Q.1 Create Table Name : Student and Exam

➔ **Create database:-**

```
Create database school;
```

➔ **Create student table:-**

```
CREATE TABLE student(rollno int PRIMARY KEY,
 name varchar(20),
branch varchar(20));
```

➔ **Insert data in student table:-**

```
INSERT INTO student VALUES(1,"Jay","Computer science"),
(2,"Shuhani","Electronic and com"),
(3,"Krirti","Electronic and com");
```

➔ **Create exam table:-**

```
CREATE TABLE exam( Rollno int,
FOREIGN KEY(Rollno) REFERENCES school(rollno),
s_code varchar(20),
marks  int,
p_code varchar(20));
```

➔ **Insert data in exam table:-**

```
INSERT INTO exam VALUES
(1,"CS11",50,"CS"),
(1,"CS12",50,"CS"),
(2,"EC101",66,"EC"),
```

(2,"EC102",70,"EC"),

(3,"EC101",45,"EC"),

(3,"EC102",50,"EC");

Q.2 Create table given below: Employee and IncentiveTable

➔ **Create database:-**

```
CREATE  DATABASE  employee;
```

➔ **Create employee table:-**

```
CREATE TABLE employee(employee_id int PRIMARY KEY AUTO_INCREMENT,
        first_name varchar(20),
        last_name varchar(20),
        salary bigint,
        joining_date datetime,
        department varchar(20));
```

➔ **Insert data in employee table:-**

```
INSERT INTO employee VALUES
        (1,"john","abraham",1000000,'2013-01-01 12.00.00
AM',"banking"),
        (2,"michael","clarke",800000,'2013-01-01 12.00.00
AM',"insurance"),
        (3,"roy","thoms",700000,'2013-01-01 12.00.00
AM',"banking"),
```

```
                (4,"tom","jose",600000,'2013-01-01 12.00.00
AM',"insurance"),
                (5,"jerry","pinto",650000,'2013-01-01 12.00.00
AM',"insurance"),
                (6,"philip","mathew",750000,'2013-01-01 12.00.00
AM',"services"),
                (7,"testname1",123,650000,'2013-01-01 12.00.00
AM',"services"),
                (8,"testname2","lname%",600000,'2013-01-01 12.00.00
AM',"insurance");
```

➔ **Create incentive table:-**

```
CREATE TABLE incentive( employee_ref_id int  ,
  FOREIGN KEY( employee_ref_id) REFERENCES employee(employee_id),
  incentive_data datetime,
  incentive_amount  bigint
  );
```

➔ **Insert data in incentive table:-**

```
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);
```

Q.3 Get First_Name from employee table using Tom name "Employee Name".

➔ SELECT first_name AS Tom

FROM employee;

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

➔ SELECT first_name,joining_date,salary FROM employee;

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

➔ SELECT * FROM employee ORDER BY first_name ASC;
➔ SELECT * FROM employee ORDER BY salary DESC;

Q.6 Get employee details from employee table whose first name contains 'J'.

➔ SELECT * FROM employee
WHERE first_name LIKE '%J%';

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

➔ SELECT department,MAX(salary) FROM employee;

Q.8 salaryascending?

➔ SELECT * FROM employee ORDER BY salary ASC;

Q.9 Select first_name, incentive amount from employee and incentives table forthose employees who have incentives and incentive amount greater than3000

➔ SELECT first_name,incentive_amount FROM employee,incentive WHERE employee.employee_id = incentive.employee_ref_id and incentive.incentive_amount>3000;

Q.10 Create After Insert trigger on Employee table which insert records inviewtable

```
DELIMITER $$

CREATE TRIGGER trg_after_employee_insert

AFTER INSERT

    ON Employee

FOR EACH ROW

BEGIN

  INSERT INTO InViewTable
(inview_id_id,inview_first_name,inview_last_name,inview_salary,inview_joini
ng_date,inview_department)

  VALUES (NEW.employee_id, NEW.first_name,
NEW.last_name,NEW.salary,NEW.joining_date,NEW.department);

END;
```

Q.11 Create table given below: Salesperson and Customer

➔ **Create database:-**

```
CREATE DATABASE sales;
```

➔ **Create salseperson table:-**

```
CREATE TABLE salesperson(sno int PRIMARY KEY,
                         sname varchar(20),
                         city varchar(20),
                         comm decimal(2,2));
```

➔ **Insert data in salseperson table:-**

➔ INSERT INTO salesperson values(1001,”peel”,”london”,.12),

(1002,”seeres”,”san jose”,.13),

(1004,”motika”,”london”,.11),
(1007,”rafkin”,”barcelona”,.15),
(1003,”axelrod”,”new york”,.1),

➔ **Create customer table:-**

```
CREATE TABLE customer(cnm int PRIMARY KEY, cname varchar(20),city
varchar(20),rating int,sno int,FOREIGN KEY(sno) REFERENCES
salesperson(sno));
```

➔ **Insert data in customer table:-**

```
INSERT INTO customer  VALUES(201,”hoffman”,”london”,100,1001),
                            (202,”glovanne”,”roe”,200,1003),
                            (203,”liu”,”san jose”,300,1002),
                            (204,”grass”,”barcelona”,100,1002),
                                 (206,”clemens”,”london”,300,1007),
                                (207,”pereira”,”roe”,100,1004);
```

Q.13.All orders for more than $1000

➔ SELECT * FROM customer WHERE sno>1000;

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

➔ SELECT sname,city FROM salesperson WHERE city = 'London' AND comm > 0.12;

Q.15 All salespeople either in Barcelona or in London

➔ SELECT *FROM salesperson WHERE city IN ('Barcelona', 'London');

Q.16.All salespeople with commission between 0.10 and 0.12.(Boundary valuesshould be excluded)

➔ SELECT *FROM salesperson WHERE comm >= 0.10 AND comm <= 0.12;

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

➔ SELECT *FROM customer WHERE (rating > 100 OR city = 'Rome');

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

➔ **Create Database:-**

CREATE DATABASE salesman;

➔ **Create sales table:-**

CREATE TABLE sales(salesman_id int PRIMARY KEY,name varchar(20),city varchar(20),commission decimal(2,2));

➔ **Inserte data sales table:-**

INSERT INTO sales VALUES(5001,"james hoog","new york",00.15),
(5002,"nail knite","paris",00.13),
(5005,"pit alex","new york",00.11),
(5006,"mc layon","paris",00.14),
(5007,"paul adam","rome",00.13),
(5003,"lauson hen","san jose",00.12);

➔ **Fatch all data:-**

SELECT * FROM sales;

Q.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.

➔ **Create orders table:-**

CREATE TABLE orders (ord_no int PRIMARY KEY,
purch_amt decimal(4,2),
ord_date date,customer_id int,
salesman_id int,
FOREIGN KEY(salesman_id)REFERENCES sales(salesman_id));

➔ **Inserte data in orders table:-**

INSERT INTO orders VALUES(70001,150.5,'2012-10-05',3005,5002),

(70009,270.65,'2012-09-10',3001,5005),

(70002,65.26,'2012-10-05',3002,5001),

(70004,110.5,'2012-08-17',3009,5003),

(70007,948.5,'2012-09-10',3005,5002),

(70005,2400.5,'2012-07-27',3007,5001),

(70008,5760,'2012-09-10',3002,5001),

(70010,1983.43,'2012-10-10',3004,5006),

(70003,2480.4,'2012-10-10',3009,5003),

(70012,250.45,'2012-06-27',3008,5002),

(70011,75.29,'2012-08-17',3003,5007),

(70013,3045.6,'2012-04-25',3002,5001);

➔ SELECT ord_no,ord_date,purch_amt FROM orders WHERE salesman_id=5001;

Q.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.

➔ **Create table:-**

```
CREATE TABLE  item_mast(pro_id int PRIMARY KEY AUTO_INCREMENT,
                pro_name varchar(20),
                pro_price DECIMAL(10,2),
                pro_com int);
```

➔ **Insert data in table:-**

```
INSERT INTO item_mast VALUES(101,"mother board",3200.00,15),
                            (102,"key board",450.00,16),
                            (103,"zip drive",250.00,14),
                            (104,"speaker",550.00,16),
                            (105,"moniter",5000.00,11),
                            (106,"dvd drive",900.00,12),
                            (107,"cd drive",800.00,12),
                            (108,"printer",2600.00,13),
                            (109,"refillb cartridge",350.00,13),
                            (110,"mouse",250.00,12);
```

➔ SELECT * FROM item_mast WHERE pro_price>=200 AND pro_price<=600;

Q.21 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;

Q.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.

➔ SELECT pro_name,pro_price FROM item_mast WHERE pro_price>=200;
➔ SELECT pro_name,pro_price FROM item_mast ORDER BY pro_price DESC;
➔ SELECT pro_name,pro_price FROM item_mast ORDER BY pro_name ASC;

Q.22 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 'Price in Rs.'FROM item_mast;

Q.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.

➔ SELECT pro_com, AVG(pro_price) FROM item_mast;