1. write a SQL query to find the salesperson and customer who reside in the same city.

Return Salesman, cust\_name and city

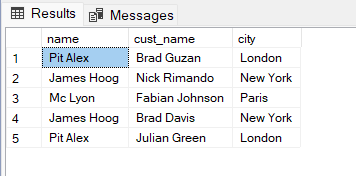
ANS.

SELECT S.name,C.cust\_name, S.city

FROM salesman S

CROSS JOIN customer C

WHERE S.city=C.CITY;



2. write a SQL query to find those orders where the order amount exists between 500

and 2000. Return ord\_no, purch\_amt, cust\_name, city

ANS.

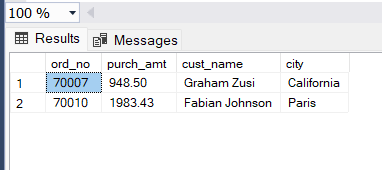
SELECT O.ord\_no, O.purch\_amt, C.cust\_name, C.city

FROM orders O

LEFT JOIN customer C

ON O.customer\_id=C.customer\_id

WHERE O.purch\_amt>=500 AND O.purch\_amt<=2000



3. write a SQL query to find the salesperson(s) and the customer(s) he represents.

Return Customer Name, city, Salesman, commission

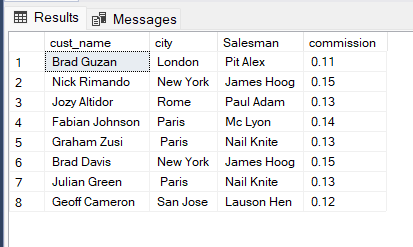
ANS.

SELECT C.cust\_name, S.city, S.name as Salesman, S.commission

FROM customer C

LEFT JOIN salesman S

ON C.salesman\_id = S.salesman\_id



4. write a SQL query to find salespeople who received commissions of more than 12

percent from the company. Return Customer Name, customer city, Salesman,

commission.

ans.

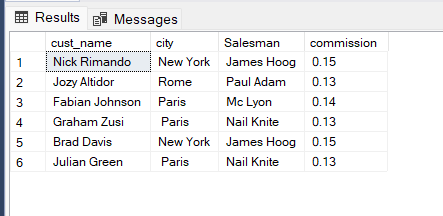
SELECT C.cust\_name, S.city, S.name as Salesman, S.commission

FROM customer C

LEFT JOIN salesman S

ON C.salesman\_id = S.salesman\_id

WHERE S.commission > 0.12



5. write a SQL query to locate those salespeople who do not live in the same city where

their customers live and have received a commission of more than 12% from the

company. Return Customer Name, customer city, Salesman, salesman city,

commission

ans.

SELECT C.cust\_name, C.city, S.name as Salesman,S.city, S.commission

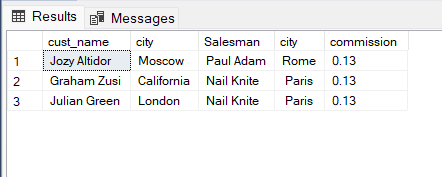
FROM customer C

LEFT JOIN salesman S

ON C.salesman\_id = S.salesman\_id

WHERE S.commission > 0.12

AND S.city <> C.city



6. write a SQL query to find the details of an order. Return ord\_no, ord\_date,

purch\_amt, Customer Name, grade, Salesman, commission

ans.

SELECT O.ord\_no,O.ord\_date, O.purch\_amt, C.cust\_name,C.grade,S.name as Salesman,S.commission

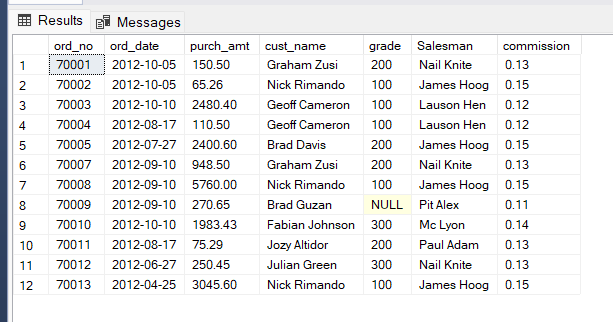
FROM orders O

LEFT JOIN customer C

ON O.customer\_id = C.customer\_id

LEFT JOIN salesman S

ON O.salesman\_id = S.salesman\_id



7. Write a SQL statement to join the tables salesman, customer and orders so that the

same column of each table appears once and only the relational rows are returned.

ans.

SELECT O.ord\_no,O.ord\_date,O.purch\_amt,O.customer\_id,C.cust\_name,C.city AS CustomerCity,C.grade,O.salesman\_id,S.name as Salesman,S.city as SalesmanCity,S.commission

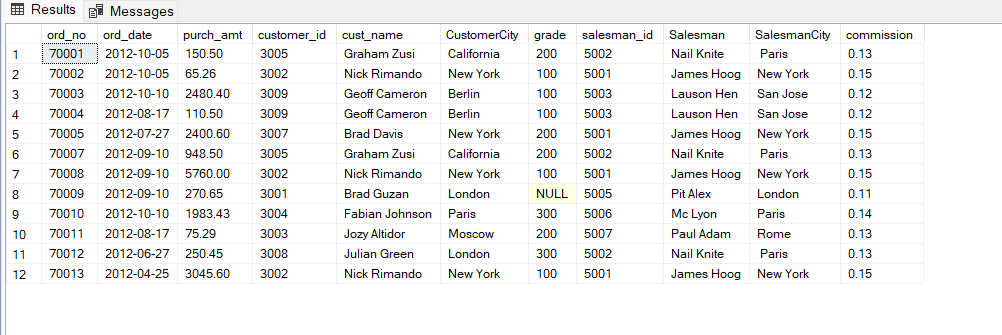
FROM orders O

INNER JOIN salesman S

ON O.salesman\_id= S.salesman\_id

INNER JOIN customer C

ON O.customer\_id = C.customer\_id



8. write a SQL query to display the customer name, customer city, grade, salesman,

salesman city. The results should be sorted by ascending customer\_id.

ans.

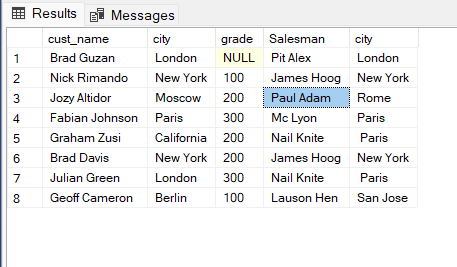
SELECT C.cust\_name, C.city,C.grade, S.name as Salesman,S.city

FROM customer C

LEFT JOIN salesman S

ON C.salesman\_id = S.salesman\_id

ORDER BY C.customer\_id ASC



9. write a SQL query to find those customers with a grade less than 300. Return

cust\_name, customer city, grade, Salesman, salesmancity. The result should be

ordered by ascending customer\_id.

ans.

SELECT C.cust\_name, C.city,C.grade, S.name as Salesman,S.city as SalesmanCity

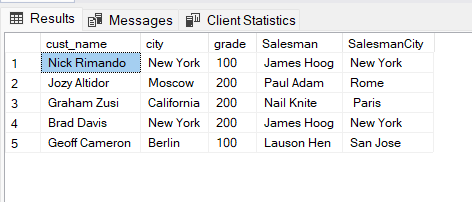
FROM customer C

LEFT JOIN salesman S

ON C.salesman\_id = S.salesman\_id

WHERE C.grade<300

ORDER BY C.customer\_id ASC



10. Write a SQL statement to make a report with customer name, city, order number,

order date, and order amount in ascending order according to the order date to

determine whether any of the existing customers have placed an order or not

ans.

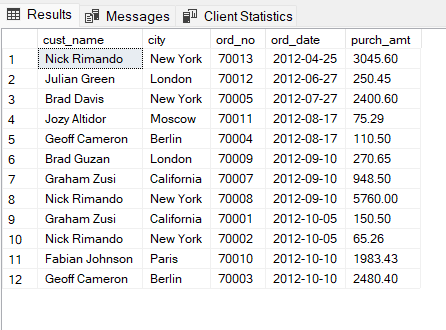
SELECT C.cust\_name, C.city, O.ord\_no, O.ord\_date,O.purch\_amt

FROM customer C

LEFT JOIN orders O

ON C.customer\_id = o.customer\_id

ORDER BY o.ord\_date ASC



11. Write a SQL statement to generate a report with customer name, city, order number,

order date, order amount, salesperson name, and commission to determine if any of

the existing customers have not placed orders or if they have placed orders through

their salesman or by themselves

ans.

SELECT C.cust\_name, C.city, O.ord\_no, O.ord\_date, O.purch\_amt, S.name as Salesman, S.commission

FROM customer C

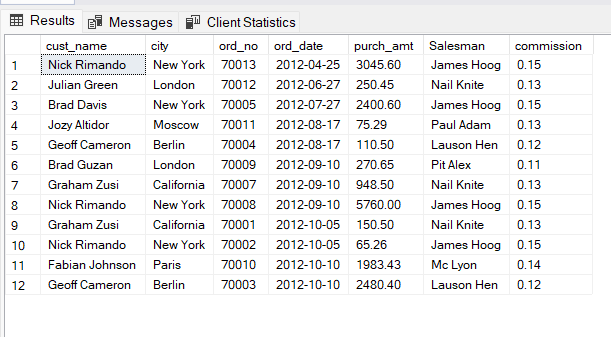
LEFT JOIN orders O

ON C.customer\_id = o.customer\_id

LEFT JOIN salesman S

ON C.salesman\_id = S.salesman\_id

ORDER BY o.ord\_date ASC



12. Write a SQL statement to generate a list in ascending order of salespersons who

work either for one or more customers or have not yet joined any of the customers

ans.

SELECT C.cust\_name, s.name AS Salesman, S.city

FROM customer C

LEFT JOIN salesman S

ON C.salesman\_id = S.salesman\_id

ORDER BY S.salesman\_id



13. write a SQL query to list all salespersons along with customer name, city, grade,

order number, date, and amount.

ans.

SELECT S.name as Salesman,cust\_name,C.city,C.grade,O.ord\_no,O.ord\_date, O.purch\_amt

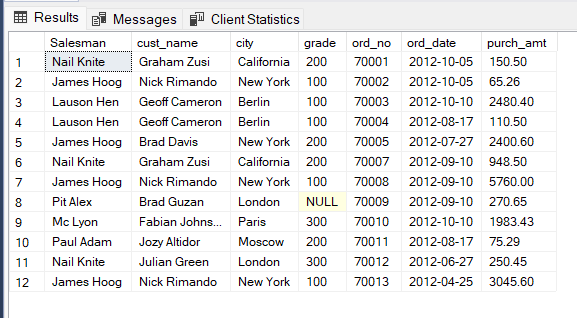
FROM orders O

LEFT JOIN customer C

ON O.customer\_id = C.customer\_id

LEFT JOIN salesman S

ON O.salesman\_id = S.salesman\_id



14. Write a SQL statement to make a list for the salesmen who either work for one or more customers or yet to join any of the customers. The customer may have placed, either one or more orders on or above order amount 2000 and must have a grade, or he may not have placed any order to the associated supplier.

ANS.

SELECT S.name as Salesman, S.commission, C.cust\_name,C.grade, O.ord\_no, O.ord\_date, O.purch\_amt

FROM customer C

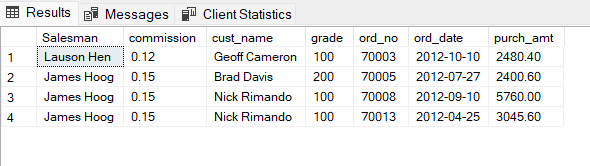
LEFT JOIN orders O

ON C.customer\_id = o.customer\_id

LEFT JOIN salesman S

ON C.salesman\_id = S.salesman\_id

WHERE grade IS NOT NULL AND purch\_amt>2000



15. Write a SQL statement to generate a list of all the salesmen who either work for one or more customers or have yet to join any of them. The customer may have placed one or more orders at or above order amount 2000, and must have a grade, or he may not have placed any orders to the associated supplier.

SELECT S.name as Salesman, S.commission, C.cust\_name,C.grade, O.ord\_no, O.ord\_date, O.purch\_amt

FROM customer C

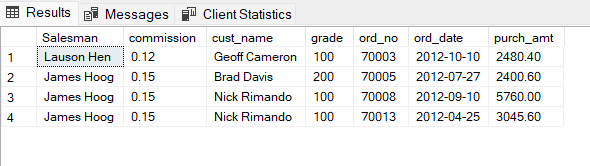
LEFT JOIN orders O

ON C.customer\_id = o.customer\_id

LEFT JOIN salesman S

ON C.salesman\_id = S.salesman\_id

WHERE grade IS NOT NULL AND purch\_amt>2000



16. Write a SQL statement to generate a report with the customer name, city, order no. order date, purchase amount for only those customers on the list who must have a grade and placed one or more orders or which order(s) have been placed by the customer who neither is on the list nor has a grade.

ANS.

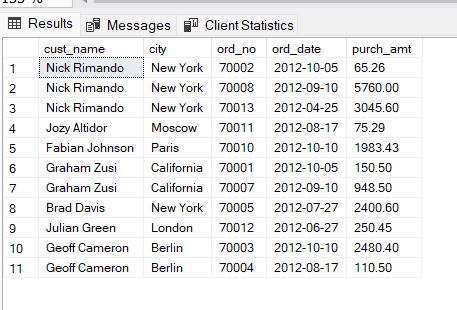
SELECT C.cust\_name,C.city, O.ord\_no,O.ord\_date,O.purch\_amt

FROM customer C

LEFT JOIN orders O

ON O.customer\_id= C.customer\_id

WHERE GRADE IS NOT NULL



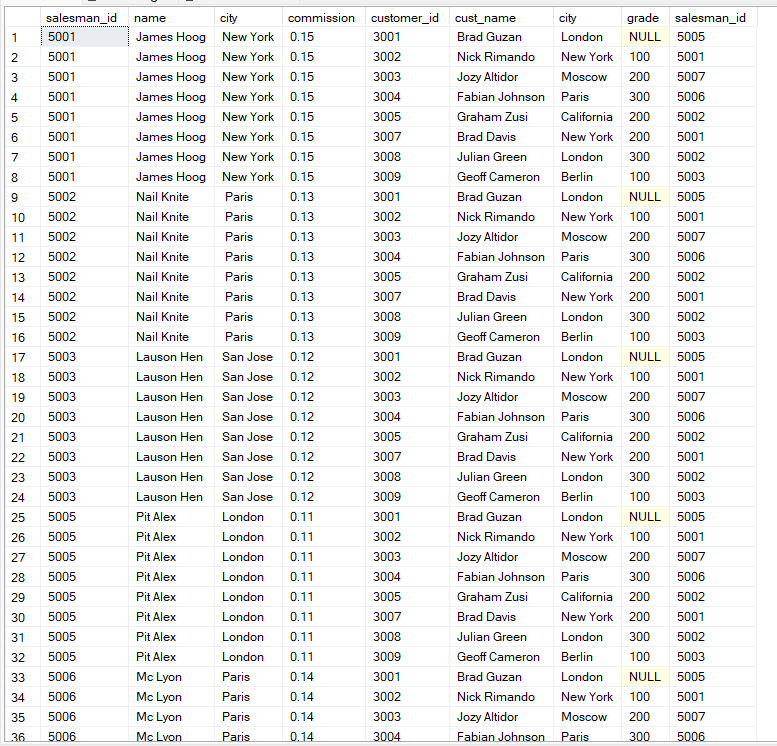
17. Write a SQL query to combine each row of the salesman table with each row of the customer table

ANS.

SELECT \*

FROM salesman

CROSS JOIN customer





18. Write a SQL statement to create a Cartesian product between salesperson and customer, i.e. each salesperson will appear for all customers and vice versa for that salesperson who belongs to that city

ANS.

SELECT \*

FROM salesman

CROSS JOIN customer

WHERE salesman.city IS NOT NULL

SAME AS 17

19. Write a SQL statement to create a Cartesian product between salesperson and customer, i.e. each salesperson will appear for every customer and vice versa for those salesmen who belong to a city and customers who require a grade

ANS.

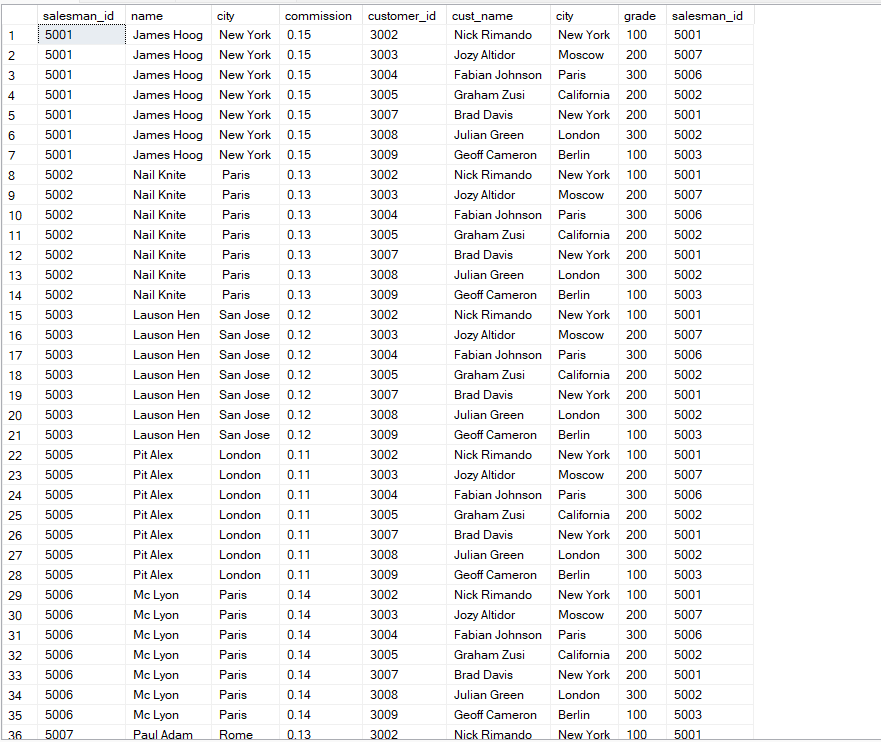
SELECT \*

FROM salesman

CROSS JOIN customer

WHERE salesman.city IS NOT NULL

AND grade IS NOT NULL





20. Write a SQL statement to make a Cartesian product between salesman and customer i.e. each salesman will appear for all customers and vice versa for those salesmen who must belong to a city which is not the same as his customer and the customers should have their own grade

ANS.

SELECT \*

FROM salesman

CROSS JOIN customer

WHERE salesman.city <> customer.city

AND grade IS NOT NULL



