1. Create Table Name : Student and Exam

Student table:-

```
        CREATE TABLE student(ROllno int,Name char(20),Branch char(30));
ALTER table student1 add PRIMARY KEY(ROllno);
```

Insert data:-

```
        INSERT into student VALUES
        (1,'jay','computer science'),
        (2,'suhani','electronic and com'),
        (3,'kriti','electronic and com');
```

Exam table:-

```
        CREATE table exam(ROllno int,S_code varchar(20),Marks int,P_code
varchar(20));
```

Add Foreign key:-

```
        ALTER TABLE exam add FOREIGN key (Rollno) REFERENCES student2
(ROllno);
```

Insert data:-

```
        INSERT INTO exam VALUES ('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

Create table employee:-

```
        CREATE table employee (employee_id int PRIMARY KEY
AUTO_INCREMENT,first_name char(20),last_name varchar(20),salary
int,joining_date datetime,deparment char(30));
```

Insert data:-

```
        INSERT INTO `employee` (`first_name`, `last_name`, `salary`,
`joining_date`, `deparment`) VALUES
('john','abraham',1000000,'2013-01-01 12:00:00','Banking'),
('michael','clarke',800000,'2013-01-01 12:00:00','Insurance'),
('Roy','Thomas',700000,'2013-02-01 12:00:00','Banking'),
('Tom','Jose',600000,'2013-02-01 12:00:00','Insurance'),
('Jerry','pinro',650000,'2013-02-01 12:00:00','Insurance'),
```

('Philip','Mathew',750000,'2013-01-01 12:00:00','Services'),
('TestName1','123',650000,'2013-01-01 12:00:00','Services'),
('TestName2','Lname%',600000,'2013-02-01 12:00:00','Insuance');

Create table incentive:-

CREATE TABLE Incentive (employee_ref_id int,Incentive_dare datetime,Incentive_amount int);

Insert data:-

INSERT INTO incentive VALUES (1,'2013-02-01','5000'), (2,'2013-02-01',3000), (3,'2013-02-01',4000), (1,'2013-01-01',4500), (2,'2013-01-01',3500);

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

   SELECT first_name FROM employee WHERE first_name='tom'

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

   SELECT first_name,joining_date,salary from employee

5. Get all employee details from the employee table order by First_Name

   SELECT * FROM employee ORDER by first_name asc ,salary DESC

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

   SELECT * from employee WHERE first_name LIKE 'j%'

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

   SELECT MAX(salary) from employee ORDER BY salary AS;

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

   SELECT first_name from employee WHERE salary>3000;

9. Create After Insert trigger on Employee table which insert records in viewtable

```
DELIMITER $$
CREATE TRIGGER em AFTER INSERT on employee FOR EACH ROW
BEGIN
INSERT into viewtable set id=new.employee_id,
f_name=new.first_name,
l_name=new.last_name,
salary=new.salary,
jo_date=new.joining_date,
deparment=new.deparment;
END $$
```

10. Create table given below: Salesperson and Customer

Create table salesperson

```
CREATE table Salesperson(SNO int PRIMARY KEY
AUTO_INCREMENT,SNAME CHAR(20),CITY CHAR(30),COMM
VARCHAR(20));
```

Insert data

```
INSERT INTO `salesperson` (`SNAME`, `CITY`, `COMM`) VALUES
('peel','london','.12'), ('serres','sanjose','.13'), ('motike','london','.11'),
('rafkin','barcelona','.15'), ('axeirod','new york','.1');
```

Create table CUSTOMER

```
CREATE TABLE CUSTOMER(CNM INT PRIMARY KEY,CNAME
CHAR(20),CITY CHAR(30),RATING INT,SNO INT);
```

Add foreign

```
ALTER TABLE customer ADD FOREIGN key(sno) REFERENCES
salesperson (sno);
```

Insert data

```
INSERT INTO `customer` VALUES
(‘201’,'Hoffman','London',’100’,’1001’),
```

('202','Giovanne','Roe','200','1003'), ('203','Liu','Aan jose','300','1002'), ('204','Grass','Barcelona','100','1002'), ('206','Clemens','London','300','1007'), ('207','Pereira','Roe','100','1004');

11. All orders for more than $1000.

```
SELECT * from orders WHERE purch_amt>1000;
```

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

```
Select SNAME,CITY from Salesperson where COMM>0.12
```

13. All salespeople either in Barcelona or in Londo

```
Select * from Salesperson where  CITY='london' OR
CITY='barcelona';
```

14. All salespeople with commission between 0.10 and 0.12. (Boundary valuesshould be excluded)

```
Select * from Salesperson where comm between 0.10 and
0.12;
```

15. All customers excluding those with rating <= 100 unless they are located inRome

```
SELECT * FROM customer WHERE RATING<=100 AND
CITY='roe';
```

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

```
SELECT * FROM salesperson;
```

17. 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.

```
SELECT ord_no,order_date,purch_amt FROM orders
WHERE salesman_id=5001;
```

18. 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.

```
Select * from item_mast where pro_price between 200 and
600;
```

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

```
Select avg(pro_price) from item_mast where pro_com = 16;
```

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

```
SELECT pro_name as item_name,'pro price' as 'prise in Rs' FROM
item_mast
```

21. 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.

```
  Select pro_name, pro_price from item_mast where pro_price >= 250
and order by pro_price desc;
```

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

```
Select avg(pro_price),pro_com from item_mast
```