|  |  |
| --- | --- |
| **q.1** | **Create Table Name Student and Exam:** |
|  | **\*CREATE TABLE STUDENT USING PRIMARY KEY:**  CREATE TABLE Student (  Rollno int,  Name varchar (20),  Branch varchar (10),  PRIMARY KEY(Rollno)  );  **\*INSERT VALUES IN STUDENT TABLE:**  INSERT INTO student VALUES (1,'jay','computer science');  INSERT INTO student VALUES (2,'suhani','Electronic and com.');  INSERT INTO student VALUES (3,'kriti','Electronic and com.');    **\*CREATE TABLE EXAM USING FOREIGN KEY:**  CREATE TABLE EXAM (  Rollno int,  S\_code varchar (8),  Marks int,  P\_code varchar (5),  FOREIGN KEY(Rollno) REFERENCES student (Rollno)  );  **\*INSERT VALUE IN EXAM TABL:**  INSERT INTO exam VALUES(1,'CS11,50,'CS');  INSERT INTO exam VALUES(1,'CS12,60,'CS');  INSERT INTO exam VALUES(2,'EC101,66,'EC');  INSERT INTO exam VALUES(2,'EC102,70,'EC');  INSERT INTO exam VALUES(3,'EC101,45,'EC');  INSERT INTO exam VALUES(3,'EC102,50,'EC'); |
|  |  |
| **Q.2** | **Create table given below: Employee and Incentive Table:** |
|  | **\*CREATE TABLE EMPLOYEE:**  CREATE TABLE Employee (  Employee\_id int PRIMARY KEY,  First\_Name varchar (20),  Last\_Name varchar (20),  salary int,  joining\_date text,  Department varchar (20)  );  **\*Insert value in EMPLOYEE table:**  INSERT INTO employee VALUES (1,'john','Abraham',10000000,'01-jan-13 12.00.00 AM’, ‘Banking');  INSERT INTO employee VALUES (2,'Michael','Clarke',800000,'01-jan-13 12.00.00 AM’,’ Insurance');  INSERT INTO employee VALUES (3,'Roy','Thomas',700000,'01-jan-13 12.00.00 AM’,  ‘Banking');  INSERT INTO employee VALUES (4,'Tom','Jose',600000,'01-jan-13 12.00.00 AM’,  ‘Insurance');  INSERT INTO employee VALUES (5,'Jerry','Pinto',650000,'01-jan-13 12.00.00 AM’,  ‘Insurance');  INSERT INTO employee VALUES (6,'Philip','Mathew',750000,'01-jan-13 12.00.00 AM’,  ‘Services');  INSERT INTO employee VALUES (7,'TestName1','123',650000,'01-jan-13 12.00.00 AM’,  ‘Services');  INSERT INTO employee VALUES (8,'TestName2','Lname%',600000,'01-jan-13 12.00.00 AM’,  ‘Insurance');    **\*CREATE TABLE INCENTIVE:**  CREATE TABLE Incentive  (  Employee\_ref\_id int,  Incentive\_date varchar (20),  Incentive\_amount int  ); |
| **Q.3**  **Q.4** | **\*Insert value in INCENTIVE table:**  INSERT INTO incentive VALUES(1,'13-FEB-01',5000);  INSERT INTO incentive VALUES(2,'13-FEB-01',3000);  INSERT INTO incentive VALUES(3,'13-FEB-01',4000);  INSERT INTO incentive VALUES(1,'13-JAN-01',4500);  INSERT INTO incentive VALUES(2,'13-JAN-01',3500);    **Get First\_Name from employee table using Tom name “Employee Name”.**  **ANS=**SELECT First\_Name FROM employee WHERE First\_Name='Tom';    **Get FIRST\_NAME, Joining Date, and Salary from employee table**.  **ANS=** SELECT First\_Name, joining\_date, salary FROM employee; |
| **Q.5**  **Q.6**  **Q.7**  **Q.8**  **Q.9** | **Get all employee details from the employee table order by First\_Name Ascending and Salary descending?**  **ANS=** SELECT \* FROM employee ORDER BY First\_Name, salary DESC;    **Get employee details from employee table whose first name contains ‘J’.**  **ANS=** SELECT \* FROM employee WHERE First\_name LIKE'j%';    **Get department wise maximum salary from employee table order by salary ascending?**  **ANS=** SELECT MAX (salary) FROM Employee;    **salary ascending?**  **ANS=** SELECT \* FROM Employee ORDER BY salary DESC;    **Select first\_name, incentive amount from employee and incentives table for**  **those employees who have incentives and incentive amount greater than 3000.**  **ANS=** SELECT e. First\_Name, i. incentive amount FROM employee e JOIN  incentive i ON e. Employee\_id=i. Employee\_ref\_id WHERE i.Incentive\_amount>  3000; |
| **Q.10**  **Q.11** | **Create After Insert trigger on Employee table which insert records in viewtable.**  **ANS=**  DELIMITER &&  CREATE TRIGGER T\_A AFTER INSERT ON employee  FOR EACH ROW  BEGIN  INSERT INTO viewt (Employee\_id, First\_Name, Last\_Name, salary,Department)  VALUES (NEW.Employee\_id, NEW.First\_Name, NEW.Last\_Name, NEW. Salary,  NEW.Department);  END  **Create table given below: Salesperson and Customer.**    **\*Create table salesperson:**  CREATE TABLE SALSEPERSON  (  PKS\_No INT PRIMARY key,  S\_name VARCHAR (40),  City VARCHAR (40),  COMM VARCHAR (40)  );  **\*Insert value in SALESPERSON table:**  INSERT INTO SALSEPERSON VALUES (1001,'peel','London','0.12');  INSERT INTO SALSEPERSON VALUES (1002,'Serres','Sen josh','0.13');  INSERT INTO SALSEPERSON VALUES (1004,'Motika','London','0.11');  INSERT INTO SALSEPERSON VALUES (1007,'Refkin','Barcelona','0.15');  INSERT INTO SALSEPERSON VALUES (1003,'Axelrod','New York','0.1'); |
|  | **\*Create table customer:**  CREATE TABLE CUSTOMER  (  PK\_CNM int PRIMARY KEY,  c\_name VARCHAR (100),  City VARCHAR (100),  RATING int,  FK\_SNo INT  );  **\*Insert value in customer table:**  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 josh',300,1002);  INSERT INTO CUSTOMER VALUES (204,'Grass','Barcelona',100,1002);  INSERT INTO CUSTOMER VALUES (206,'Clemens','London',300,1007);  INSERT INTO CUSTOMER VALUES (207,'Pereira','Roe',100,1004); |
| **Q.12**  **Q.13**  **Q.14**  **Q.15**  **Q.16** | **Retrieve the below data from above table.**  **All orders for more than $1000.**  **ANS=**select \* FROM employee WHERE orders>1000;  **Names and cities of all salespeople in London with commission above 0.12.**  **ANS**= SELECT S\_name, City FROM salseperson WHERE City='london' AND COMM>0.12;    **All salespeople either in Barcelona or in London.**  **ANS**= SELECT S\_name, City FROM SALSEPERSON WHERE City='Barcelona'  OR City='London';    **All salespeople with commission between 0.10 and 0.12. (Boundary values should be excluded).**  **ANS**= SELECT S\_name, COMM FROM salseperson WHERE COMM BETWEEN  0.1 AND 0.12 |
| **Q.17**  **Q.18** | **All customers excluding those with rating <= 100 unless they are located in Rome.**  **ANS**= SELECT \* FROM customer WHERE RATING<=100 AND NOT City='ROME';    **Write a SQL statement that displays all the information about all salespeople.**  INSERT INTO salseperson VALUES (5002,'James Hoog','Peris',0.13);  INSERT INTO salseperson VALUES (5002,'James Hoog','Peris',0.13);  INSERT INTO salseperson VALUES (5005,'Nail Knite','London',0.11);  INSERT INTO salseperson VALUES (5006,'Pit Alex','Paris',0.14);  INSERT INTO salseperson VALUES (5007,'Mc Lyon','Rome',0.13);  INSERT INTO salseperson VALUES (5003,'Lauson Hen', 'San Jose',0.12);  **ANS**= SELECT \* FROM salseperson;    **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**.  **ANS**=    **\*Create table orders:**  CREATE TABLE orders  (  ord\_no int,  purch\_amt text,  ord\_date date,  customer\_id int,  salesman\_id int  );  **\*Insert value in orders table:**  INSERT INTO orders VALUES (70001,150.5,'2012-10-05',3005,5002);  INSERT INTO orders VALUES (70009,270.65,'2012-09-10',3001,5005);  INSERT INTO orders VALUES (70002,65.26,'2012-10-05',3002,5001);  INSERT INTO orders VALUES (70004,110.5,'2012-08-17',3009,5003);  INSERT INTO orders VALUES (70007,948.5,'2012-09-10',3005,5002);  INSERT INTO orders VALUES (70005,2400.6,'2012-07-27',3007,5001);  INSERT INTO orders VALUES (70008,5760,'2012-09-10',3002,5001);  INSERT INTO orders VALUES (70010,1983.43,'2012-10-10',3004,5006);  INSERT INTO orders VALUES (70003,2480.4,'2012-10-10',3009,5003);  INSERT INTO orders VALUES (70012,250.45,'2012-06-27',3008,5002);  INSERT INTO orders VALUES (70011,75.29,'2012-08-17',3003,5007);  INSERT INTO orders VALUES (70013,3045.6,'2012-04-25',3002,5001);    **ANS**= SELECT ord\_no, ord\_date, purch\_amt FROM orders WHERE salesman\_id=5001; |
| **Q.20**  **Q.21** | **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.**  **ANS**= SELECT ord\_no, ord\_date ,purch\_amt FROM orders WHERE salesman\_id=5001;    **From the following table, write a SQL query to calculate the average price for a**  **manufacturer code of 16. Return avg.**    **\*Create table item\_mast:**  CREATE TABLE item\_mast  (Pro\_id int,  pro\_name varchar (30),  pro\_price varchar (30),  pro\_com int  );    **\*Insert value in item\_mast table:**  INSERT INTO item\_mast VALUES (101,'Mother Board',3200.00,15);  INSERT INTO item\_mast VALUES (102,'Key Board',450.00,16);  INSERT INTO item\_mast VALUES (103,'ZIP Drive',250.00,14);  INSERT INTO item\_mast VALUES (104,'Speaker',550.00,16);  INSERT INTO item\_mast VALUES (105,'Monitor',5000.00,11);  INSERT INTO item\_mast VALUES (106,'DVD drive',900.00,12);  INSERT INTO item\_mast VALUES (107,'CD drive',800.00,12);  INSERT INTO item\_mast VALUES (108,'Printer',2600.00,13);  INSERT INTO item\_mast VALUES (109,'Refill catridge',350.00,13);  INSERT INTO item\_mast VALUES (110,'Mouse',250.00,12);    ANS=SELECT AVG(pro\_price) FROM item\_mast WHERE pro\_com=16;    **Q.22)**  **From the following table, write a SQL query to display the pro\_name as 'Item Name'**  **and pro\_prices 'Price in Rs.'**  **ANS**= SELECT pro\_name "item Name", pro\_price "price in Rs" FROM item\_mast;    **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**.  **ANS**=1) SELECT \* FROM item\_mast WHERE pro\_price>=200 AND pro\_price<=600; |
| **Q.24** | **ANS**= 2) SELECT pro\_name , pro\_price FROM item\_mast WHERE pro\_price >= 250 ORDER BY  pro\_price DESC, pro\_name ASC;    **From the following table, write a SQL query to calculate average price of the items for each company. Return average price and companycode.**  **ANS**= SELECT pro\_com, AVG (pro\_price) "AVG\_price"FROM item\_mast GROUP BY pro\_com ASC; |