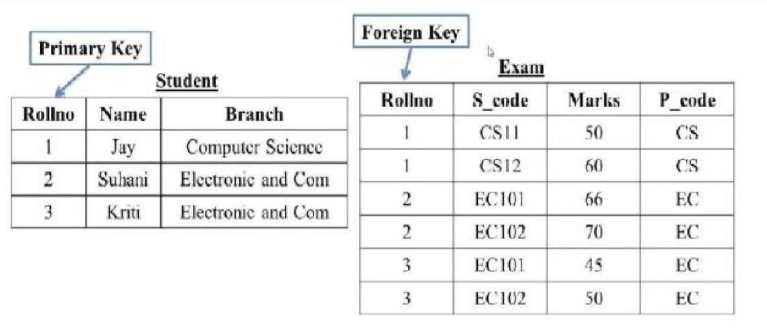
# Sql-query assignment

## By:- jaimin trivedi

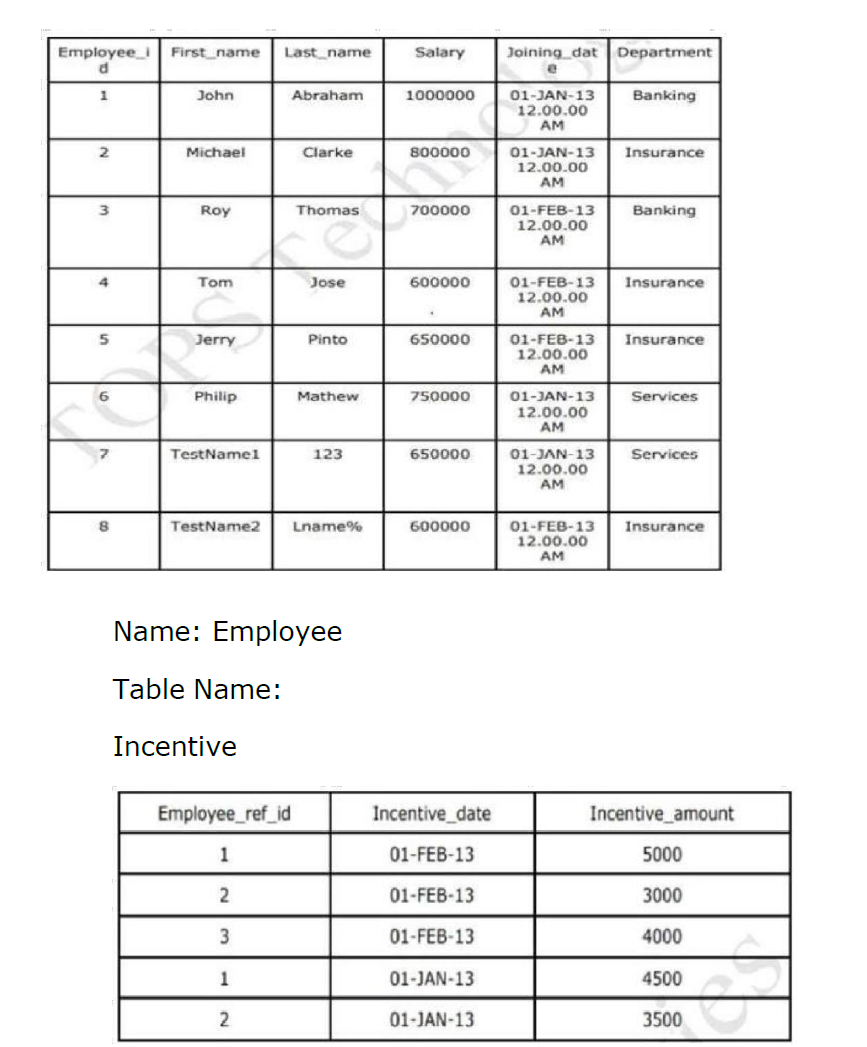
[[1]](#footnote-0)1. Create Table Name : Student and Exam



Ans

| Student table | CREATE TABLE Student (  Roll\_no INT AUTO\_INCREMENT PRIMARY KEY,  name VARCHAR(255) NOT NULL,  branch VARCHAR(255) NOT NULL  ); |
| --- | --- |
| Exam table | CREATE TABLE Exam (  Roll\_no INT AUTO\_INCREMENT,  s\_code int NOT NULL,  mark int ,  p\_code VARCHAR(255) NOT NULL,  Foreign KEY (Roll\_no) REFERENCES student(Roll\_no)  ); |

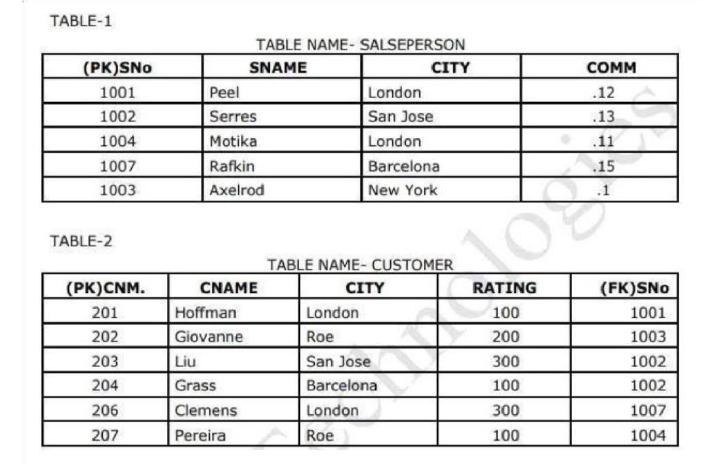
2. Create table given below: Employee and IncentiveTable



| Employee table  [[2]](#footnote-1)  Incentent table | CREATE TABLE Employee (  Employee\_id INT NOT NULL UNIQUE AUTO\_INCREMENT,  first\_name VARCHAR(30),  last\_name VARCHAR(30),  salary INT,  joining\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP NOT NULL,  department VARCHAR(50),  PRIMARY KEY (Employee\_id)  );  CREATE TABLE incentive (  Employee\_ref\_id INT,  incentive\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP NOT NULL,  incentent\_amount INT NOT NULL,  FOREIGN KEY (Employee\_ref\_id) REFERENCES Employee(Employee\_id)  ); |
| --- | --- |

| 3. Get First\_Name from employee table using Tom name “Employee Name”. | SELECT \* 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 ascending and Salary descending? | 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  8. salaryascending? | SELECT department, MAX(salary) AS max\_salary  FROM employee  GROUP BY department  ORDER BY max\_salary ASC; |
| 9. Select first\_name, incentive amount from employee and incentives table forthose employees who have incentives and incentive amount greater than 3000 | SELECT e.first\_name, i.amount AS incentive\_amount  FROM employee e  INNER JOIN incentive i ON e.Employee\_id = i.Employee\_id  WHERE i.amount > 3000; |

11. Create table given below: Salesperson and Customer



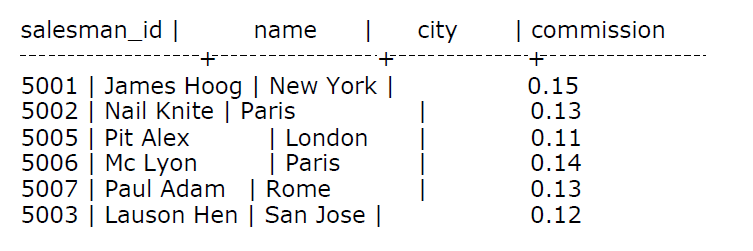
[[3]](#footnote-2)

| Create salesperson table | CREATE TABLE salesperson  (  sno int UNION NOT null,  sname varchar(30),  city varchar(35),  comm float,  PRIMARY KEY (sno)  ); |
| --- | --- |
| Create customer Table | CREATE TABLE customer  (  cno int UNION NOT null,  cname varchar(30),  city varchar(35),  rating int,  sno int    PRIMARY KEY (cno)  FOREIGN KEY (sno) REFERENCES salesperson(sno)  ); |

[[4]](#footnote-3)

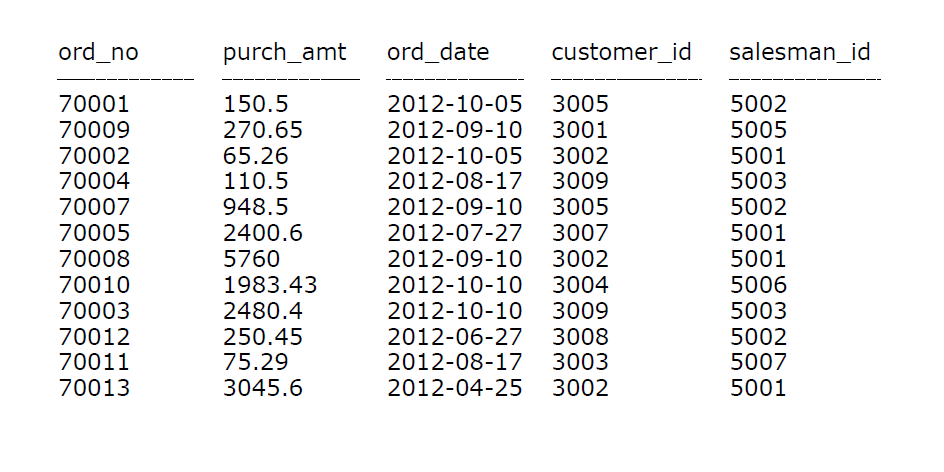
| 12. Retrieve the below data from above table  13.All orders for more than $1000. | SELECT \*  FROM customer  WHERE ORDER\_value > 1000; |
| --- | --- |
| 14.Names and cities of all salespeople in London with commission above 0.12 | SELECT sname , city from salesperson WHERE comm >0.12; |
| 15.All salespeople either in Barcelona or in London | SELECT \* FROM salespersom  WHERE city = 'barcelona' or city='london'; |
| 16. 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; |
| 17. All customers excluding those with rating <= 100 unless they are located inRome | SELECT \* from customer  WHERE rating <= 100 or city = 'rome'; |

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



| SELECT \* FROM salsepeople |
| --- |

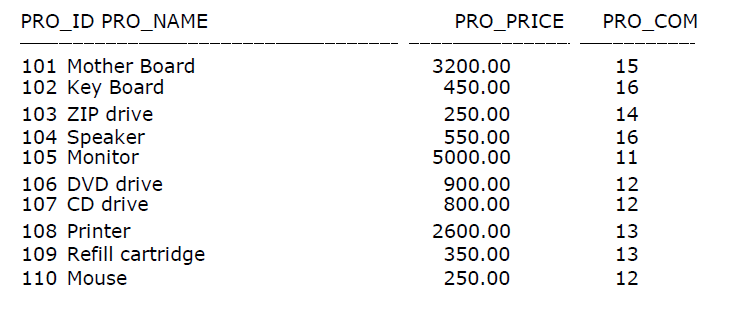
19. From the following table, write a SQL query to find orders that are delivered by a salesperson with ID. 5001. Return ord\_no, ord\_date, purch\_amt.



| SELECT ord\_no, ord\_date , purch\_amt  FROM orders  [[5]](#footnote-4)WHERE salesman\_id = 5001; |
| --- |

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.

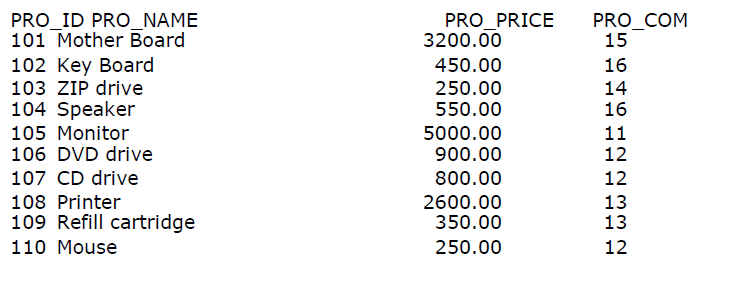
Sample table: item\_mast



| 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

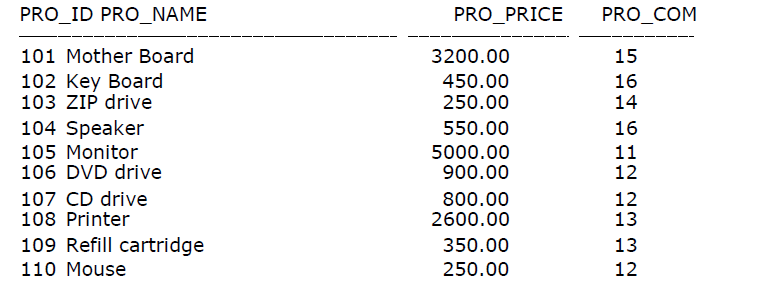


| SELECT AVG (pro\_price) as AVG  FROM item\_mast  WHERE pro\_com = 16; |
| --- |

[[6]](#footnote-5)

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

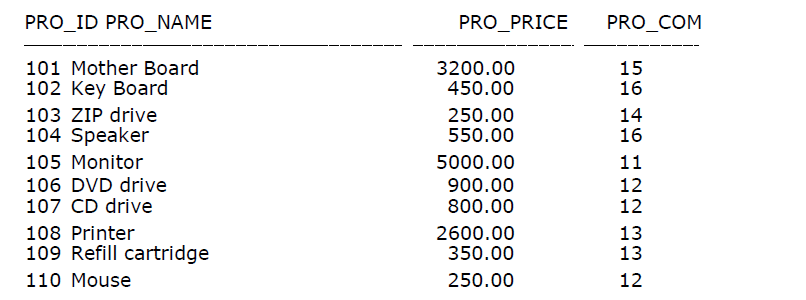
Sample table: item\_mast



| SELECT pro\_name as 'item name',  concat('price in rs .',fromat(pro\_price,2))  AS 'price in rs.'  from item mast;  [[7]](#footnote-6) |
| --- |

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.

Sample table: item\_mast



| SELECT pro\_name , pro\_price  FROM item\_mast  WHERE pro\_price >=250.00  ORDER BY pro\_price DESC , pro\_name ASC |
| --- |

1. [↑](#footnote-ref-0)
2. [↑](#footnote-ref-1)
3. [↑](#footnote-ref-2)
4. [↑](#footnote-ref-3)
5. [↑](#footnote-ref-4)
6. [↑](#footnote-ref-5)
7. [↑](#footnote-ref-6)