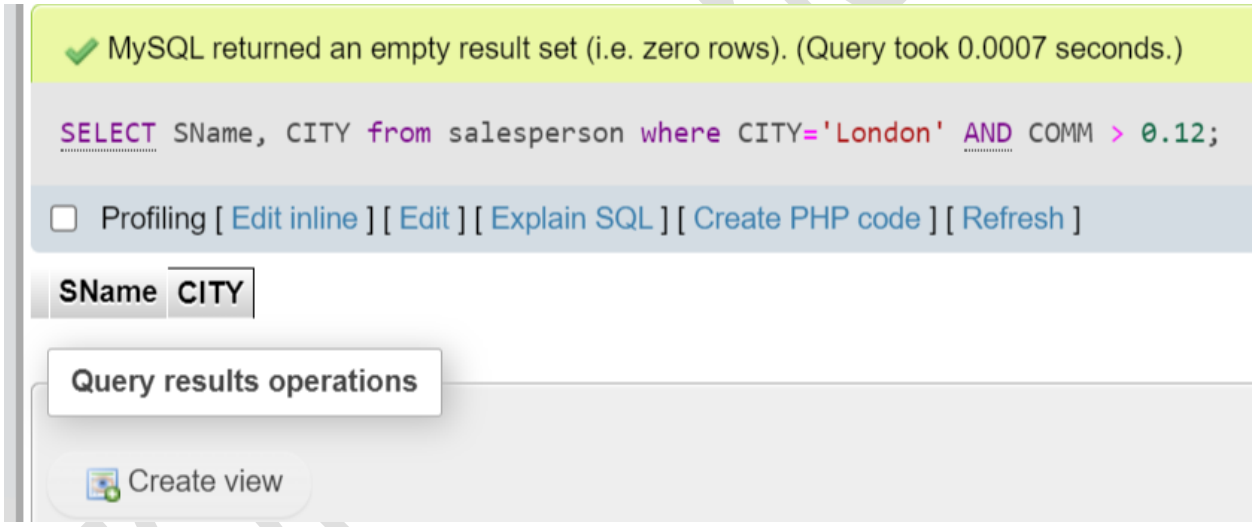
13. All Customer name whose rating is more than 100.

Answer :-
```
SELECT * FROM customer where RATING > 100;
```

CNM	CNAME	CITY	RATING	SNO
202	Giovanne	Roe	200	1003
203	Liu	San Jose	300	1002
206	Ciemens	London	300	1007

14.Names and cities of all salespeople in London with commission above 0.12.

Answer :-
```
SELECT SName, CITY from salesperson where CITY='London' AND COMM > 0.12;
```



15.All salespeople either in Barcelona or in London.

Answer :-
```
SELECT * from salesperson WHERE CITY = 'Barcelona' OR CITY = 'London';
```

SNO	SNAME	CITY	COMM
1001	Peel	London	0.12
1004	Motika	London	0.11
1007	Rafkin	Barcelona	0.15

16. All salespeople with commission between 0.10 and 0.12. (Boundary values should be excluded).

Answer :-
```
SELECT * from salesperson where COMM BETWEEN 0.10 AND 0.12;
```

SNO	SNAME	CITY	COMM
1001	Peel	London	0.12
1003	Axelord	New York	0.1
1004	Motika	London	0.11

17. All customers excluding those with rating <= 100 unless they are located in Rome.

Answer :-
```
SELECT * FROM customer WHERE RATING <= 100 AND CITY = 'Roe';
```

CNM	CNAME	CITY	RATING	SNO
207	Pereira	Roe	100	1004

18. Write a SQL statement that displays all the information about all salespeople.

Answer :-

```
salesman_id |    name     |   city    | commission
-------------------------------+---------------------------+-----------------------+----------------------------
5001 | James Hoog | New York |        0.15
5002 | Nail Knite | Paris    |        0.13
5005 | Pit Alex   | London   |        0.11
5006 | Mc Lyon    | Paris    |        0.14
5007 | Paul Adam  | Rome     |        0.13
5003| Lauson Hen | San Jose |        0.12
```

```sql
create TABLE salespeople
(
    salesman_id int PRIMARY KEY,
    name varchar(20),
    city varchar(20),
    commission float
    );
```

```sql
insert into salespeople VALUES (5001,'James Hong','New York',0.15);
insert into salespeople VALUES (5002,'Nail nite','Paris',0.13);
insert into salespeople VALUES (5005,'Pit Alex','London',0.11);
insert into salespeople VALUES (5006,'Mc Lyon','Paris',0.14);
insert into salespeople VALUES (5007,'Paul Adam','Rome',0.13);
insert into salespeople VALUES (5003,'Lauson Hen','San Jose',0.12);
```

salesman_id	name	city	commission
5001	James Hong	New York	0.15
5002	Nail nite	Paris	0.13
5003	Lauson Hen	San Jose	0.12
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13

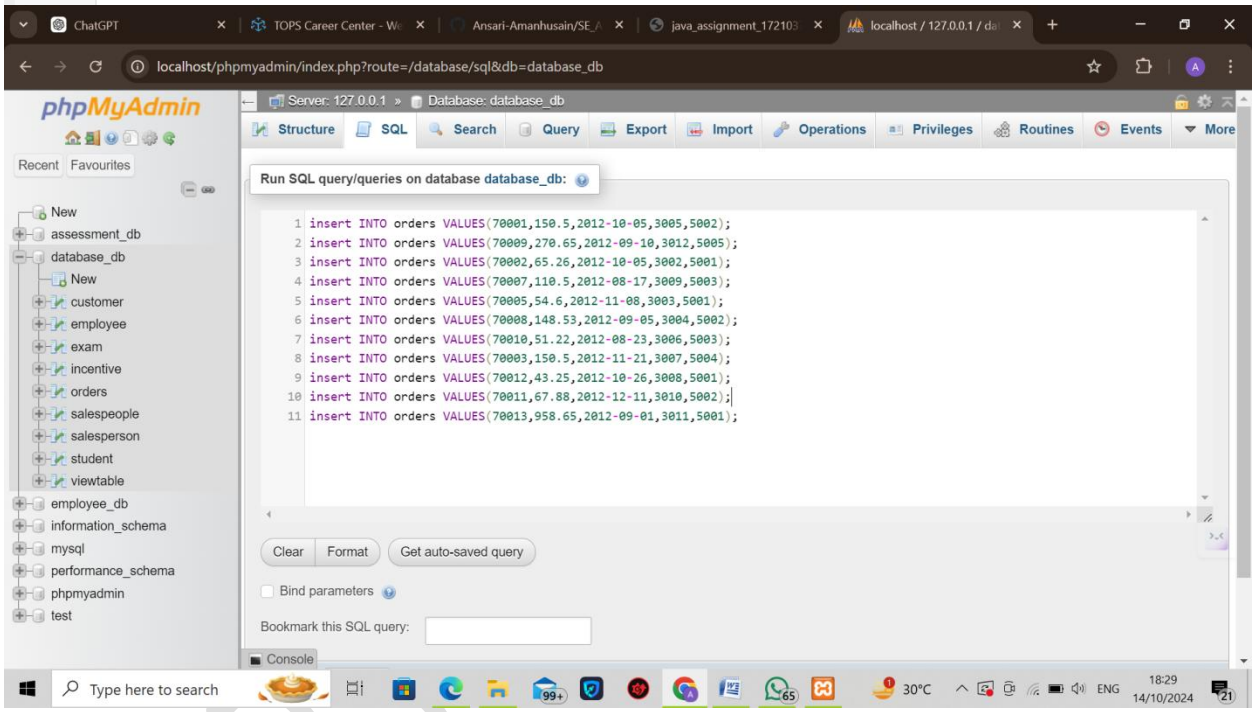
19. From the following table, write a SQL query to find orders that are delivered by a salesperson with ID. 5001. Return ord_no, ord_date, purch_amt.

Answer :-

MODULE 4 :- Introduction to Database

```sql
CREATE TABLE Orders
(
    ord_no int,
    purch_amt float,
    ord_date date,
    customer_id int,
    salesman_id int,
    PRIMARY KEY(ord_no), FOREIGN KEY(salesman_id) REFERENCES salespeople(salesman_id)

    );
```



ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	1997	3005	5002
70002	65.26	1997	3002	5001
70005	54.6	1993	3003	5001
70007	110.5	1987	3009	5003
70008	148.53	1998	3004	5002
70009	270.65	1993	3012	5005
70010	51.22	1981	3006	5003

```
SELECT ord_no,ord_date,purch_amt from orders WHERE salesman_id=5001;
```

ord_no	ord_date	purch_amt
70002	1997	65.26
70005	1993	54.6

20. From the following table, write a SQL query to select a range of products whose price is in the range Rs.200 to Rs.600. Begin and end values are included. Return pro_id, pro_name, pro_price, and pro_com.

Sample table: item_mast

PRO_ID	PRO_NAME	PRO_PRICE	PRO_COM
101	Mother Board	3200.00	15
102	Key Board	450.00	16
103	ZIP drive	250.00	14
104	Speaker	550.00	16
105	Monitor	5000.00	11
106	DVD drive	900.00	12
107	CD drive	800.00	12
108	Printer	2600.00	13
109	Refill cartridge	350.00	13
110	Mouse	250.00	12

Answer :-

```
create TABLE item_mast
(
    pro_id int PRIMARY KEY,
    pro_name varchar(30),
    pro_price int,
    pro_comm int
    );
```

```
insert INTO item_mast VALUES (101,'Mother Board',3200,15);
insert INTO item_mast VALUES (102,'Key Board',450,16);
insert INTO item_mast VALUES (103,'Zip drive',250,14);
insert INTO item_mast VALUES (104,'Speaker',550,16);
insert INTO item_mast VALUES (105,'Monitor',5000,11);
insert INTO item_mast VALUES (106,'DVD drive',900,12);
insert INTO item_mast VALUES (107,'CD drive',800,12);
insert INTO item_mast VALUES (108,'Printer',2600,13);
insert INTO item_mast VALUES (109,'Refill cartridge',350,13);
insert INTO item_mast VALUES (110,'Mouse',250,12);
```

```
SELECT * FROM `item_mast` WHERE pro_price BETWEEN 200 AND 600;
```

pro_id	pro_name	pro_price	pro_comm
102	Key Board	450	16
103	Zip drive	250	14
104	Speaker	550	16
109	Refill cartridge	350	13
110	Mouse	250	12

21. From the following table, write a SQL query to calculate the average price for a manufacturer code of 16. Return avg.

Answer :-

```
SELECT AVG(pro_price) AS average from item_mast where pro_comm=16;
```

average
500.0000

22. From the following table, write a SQL query to display the pro_name as 'Item Name' and pro_priceas 'Price in Rs.'

Answer :-

```
SELECT pro_name AS "item_name", pro_price AS "Price in
RS" from item_mast;
```

item_name	Price in RS
Mother Board	3200
Key Board	450
Zip drive	250
Speaker	550
Monitor	5000
DVD drive	900
CD drive	800
Printer	2600
Refill cartridge	350
Mouse	250

23. From the following table, write a SQL query to find the items whose prices are higher than or equal to $250. Order the result by product price in descending, then product name in ascending. Return pro_name and pro_price.

Answer :-

```
SELECT pro_name, pro_price from item_mast where pro_price >= 250 order
 BY pro_name, pro_price DESC;
```

pro_name ▲ 1	pro_price ▼ 2
CD drive	800
DVD drive	900
Key Board	450
Monitor	5000
Mother Board	3200
Mouse	250
Printer	2600
Refill cartridge	350
Speaker	550
Zip drive	250

24. From the following table, write a SQL query to calculate average price of the items for each company. Return average price and company code.

Answer :-

```
SELECT pro_comm AS "Company code", AVG(pro_price) AS "Average
Price" FROM item_mast GROUP BY pro_comm;
```

Company code	Average Price
11	5000.0000
12	650.0000
13	1475.0000
14	250.0000
15	3200.0000
16	500.0000