# SQL QUERIES

# ASSIGNMENT



## SUBMITTED BY:

PRIYA PAREKH

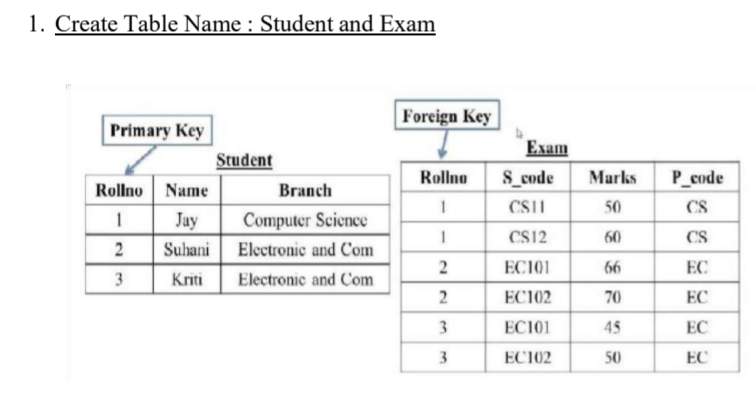
## SUBMITTED TO:

CHINMAYEE MAM

BACKEND DEVELOPMENT TOPS TECHNOLOGIES

1. Create Table Name: Student and Exam

Answer:



CREATE TABLE Student (

Rollno INT PRIMARY KEY,

Name VARCHAR(255),

Branch VARCHAR(255)

);

CREATE TABLE Exam (

Rollno INT,

S\_code VARCHAR(255),

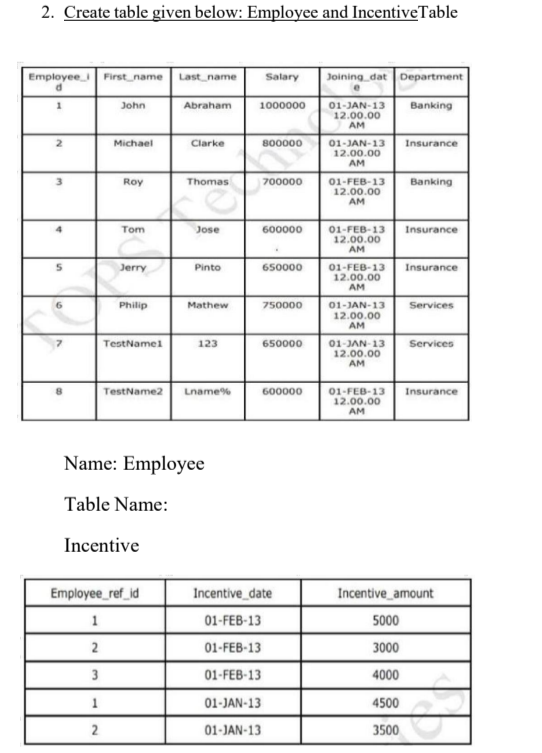
Marks INT,

P\_code VARCHAR(255),

FOREIGN KEY (Rollno) REFERENCES Student(Rollno)

);

1. Create table given below: Employee and Incentive Table



CREATE TABLE Employee (

Employee\_id INT PRIMARY KEY,

First\_name VARCHAR(255),

Last\_name VARCHAR(255),

Salary INT,

Joining\_date DATE,

Department VARCHAR(255)

);

INSERT INTO Employee (Employee\_id, First\_name, Last\_name, Salary, Joining\_date, Department) VALUES

(1, 'John', 'Abraham', 1000000, '2013-01-13', 'Banking'),

(2, 'Michael', 'Clarke', 800000, '2013-01-13', 'Insurance'),

(3, 'Roy', 'Thomas', 700000, '2013-02-13', 'Banking'),

(4, 'Tom', 'Jose', 600000, '2013-02-13', 'Insurance'),

(5, 'Jerry', 'Pinto', 650000, '2013-02-13', 'Insurance'),

(6, 'Philip', 'Mathew', 750000, '2013-01-13', 'Services'),

(7, 'TestName1', '123', 650000, '2013-01-13', 'Services'),

(8, 'TestName2', 'Lname%', 600000, '2013-02-13', 'Insurance');

CREATE TABLE Incentive (

Employee\_ref\_id INT,

Incentive\_date DATE,

Incentive\_amount INT

);

INSERT INTO Incentive (Employee\_ref\_id, Incentive\_date, Incentive\_amount) VALUES

(1, '2013-02-13', 5000),

(2, '2013-02-13', 3000),

(3, '2013-02-13', 4000),

(1, '2013-01-13', 4500),

1. Get First Name from the employee table using Tom name “Employee Name”.

Answer:

SELECT First\_name

FROM Employee

WHERE Last\_name = 'Jose';

4. Get FIRST\_NAME, Joining Date, and Salary from employee table.

Answer:

SELECT

First\_name,

Joining\_dat,

Salary

FROM Employee;

1. Get all employee details from the employee table order by First\_Name

Answer:

SELECT \* FROM Employee ORDER BY First\_name ASC, Salary DESC

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

Answer:

SELECT \*

FROM Employee

WHERE First\_name LIKE '%J%';

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

Answer:

SELECT Department, MAX(Salary) AS MaxSalary

FROM Employee

GROUP BY Department

ORDER BY MaxSalary ASC;

8. Select first\_name, incentive amount from the employee and incentives table

for those employees who have incentives and incentive amounts greater than 3000

Answer:

SELECT e.first\_name, i.Incentive\_amount

FROM Employee e

JOIN Incentive i ON

e.Employee\_i\_d = i.Employee\_ref\_id

WHERE i.Incentive\_amount > 3000;

9. Create after Insert trigger on the Employee table which inserts records in

Viewable

Answer:

CREATE TRIGGER trg\_Employee\_Insert

AFTER INSERT ON Employee

AS

BEGIN

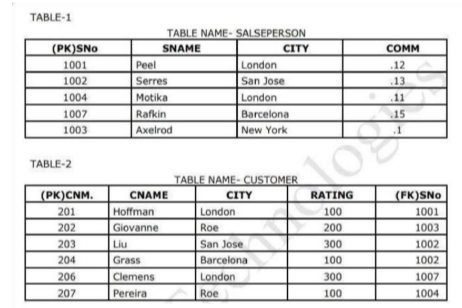
INSERT INTO Incentive (Employee\_ref\_id, Incentive\_date, Incentive\_amount)

SELECT i.Employee\_id, GETDATE(), 5000

FROM inserted i;

END;

10. Create the table given below: Salesperson and Customer



Answer:

CREATE TABLE Salesperson (

SNo INT PRIMARY KEY,

SNAME VARCHAR(255),

CITY VARCHAR(255),

COMM DECIMAL(4,2)

);

CREATE TABLE Customer (

CNM INT PRIMARY KEY,

CNAME VARCHAR(255),

CITY VARCHAR(255),

RATING INT,

SNo INT,

FOREIGN KEY (SNo) REFERENCES Salesperson(SNo)

);

INSERT INTO Salesperson (SNo, SNAME, CITY, COMM) VALUES

(1001, 'Peel', 'London', 0.12),

(1002, 'Serres', 'San Jose', 0.13),

(1004, 'Motika', 'London', 0.11),

(1007, 'Rafkin', 'Barcelona', 0.15),

(1003, 'Axelrod', 'New York', 0.1);

INSERT INTO Customer (CNM, CNAME, CITY, RATING, SNo) VALUES

(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);

12. Retrieve the below data from the above table

Answer:

SELECT c.CNAME AS CustomerName, s.SNAME AS SalespersonName

FROM CUSTOMER c

JOIN SALESPERSONS ON c.SNo = s.SNo;

13. All orders for more than $1000.

Answer:

SELECT \*

FROM Orders

WHERE amount > 1000;

14.Names and cities of all salespeople in London with commission above 0.1215.All salespeople either in Barcelona or in London

Answer:

SELECT SNAME, CITY

FROM SALESPERSON

WHERE CITY = 'London' AND COMM > 0.1215

UNION

SELECT SNAME, CITY

FROM SALESPERSON

WHERE CITY IN ('Barcelona', 'London');

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

Answer:

SELECT

CNAME,

CITY,

RATING,

SNAME

FROM CUSTOMER AS C

JOIN SALESPERSON AS S

ON C.(FK)SNo = S.SNo;

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

in Rome

Answer:

SELECT \*

FROM Customer

WHERE RATING > 100

UNION

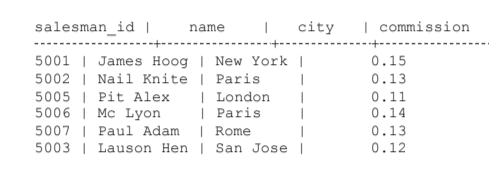
SELECT \*

FROM Customer

WHERE CITY = 'Rome'

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

Salespeople



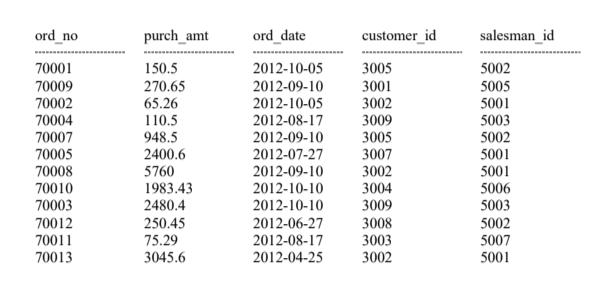
Answer:

SELECT \*

FROM Salesperson;

19. From the following table, write a SQL query to find orders that a salesperson with ID delivers. 5001. Return ord\_no, ord\_date, purch\_amt.

Sample table: orders



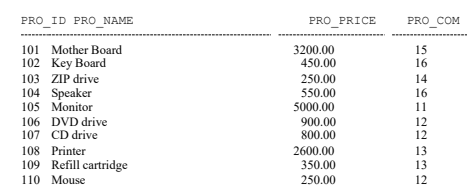
SELECT ord\_no, ord\_date, purch\_amt

FROM orders

WHERE salesman\_id = 5001;

20. From the following table, write an SQL query to select a range of products whose prices range from 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



SELECT pro\_id, pro\_name, pro\_price, pro\_com

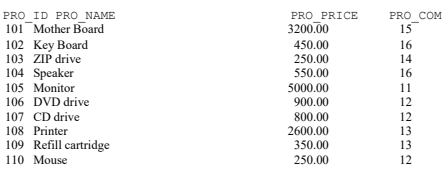
FROM item\_mast

WHERE pro\_price BETWEEN 200 AND 600;

21. From the following table, write a SQL query to calculate the average

price for a manufacturer code of 16. Return avg.

Sample table: item\_mast



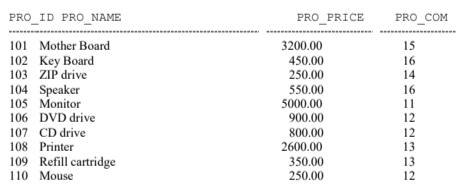
SELECT AVG(PRO\_PRICE)

FROM item\_mast

WHERE PRO\_COM = 16;

22. From the following table, write a SQL query to display the pro\_name as 'Item Name' and pro\_priceas 'Price in Rs.'

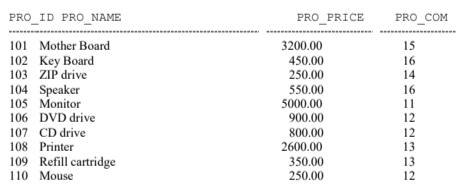
Sample table: item\_mast



SELECT PRO\_NAME AS 'Item Name', PRO\_PRICE AS 'Price in Rs.'

FROM item\_mast;

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 by product name in ascending. Return pro\_name and pro\_price.



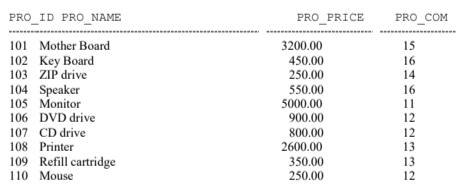
SELECT pro\_name, pro\_price

FROM item\_mast

WHERE pro\_price >= 250

ORDER BY pro\_price DESC, pro\_name ASC;

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



SELECT PRO\_COM, AVG(PRO\_PRICE) AS average\_price

FROM your\_table

GROUP BY PRO\_COM;