DEV VIPULKUMAR PATEL

SQL - Assignnmet-2

NAME: PATEL DEV VIPULKUMAR

- **Create Table:**

  QUERY:

- **Salesman Table**

```
CREATE TABLE salesman(
salesman_id int NOT NULL PRIMARY KEY,
name nvarchar(50) NOT NULL,
city nvarchar(50),
commission int,
)
```

- **Customer Table**

```
CREATE TABLE customer(
customer_id int NOT NULL PRIMARY KEY,
cust_name nvarchar(50),
city nvarchar(50),
grade int,
salesman_id int,
);

ALTER TABLE customer
ADD FOREIGN KEY (salesman_id) REFERENCES salesman(salesman_id);
```

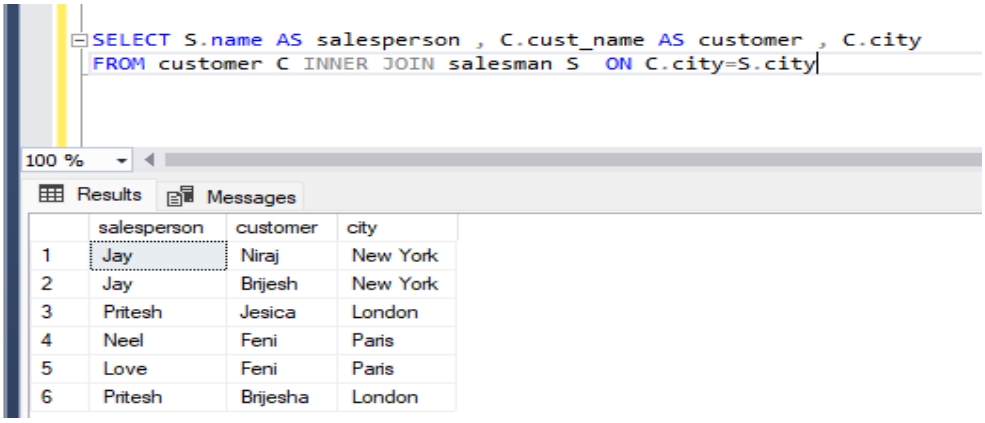
- **Order Table**

```
CREATE TABLE orders(
ord_no int NOT NULL PRIMARY KEY,
purch_amt money NOT NULL,
ord_date date NOT NULL,
customer_id int NOT NULL,
salesman_id int NOT NULL,
FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
FOREIGN KEY (salesman_id) REFERENCES salesman(salesman_id),
);
```

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

  QUERY:1

```
SELECT S.name AS salesperson , C.cust_name AS customer , C.city
FROM customer C INNER JOIN salesman S  ON C.city=S.city
```

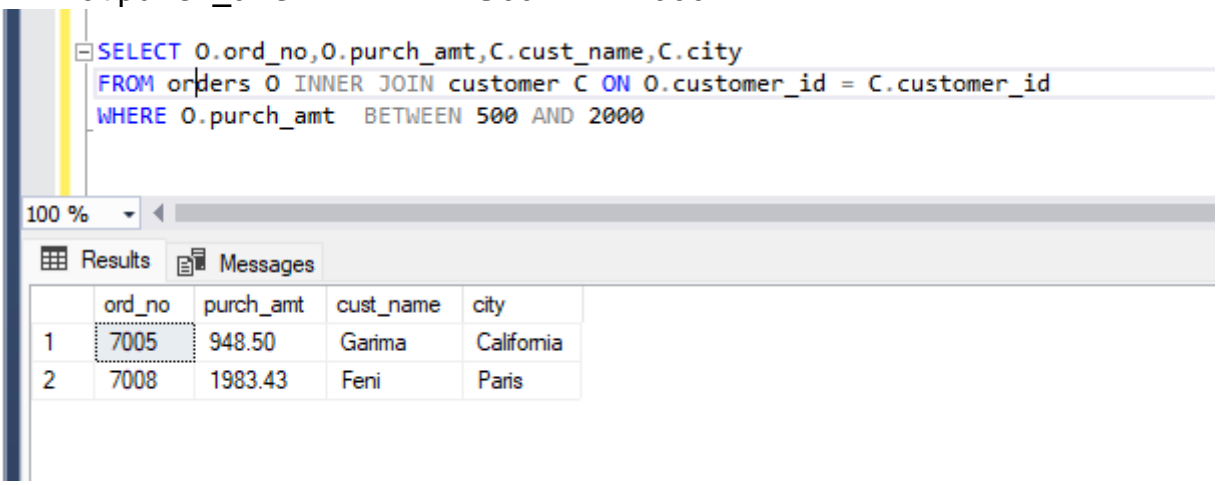


	salesperson	customer	city
1	Jay	Niraj	New York
2	Jay	Brijesh	New York
3	Pritesh	Jesica	London
4	Neel	Feni	Paris
5	Love	Feni	Paris
6	Pritesh	Brijesha	London

- **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.**

  QUERY:2

```
SELECT O.ord_no,O.purch_amt,C.cust_name,C.city
FROM orders O INNER JOIN customer C ON O.customer_id =
C.customer_id
WHERE O.purch_amt  BETWEEN 500 AND 2000
```

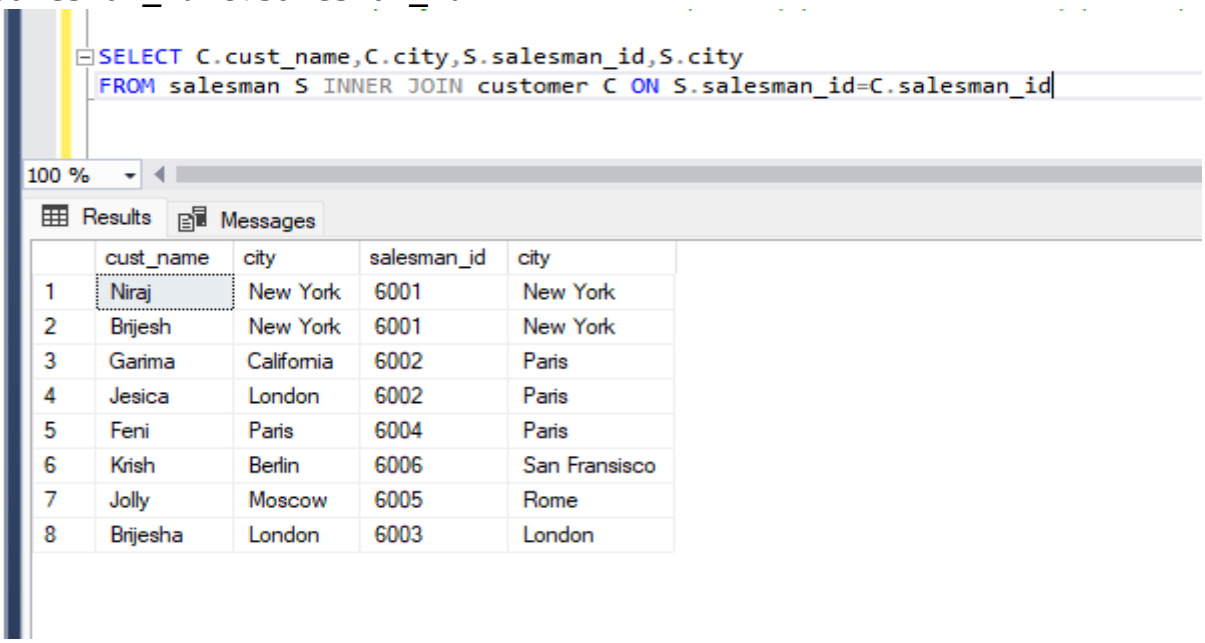


	ord_no	purch_amt	cust_name	city
1	7005	948.50	Garima	California
2	7008	1983.43	Feni	Paris

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

  QUERY:3

```
SELECT C.cust_name,C.city,S.salesman_id,S.city
FROM salesman S INNER JOIN customer C ON
S.salesman_id=C.salesman_id
```



	cust_name	city	salesman_id	city
1	Niraj	New York	6001	New York
2	Brijesh	New York	6001	New York
3	Garima	California	6002	Paris
4	Jesica	London	6002	Paris
5	Feni	Paris	6004	Paris
6	Krish	Berlin	6006	San Fransisco
7	Jolly	Moscow	6005	Rome
8	Brijesha	London	6003	London

- **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.**

  QUERY:4

```
SELECT C.cust_name,C.city,S.name AS Salesman ,S.commission
FROM customer C INNER JOIN salesman S ON C.salesman_id =
S.salesman_id
WHERE S.commission >12
```

```
SELECT C.cust_name,C.city,S.name AS Salesman ,S.commission
FROM customer C INNER JOIN salesman S ON C.salesman_id = S.salesman_id
WHERE S.commission >12
```

100 %

Results | Messages

	cust_name	city	Salesman	commission
1	Niraj	New York	Jay	15
2	Brijesh	New York	Jay	15
3	Garima	California	Neel	13
4	Jesica	London	Neel	13
5	Feni	Paris	Love	14
6	Jolly	Moscow	Paras	13

- **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.**

  QUERY:5

```
SELECT C.cust_name,C.city AS [Customer City],S.name AS
Salesman,S.city AS [Salesman City] ,S.commission
FROM customer C INNER JOIN salesman S ON C.salesman_id =
S.salesman_id
WHERE S.commission >12 AND C.city != S.city
```

```
SELECT C.cust_name,C.city AS [Customer City],S.name AS Salesman,S.city AS [Salesman City] ,S.commission
FROM customer C INNER JOIN salesman S ON C.salesman_id = S.salesman_id
WHERE S.commission >12 AND C.city != S.city
```

100 %

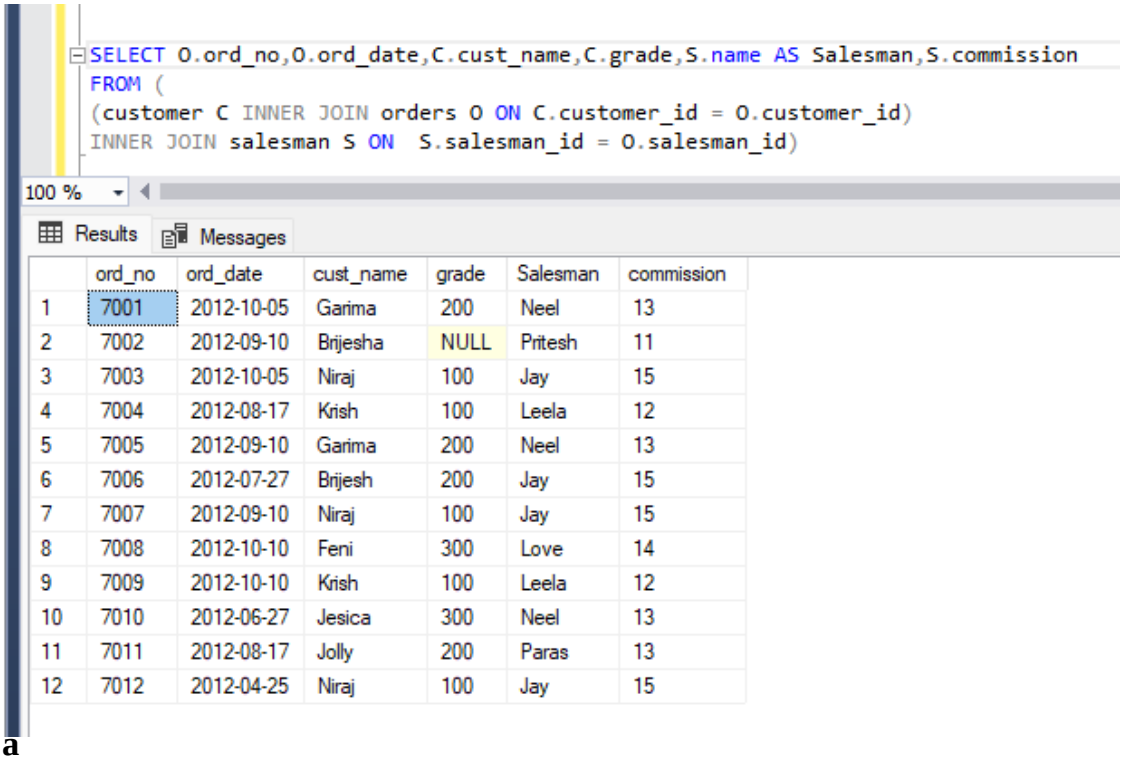
Results | Messages

	cust_name	Customer City	Salesman	Salesman City	commission
1	Garima	California	Neel	Paris	13
2	Jesica	London	Neel	Paris	13
3	Jolly	Moscow	Paras	Rome	13

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

  QUERY:6

```
SELECT O.ord_no,O.ord_date,C.cust_name,C.grade,S.name AS
Salesman,S.commission
FROM (
(customer C INNER JOIN orders O ON C.customer_id = O.customer_id)
INNER JOIN salesman S ON  S.salesman_id = O.salesman_id)
```

- 

```
SELECT O.ord_no,O.ord_date,C.cust_name,C.grade,S.name AS Salesman,S.commission
FROM (
(customer C INNER JOIN orders O ON C.customer_id = O.customer_id)
INNER JOIN salesman S ON  S.salesman_id = O.salesman_id)
```

100 %

Results | Messages

	ord_no	ord_date	cust_name	grade	Salesman	commission
1	7001	2012-10-05	Garima	200	Neel	13
2	7002	2012-09-10	Brijesha	NULL	Pritesh	11
3	7003	2012-10-05	Niraj	100	Jay	15
4	7004	2012-08-17	Krish	100	Leela	12
5	7005	2012-09-10	Garima	200	Neel	13
6	7006	2012-07-27	Brijesh	200	Jay	15
7	7007	2012-09-10	Niraj	100	Jay	15
8	7008	2012-10-10	Feni	300	Love	14
9	7009	2012-10-10	Krish	100	Leela	12
10	7010	2012-06-27	Jesica	300	Neel	13
11	7011	2012-08-17	Jolly	200	Paras	13
12	7012	2012-04-25	Niraj	100	Jay	15

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.

QUERY: 7

```
SELECT O.* , C.cust_name as "customer name" , C.city as "customer
city" , C.grade , S.name as "salesman name" , S.city as "salesman
city" , S.commission
FROM ( (customer C INNER JOIN orders O ON C.customer_id =
O.customer_id )
INNER JOIN salesman S ON S.salesman_id = O.salesman_id)
```

```
SELECT O.* , C.cust_name as "customer name" , C.city as "customer city" , C.grade , S.name as "salesman name" , S.city as "salesman city" , S.commission
FROM ( (customer C INNER JOIN orders O ON C.customer_id = O.customer_id )
INNER JOIN salesman S ON S.salesman_id = O.salesman_id)
```

100 %

Results | Messages

	ord_no	purch_amt	ord_date	customer_id	salesman_id	customer name	customer city	grade	salesman name	salesman city	commission
1	7001	150.50	2012-10-05	3003	6002	Garima	California	200	Neel	Paris	13
2	7002	270.65	2012-09-10	3008	6003	Brijesha	London	NULL	Pritesh	London	11
3	7003	65.26	2012-10-05	3001	6001	Niraj	New York	100	Jay	New York	15
4	7004	110.50	2012-08-17	3006	6006	Krish	Berlin	100	Leela	San Fransisco	12
5	7005	948.50	2012-09-10	3003	6002	Garima	California	200	Neel	Paris	13
6	7006	2400.60	2012-07-27	3002	6001	Brijesh	New York	200	Jay	New York	15
7	7007	5760.00	2012-09-10	3001	6001	Niraj	New York	100	Jay	New York	15
8	7008	1983.43	2012-10-10	3005	6004	Feni	Paris	300	Love	Paris	14
9	7009	2480.40	2012-10-10	3006	6006	Krish	Berlin	100	Leela	San Fransisco	12
10	7010	250.45	2012-06-27	3004	6002	Jesica	London	300	Neel	Paris	13
11	7011	75.29	2012-08-17	3007	6005	Jolly	Moscow	200	Paras	Rome	13
12	7012	3045.60	2012-04-25	3001	6001	Niraj	New York	100	Jay	New York	15

SQL query to display the customer name, customer city, grade, salesman, salesman city. The results should be sorted by ascending customer_id.

QUERY:8

```
SELECT C.cust_name,C.city AS [Customer City],C.grade,S.name AS Salesman , S.city AS [Salesman City]
FROM customer C INNER JOIN salesman S ON C.salesman_id = S.salesman_id
ORDER BY C.customer_id ASC
```

```
SELECT C.cust_name,C.city AS [Customer City],C.grade,S.name AS Salesman , S.city AS [Salesman City]
FROM customer C INNER JOIN salesman S ON C.salesman_id = S.salesman_id
ORDER BY C.customer_id ASC
```

100 %

Results | Messages

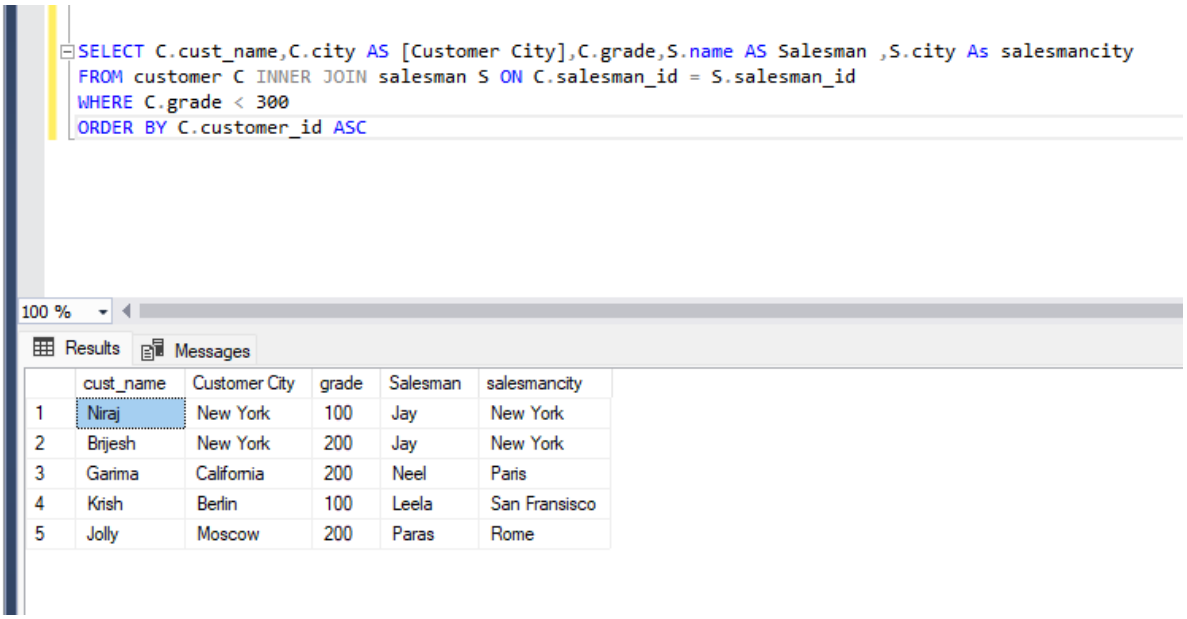
	cust_name	Customer City	grade	Salesman	Salesman City
1	Niraj	New York	100	Jay	New York
2	Brijesh	New York	200	Jay	New York
3	Garima	California	200	Neel	Paris
4	Jesica	London	300	Neel	Paris
5	Feni	Paris	300	Love	Paris
6	Krish	Berlin	100	Leela	San Fransisco
7	Jolly	Moscow	200	Paras	Rome
8	Brijesha	London	NULL	Pritesh	London

- **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.

QUERY:9

```
SELECT C.cust_name,C.city AS [Customer City],C.grade,S.name AS
Salesman ,S.city As salesmancity
FROM customer C INNER JOIN salesman S ON C.salesman_id =
S.salesman_id
WHERE C.grade < 300
ORDER BY C.customer_id ASC
```



	cust_name	Customer City	grade	Salesman	salesmancity
1	Niraj	New York	100	Jay	New York
2	Brijesh	New York	200	Jay	New York
3	Garima	California	200	Neel	Paris
4	Krish	Berlin	100	Leela	San Fransisco
5	Jolly	Moscow	200	Paras	Rome

- **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**

QUERY:10

```
SELECT C.cust_name AS [customer name],C.city AS [customer city] ,
O.ord_date AS "order date" , O.purch_amt AS [order amount]
FROM customer C LEFT OUTER JOIN orders O ON C.customer_id =
O.customer_id
ORDER BY O.ord_date
```

```
SELECT C.cust_name AS [customer name],C.city AS [customer city] , O.ord_date AS "order date" , O.purch_amt AS [order amount]
FROM customer C LEFT OUTER JOIN orders O ON C.customer_id = O.customer_id
ORDER BY O.ord_date
```

100 %

Results | Messages

	customer name	customer city	order date	order amount
1	Niraj	New York	2012-04-25	3045.60
2	Jesica	London	2012-06-27	250.45
3	Brijesh	New York	2012-07-27	2400.60
4	Krish	Berlin	2012-08-17	110.50
5	Jolly	Moscow	2012-08-17	75.29
6	Brijesha	London	2012-09-10	270.65
7	Garima	California	2012-09-10	948.50
8	Niraj	New York	2012-09-10	5760.00
9	Niraj	New York	2012-10-05	65.26
10	Garima	California	2012-10-05	150.50
11	Krish	Berlin	2012-10-10	2480.40
12	Feni	Paris	2012-10-10	1983.43

- **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**

QUERY:11

```
SELECT C.cust_name AS [customer name] , C.city,O.ord_no AS "order number", O.ord_date,S.name AS "salesperson name", S.commission
FROM ((customer C LEFT OUTER JOIN orders O ON C.customer_id = O.customer_id )
LEFT OUTER JOIN salesman S ON C.salesman_id=S.salesman_id )
```

```
SELECT C.cust_name AS [customer name] , C.city,O.ord_no AS "order number", O.ord_date,S.name AS "salesperson name", S.commission
FROM ((customer C LEFT OUTER JOIN orders O ON C.customer_id = O.customer_id )
LEFT OUTER JOIN salesman S ON C.salesman_id=S.salesman_id )
```

100 %

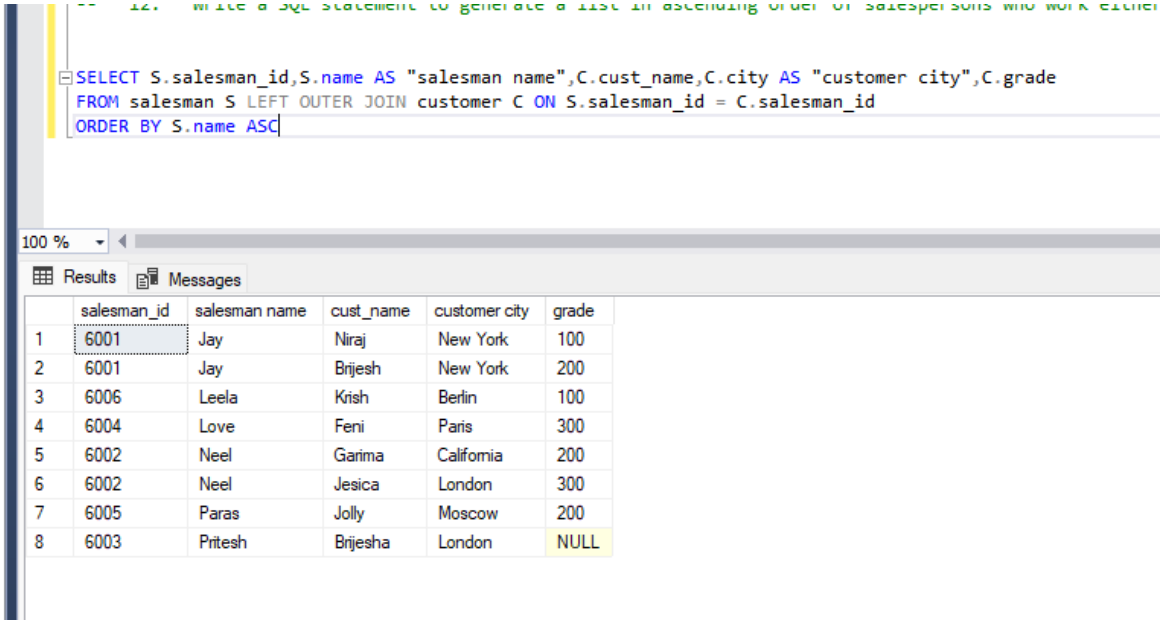
Results | Messages

	customer name	city	order number	ord_date	salesperson name	commission
1	Niraj	New York	7003	2012-10-05	Jay	15
2	Niraj	New York	7007	2012-09-10	Jay	15
3	Niraj	New York	7012	2012-04-25	Jay	15
4	Brijesh	New York	7006	2012-07-27	Jay	15
5	Garima	California	7001	2012-10-05	Neel	13
6	Garima	California	7005	2012-09-10	Neel	13
7	Jesica	London	7010	2012-06-27	Neel	13
8	Feni	Paris	7008	2012-10-10	Love	14
9	Krish	Berlin	7004	2012-08-17	Leela	12
10	Krish	Berlin	7009	2012-10-10	Leela	12
11	Jolly	Moscow	7011	2012-08-17	Paras	13
12	Brijesha	London	7002	2012-09-10	Pritesh	11

- **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**

QUERY:12

```
SELECT S.salesman_id,S.name AS "salesman name",C.cust_name,C.city
AS "customer city",C.grade
FROM salesman S LEFT OUTER JOIN customer C
ON S.salesman_id = C.salesman_id
ORDER BY S.name ASC
```



```
SELECT S.salesman_id,S.name AS "salesman name",C.cust_name,C.city AS "customer city",C.grade
FROM salesman S LEFT OUTER JOIN customer C ON S.salesman_id = C.salesman_id
ORDER BY S.name ASC
```

100 %

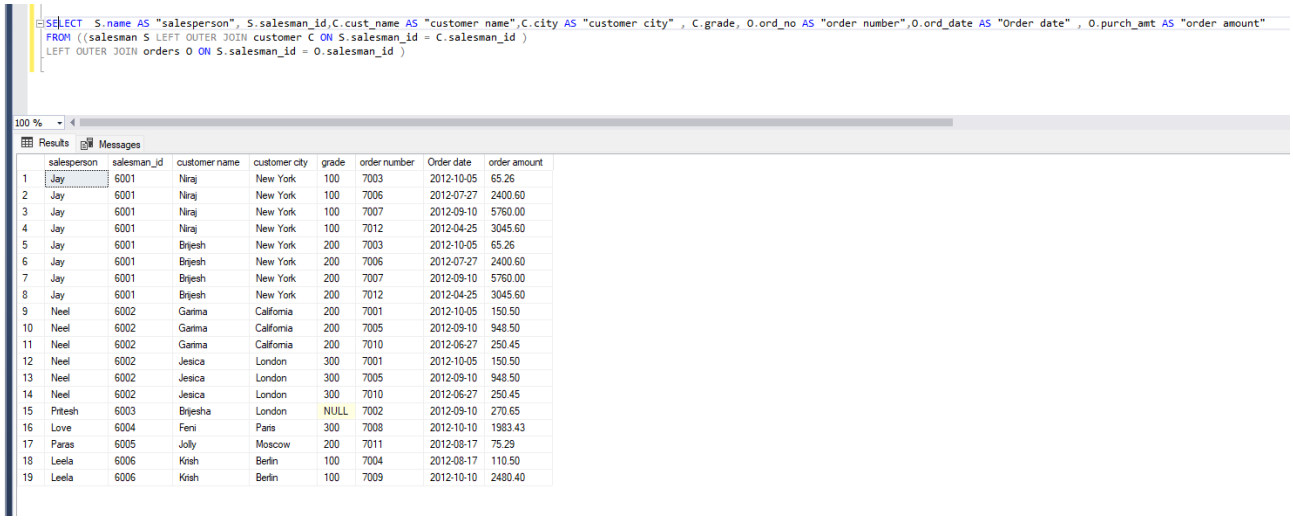
Results | Messages

	salesman_id	salesman name	cust_name	customer city	grade
1	6001	Jay	Niraj	New York	100
2	6001	Jay	Brijesh	New York	200
3	6006	Leela	Krish	Berlin	100
4	6004	Love	Feni	Paris	300
5	6002	Neel	Garima	California	200
6	6002	Neel	Jesica	London	300
7	6005	Paras	Jolly	Moscow	200
8	6003	Pritesh	Brijesha	London	NULL

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

QUERY:13

```
SELECT  S.name AS "salesperson", S.salesman_id,C.cust_name AS
"customer name",C.city AS "customer city" , C.grade, O.ord_no AS
"order number",O.ord_date AS "Order date" , O.purch_amt AS "order
amount"
FROM ((salesman S LEFT OUTER JOIN customer C ON S.salesman_id =
C.salesman_id )
LEFT OUTER JOIN orders O ON S.salesman_id = O.salesman_id )
```



```
SELECT  S.name AS "salesperson", S.salesman_id,C.cust_name AS "customer name",C.city AS "customer city" , C.grade, O.ord_no AS "order number",O.ord_date AS "Order date" , O.purch_amt AS "order amount"
FROM ((salesman S LEFT OUTER JOIN customer C ON S.salesman_id = C.salesman_id )
LEFT OUTER JOIN orders O ON S.salesman_id = O.salesman_id )
```

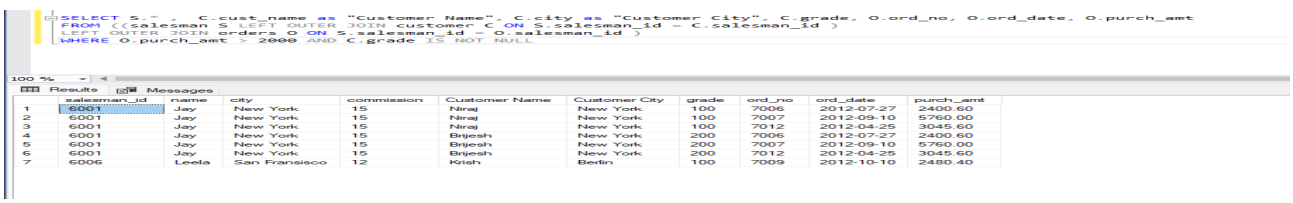
100 % — Results | Messages

	salesperson	salesman_id	customer name	customer city	grade	order number	Order date	order amount
1	Jay	6001	Niraj	New York	100	7003	2012-10-05	65.26
2	Jay	6001	Niraj	New York	100	7006	2012-07-27	2400.60
3	Jay	6001	Niraj	New York	100	7007	2012-09-10	5760.00
4	Jay	6001	Niraj	New York	100	7012	2012-04-25	3045.60
5	Jay	6001	Brijesh	New York	200	7003	2012-10-05	65.26
6	Jay	6001	Brijesh	New York	200	7006	2012-07-27	2400.60
7	Jay	6001	Brijesh	New York	200	7007	2012-09-10	5760.00
8	Jay	6001	Brijesh	New York	200	7012	2012-04-25	3045.60
9	Neel	6002	Garima	California	200	7001	2012-10-05	150.50
10	Neel	6002	Garima	California	200	7005	2012-09-10	948.50
11	Neel	6002	Garima	California	200	7010	2012-06-27	250.45
12	Neel	6002	Jesica	London	300	7001	2012-10-05	150.50
13	Neel	6002	Jesica	London	300	7005	2012-09-10	948.50
14	Neel	6002	Jesica	London	300	7010	2012-06-27	250.45
15	Pritesh	6003	Brijesha	London	NULL	7002	2012-09-10	270.65
16	Love	6004	Feni	Paris	300	7008	2012-10-10	1983.43
17	Paras	6005	Jolly	Moscow	200	7011	2012-08-17	75.29
18	Leela	6006	Krish	Berlin	100	7004	2012-08-17	110.50
19	Leela	6006	Krish	Berlin	100	7009	2012-10-10	2480.40

- **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.**

QUERY:14

```
SELECT S.* ,  C.cust_name as "Customer Name", C.city as "Customer City", C.grade, O.ord_no, O.ord_date, O.purch_amt
FROM ((salesman S LEFT OUTER JOIN customer C ON S.salesman_id = C.salesman_id )
LEFT OUTER JOIN orders O ON S.salesman_id = O.salesman_id )
WHERE O.purch_amt > 2000 AND C.grade IS NOT NULL
```



```
SELECT S.* ,  C.cust_name as "Customer Name", C.city as "Customer City", C.grade, O.ord_no, O.ord_date, O.purch_amt
FROM ((salesman S LEFT OUTER JOIN customer C ON S.salesman_id = C.salesman_id )
LEFT OUTER JOIN orders O ON S.salesman_id = O.salesman_id )
WHERE O.purch_amt > 2000 AND C.grade IS NOT NULL
```

100 % — Results | Messages

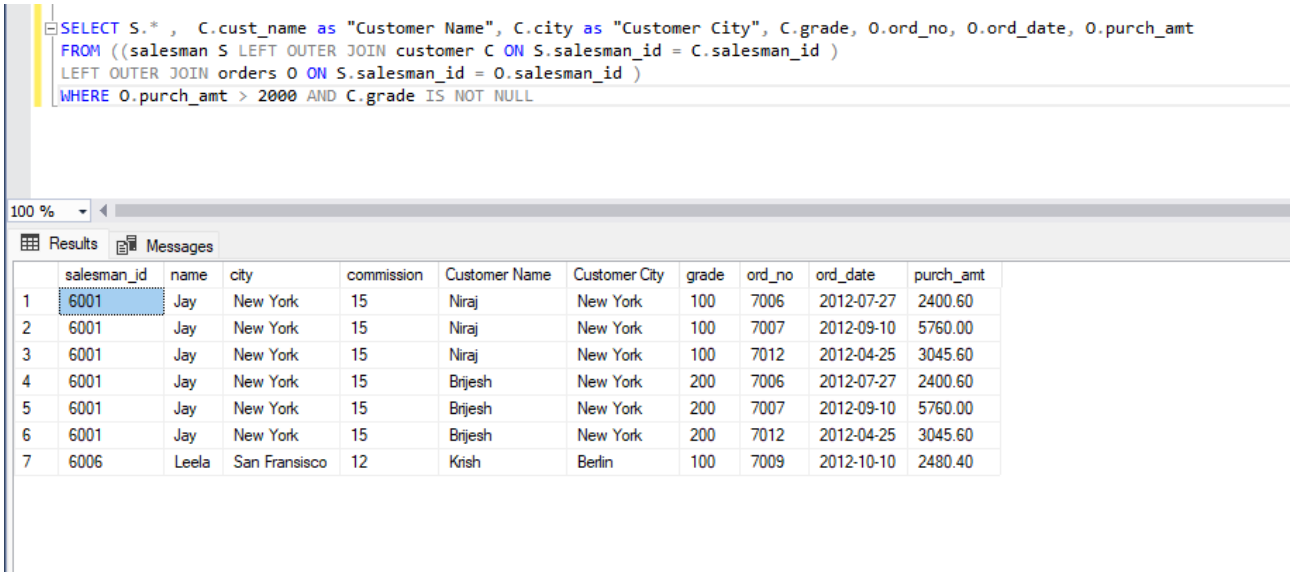
	salesman_id	name	city	commission	Customer Name	Customer City	grade	ord_no	ord_date	purch_amt
1	6001	Jay	New York	15	Niraj	New York	100	7006	2012-07-27	2400.60
2	6001	Jay	New York	15	Niraj	New York	100	7007	2012-09-10	5760.00
3	6001	Jay	New York	15	Niraj	New York	100	7012	2012-04-25	3045.60
4	6001	Jay	New York	15	Brijesh	New York	200	7006	2012-07-27	2400.60
5	6001	Jay	New York	15	Brijesh	New York	200	7007	2012-09-10	5760.00
6	6001	Jay	New York	15	Brijesh	New York	200	7012	2012-04-25	3045.60
7	6006	Leela	San Fransisco	12	Krish	Berlin	100	7009	2012-10-10	2480.40

- **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.**

DEV VIPULKUMAR PATEL

QUERY:15

```
SELECT S.* ,  C.cust_name as "Customer Name", C.city as "Customer
City", C.grade, O.ord_no, O.ord_date, O.purch_amt
FROM ((salesman S LEFT OUTER JOIN customer C ON S.salesman_id =
C.salesman_id )
LEFT OUTER JOIN orders O ON S.salesman_id = O.salesman_id )
WHERE O.purch_amt > 2000 AND C.grade IS NOT NULL
```



```
SELECT S.* ,  C.cust_name as "Customer Name", C.city as "Customer City", C.grade, O.ord_no, O.ord_date, O.purch_amt
FROM ((salesman S LEFT OUTER JOIN customer C ON S.salesman_id = C.salesman_id )
LEFT OUTER JOIN orders O ON S.salesman_id = O.salesman_id )
WHERE O.purch_amt > 2000 AND C.grade IS NOT NULL
```

100 %

Results | Messages

	salesman_id	name	city	commission	Customer Name	Customer City	grade	ord_no	ord_date	purch_amt
1	6001	Jay	New York	15	Niraj	New York	100	7006	2012-07-27	2400.60
2	6001	Jay	New York	15	Niraj	New York	100	7007	2012-09-10	5760.00
3	6001	Jay	New York	15	Niraj	New York	100	7012	2012-04-25	3045.60
4	6001	Jay	New York	15	Brijesh	New York	200	7006	2012-07-27	2400.60
5	6001	Jay	New York	15	Brijesh	New York	200	7007	2012-09-10	5760.00
6	6001	Jay	New York	15	Brijesh	New York	200	7012	2012-04-25	3045.60
7	6006	Leela	San Fransisco	12	Krish	Berlin	100	7009	2012-10-10	2480.40

- **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.**

QUERY:16

```
SELECT C.cust_name AS "customer name", C.city, O.ord_no,
O.ord_date,O.purch_amt
FROM customer C FULL OUTER JOIN orders O ON C.customer_id =
O.customer_id
WHERE C.grade IS NOT NULL
```

```
SELECT C.cust_name AS "customer name", C.city, O.ord_no, O.ord_date,O.purch_amt
FROM customer C FULL OUTER JOIN orders O ON C.customer_id = O.customer_id
WHERE C.grade IS NOT NULL
```

100 %

Results | Messages

	customer name	city	ord_no	ord_date	purch_amt
1	Niraj	New York	7003	2012-10-05	65.26
2	Niraj	New York	7007	2012-09-10	5760.00
3	Niraj	New York	7012	2012-04-25	3045.60
4	Brijesh	New York	7006	2012-07-27	2400.60
5	Garima	California	7001	2012-10-05	150.50
6	Garima	California	7005	2012-09-10	948.50
7	Jesica	London	7010	2012-06-27	250.45
8	Feni	Paris	7008	2012-10-10	1983.43
9	Krish	Berlin	7004	2012-08-17	110.50
10	Krish	Berlin	7009	2012-10-10	2480.40
11	Jolly	Moscow	7011	2012-08-17	75.29

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

QUERY:17

```
SELECT S.salesman_id , S.city AS "salesman city", S.name AS
"salesman name" , S.commission , C.cust_name AS "customer name" ,
C.city AS "customer city" , C.grade , C.customer_id
FROM salesman S CROSS JOIN customer C
```

DEV VIPULKUMAR PATEL

```
SELECT S.salesman_id , S.city AS "salesman city", S.name AS "salesman name" , S.commission , C.cust_name AS "customer name" , C.city AS "customer city" , C.grade , C.customer_id
FROM salesman S CROSS JOIN customer C
```

100 %

Results | Messages

	salesman_id	salesman city	salesman name	commission	customer name	customer city	grade	customer_id
1	6001	New York	Jay	15	Niraj	New York	100	3001
2	6001	New York	Jay	15	Brijesh	New York	200	3002
3	6001	New York	Jay	15	Garima	California	200	3003
4	6001	New York	Jay	15	Jesica	London	300	3004
5	6001	New York	Jay	15	Feni	Paris	300	3005
6	6001	New York	Jay	15	Krish	Berlin	100	3006
7	6001	New York	Jay	15	Jolly	Moscow	200	3007
8	6001	New York	Jay	15	Brijesha	London	NULL	3008
9	6002	Paris	Neel	13	Niraj	New York	100	3001
10	6002	Paris	Neel	13	Brijesh	New York	200	3002
11	6002	Paris	Neel	13	Garima	California	200	3003
12	6002	Paris	Neel	13	Jesica	London	300	3004
13	6002	Paris	Neel	13	Feni	Paris	300	3005
14	6002	Paris	Neel	13	Krish	Berlin	100	3006
15	6002	Paris	Neel	13	Jolly	Moscow	200	3007
16	6002	Paris	Neel	13	Brijesha	London	NULL	3008
17	6003	London	Pritesh	11	Niraj	New York	100	3001
18	6003	London	Pritesh	11	Brijesh	New York	200	3002
19	6003	London	Pritesh	11	Garima	California	200	3003
20	6003	London	Pritesh	11	Jesica	London	300	3004
21	6003	London	Pritesh	11	Feni	Paris	300	3005
22	6003	London	Pritesh	11	Krish	Berlin	100	3006
23	6003	London	Pritesh	11	Jolly	Moscow	200	3007
24	6003	London	Pritesh	11	Brijesha	London	NULL	3008
25	6004	Paris	Love	14	Niraj	New York	100	3001
26	6004	Paris	Love	14	Brijesh	New York	200	3002
27	6004	Paris	Love	14	Garima	California	200	3003
28	6004	Paris	Love	14	Jesica	London	300	3004
29	6004	Paris	Love	14	Feni	Paris	300	3005
30	6004	Paris	Love	14	Krish	Berlin	100	3006
31	6004	Paris	Love	14	Jolly	Moscow	200	3007
32	6004	Paris	Love	14	Brijesha	London	NULL	3008
33	6005	Rome	Paras	13	Niraj	New York	100	3001
34	6005	Rome	Paras	13	Brijesh	New York	200	3002

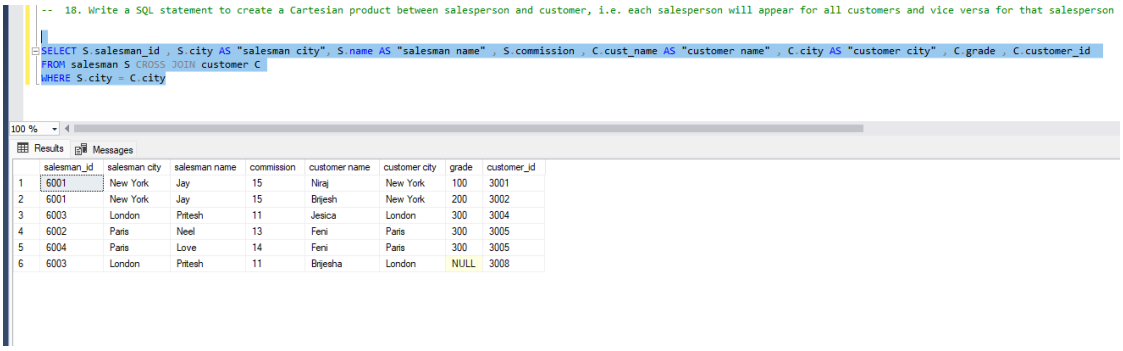
34	6005	Rome	Paras	13	Brijesh	New York	200	3002
35	6005	Rome	Paras	13	Garima	California	200	3003
36	6005	Rome	Paras	13	Jesica	London	300	3004
37	6005	Rome	Paras	13	Feni	Paris	300	3005
38	6005	Rome	Paras	13	Krish	Berlin	100	3006
39	6005	Rome	Paras	13	Jolly	Moscow	200	3007
40	6005	Rome	Paras	13	Brijesha	London	NULL	3008
41	6006	San Fransis...	Leela	12	Niraj	New York	100	3001
42	6006	San Fransis...	Leela	12	Brijesh	New York	200	3002
43	6006	San Fransis...	Leela	12	Garima	California	200	3003
44	6006	San Fransis...	Leela	12	Jesica	London	300	3004
45	6006	San Fransis...	Leela	12	Feni	Paris	300	3005
46	6006	San Fransis...	Leela	12	Krish	Berlin	100	3006
47	6006	San Fransis...	Leela	12	Jolly	Moscow	200	3007
48	6006	San Fransis...	Leela	12	Brijesha	London	NULL	3008

DEV VIPULKUMAR PATEL

- **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**

QUERY:18

```
SELECT S.salesman_id , S.city AS "salesman city", S.name AS
"salesman name" , S.commission , C.cust_name AS "customer name" ,
C.city AS "customer city" , C.grade , C.customer_id
FROM salesman S CROSS JOIN customer C
WHERE S.city = C.city
```



```
-- 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

SELECT S.salesman_id , S.city AS "salesman city", S.name AS "salesman name" , S.commission , C.cust_name AS "customer name" , C.city AS "customer city" , C.grade , C.customer_id
FROM salesman S CROSS JOIN customer C
WHERE S.city = C.city
```

100 %

Results | Messages

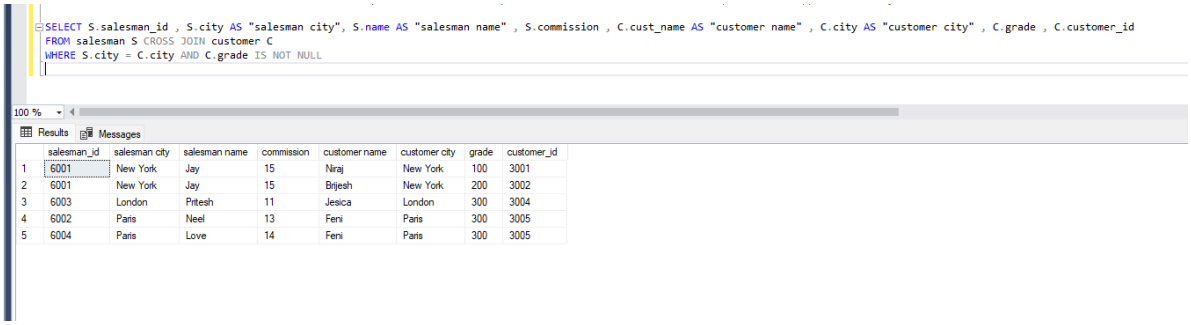
	salesman_id	salesman city	salesman name	commission	customer name	customer city	grade	customer_id
1	6001	New York	Jay	15	Niraj	New York	100	3001
2	6001	New York	Jay	15	Brijesh	New York	200	3002
3	6003	London	Pritesh	11	Jesica	London	300	3004
4	6002	Paris	Neel	13	Feni	Paris	300	3005
5	6004	Paris	Love	14	Feni	Paris	300	3005
6	6003	London	Pritesh	11	Brijesha	London	NULL	3008

- **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**

QUERY:19

```
SELECT S.salesman_id , S.city AS "salesman city", S.name AS
"salesman name" , S.commission , C.cust_name AS "customer name" ,
C.city AS "customer city" , C.grade , C.customer_id
FROM salesman S CROSS JOIN customer C
WHERE S.city = C.city AND C.grade IS NOT NULL
```

DEV VIPULKUMAR PATEL



```
SELECT S.salesman_id , S.city AS "salesman city", S.name AS "salesman name" , S.commission , C.cust_name AS "customer name" , C.city AS "customer city" , C.grade , C.customer_id
FROM salesman S CROSS JOIN customer C
WHERE S.city = C.city AND C.grade IS NOT NULL
```

100 %

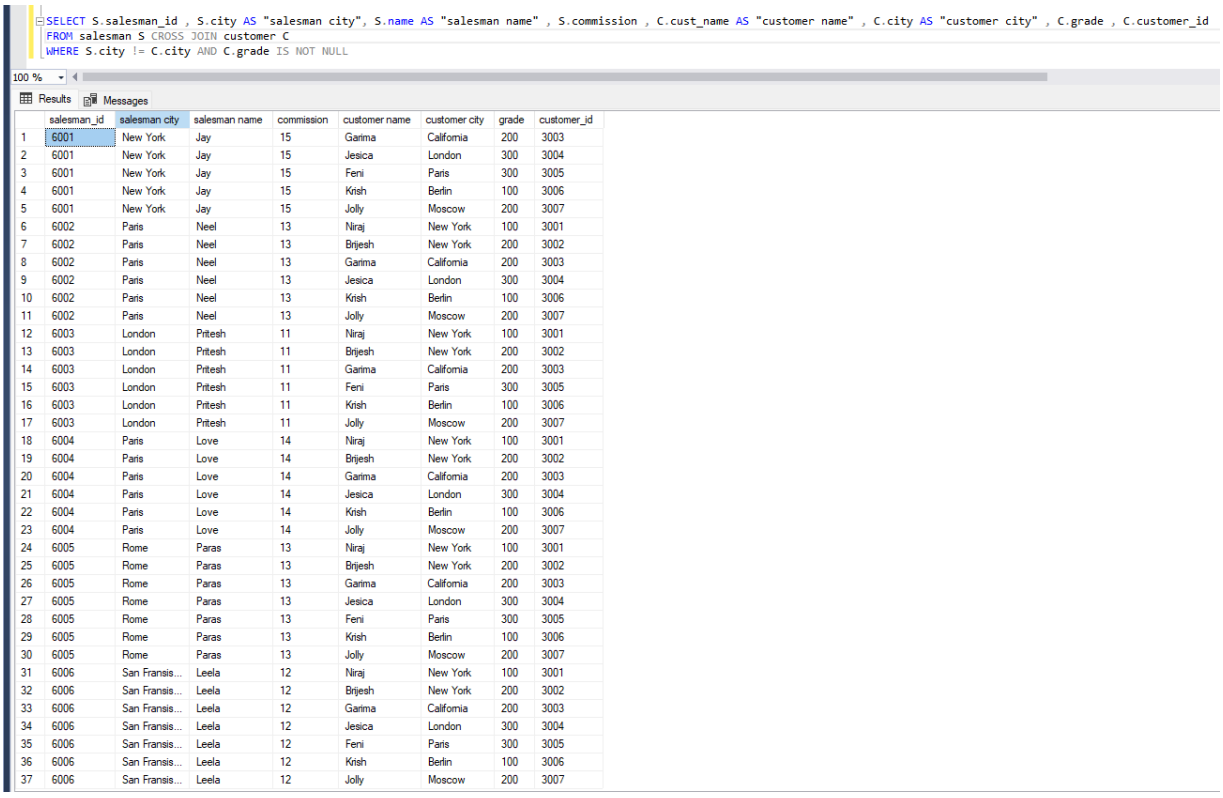
Results | Messages

	salesman_id	salesman city	salesman name	commission	customer name	customer city	grade	customer_id
1	6001	New York	Jay	15	Niraj	New York	100	3001
2	6001	New York	Jay	15	Brijesh	New York	200	3002
3	6003	London	Pritesh	11	Jesica	London	300	3004
4	6002	Paris	Neel	13	Feni	Paris	300	3005
5	6004	Paris	Love	14	Feni	Paris	300	3005

- **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**

QUERY:20

```
SELECT S.salesman_id , S.city AS "salesman city", S.name AS "salesman name" , S.commission , C.cust_name AS "customer name" , C.city AS "customer city" , C.grade , C.customer_id
FROM salesman S CROSS JOIN customer C
WHERE S.city != C.city AND C.grade IS NOT NULL
```



100 %

Results | Messages

	salesman_id	salesman city	salesman name	commission	customer name	customer city	grade	customer_id
1	6001	New York	Jay	15	Garima	California	200	3003
2	6001	New York	Jay	15	Jesica	London	300	3004
3	6001	New York	Jay	15	Feni	Paris	300	3005
4	6001	New York	Jay	15	Krish	Berlin	100	3006
5	6001	New York	Jay	15	Jolly	Moscow	200	3007
6	6002	Paris	Neel	13	Niraj	New York	100	3001
7	6002	Paris	Neel	13	Brijesh	New York	200	3002
8	6002	Paris	Neel	13	Garima	California	200	3003
9	6002	Paris	Neel	13	Jesica	London	300	3004
10	6002	Paris	Neel	13	Krish	Berlin	100	3006
11	6002	Paris	Neel	13	Jolly	Moscow	200	3007
12	6003	London	Pritesh	11	Niraj	New York	100	3001
13	6003	London	Pritesh	11	Brijesh	New York	200	3002
14	6003	London	Pritesh	11	Garima	California	200	3003
15	6003	London	Pritesh	11	Feni	Paris	300	3005
16	6003	London	Pritesh	11	Krish	Berlin	100	3006
17	6003	London	Pritesh	11	Jolly	Moscow	200	3007
18	6004	Paris	Love	14	Niraj	New York	100	3001
19	6004	Paris	Love	14	Brijesh	New York	200	3002
20	6004	Paris	Love	14	Garima	California	200	3003
21	6004	Paris	Love	14	Jesica	London	300	3004
22	6004	Paris	Love	14	Krish	Berlin	100	3006
23	6004	Paris	Love	14	Jolly	Moscow	200	3007
24	6005	Rome	Paras	13	Niraj	New York	100	3001
25	6005	Rome	Paras	13	Brijesh	New York	200	3002
26	6005	Rome	Paras	13	Garima	California	200	3003
27	6005	Rome	Paras	13	Jesica	London	300	3004
28	6005	Rome	Paras	13	Feni	Paris	300	3005
29	6005	Rome	Paras	13	Krish	Berlin	100	3006
30	6005	Rome	Paras	13	Jolly	Moscow	200	3007
31	6006	San Fransis...	Leela	12	Niraj	New York	100	3001
32	6006	San Fransis...	Leela	12	Brijesh	New York	200	3002
33	6006	San Fransis...	Leela	12	Garima	California	200	3003
34	6006	San Fransis...	Leela	12	Jesica	London	300	3004
35	6006	San Fransis...	Leela	12	Feni	Paris	300	3005
36	6006	San Fransis...	Leela	12	Krish	Berlin	100	3006
37	6006	San Fransis...	Leela	12	Jolly	Moscow	200	3007