Name-Ashish Bhangale

Batch-A1

Rollno-07

Assignment no-5

Title-Design at least 10 SQL queries for suitable database application using SQL DML

statements: all types of Join, Sub-Query and View.Problem Statement:

Problem statement-A)

Create tables with primary key , foreign key constraint as shown in above schema, salesman\_id is

foreign key in customer table

Salesman

salesman\_id | name | city | commission

-------------+------------+----------+------------

5001 | James Hoog | New York | 0.15

5002 | Nail Knite | Paris | 0.13

5005 | Pit Alex | London | 0.11

5006 | Mc Lyon | Paris | 0.14

5007 | Paul Adam | Rome | 0.13

5003 | Lauson Hen | San Jose | 0.12

Customer

customer\_id | cust\_name | city | grade | salesman\_id

-------------+----------------+------------+-------+------------

-

3002 | Nick Rimando | New York | 100 | 5001

3007 | Brad Davis | New York | 200 | 5001

3005 | Graham Zusi | California | 200 | 5002

3008 | Julian Green | London | 300 | 5002

3004 | Fabian Johnson | Paris | 300 | 5006

3009 | Geoff Cameron | Berlin | 100 | 5003

3003 | Jozy Altidor | Moscow | 200 | 5007

3001 | Brad Guzan | London | | 5003

Orders

ord\_nopurch\_amtord\_datecustomer\_idsalesman\_id

---------- ---------- ---------- ----------- -----------

70001 150.5 2012-10-05 3005 5002

70009 270.65 2012-09-10 3001 5003

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 2400.6 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

Queries: (use appropriate join types and join condition like, cross join, natural join, left , right , full

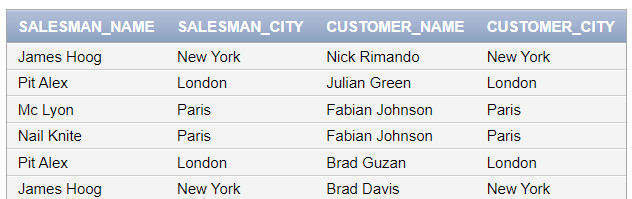
outer join, Join... on, join...using)

1. **Find the salesmen and customers with their name and cities, who belongs to the same city.**

SELECT s.name AS salesman\_name, s.city AS salesman\_city, c.cust\_name AS customer\_name, c.city AS customer\_city

FROM Salesman s

JOIN Customer c ON s.city = c.city;



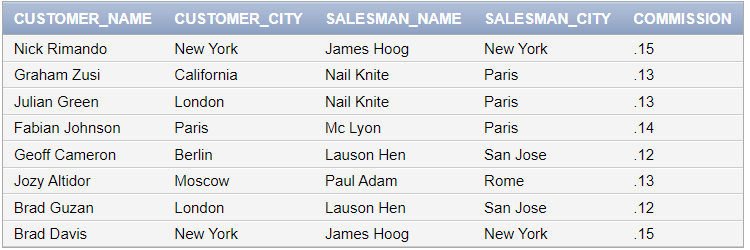
**2) Prepare a list for which salesman are working for which customer along with city and**

**commissions earned by the salesman.**

SELECT c.cust\_name AS customer\_name, c.city AS customer\_city, s.name AS salesman\_name, s.city AS salesman\_city, s.commission

FROM Customer c

JOIN Salesman s ON c.salesman\_id = s.salesman\_id;



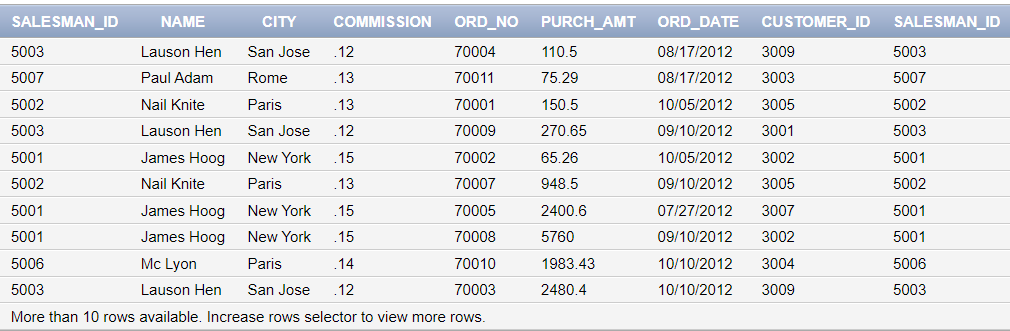
**3) Write a query to make a join on the tables salesman and orders in such a form that the same**

**column of each table will appear once and only the relational rows will come.**

SELECT s.\*, o.\*

FROM Salesman s

JOIN Orders o ON s.salesman\_id = o.salesman\_id;



**4) Prepare a list for the salesmen who works either for one or more customer or not yet join under**

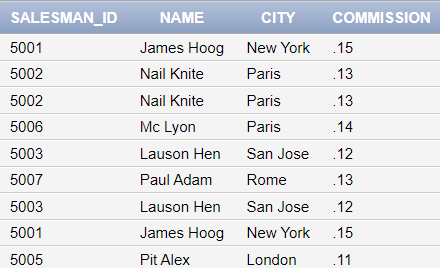
**any of the customers**

SELECT s.\*

FROM Salesman s

LEFT JOIN Customer c ON s.salesman\_id = c.salesman\_id

WHERE c.salesman\_id IS NOT NULL OR c.salesman\_id IS NULL;



**5) Prepare a list in ascending order for the salesmen who works either for one or more customer**

**or not yet join under any of the customers**

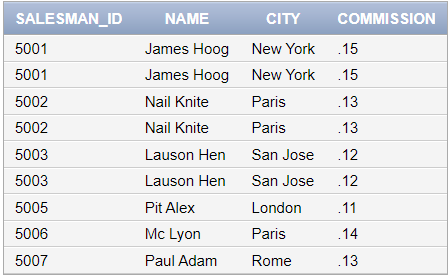
SELECT s.\*

FROM Salesman s

LEFT JOIN Customer c ON s.salesman\_id = c.salesman\_id

WHERE c.salesman\_id IS NOT NULL OR c.salesman\_id IS NULL

ORDER BY s.salesman\_id;



B) employee (employee-name, street, city)

works (employee-name, company-name, salary)

company (company-name, city)

manages (employee-name, manager-name)

Insert appropriate data in the tables.

Queries:

Solve Query 1 to 4 using appropriate operator in, not in , all, some , exists etc

**Q1)Find the names, street address, and cities of residence of all employees who work for Tata**

and earn more than $10,000

SELECT e.employee\_name, e.street, e.city

FROM employee e

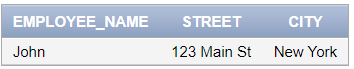
WHERE e.employee\_name IN (

SELECT w.employee\_name

FROM works w

WHERE w.company\_name = 'Tata' AND w.salary > 10000

);



**Q2)Find all employees in the database who do not work for Tata**

SELECT employee\_name

FROM employee

WHERE employee\_name NOT IN (

SELECT employee\_name

FROM works

WHERE company\_name = 'Tata'

);



**Q3) Find all employees in the database who earn more than every employee of Reliance**

SELECT e.employee\_name, e.street, e.city

FROM employee e

WHERE EXISTS (

SELECT \*

FROM works w

WHERE w.company\_name = 'Reliance' AND e.employee\_name <> w.employee\_name AND e.salary > w.salary

);

**Q4) Find the company that has the smallest payroll**

SELECT company\_name

FROM works

GROUP BY company\_name

HAVING SUM(salary) = (

SELECT MIN(total\_salary)

FROM (

SELECT SUM(salary) AS total\_salary

FROM works

GROUP BY company\_name

)

);



**Q5) Create a view for names and cities of residence of all employees who work for First Bank**

**Corporation**

CREATE VIEW First\_Bank\_Employees AS

SELECT e.employee\_name, e.city

FROM employee e

JOIN works w ON e.employee\_name = w.employee\_name

WHERE w.company\_name = 'First Bank Corporation';