Assignment-2

Retrieve data using join with where clause

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

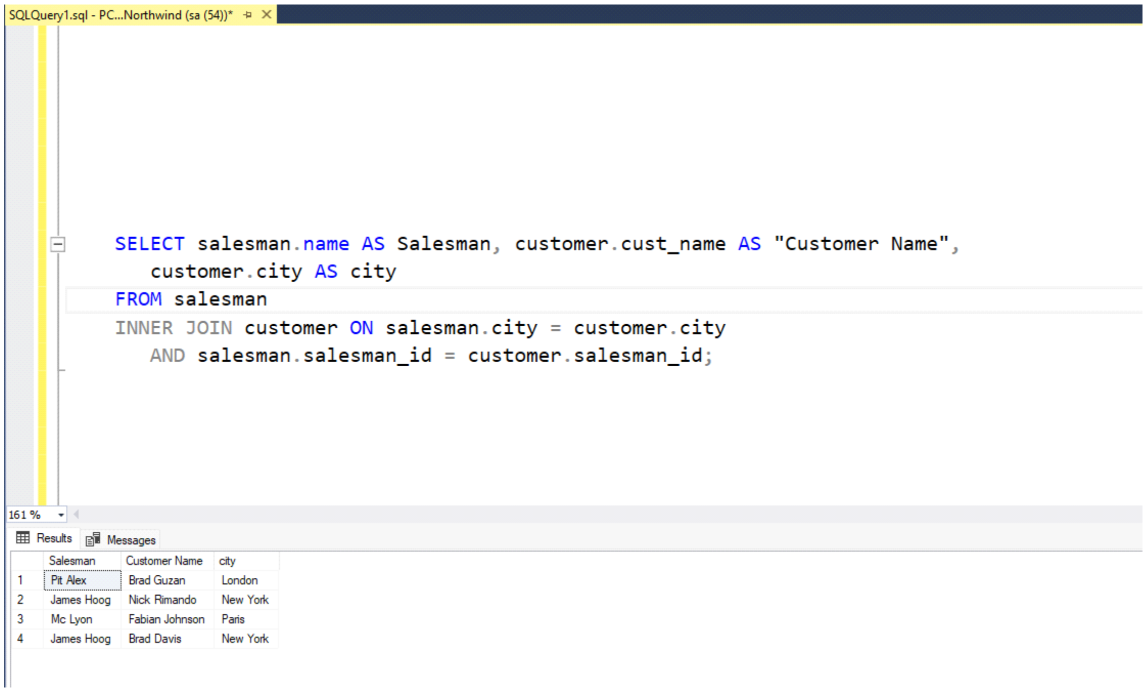
SELECT salesman.name AS Salesman, customer.cust_name AS "Customer Name",

    customer.city AS city

FROM salesman

INNER JOIN customer ON salesman.city = customer.city

    AND salesman.salesman_id = customer.salesman_id;



SQLQuery1.sql - PC...Northwind (sa (54))*

```
SELECT salesman.name AS Salesman, customer.cust_name AS "Customer Name",
   customer.city AS city
FROM salesman
INNER JOIN customer ON salesman.city = customer.city
   AND salesman.salesman_id = customer.salesman_id;
```

161 %

Results | Messages

	Salesman	Customer Name	city
1	Pit Alex	Brad Guzan	London
2	James Hoog	Nick Rimando	New York
3	Mc Lyon	Fabian Johnson	Paris
4	James Hoog	Brad Davis	New York

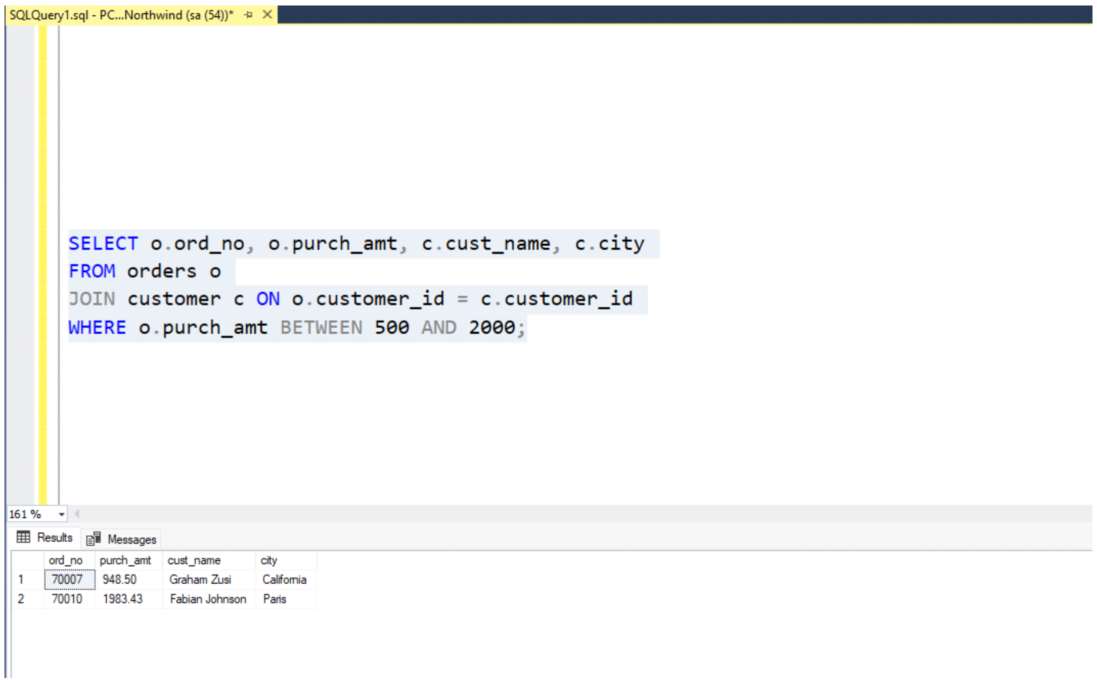
2. write a SQL query to find those orders where the order amount exists between 500 and 2000. Return ord_no, purch_amt, cust_name, city.

SELECT o.ord_no, o.purch_amt, c.cust_name, c.city

FROM orders o

JOIN customer c ON o.customer_id = c.customer_id

WHERE o.purch_amt BETWEEN 500 AND 2000;



```
SELECT o.ord_no, o.purch_amt, c.cust_name, c.city
FROM orders o
JOIN customer c ON o.customer_id = c.customer_id
WHERE o.purch_amt BETWEEN 500 AND 2000;
```

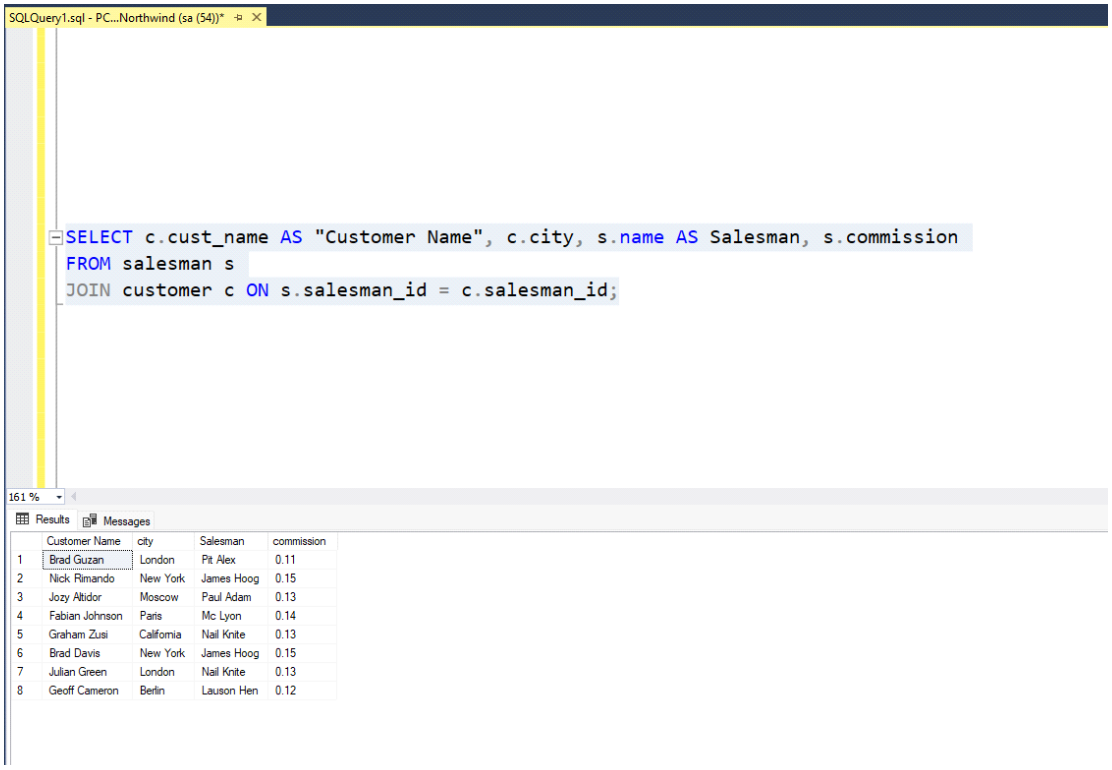
	ord_no	purch_amt	cust_name	city
1	70007	948.50	Graham Zusi	California
2	70010	1983.43	Fabian Johnson	Paris

3. write a SQL query to find the salesperson(s) and the customer(s) he represents. Return Customer Name, city, Salesman, commission.

SELECT c.cust_name AS "Customer Name", c.city, s.name AS Salesman, s.commission

FROM salesman s

JOIN customer c ON s.salesman_id = c.salesman_id;



```sql
SELECT c.cust_name AS "Customer Name", c.city, s.name AS Salesman, s.commission
FROM salesman s
JOIN customer c ON s.salesman_id = c.salesman_id;
```

	Customer Name	city	Salesman	commission
1	Brad Guzan	London	Pit Alex	0.11
2	Nick Rimando	New York	James Hoog	0.15
3	Jozy Altidor	Moscow	Paul Adam	0.13
4	Fabian Johnson	Paris	Mc Lyon	0.14
5	Graham Zusi	California	Nail Knite	0.13
6	Brad Davis	New York	James Hoog	0.15
7	Julian Green	London	Nail Knite	0.13
8	Geoff Cameron	Berlin	Lauson Hen	0.12

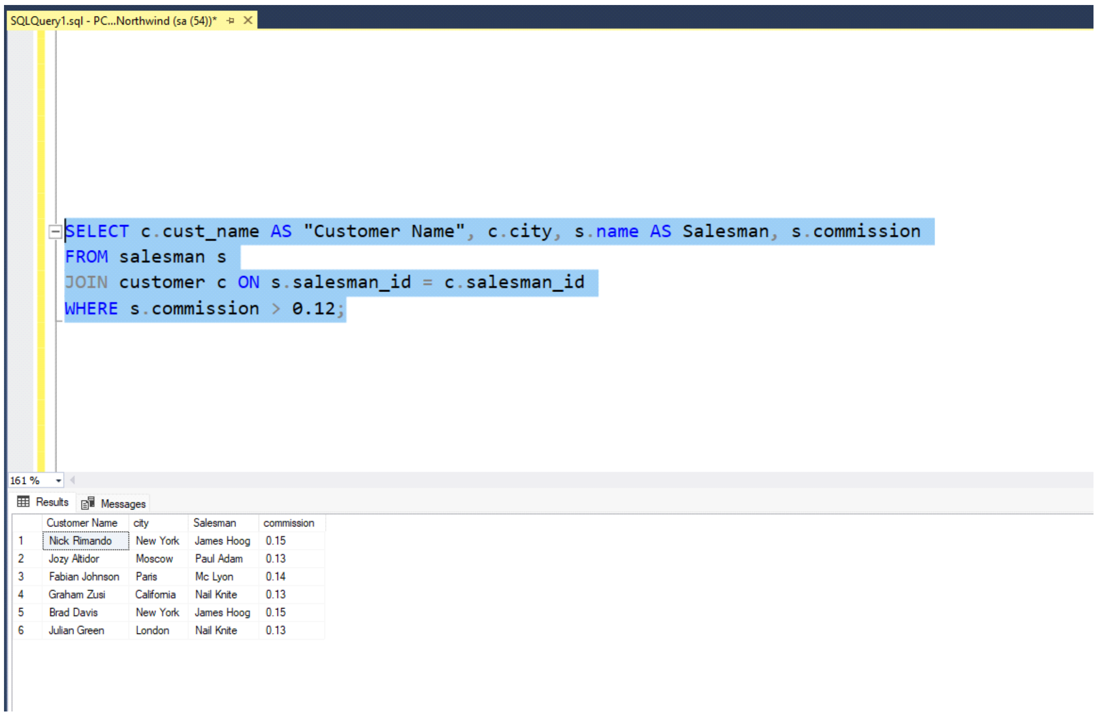
4. write a SQL query to find salespeople who received commissions of more than 12 percent from the company. Return Customer Name, customer city, Salesman, commission.

SELECT c.cust_name AS "Customer Name", c.city, s.name AS Salesman, s.commission

FROM salesman s

JOIN customer c ON s.salesman_id = c.salesman_id

WHERE s.commission > 0.12;

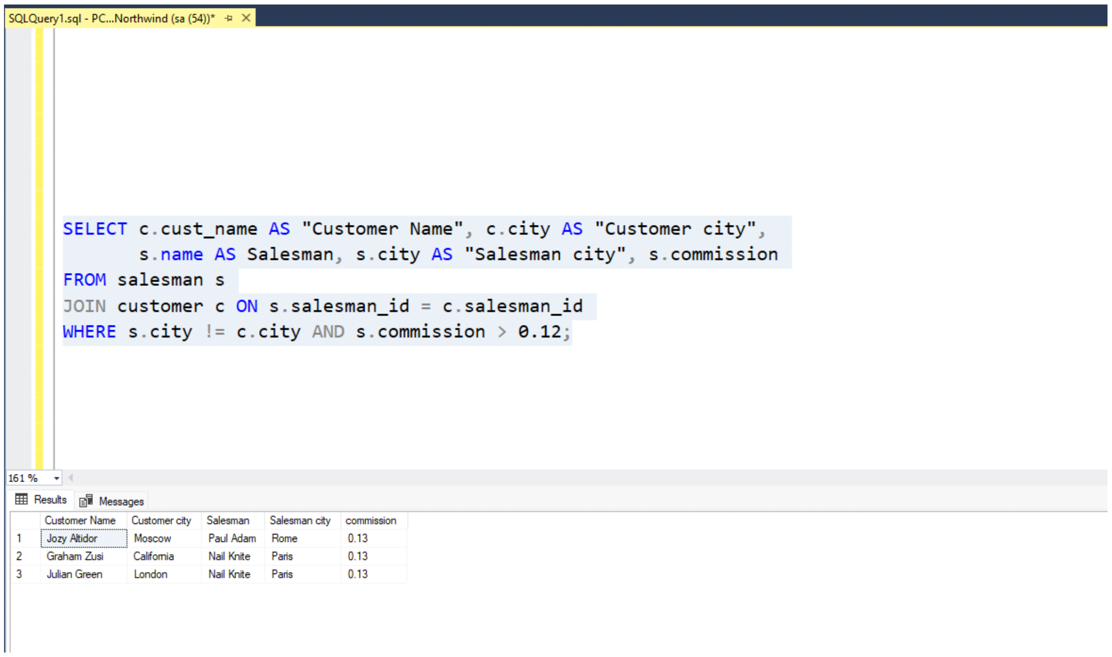


```sql
SELECT c.cust_name AS "Customer Name", c.city, s.name AS Salesman, s.commission
FROM salesman s
JOIN customer c ON s.salesman_id = c.salesman_id
WHERE s.commission > 0.12;
```

	Customer Name	city	Salesman	commission
1	Nick Rimando	New York	James Hoog	0.15
2	Jozy Altidor	Moscow	Paul Adam	0.13
3	Fabian Johnson	Paris	Mc Lyon	0.14
4	Graham Zusi	California	Nail Knite	0.13
5	Brad Davis	New York	James Hoog	0.15
6	Julian Green	London	Nail Knite	0.13

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.

```
SELECT c.cust_name AS "Customer Name", c.city AS "Customer city",
    s.name AS Salesman, s.city AS "Salesman city", s.commission
FROM salesman s
JOIN customer c ON s.salesman_id = c.salesman_id
WHERE s.city != c.city AND s.commission > 0.12;
```

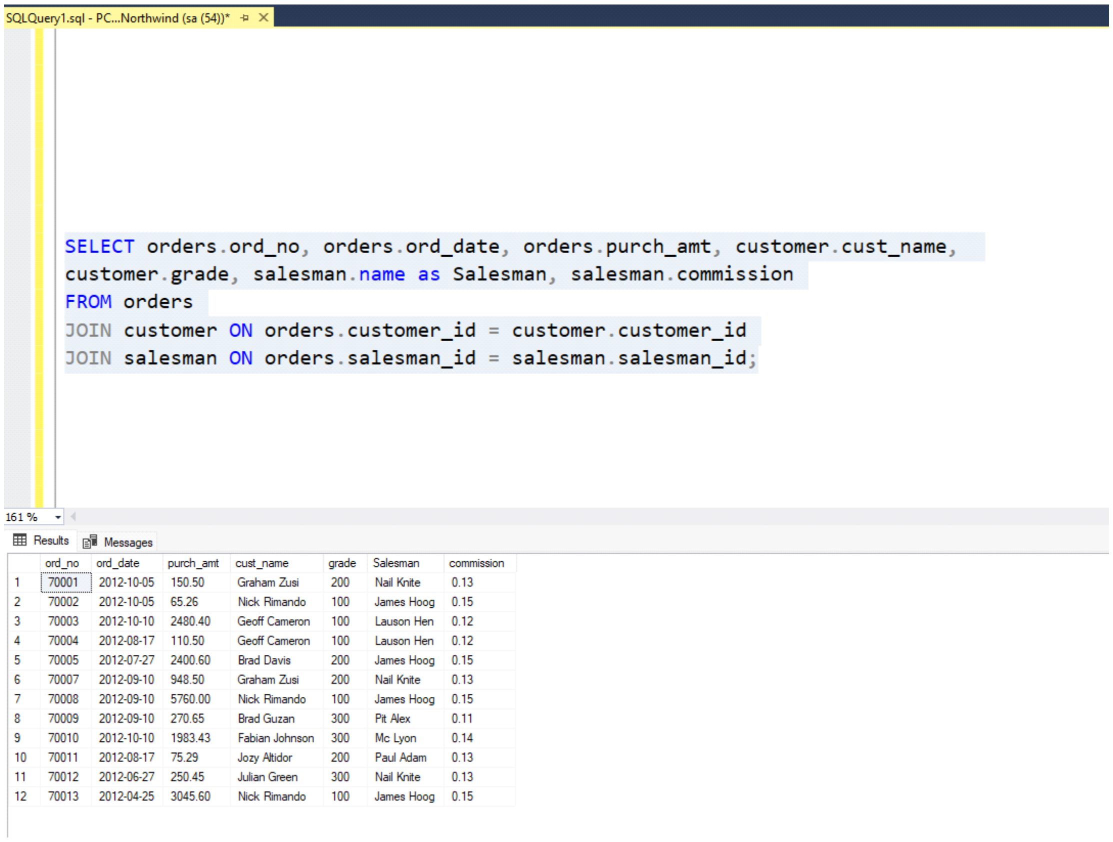


```
SELECT c.cust_name AS "Customer Name", c.city AS "Customer city",
       s.name AS Salesman, s.city AS "Salesman city", s.commission
FROM salesman s
JOIN customer c ON s.salesman_id = c.salesman_id
WHERE s.city != c.city AND s.commission > 0.12;
```

	Customer Name	Customer city	Salesman	Salesman city	commission
1	Jozy Altidor	Moscow	Paul Adam	Rome	0.13
2	Graham Zusi	Califomia	Nail Knite	Paris	0.13
3	Julian Green	London	Nail Knite	Paris	0.13

6. write a SQL query to find the details of an order. Return ord_no, ord_date, purch_amt, Customer Name, grade, Salesman, commission.

```
SELECT orders.ord_no, orders.ord_date, orders.purch_amt,
customer.cust_name,

customer.grade, salesman.name as Salesman, salesman.commission

FROM orders

JOIN customer ON orders.customer_id = customer.customer_id

JOIN salesman ON orders.salesman_id = salesman.salesman_id;
```



SQLQuery1.sql - PC...Northwind (sa (54))*

```
SELECT orders.ord_no, orders.ord_date, orders.purch_amt, customer.cust_name,
customer.grade, salesman.name as Salesman, salesman.commission
FROM orders
JOIN customer ON orders.customer_id = customer.customer_id
JOIN salesman ON orders.salesman_id = salesman.salesman_id;
```

161 %

Results | Messages

	ord_no	ord_date	purch_amt	cust_name	grade	Salesman	commission
1	70001	2012-10-05	150.50	Graham Zusi	200	Nail Knite	0.13
2	70002	2012-10-05	65.26	Nick Rimando	100	James Hoog	0.15
3	70003	2012-10-10	2480.40	Geoff Cameron	100	Lauson Hen	0.12
4	70004	2012-08-17	110.50	Geoff Cameron	100	Lauson Hen	0.12
5	70005	2012-07-27	2400.60	Brad Davis	200	James Hoog	0.15
6	70007	2012-09-10	948.50	Graham Zusi	200	Nail Knite	0.13
7	70008	2012-09-10	5760.00	Nick Rimando	100	James Hoog	0.15
8	70009	2012-09-10	270.65	Brad Guzan	300	Pit Alex	0.11
9	70010	2012-10-10	1983.43	Fabian Johnson	300	Mc Lyon	0.14
10	70011	2012-08-17	75.29	Jozy Altidor	200	Paul Adam	0.13
11	70012	2012-06-27	250.45	Julian Green	300	Nail Knite	0.13
12	70013	2012-04-25	3045.60	Nick Rimando	100	James Hoog	0.15

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.

SELECT salesman.salesman_id, salesman.name as Salesman, salesman.city as salesman_city,

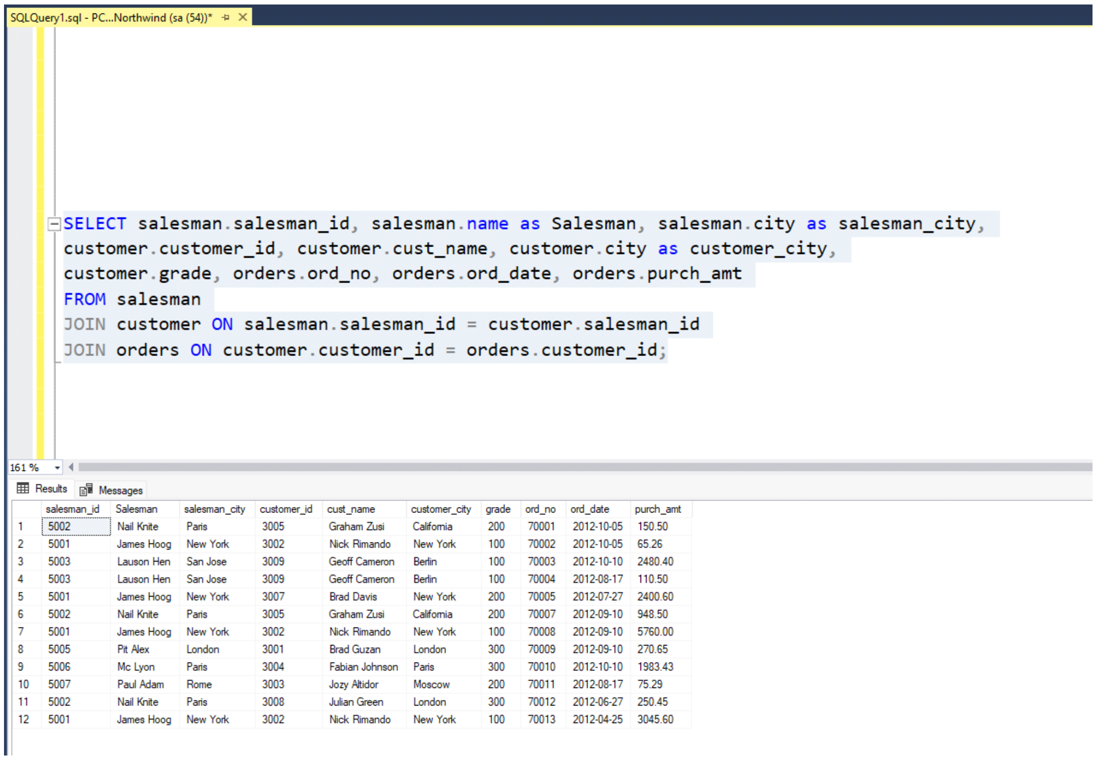
customer.customer_id, customer.cust_name, customer.city as customer_city,

customer.grade, orders.ord_no, orders.ord_date, orders.purch_amt

FROM salesman

JOIN customer ON salesman.salesman_id = customer.salesman_id

JOIN orders ON customer.customer_id = orders.customer_id;



SQLQuery1.sql - PC...Northwind (sa (54))*

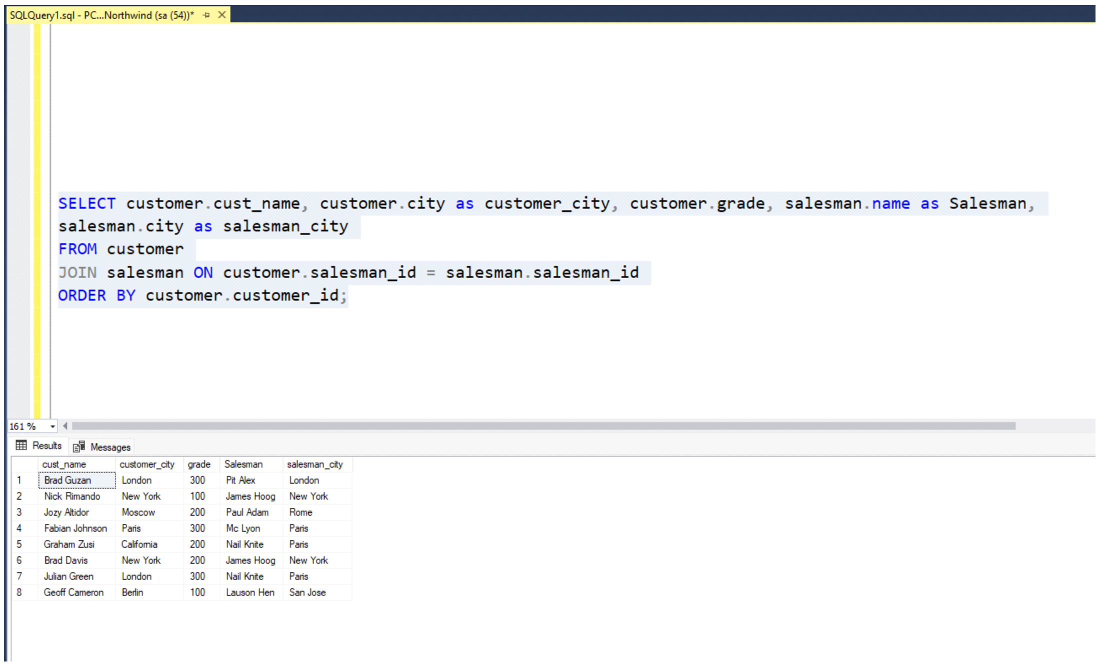
```sql
SELECT salesman.salesman_id, salesman.name as Salesman, salesman.city as salesman_city,
customer.customer_id, customer.cust_name, customer.city as customer_city,
customer.grade, orders.ord_no, orders.ord_date, orders.purch_amt
FROM salesman
JOIN customer ON salesman.salesman_id = customer.salesman_id
JOIN orders ON customer.customer_id = orders.customer_id;
```

161 %

Results | Messages

	salesman_id	Salesman	salesman_city	customer_id	cust_name	customer_city	grade	ord_no	ord_date	purch_amt
1	5002	Nail Knite	Paris	3005	Graham Zusi	California	200	70001	2012-10-05	150.50
2	5001	James Hoog	New York	3002	Nick Rimando	New York	100	70002	2012-10-05	65.26
3	5003	Lauson Hen	San Jose	3009	Geoff Cameron	Berlin	100	70003	2012-10-10	2480.40
4	5003	Lauson Hen	San Jose	3009	Geoff Cameron	Berlin	100	70004	2012-08-17	110.50
5	5001	James Hoog	New York	3007	Brad Davis	New York	200	70005	2012-07-27	2400.60
6	5002	Nail Knite	Paris	3005	Graham Zusi	California	200	70007	2012-09-10	948.50
7	5001	James Hoog	New York	3002	Nick Rimando	New York	100	70008	2012-09-10	5760.00
8	5005	Pit Alex	London	3001	Brad Guzan	London	300	70009	2012-09-10	270.65
9	5006	Mc Lyon	Paris	3004	Fabian Johnson	Paris	300	70010	2012-10-10	1983.43
10	5007	Paul Adam	Rome	3003	Jozy Altidor	Moscow	200	70011	2012-08-17	75.29
11	5002	Nail Knite	Paris	3008	Julian Green	London	300	70012	2012-06-27	250.45
12	5001	James Hoog	New York	3002	Nick Rimando	New York	100	70013	2012-04-25	3045.60

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.



```sql
SELECT customer.cust_name, customer.city as customer_city, customer.grade, salesman.name as Salesman,
salesman.city as salesman_city
FROM customer
JOIN salesman ON customer.salesman_id = salesman.salesman_id
ORDER BY customer.customer_id;
```

	cust_name	customer_city	grade	Salesman	salesman_city
1	Brad Guzan	London	300	Pit Alex	London
2	Nick Rimando	New York	100	James Hoog	New York
3	Jozy Altidor	Moscow	200	Paul Adam	Rome
4	Fabian Johnson	Paris	300	Mc Lyon	Paris
5	Graham Zusi	California	200	Nail Knite	Paris
6	Brad Davis	New York	200	James Hoog	New York
7	Julian Green	London	300	Nail Knite	Paris
8	Geoff Cameron	Berlin	100	Lauson Hen	San Jose

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.

SELECT customer.cust_name, customer.city as customer_city, customer.grade, salesman.name as Salesman,

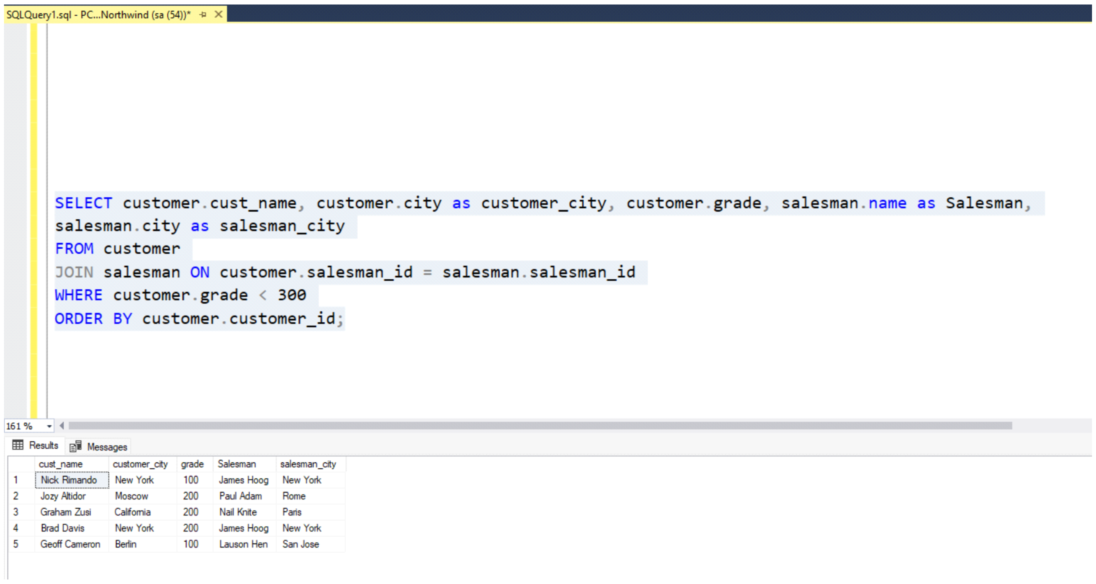
salesman.city as salesman_city

FROM customer

JOIN salesman ON customer.salesman_id = salesman.salesman_id

WHERE customer.grade < 300

ORDER BY customer.customer_id;



SQLQuery1.sql - PC...Northwind (sa (54))*

```
SELECT customer.cust_name, customer.city as customer_city, customer.grade, salesman.name as Salesman,
salesman.city as salesman_city
FROM customer
JOIN salesman ON customer.salesman_id = salesman.salesman_id
WHERE customer.grade < 300
ORDER BY customer.customer_id;
```

161 %

Results | Messages

	cust_name	customer_city	grade	Salesman	salesman_city
1	Nick Rimando	New York	100	James Hoog	New York
2	Jozy Altidor	Moscow	200	Paul Adam	Rome
3	Graham Zusi	California	200	Nail Knite	Paris
4	Brad Davis	New York	200	James Hoog	New York
5	Geoff Cameron	Berlin	100	Lauson Hen	San Jose

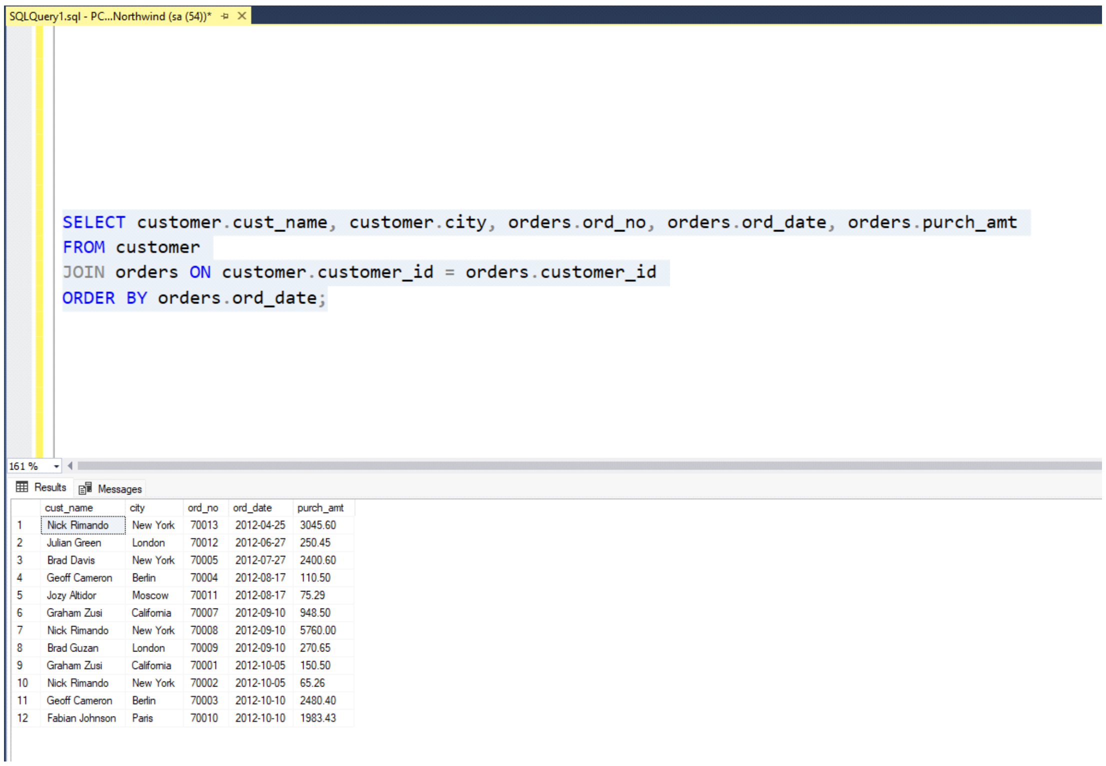
10. Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the order date to determine whether any of the existing customers have placed an order or not.

SELECT customer.cust_name, customer.city, orders.ord_no, orders.ord_date, orders.purch_amt

FROM customer

JOIN orders ON customer.customer_id = orders.customer_id

ORDER BY orders.ord_date;


SQLQuery1.sql - PC...Northwind (sa (54))*

```
SELECT customer.cust_name, customer.city, orders.ord_no, orders.ord_date, orders.purch_amt
FROM customer
JOIN orders ON customer.customer_id = orders.customer_id
ORDER BY orders.ord_date;
```

161 %

Results | Messages

	cust_name	city	ord_no	ord_date	purch_amt
1	Nick Rimando	New York	70013	2012-04-25	3045.60
2	Julian Green	London	70012	2012-06-27	250.45
3	Brad Davis	New York	70005	2012-07-27	2400.60
4	Geoff Cameron	Berlin	70004	2012-08-17	110.50
5	Jozy Altidor	Moscow	70011	2012-08-17	75.29
6	Graham Zusi	California	70007	2012-09-10	948.50
7	Nick Rimando	New York	70008	2012-09-10	5760.00
8	Brad Guzan	London	70009	2012-09-10	270.65
9	Graham Zusi	California	70001	2012-10-05	150.50
10	Nick Rimando	New York	70002	2012-10-05	65.26
11	Geoff Cameron	Berlin	70003	2012-10-10	2480.40
12	Fabian Johnson	Paris	70010	2012-10-10	1983.43

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.

SELECT customer.cust_name, customer.city, orders.ord_no, orders.ord_date, orders.purch_amt,

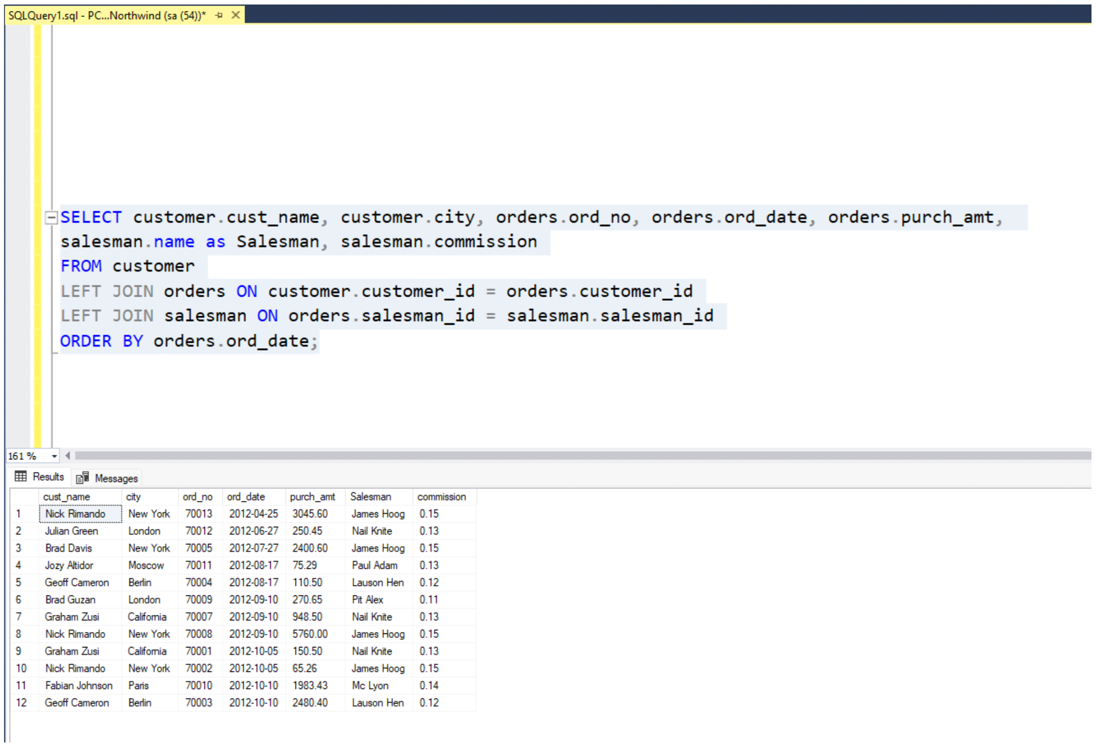
salesman.name as Salesman, salesman.commission

FROM customer

LEFT JOIN orders ON customer.customer_id = orders.customer_id

LEFT JOIN salesman ON orders.salesman_id = salesman.salesman_id

ORDER BY orders.ord_date;



```
SELECT customer.cust_name, customer.city, orders.ord_no, orders.ord_date, orders.purch_amt,
salesman.name as Salesman, salesman.commission
FROM customer
LEFT JOIN orders ON customer.customer_id = orders.customer_id
LEFT JOIN salesman ON orders.salesman_id = salesman.salesman_id
ORDER BY orders.ord_date;
```

	cust_name	city	ord_no	ord_date	purch_amt	Salesman	commission
1	Nick Rimando	New York	70013	2012-04-25	3045.60	James Hoog	0.15
2	Julian Green	London	70012	2012-06-27	250.45	Nail Knite	0.13
3	Brad Davis	New York	70005	2012-07-27	2400.60	James Hoog	0.15
4	Jozy Altidor	Moscow	70011	2012-08-17	75.29	Paul Adam	0.13
5	Geoff Cameron	Berlin	70004	2012-08-17	110.50	Lauson Hen	0.12
6	Brad Guzan	London	70009	2012-09-10	270.65	Pit Alex	0.11
7	Graham Zusi	California	70007	2012-09-10	948.50	Nail Knite	0.13
8	Nick Rimando	New York	70008	2012-09-10	5760.00	James Hoog	0.15
9	Graham Zusi	California	70001	2012-10-05	150.50	Nail Knite	0.13
10	Nick Rimando	New York	70002	2012-10-05	65.26	James Hoog	0.15
11	Fabian Johnson	Paris	70010	2012-10-10	1983.43	Mc Lyon	0.14
12	Geoff Cameron	Berlin	70003	2012-10-10	2480.40	Lauson Hen	0.12

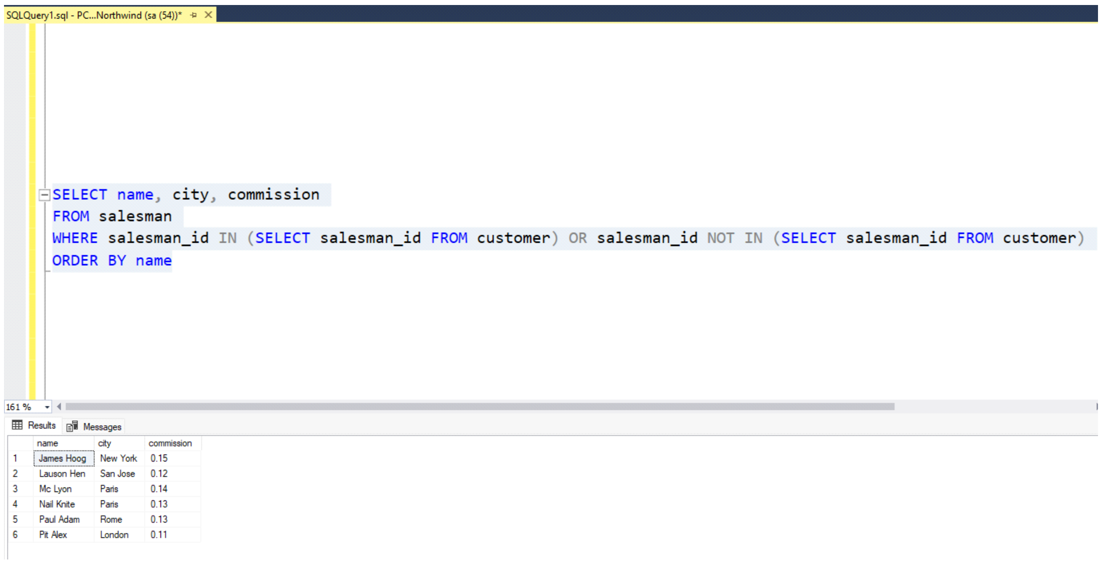
12. Write a SQL statement to generate a list in ascending order of salespersons who work either for one or more customers or have not yet joined any of the customers.

SELECT name, city, commission

FROM salesman

WHERE salesman_id IN (SELECT salesman_id FROM customer) OR salesman_id NOT IN (SELECT salesman_id FROM customer)

ORDER BY name

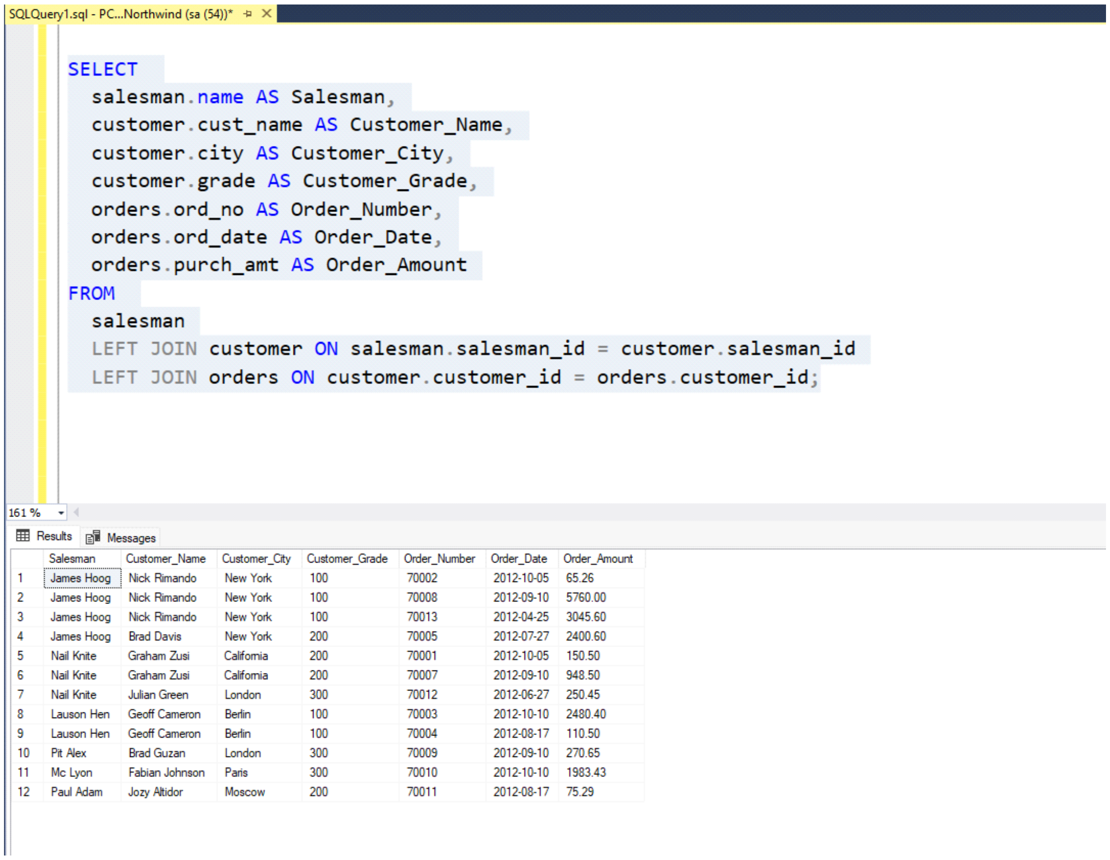


```
SELECT name, city, commission
FROM salesman
WHERE salesman_id IN (SELECT salesman_id FROM customer) OR salesman_id NOT IN (SELECT salesman_id FROM customer)
ORDER BY name
```

	name	city	commission
1	James Hoog	New York	0.15
2	Lauson Hen	San Jose	0.12
3	Mc Lyon	Paris	0.14
4	Nail Knite	Paris	0.13
5	Paul Adam	Rome	0.13
6	Pit Alex	London	0.11

13. write a SQL query to list all salespersons along with customer name, city, grade, order number, date, and amount.

```
SELECT
  salesman.name AS Salesman,
  customer.cust_name AS Customer_Name,
  customer.city AS Customer_City,
  customer.grade AS Customer_Grade,
  orders.ord_no AS Order_Number,
  orders.ord_date AS Order_Date,
  orders.purch_amt AS Order_Amount
FROM
  salesman
  LEFT JOIN customer ON salesman.salesman_id = customer.salesman_id
  LEFT JOIN orders ON customer.customer_id = orders.customer_id;
```

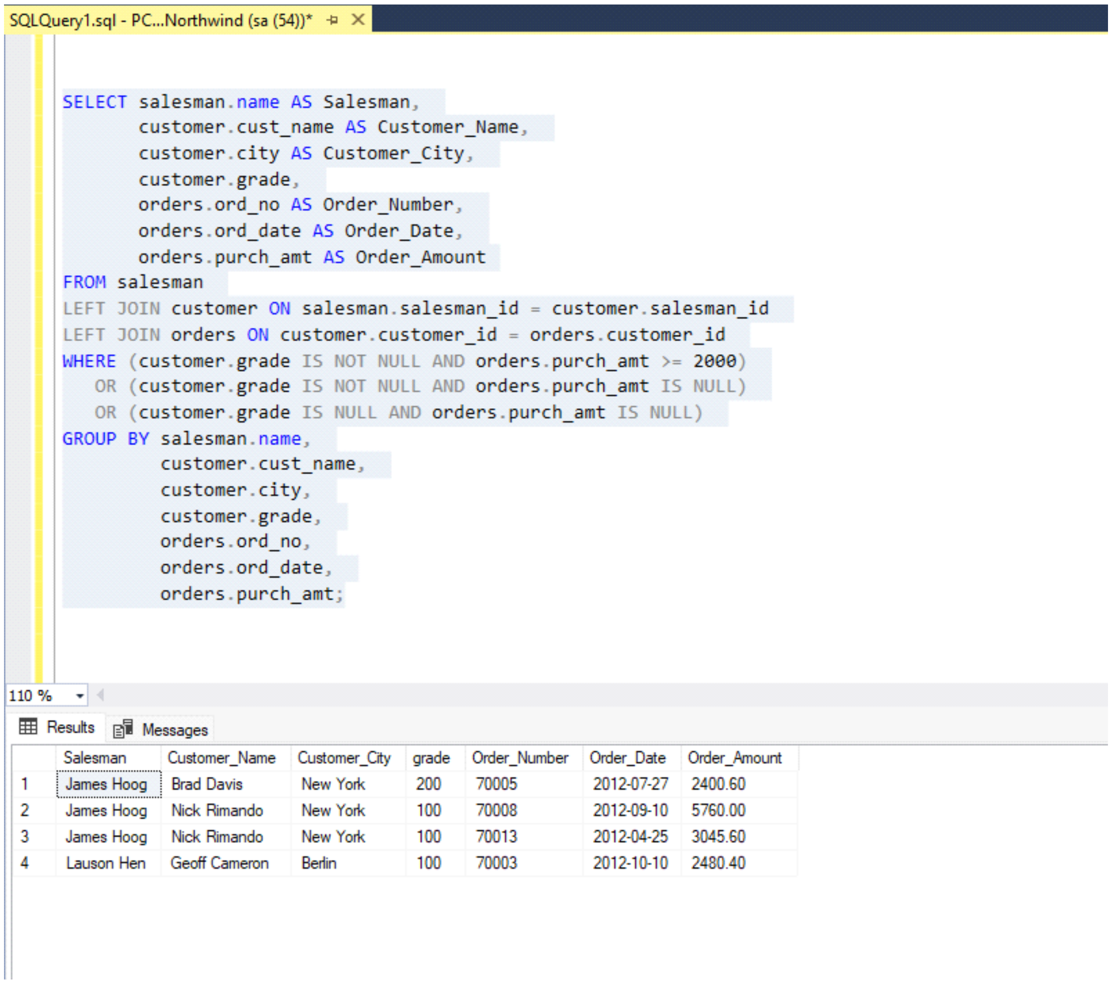


```sql
SELECT
  salesman.name AS Salesman,
  customer.cust_name AS Customer_Name,
  customer.city AS Customer_City,
  customer.grade AS Customer_Grade,
  orders.ord_no AS Order_Number,
  orders.ord_date AS Order_Date,
  orders.purch_amt AS Order_Amount
FROM
  salesman
  LEFT JOIN customer ON salesman.salesman_id = customer.salesman_id
  LEFT JOIN orders ON customer.customer_id = orders.customer_id;
```

	Salesman	Customer_Name	Customer_City	Customer_Grade	Order_Number	Order_Date	Order_Amount
1	James Hoog	Nick Rimando	New York	100	70002	2012-10-05	65.26
2	James Hoog	Nick Rimando	New York	100	70008	2012-09-10	5760.00
3	James Hoog	Nick Rimando	New York	100	70013	2012-04-25	3045.60
4	James Hoog	Brad Davis	New York	200	70005	2012-07-27	2400.60
5	Nail Knite	Graham Zusi	California	200	70001	2012-10-05	150.50
6	Nail Knite	Graham Zusi	California	200	70007	2012-09-10	948.50
7	Nail Knite	Julian Green	London	300	70012	2012-06-27	250.45
8	Lauson Hen	Geoff Cameron	Berlin	100	70003	2012-10-10	2480.40
9	Lauson Hen	Geoff Cameron	Berlin	100	70004	2012-08-17	110.50
10	Pit Alex	Brad Guzan	London	300	70009	2012-09-10	270.65
11	Mc Lyon	Fabian Johnson	Paris	300	70010	2012-10-10	1983.43
12	Paul Adam	Jozy Altidor	Moscow	200	70011	2012-08-17	75.29

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.

```
SELECT salesman.name AS Salesman,
       customer.cust_name AS Customer_Name,
       customer.city AS Customer_City,
       customer.grade,
       orders.ord_no AS Order_Number,
       orders.ord_date AS Order_Date,
       orders.purch_amt AS Order_Amount
FROM salesman
LEFT JOIN customer ON salesman.salesman_id = customer.salesman_id
LEFT JOIN orders ON customer.customer_id = orders.customer_id
WHERE (customer.grade IS NOT NULL AND orders.purch_amt >= 2000)
   OR (customer.grade IS NOT NULL AND orders.purch_amt IS NULL)
   OR (customer.grade IS NULL AND orders.purch_amt IS NULL)
GROUP BY salesman.name,
         customer.cust_name,
         customer.city,
         customer.grade,
         orders.ord_no,
         orders.ord_date,
         orders.purch_amt;
```


SQLQuery1.sql - PC...Northwind (sa (54))*

```
SELECT salesman.name AS Salesman,
       customer.cust_name AS Customer_Name,
       customer.city AS Customer_City,
       customer.grade,
       orders.ord_no AS Order_Number,
       orders.ord_date AS Order_Date,
       orders.purch_amt AS Order_Amount
FROM salesman
LEFT JOIN customer ON salesman.salesman_id = customer.salesman_id
LEFT JOIN orders ON customer.customer_id = orders.customer_id
WHERE (customer.grade IS NOT NULL AND orders.purch_amt >= 2000)
   OR (customer.grade IS NOT NULL AND orders.purch_amt IS NULL)
   OR (customer.grade IS NULL AND orders.purch_amt IS NULL)
GROUP BY salesman.name,
         customer.cust_name,
         customer.city,
         customer.grade,
         orders.ord_no,
         orders.ord_date,
         orders.purch_amt;
```

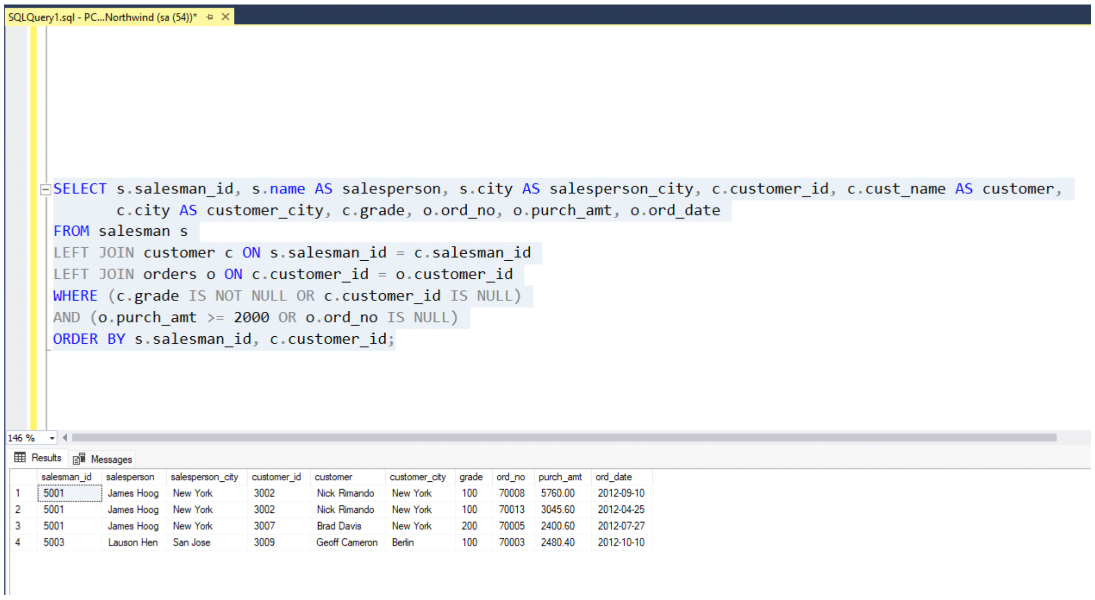
110 %

Results Messages

	Salesman	Customer_Name	Customer_City	grade	Order_Number	Order_Date	Order_Amount
1	James Hoog	Brad Davis	New York	200	70005	2012-07-27	2400.60
2	James Hoog	Nick Rimando	New York	100	70008	2012-09-10	5760.00
3	James Hoog	Nick Rimando	New York	100	70013	2012-04-25	3045.60
4	Lauson Hen	Geoff Cameron	Berlin	100	70003	2012-10-10	2480.40

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.salesman_id, s.name AS salesperson, s.city AS salesperson_city, c.customer_id, c.cust_name AS customer, c.city AS customer_city, c.grade, o.ord_no, o.purch_amt, o.ord_date

FROM salesman s

LEFT JOIN customer c ON s.salesman_id = c.salesman_id

LEFT JOIN orders o ON c.customer_id = o.customer_id

WHERE (c.grade IS NOT NULL OR c.customer_id IS NULL)

AND (o.purch_amt >= 2000 OR o.ord_no IS NULL)

ORDER BY s.salesman_id, c.customer_id;
```


SQLQuery1.sql - PC...Northwind (sa (54))*

```
SELECT s.salesman_id, s.name AS salesperson, s.city AS salesperson_city, c.customer_id, c.cust_name AS customer,
       c.city AS customer_city, c.grade, o.ord_no, o.purch_amt, o.ord_date
FROM salesman s
LEFT JOIN customer c ON s.salesman_id = c.salesman_id
LEFT JOIN orders o ON c.customer_id = o.customer_id
WHERE (c.grade IS NOT NULL OR c.customer_id IS NULL)
AND (o.purch_amt >= 2000 OR o.ord_no IS NULL)
ORDER BY s.salesman_id, c.customer_id;
```

146 %

Results | Messages

	salesman_id	salesperson	salesperson_city	customer_id	customer	customer_city	grade	ord_no	purch_amt	ord_date
1	5001	James Hoog	New York	3002	Nick Rimando	New York	100	70008	5760.00	2012-09-10
2	5001	James Hoog	New York	3002	Nick Rimando	New York	100	70013	3045.60	2012-04-25
3	5001	James Hoog	New York	3007	Brad Davis	New York	200	70005	2400.60	2012-07-27
4	5003	Lauson Hen	San Jose	3009	Geoff Cameron	Berlin	100	70003	2480.40	2012-10-10

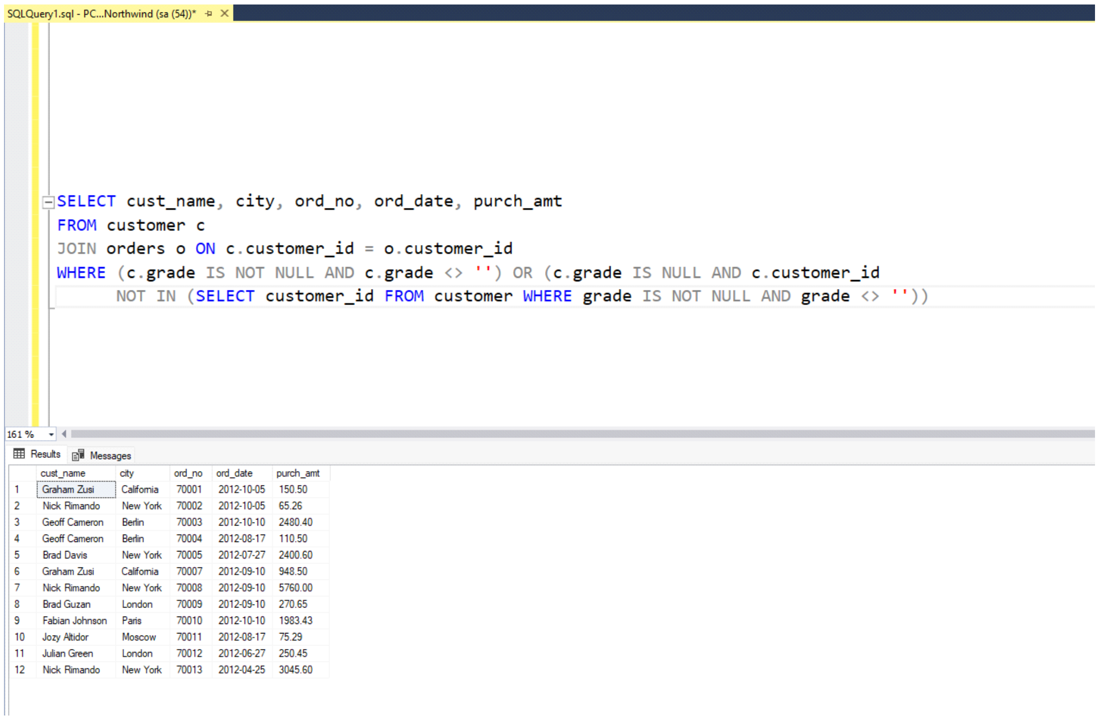
16. Write a SQL statement to generate a report with the customer name, city, order no. order date, purchase amount for only those customers on the list who must have a grade and placed one or more orders or which order(s) have been placed by the customer who neither is on the list nor has a grade.

SELECT cust_name, city, ord_no, ord_date, purch_amt

FROM customer c

JOIN orders o ON c.customer_id = o.customer_id

WHERE (c.grade IS NOT NULL AND c.grade <> '') OR (c.grade IS NULL AND c.customer_id NOT IN (SELECT customer_id FROM customer WHERE grade IS NOT NULL AND grade <> ''))



SQLQuery1.sql - PC...Northwind (sa (54))*

```
SELECT cust_name, city, ord_no, ord_date, purch_amt
FROM customer c
JOIN orders o ON c.customer_id = o.customer_id
WHERE (c.grade IS NOT NULL AND c.grade <> '') OR (c.grade IS NULL AND c.customer_id
      NOT IN (SELECT customer_id FROM customer WHERE grade IS NOT NULL AND grade <> ''))
```

161 %

Results | Messages

	cust_name	city	ord_no	ord_date	purch_amt
1	Graham Zusi	California	70001	2012-10-05	150.50
2	Nick Rimando	New York	70002	2012-10-05	65.26
3	Geoff Cameron	Berlin	70003	2012-10-10	2480.40
4	Geoff Cameron	Berlin	70004	2012-08-17	110.50
5	Brad Davis	New York	70005	2012-07-27	2400.60
6	Graham Zusi	California	70007	2012-09-10	948.50
7	Nick Rimando	New York	70008	2012-09-10	5760.00
8	Brad Guzan	London	70009	2012-09-10	270.65
9	Fabian Johnson	Paris	70010	2012-10-10	1983.43
10	Jozy Altidor	Moscow	70011	2012-08-17	75.29
11	Julian Green	London	70012	2012-06-27	250.45
12	Nick Rimando	New York	70013	2012-04-25	3045.60

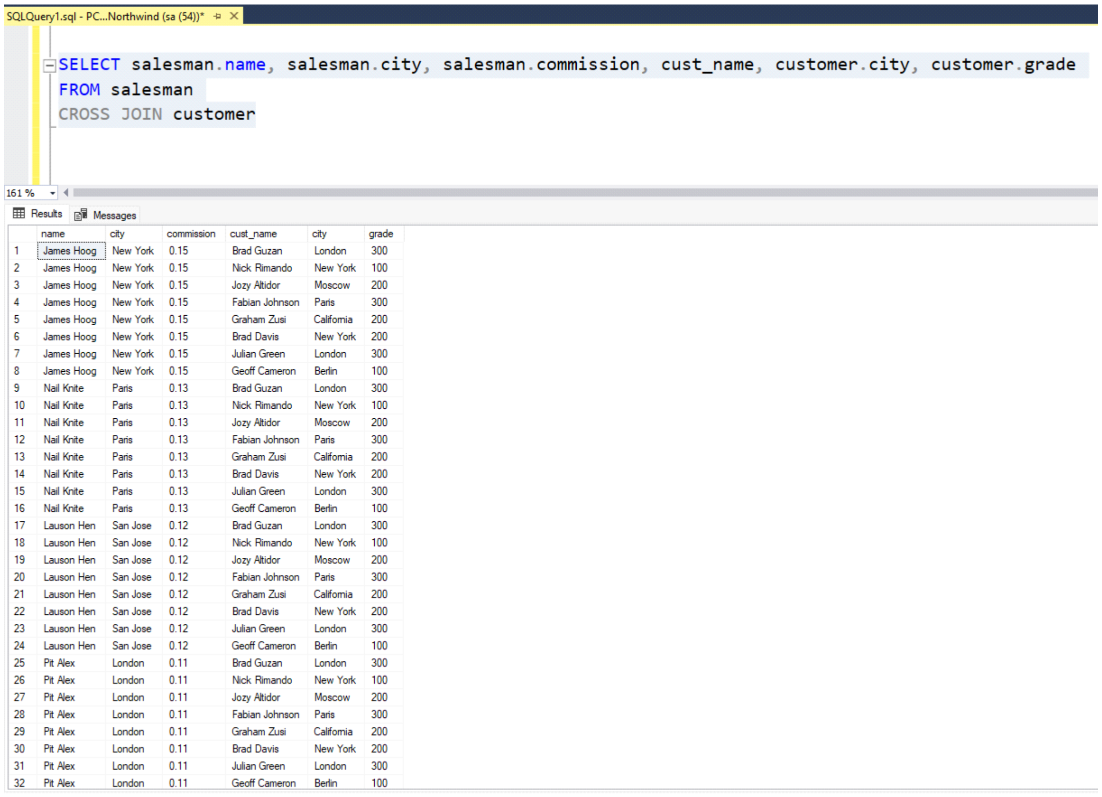
17. Write a SQL query to combine each row of the salesman table with each row

of the customer table.

SELECT salesman.name, salesman.city, salesman.commission, cust_name, customer.city, customer.grade

FROM salesman

CROSS JOIN customer



SQLQuery1.sql - PC...Northwind (sa (54))*

```
SELECT salesman.name, salesman.city, salesman.commission, cust_name, customer.city, customer.grade
FROM salesman
CROSS JOIN customer
```

161 %

Results | Messages

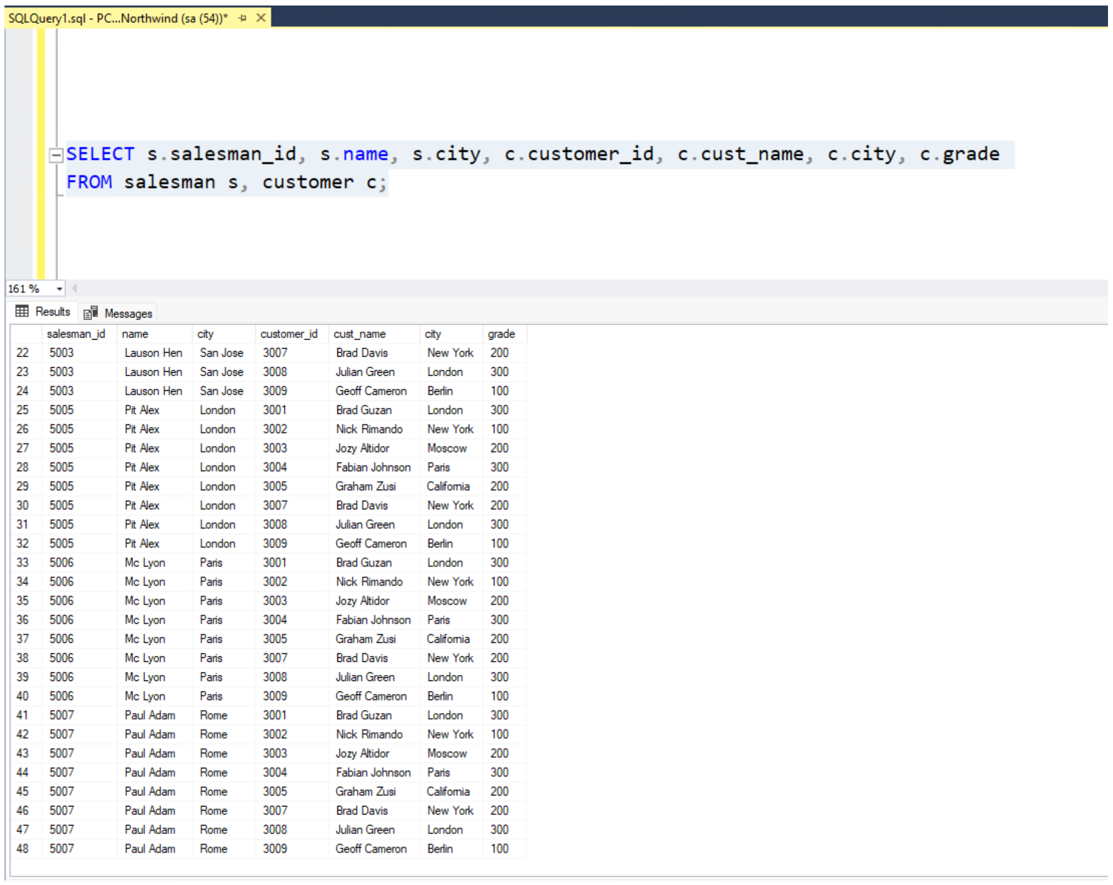
	name	city	commission	cust_name	city	grade
1	James Hoog	New York	0.15	Brad Guzan	London	300
2	James Hoog	New York	0.15	Nick Rimando	New York	100
3	James Hoog	New York	0.15	Jozy Altidor	Moscow	200
4	James Hoog	New York	0.15	Fabian Johnson	Paris	300
5	James Hoog	New York	0.15	Graham Zusi	California	200
6	James Hoog	New York	0.15	Brad Davis	New York	200
7	James Hoog	New York	0.15	Julian Green	London	300
8	James Hoog	New York	0.15	Geoff Cameron	Berlin	100
9	Nail Knite	Paris	0.13	Brad Guzan	London	300
10	Nail Knite	Paris	0.13	Nick Rimando	New York	100
11	Nail Knite	Paris	0.13	Jozy Altidor	Moscow	200
12	Nail Knite	Paris	0.13	Fabian Johnson	Paris	300
13	Nail Knite	Paris	0.13	Graham Zusi	California	200
14	Nail Knite	Paris	0.13	Brad Davis	New York	200
15	Nail Knite	Paris	0.13	Julian Green	London	300
16	Nail Knite	Paris	0.13	Geoff Cameron	Berlin	100
17	Lauson Hen	San Jose	0.12	Brad Guzan	London	300
18	Lauson Hen	San Jose	0.12	Nick Rimando	New York	100
19	Lauson Hen	San Jose	0.12	Jozy Altidor	Moscow	200
20	Lauson Hen	San Jose	0.12	Fabian Johnson	Paris	300
21	Lauson Hen	San Jose	0.12	Graham Zusi	California	200
22	Lauson Hen	San Jose	0.12	Brad Davis	New York	200
23	Lauson Hen	San Jose	0.12	Julian Green	London	300
24	Lauson Hen	San Jose	0.12	Geoff Cameron	Berlin	100
25	Pit Alex	London	0.11	Brad Guzan	London	300
26	Pit Alex	London	0.11	Nick Rimando	New York	100
27	Pit Alex	London	0.11	Jozy Altidor	Moscow	200
28	Pit Alex	London	0.11	Fabian Johnson	Paris	300
29	Pit Alex	London	0.11	Graham Zusi	California	200
30	Pit Alex	London	0.11	Brad Davis	New York	200
31	Pit Alex	London	0.11	Julian Green	London	300
32	Pit Alex	London	0.11	Geoff Cameron	Berlin	100

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.

SELECT s.salesman_id, s.name, s.city, c.customer_id, c.cust_name, c.city, c.grade

FROM salesman s, customer c;



```sql
SELECT s.salesman_id, s.name, s.city, c.customer_id, c.cust_name, c.city, c.grade
FROM salesman s, customer c;
```

	salesman_id	name	city	customer_id	cust_name	city	grade
22	5003	Lauson Hen	San Jose	3007	Brad Davis	New York	200
23	5003	Lauson Hen	San Jose	3008	Julian Green	London	300
24	5003	Lauson Hen	San Jose	3009	Geoff Cameron	Berlin	100
25	5005	Pit Alex	London	3001	Brad Guzan	London	300
26	5005	Pit Alex	London	3002	Nick Rimando	New York	100
27	5005	Pit Alex	London	3003	Jozy Altidor	Moscow	200
28	5005	Pit Alex	London	3004	Fabian Johnson	Paris	300
29	5005	Pit Alex	London	3005	Graham Zusi	California	200
30	5005	Pit Alex	London	3007	Brad Davis	New York	200
31	5005	Pit Alex	London	3008	Julian Green	London	300
32	5005	Pit Alex	London	3009	Geoff Cameron	Berlin	100
33	5006	Mc Lyon	Paris	3001	Brad Guzan	London	300
34	5006	Mc Lyon	Paris	3002	Nick Rimando	New York	100
35	5006	Mc Lyon	Paris	3003	Jozy Altidor	Moscow	200
36	5006	Mc Lyon	Paris	3004	Fabian Johnson	Paris	300
37	5006	Mc Lyon	Paris	3005	Graham Zusi	California	200
38	5006	Mc Lyon	Paris	3007	Brad Davis	New York	200
39	5006	Mc Lyon	Paris	3008	Julian Green	London	300
40	5006	Mc Lyon	Paris	3009	Geoff Cameron	Berlin	100
41	5007	Paul Adam	Rome	3001	Brad Guzan	London	300
42	5007	Paul Adam	Rome	3002	Nick Rimando	New York	100
43	5007	Paul Adam	Rome	3003	Jozy Altidor	Moscow	200
44	5007	Paul Adam	Rome	3004	Fabian Johnson	Paris	300
45	5007	Paul Adam	Rome	3005	Graham Zusi	California	200
46	5007	Paul Adam	Rome	3007	Brad Davis	New York	200
47	5007	Paul Adam	Rome	3008	Julian Green	London	300
48	5007	Paul Adam	Rome	3009	Geoff Cameron	Berlin	100

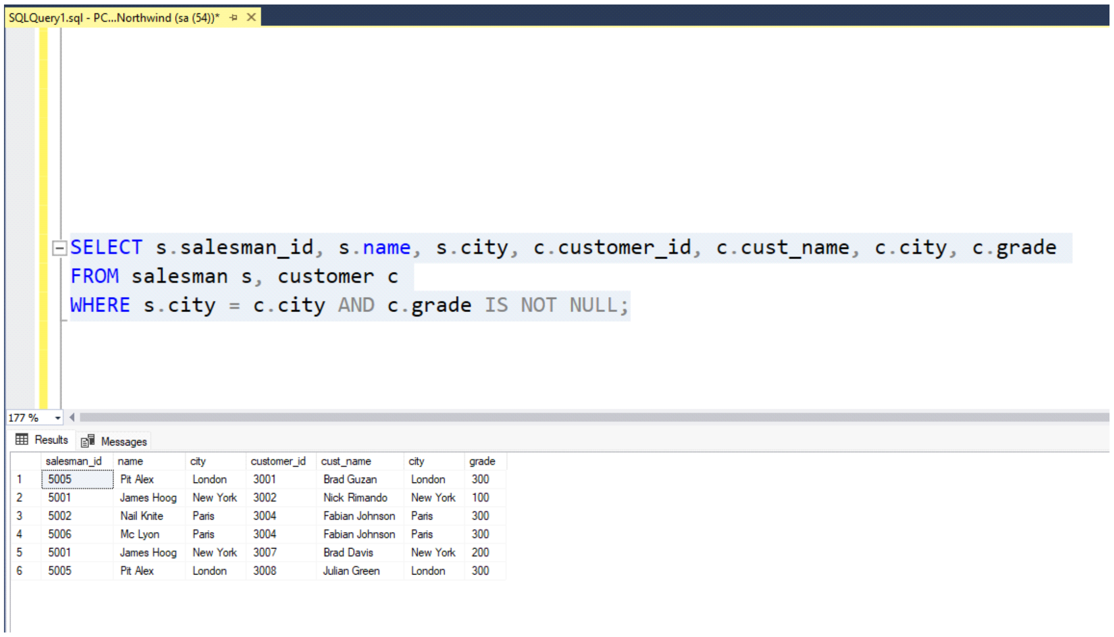
19. Write a SQL statement to create a Cartesian product between salesperson

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.

SELECT s.salesman_id, s.name, s.city, c.customer_id, c.cust_name, c.city, c.grade

FROM salesman s, customer c

WHERE s.city = c.city AND c.grade IS NOT NULL;



```sql
SELECT s.salesman_id, s.name, s.city, c.customer_id, c.cust_name, c.city, c.grade
FROM salesman s, customer c
WHERE s.city = c.city AND c.grade IS NOT NULL;
```

	salesman_id	name	city	customer_id	cust_name	city	grade
1	5005	Pit Alex	London	3001	Brad Guzan	London	300
2	5001	James Hoog	New York	3002	Nick Rimando	New York	100
3	5002	Nail Knite	Paris	3004	Fabian Johnson	Paris	300
4	5006	Mc Lyon	Paris	3004	Fabian Johnson	Paris	300
5	5001	James Hoog	New York	3007	Brad Davis	New York	200
6	5005	Pit Alex	London	3008	Julian Green	London	300

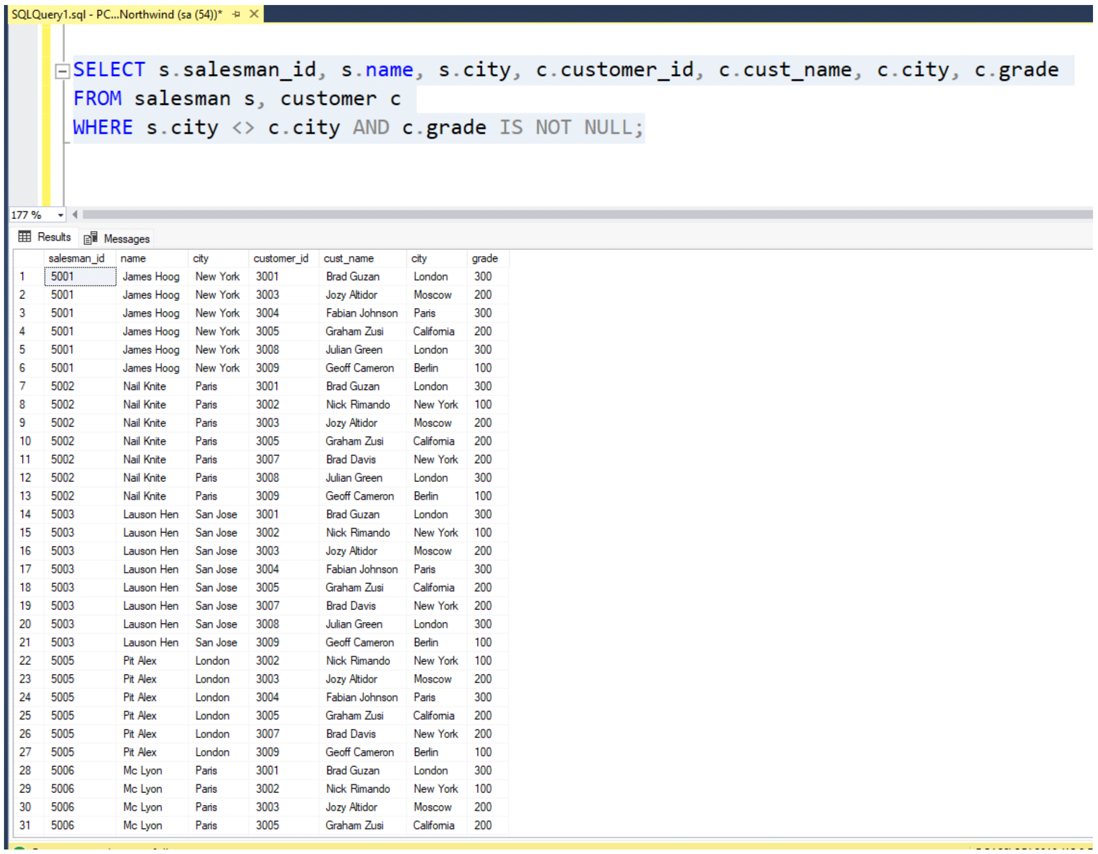
20. Write a SQL statement to make a Cartesian product between salesman and customer i.e. each salesman will appear for all customers and vice versa for

those salesmen who must belong to a city which is not the same as his customer and the customers should have their own grade.

SELECT s.salesman_id, s.name, s.city, c.customer_id, c.cust_name, c.city, c.grade

FROM salesman s, customer c

WHERE s.city <> c.city AND c.grade IS NOT NULL;



```sql
SELECT s.salesman_id, s.name, s.city, c.customer_id, c.cust_name, c.city, c.grade
FROM salesman s, customer c
WHERE s.city <> c.city AND c.grade IS NOT NULL;
```

	salesman_id	name	city	customer_id	cust_name	city	grade
1	5001	James Hoog	New York	3001	Brad Guzan	London	300
2	5001	James Hoog	New York	3003	Jozy Altidor	Moscow	200
3	5001	James Hoog	New York	3004	Fabian Johnson	Paris	300
4	5001	James Hoog	New York	3005	Graham Zusi	California	200
5	5001	James Hoog	New York	3008	Julian Green	London	300
6	5001	James Hoog	New York	3009	Geoff Cameron	Berlin	100
7	5002	Nail Knite	Paris	3001	Brad Guzan	London	300
8	5002	Nail Knite	Paris	3002	Nick Rimando	New York	100
9	5002	Nail Knite	Paris	3003	Jozy Altidor	Moscow	200
10	5002	Nail Knite	Paris	3005	Graham Zusi	California	200
11	5002	Nail Knite	Paris	3007	Brad Davis	New York	200
12	5002	Nail Knite	Paris	3008	Julian Green	London	300
13	5002	Nail Knite	Paris	3009	Geoff Cameron	Berlin	100
14	5003	Lauson Hen	San Jose	3001	Brad Guzan	London	300
15	5003	Lauson Hen	San Jose	3002	Nick Rimando	New York	100
16	5003	Lauson Hen	San Jose	3003	Jozy Altidor	Moscow	200
17	5003	Lauson Hen	San Jose	3004	Fabian Johnson	Paris	300
18	5003	Lauson Hen	San Jose	3005	Graham Zusi	California	200
19	5003	Lauson Hen	San Jose	3007	Brad Davis	New York	200
20	5003	Lauson Hen	San Jose	3008	Julian Green	London	300
21	5003	Lauson Hen	San Jose	3009	Geoff Cameron	Berlin	100
22	5005	Pit Alex	London	3002	Nick Rimando	New York	100
23	5005	Pit Alex	London	3003	Jozy Altidor	Moscow	200
24	5005	Pit Alex	London	3004	Fabian Johnson	Paris	300
25	5005	Pit Alex	London	3005	Graham Zusi	California	200
26	5005	Pit Alex	London	3007	Brad Davis	New York	200
27	5005	Pit Alex	London	3009	Geoff Cameron	Berlin	100
28	5006	Mc Lyon	Paris	3001	Brad Guzan	London	300
29	5006	Mc Lyon	Paris	3002	Nick Rimando	New York	100
30	5006	Mc Lyon	Paris	3003	Jozy Altidor	Moscow	200
31	5006	Mc Lyon	Paris	3005	Graham Zusi	California	200