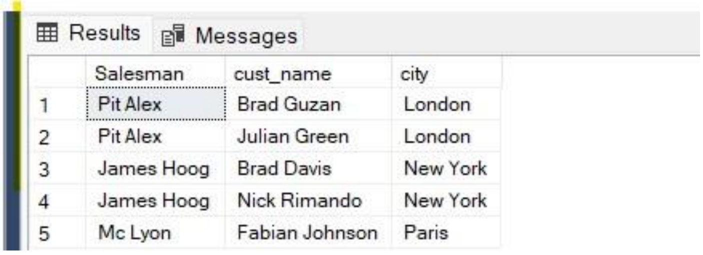
Assignment :- 2

Name : - Avinash Radadiya

SQL Query

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

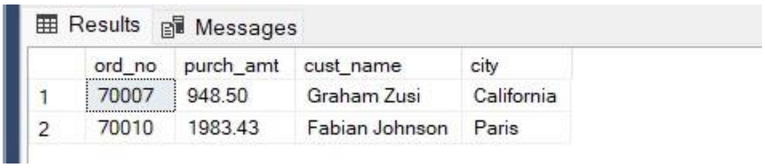
Ans : -
```
SELECT salesman.name AS "Salesman",customer.cust_name, customer.city
FROM salesman, customer
WHERE salesman.city=customer.city
order by city
```



	Salesman	cust_name	city
1	Pit Alex	Brad Guzan	London
2	Pit Alex	Julian Green	London
3	James Hoog	Brad Davis	New York
4	James Hoog	Nick Rimando	New York
5	Mc Lyon	Fabian Johnson	Paris

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

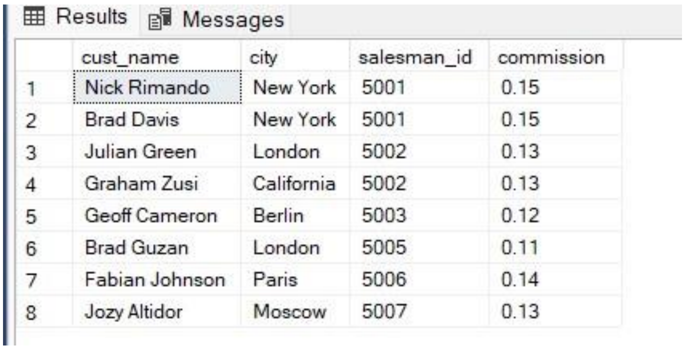
Ans:-
```
SELECT  a.ord_no,a.purch_amt,b.cust_name,b.city
FROM orders a,customer b
WHERE a.customer_id=b.customer_id AND a.purch_amt BETWEEN 500 AND 2000;
```



	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

Ans :-
```
SELECT a.cust_name,a.city,b.salesman_id,b.commission
FROM customer a,salesman b
WHERE a.salesman_id=b.salesman_id
ORDER BY a.salesman_id
```

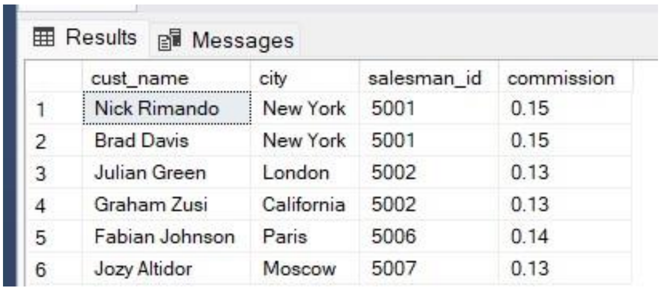


	cust_name	city	salesman_id	commission
1	Nick Rimando	New York	5001	0.15
2	Brad Davis	New York	5001	0.15
3	Julian Green	London	5002	0.13
4	Graham Zusi	California	5002	0.13
5	Geoff Cameron	Berlin	5003	0.12
6	Brad Guzan	London	5005	0.11
7	Fabian Johnson	Paris	5006	0.14
8	Jozy Altidor	Moscow	5007	0.13

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 a.cust_name,a.city,b.salesman_id,b.commission
FROM customer a,salesman b
WHERE a.salesman_id=b.salesman_id AND b.commission > 0.12
ORDER BY a.salesman_id
```

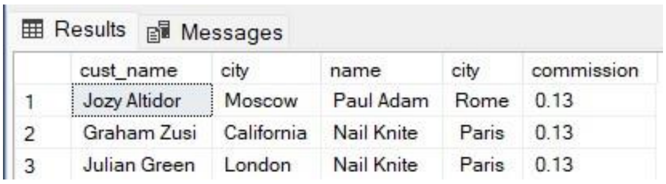


	cust_name	city	salesman_id	commission
1	Nick Rimando	New York	5001	0.15
2	Brad Davis	New York	5001	0.15
3	Julian Green	London	5002	0.13
4	Graham Zusi	California	5002	0.13
5	Fabian Johnson	Paris	5006	0.14
6	Jozy Altidor	Moscow	5007	0.13

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

Ans:-

```
SELECT a.cust_name, a.city, b.name, b.city,b.commission
FROM customer a
INNER JOIN salesman b
ON a.salesman_id=b.salesman_id
WHERE b.commission>.12
AND a.city<>b.city;
```

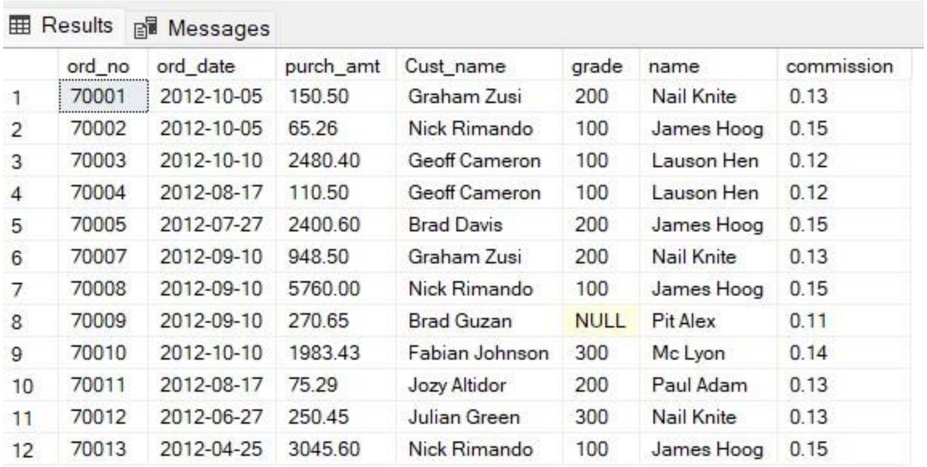


	cust_name	city	name	city	commission
1	Jozy Altidor	Moscow	Paul Adam	Rome	0.13
2	Graham Zusi	California	Nail Knite	Paris	0.13
3	Julian Green	London	Nail Knite	Paris	0.13

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

Ans:-

```
SELECT a.ord_no,a.ord_date,a.purch_amt,b.Cust_name,b.grade,c.name,c.commission
FROM orders a
INNER JOIN customer b
ON a.customer_id=b.customer_id
INNER JOIN salesman c
ON a.salesman_id=c.salesman_id
```



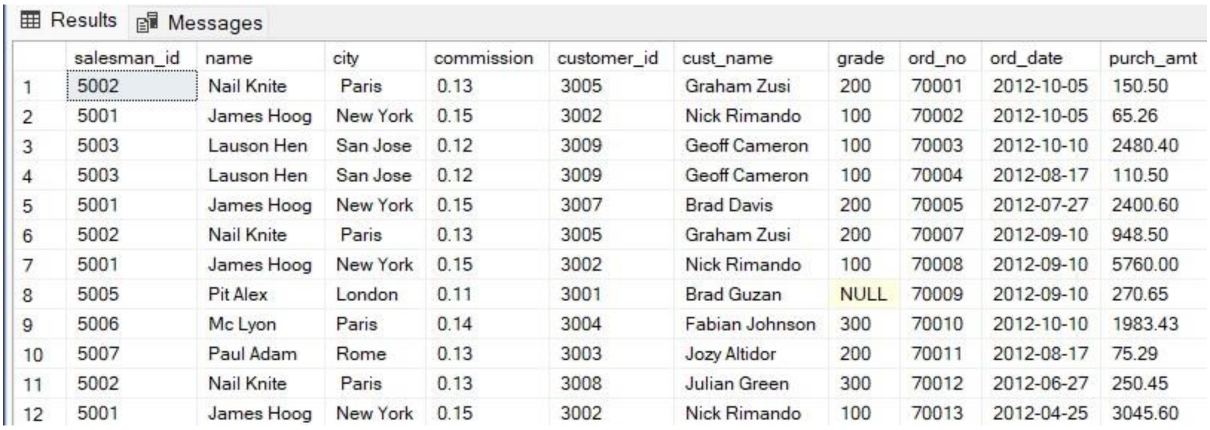
Results | Messages

	ord_no	ord_date	purch_amt	Cust_name	grade	name	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	NULL	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.

Ans:-

```
SELECT a.salesman_id,a.name,a.city,a.commission,b.customer_id,b.cust_name,
.grade,c.ord_no,c.ord_date,c.purch_amt
FROM salesman a
INNER JOIN customer b
ON a.salesman_id=b.salesman_id
INNER JOIN orders c
ON b.customer_id=c.customer_id
```



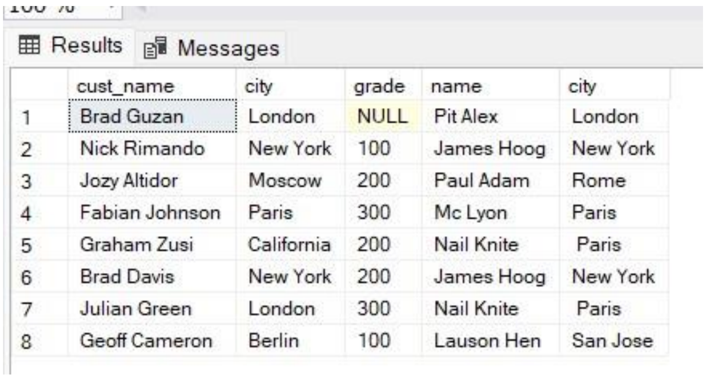
Results | Messages

	salesman_id	name	city	commission	customer_id	cust_name	grade	ord_no	ord_date	purch_amt
1	5002	Nail Knite	Paris	0.13	3005	Graham Zusi	200	70001	2012-10-05	150.50
2	5001	James Hoog	New York	0.15	3002	Nick Rimando	100	70002	2012-10-05	65.26
3	5003	Lauson Hen	San Jose	0.12	3009	Geoff Cameron	100	70003	2012-10-10	2480.40
4	5003	Lauson Hen	San Jose	0.12	3009	Geoff Cameron	100	70004	2012-08-17	110.50
5	5001	James Hoog	New York	0.15	3007	Brad Davis	200	70005	2012-07-27	2400.60
6	5002	Nail Knite	Paris	0.13	3005	Graham Zusi	200	70007	2012-09-10	948.50
7	5001	James Hoog	New York	0.15	3002	Nick Rimando	100	70008	2012-09-10	5760.00
8	5005	Pit Alex	London	0.11	3001	Brad Guzan	NULL	70009	2012-09-10	270.65
9	5006	Mc Lyon	Paris	0.14	3004	Fabian Johnson	300	70010	2012-10-10	1983.43
10	5007	Paul Adam	Rome	0.13	3003	Jozy Altidor	200	70011	2012-08-17	75.29
11	5002	Nail Knite	Paris	0.13	3008	Julian Green	300	70012	2012-06-27	250.45
12	5001	James Hoog	New York	0.15	3002	Nick Rimando	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.

Ans:-

```
SELECT a.cust_name,a.city,a.grade, b.name ,b.city
FROM customer a,salesman b
WHERE a.salesman_id=b.salesman_id
order by a.customer_id
```

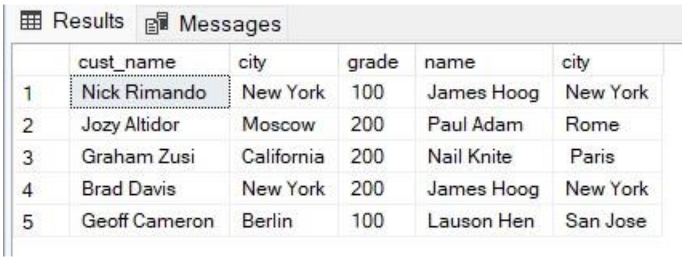


	cust_name	city	grade	name	city
1	Brad Guzan	London	NULL	Pit Alex	London
2	Nick Rimando	New York	100	James Hoog	New York
3	Jozy Altidor	Moscow	200	Paul Adam	Rome
4	Fabian Johnson	Paris	300	Mc Lyon	Paris
5	Graham Zusi	California	200	Nail Knite	Paris
6	Brad Davis	New York	200	James Hoog	New York
7	Julian 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 orered by ascending customer_id.

Ans:-

```
SELECT a.cust_name,a.city,a.grade, b.name , b.city
FROM customer a,salesman b
WHERE a.salesman_id=b.salesman_id AND a.grade<300
ORDER BY a.customer_id;
```

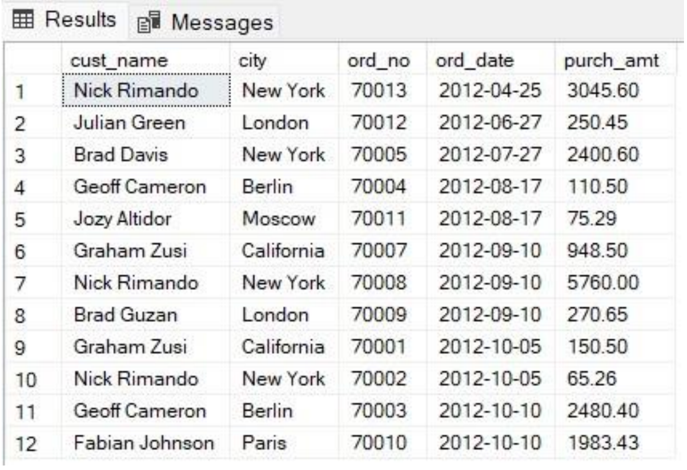


	cust_name	city	grade	name	city
1	Nick Rimando	New York	100	James Hoog	New York
2	Jozy Altidor	Moscow	200	Paul Adam	Rome
3	Graham Zusi	California	200	Nail Knite	Paris
4	Brad Davis	New York	200	James Hoog	New York
5	Geoff Cameron	Berlin	100	Lauson Hen	San Jose

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

Ans:-

```
SELECT a.cust_name,a.city, b.ord_no,b.ord_date,b.purch_amt
FROM customer a,orders b
WHERE a.customer_id=b.customer_id
order by b.ord_date;
```

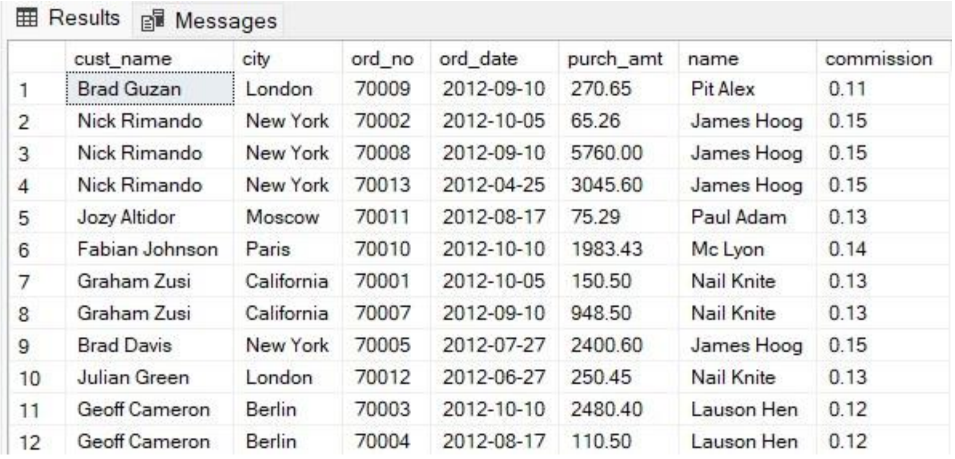


	cust_name	city	ord_no	ord_date	purch_amt
1	Nick Rimando	New York	70013	2012-04-25	3045.60
2	Julian Green	London	70012	2012-06-27	250.45
3	Brad Davis	New York	70005	2012-07-27	2400.60
4	Geoff Cameron	Berlin	70004	2012-08-17	110.50
5	Jozy Altidor	Moscow	70011	2012-08-17	75.29
6	Graham Zusi	California	70007	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

Ans:-

```
SELECT a.cust_name,a.city, b.ord_no,b.ord_date,b.purch_amt,
c.name,c.commission
FROM customer a
LEFT OUTER JOIN orders b
ON a.customer_id=b.customer_id
LEFT OUTER JOIN salesman c
ON c.salesman_id=b.salesman_id;
```

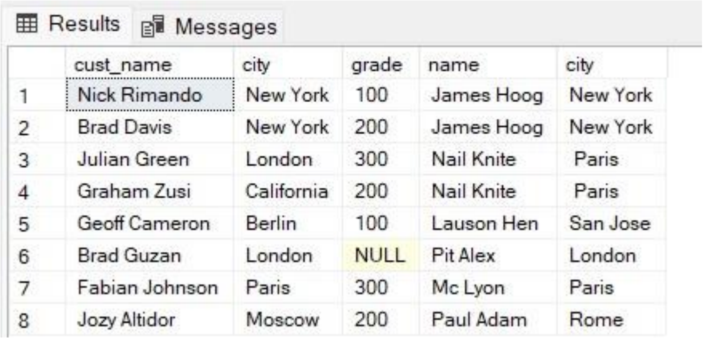


	cust_name	city	ord_no	ord_date	purch_amt	name	commission
1	Brad Guzan	London	70009	2012-09-10	270.65	Pit Alex	0.11
2	Nick Rimando	New York	70002	2012-10-05	65.26	James Hoog	0.15
3	Nick Rimando	New York	70008	2012-09-10	5760.00	James Hoog	0.15
4	Nick Rimando	New York	70013	2012-04-25	3045.60	James Hoog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

Ans:-

```
SELECT a.cust_name,a.city,a.grade, b.name, b.city
FROM customer a,salesman b
WHERE b.salesman_id=a.salesman_id
ORDER BY b.salesman_id;
```

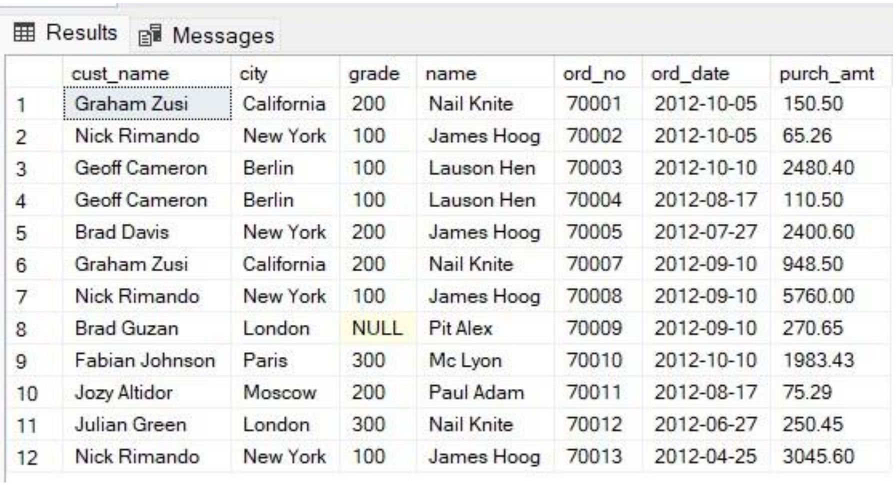


	cust_name	city	grade	name	city
1	Nick Rimando	New York	100	James Hoog	New York
2	Brad Davis	New York	200	James Hoog	New York
3	Julian Green	London	300	Nail Knite	Paris
4	Graham Zusi	California	200	Nail Knite	Paris
5	Geoff Cameron	Berlin	100	Lauson Hen	San Jose
6	Brad Guzan	London	NULL	Pit Alex	London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

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

Ans:-

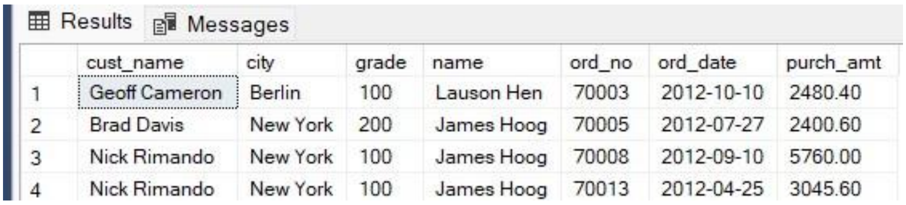
```
SELECT a.cust_name,a.city,a.grade, b.name, c.ord_no, c.ord_date, c.purch_amt
 FROM customer a ,salesman b ,orders c
 WHERE b.salesman_id=a.salesman_id AND c.customer_id=a.customer_id
```



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

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.

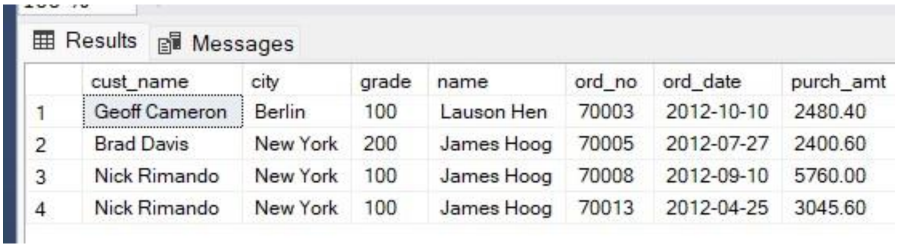
Ans:-
```
SELECT a.cust_name,a.city,a.grade, b.name, c.ord_no, c.ord_date, c.purch_amt
FROM customer a ,salesman b ,orders c
WHERE  b.salesman_id=a.salesman_id AND c.customer_id=a.customer_id AND
c.purch_amt>=2000 AND a.grade IS NOT NULL
```



	cust_name	city	grade	name	ord_no	ord_date	purch_amt
1	Geoff Cameron	Berlin	100	Lauson Hen	70003	2012-10-10	2480.40
2	Brad Davis	New York	200	James Hoog	70005	2012-07-27	2400.60
3	Nick Rimando	New York	100	James Hoog	70008	2012-09-10	5760.00
4	Nick Rimando	New York	100	James Hoog	70013	2012-04-25	3045.60

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.

Ans:-
```
SELECT a.cust_name,a.city,a.grade, b.name, c.ord_no, c.ord_date, c.purch_amt
FROM customer a ,salesman b ,orders c
WHERE  b.salesman_id=a.salesman_id AND c.customer_id=a.customer_id AND
c.purch_amt>=2000 AND a.grade IS NOT NULL
```

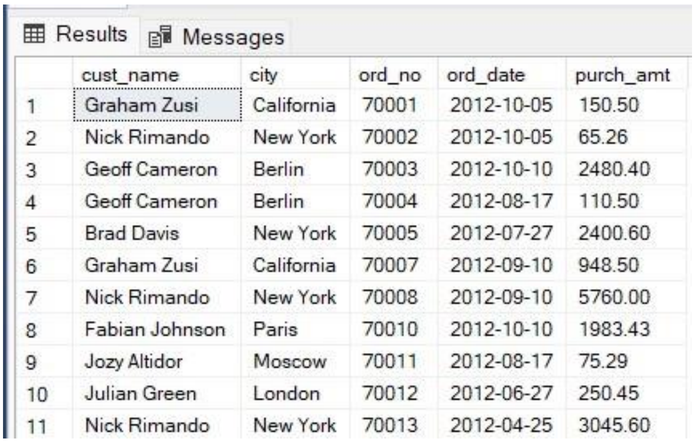


	cust_name	city	grade	name	ord_no	ord_date	purch_amt
1	Geoff Cameron	Berlin	100	Lauson Hen	70003	2012-10-10	2480.40
2	Brad Davis	New York	200	James Hoog	70005	2012-07-27	2400.60
3	Nick Rimando	New York	100	James Hoog	70008	2012-09-10	5760.00
4	Nick Rimando	New York	100	James Hoog	70013	2012-04-25	3045.60

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.

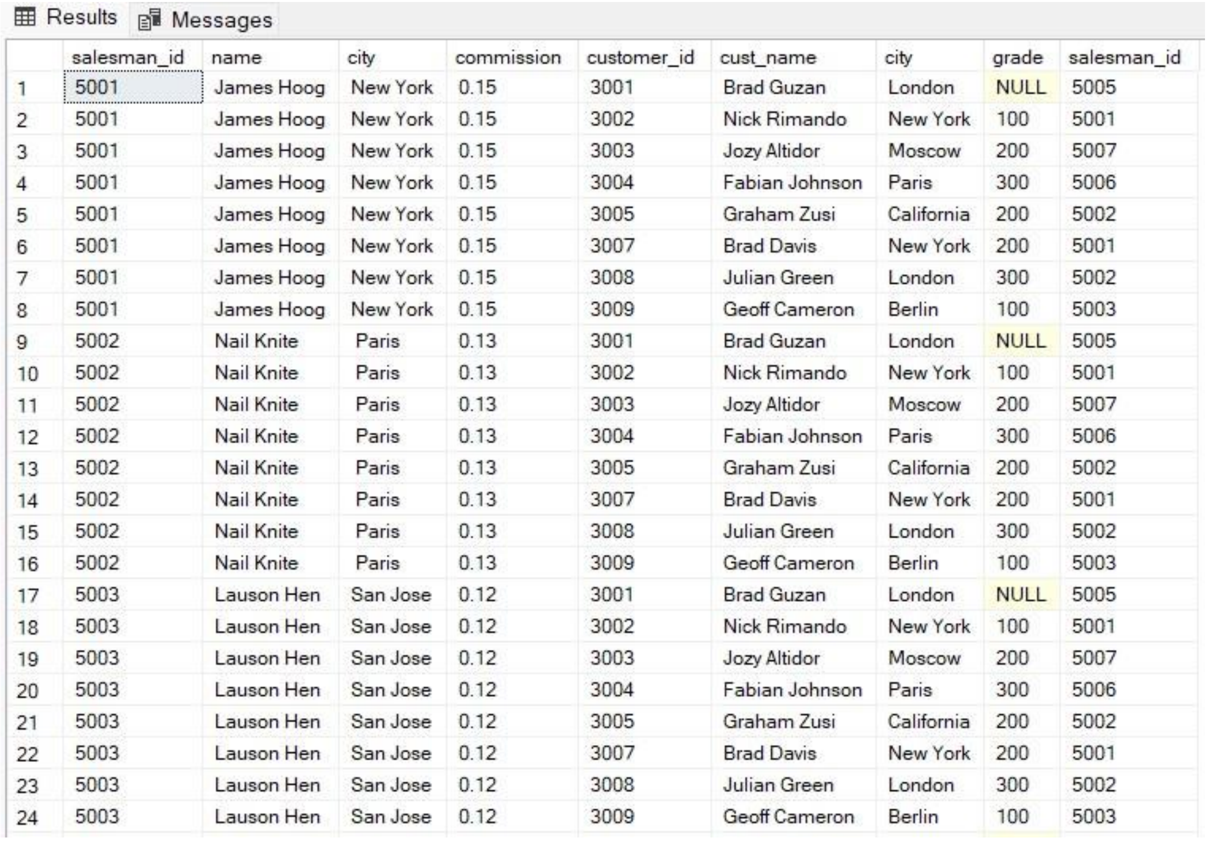
Ans:-
```
SELECT a.cust_name,a.city, b.ord_no,b.ord_date,b.purch_amt
 FROM customer a ,orders b
 WHERE a.customer_id=b.customer_id AND a.grade IS NOT NULL
```



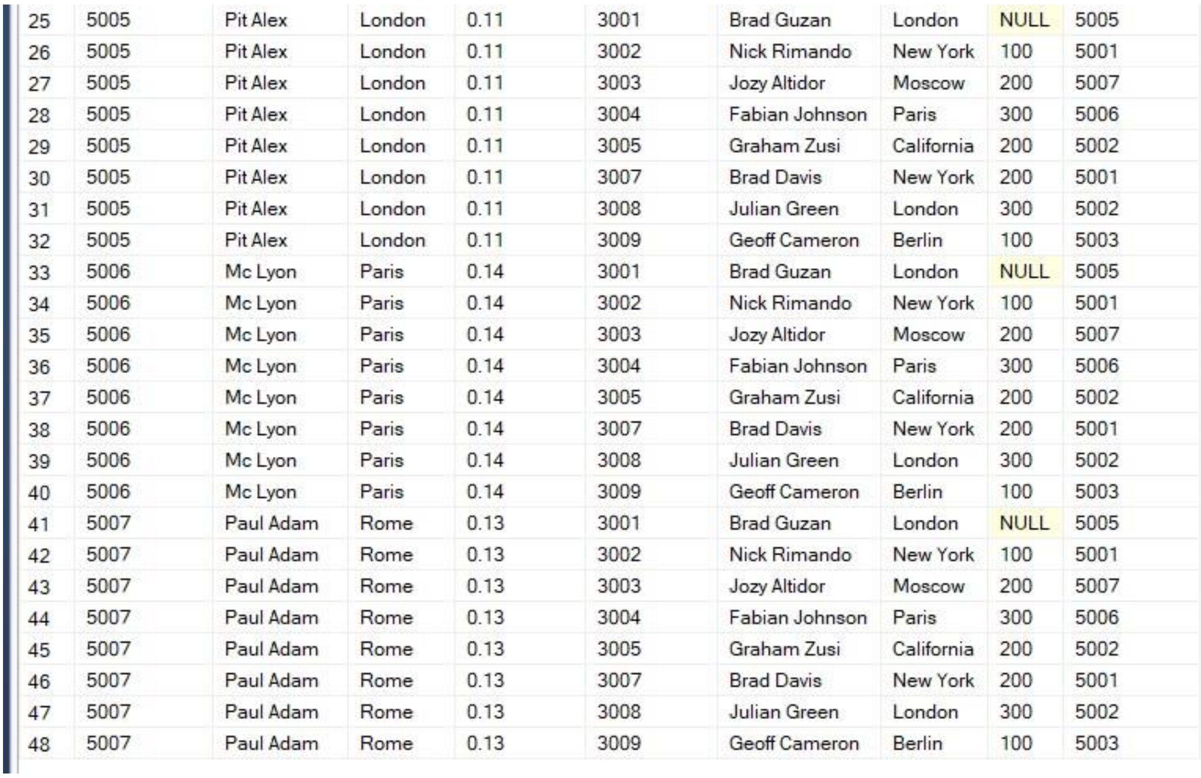
	cust_name	city	ord_no	ord_date	purch_amt
1	Graham Zusi	California	70001	2012-10-05	150.50
2	Nick Rimando	New York	70002	2012-10-05	65.26
3	Geoff Cameron	Berlin	70003	2012-10-10	2480.40
4	Geoff Cameron	Berlin	70004	2012-08-17	110.50
5	Brad Davis	New York	70005	2012-07-27	2400.60
6	Graham Zusi	California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

Ans:-
```
SELECT * FROM salesman a
 CROSS JOIN customer b;
```



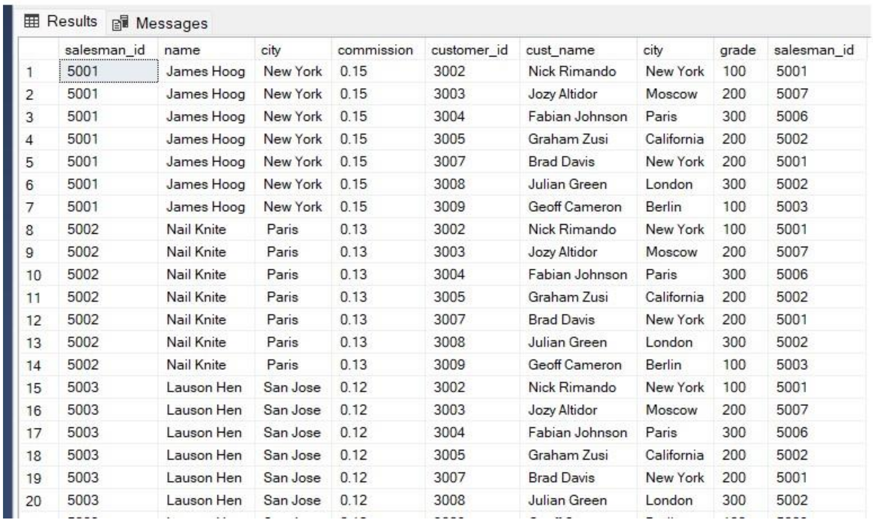
	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	5001	James Hoog	New York	0.15	3001	Brad Guzan	London	NULL	5005
2	5001	James Hoog	New York	0.15	3002	Nick Rimando	New York	100	5001
3	5001	James Hoog	New York	0.15	3003	Jozy Altidor	Moscow	200	5007
4	5001	James Hoog	New York	0.15	3004	Fabian Johnson	Paris	300	5006
5	5001	James Hoog	New York	0.15	3005	Graham Zusi	California	200	5002
6	5001	James Hoog	New York	0.15	3007	Brad Davis	New York	200	5001
7	5001	James Hoog	New York	0.15	3008	Julian Green	London	300	5002
8	5001	James Hoog	New York	0.15	3009	Geoff Cameron	Berlin	100	5003
9	5002	Nail Knite	Paris	0.13	3001	Brad Guzan	London	NULL	5005
10	5002	Nail Knite	Paris	0.13	3002	Nick Rimando	New York	100	5001
11	5002	Nail Knite	Paris	0.13	3003	Jozy Altidor	Moscow	200	5007
12	5002	Nail Knite	Paris	0.13	3004	Fabian Johnson	Paris	300	5006
13	5002	Nail Knite	Paris	0.13	3005	Graham Zusi	California	200	5002
14	5002	Nail Knite	Paris	0.13	3007	Brad Davis	New York	200	5001
15	5002	Nail Knite	Paris	0.13	3008	Julian Green	London	300	5002
16	5002	Nail Knite	Paris	0.13	3009	Geoff Cameron	Berlin	100	5003
17	5003	Lauson Hen	San Jose	0.12	3001	Brad Guzan	London	NULL	5005
18	5003	Lauson Hen	San Jose	0.12	3002	Nick Rimando	New York	100	5001
19	5003	Lauson Hen	San Jose	0.12	3003	Jozy Altidor	Moscow	200	5007
20	5003	Lauson Hen	San Jose	0.12	3004	Fabian Johnson	Paris	300	5006
21	5003	Lauson Hen	San Jose	0.12	3005	Graham Zusi	California	200	5002
22	5003	Lauson Hen	San Jose	0.12	3007	Brad Davis	New York	200	5001
23	5003	Lauson Hen	San Jose	0.12	3008	Julian Green	London	300	5002
24	5003	Lauson Hen	San Jose	0.12	3009	Geoff Cameron	Berlin	100	5003



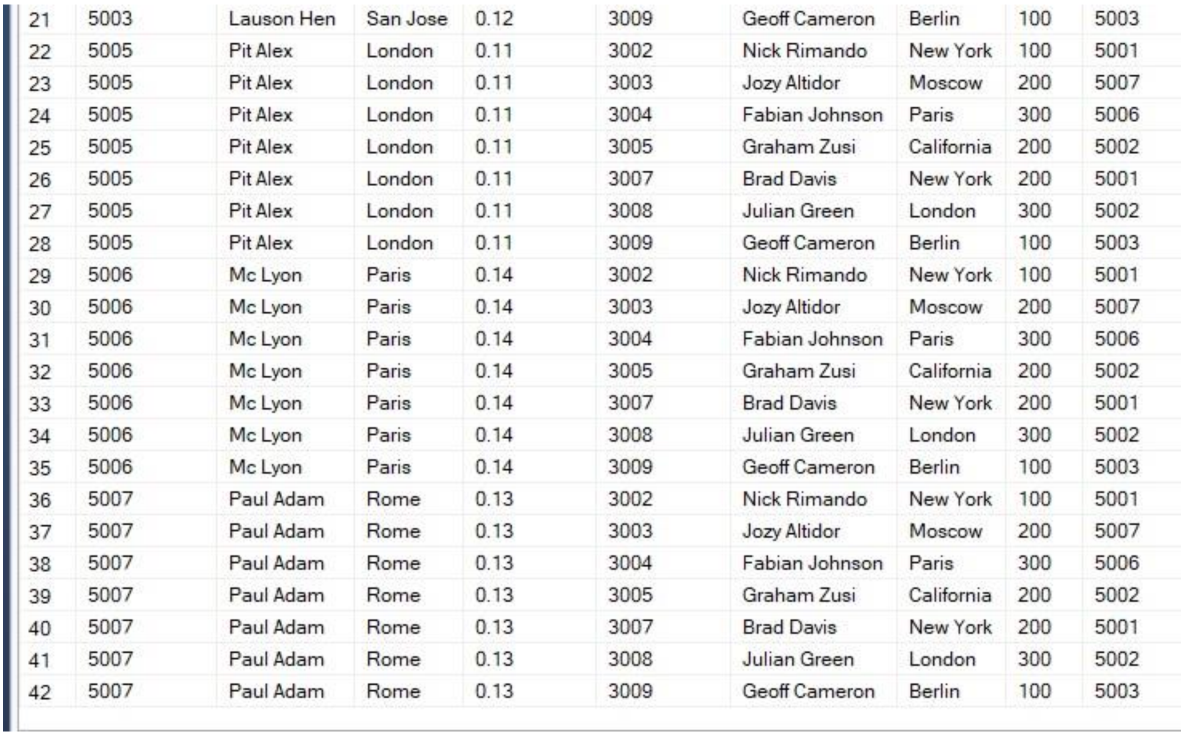
25	5005	Pit Alex	London	0.11	3001	Brad Guzan	London	NULL	5005
26	5005	Pit Alex	London	0.11	3002	Nick Rimando	New York	100	5001
27	5005	Pit Alex	London	0.11	3003	Jozy Altidor	Moscow	200	5007
28	5005	Pit Alex	London	0.11	3004	Fabian Johnson	Paris	300	5006
29	5005	Pit Alex	London	0.11	3005	Graham Zusi	California	200	5002
30	5005	Pit Alex	London	0.11	3007	Brad Davis	New York	200	5001
31	5005	Pit Alex	London	0.11	3008	Julian Green	London	300	5002
32	5005	Pit Alex	London	0.11	3009	Geoff Cameron	Berlin	100	5003
33	5006	Mc Lyon	Paris	0.14	3001	Brad Guzan	London	NULL	5005
34	5006	Mc Lyon	Paris	0.14	3002	Nick Rimando	New York	100	5001
35	5006	Mc Lyon	Paris	0.14	3003	Jozy Altidor	Moscow	200	5007
36	5006	Mc Lyon	Paris	0.14	3004	Fabian Johnson	Paris	300	5006
37	5006	Mc Lyon	Paris	0.14	3005	Graham Zusi	California	200	5002
38	5006	Mc Lyon	Paris	0.14	3007	Brad Davis	New York	200	5001
39	5006	Mc Lyon	Paris	0.14	3008	Julian Green	London	300	5002
40	5006	Mc Lyon	Paris	0.14	3009	Geoff Cameron	Berlin	100	5003
41	5007	Paul Adam	Rome	0.13	3001	Brad Guzan	London	NULL	5005
42	5007	Paul Adam	Rome	0.13	3002	Nick Rimando	New York	100	5001
43	5007	Paul Adam	Rome	0.13	3003	Jozy Altidor	Moscow	200	5007
44	5007	Paul Adam	Rome	0.13	3004	Fabian Johnson	Paris	300	5006
45	5007	Paul Adam	Rome	0.13	3005	Graham Zusi	California	200	5002
46	5007	Paul Adam	Rome	0.13	3007	Brad Davis	New York	200	5001
47	5007	Paul Adam	Rome	0.13	3008	Julian Green	London	300	5002
48	5007	Paul Adam	Rome	0.13	3009	Geoff Cameron	Berlin	100	5003

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 a
   CROSS JOIN  customer b
   WHERE a.city IS NOT NULL
   AND b.grade IS NOT NULL;
```



	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	5001	James Hoog	New York	0.15	3002	Nick Rimando	New York	100	5001
2	5001	James Hoog	New York	0.15	3003	Jozy Altidor	Moscow	200	5007
3	5001	James Hoog	New York	0.15	3004	Fabian Johnson	Paris	300	5006
4	5001	James Hoog	New York	0.15	3005	Graham Zusi	California	200	5002
5	5001	James Hoog	New York	0.15	3007	Brad Davis	New York	200	5001
6	5001	James Hoog	New York	0.15	3008	Julian Green	London	300	5002
7	5001	James Hoog	New York	0.15	3009	Geoff Cameron	Berlin	100	5003
8	5002	Nail Knite	Paris	0.13	3002	Nick Rimando	New York	100	5001
9	5002	Nail Knite	Paris	0.13	3003	Jozy Altidor	Moscow	200	5007
10	5002	Nail Knite	Paris	0.13	3004	Fabian Johnson	Paris	300	5006
11	5002	Nail Knite	Paris	0.13	3005	Graham Zusi	California	200	5002
12	5002	Nail Knite	Paris	0.13	3007	Brad Davis	New York	200	5001
13	5002	Nail Knite	Paris	0.13	3008	Julian Green	London	300	5002
14	5002	Nail Knite	Paris	0.13	3009	Geoff Cameron	Berlin	100	5003
15	5003	Lauson Hen	San Jose	0.12	3002	Nick Rimando	New York	100	5001
16	5003	Lauson Hen	San Jose	0.12	3003	Jozy Altidor	Moscow	200	5007
17	5003	Lauson Hen	San Jose	0.12	3004	Fabian Johnson	Paris	300	5006
18	5003	Lauson Hen	San Jose	0.12	3005	Graham Zusi	California	200	5002
19	5003	Lauson Hen	San Jose	0.12	3007	Brad Davis	New York	200	5001
20	5003	Lauson Hen	San Jose	0.12	3008	Julian Green	London	300	5002

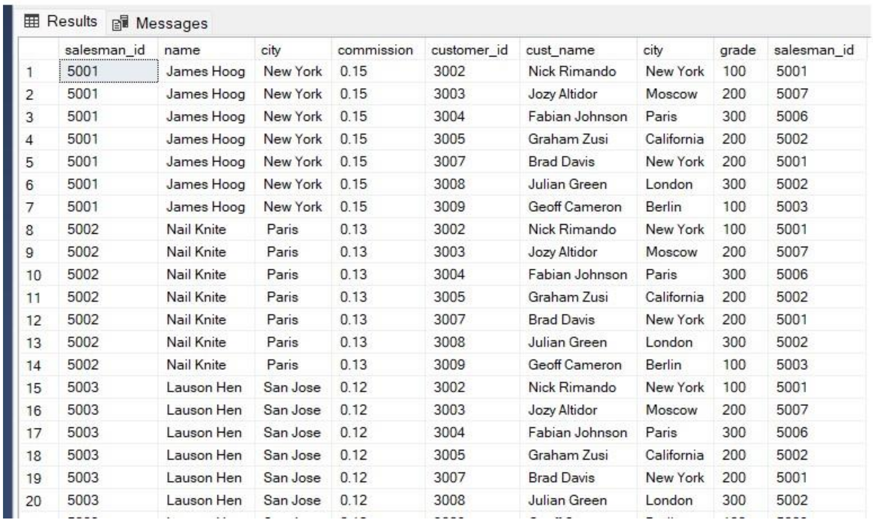


21	5003	Lauson Hen	San Jose	0.12	3009	Geoff Cameron	Berlin	100	5003
22	5005	Pit Alex	London	0.11	3002	Nick Rimando	New York	100	5001
23	5005	Pit Alex	London	0.11	3003	Jozy Altidor	Moscow	200	5007
24	5005	Pit Alex	London	0.11	3004	Fabian Johnson	Paris	300	5006
25	5005	Pit Alex	London	0.11	3005	Graham Zusi	California	200	5002
26	5005	Pit Alex	London	0.11	3007	Brad Davis	New York	200	5001
27	5005	Pit Alex	London	0.11	3008	Julian Green	London	300	5002
28	5005	Pit Alex	London	0.11	3009	Geoff Cameron	Berlin	100	5003
29	5006	Mc Lyon	Paris	0.14	3002	Nick Rimando	New York	100	5001
30	5006	Mc Lyon	Paris	0.14	3003	Jozy Altidor	Moscow	200	5007
31	5006	Mc Lyon	Paris	0.14	3004	Fabian Johnson	Paris	300	5006
32	5006	Mc Lyon	Paris	0.14	3005	Graham Zusi	California	200	5002
33	5006	Mc Lyon	Paris	0.14	3007	Brad Davis	New York	200	5001
34	5006	Mc Lyon	Paris	0.14	3008	Julian Green	London	300	5002
35	5006	Mc Lyon	Paris	0.14	3009	Geoff Cameron	Berlin	100	5003
36	5007	Paul Adam	Rome	0.13	3002	Nick Rimando	New York	100	5001
37	5007	Paul Adam	Rome	0.13	3003	Jozy Altidor	Moscow	200	5007
38	5007	Paul Adam	Rome	0.13	3004	Fabian Johnson	Paris	300	5006
39	5007	Paul Adam	Rome	0.13	3005	Graham Zusi	California	200	5002
40	5007	Paul Adam	Rome	0.13	3007	Brad Davis	New York	200	5001
41	5007	Paul Adam	Rome	0.13	3008	Julian Green	London	300	5002
42	5007	Paul Adam	Rome	0.13	3009	Geoff Cameron	Berlin	100	5003

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

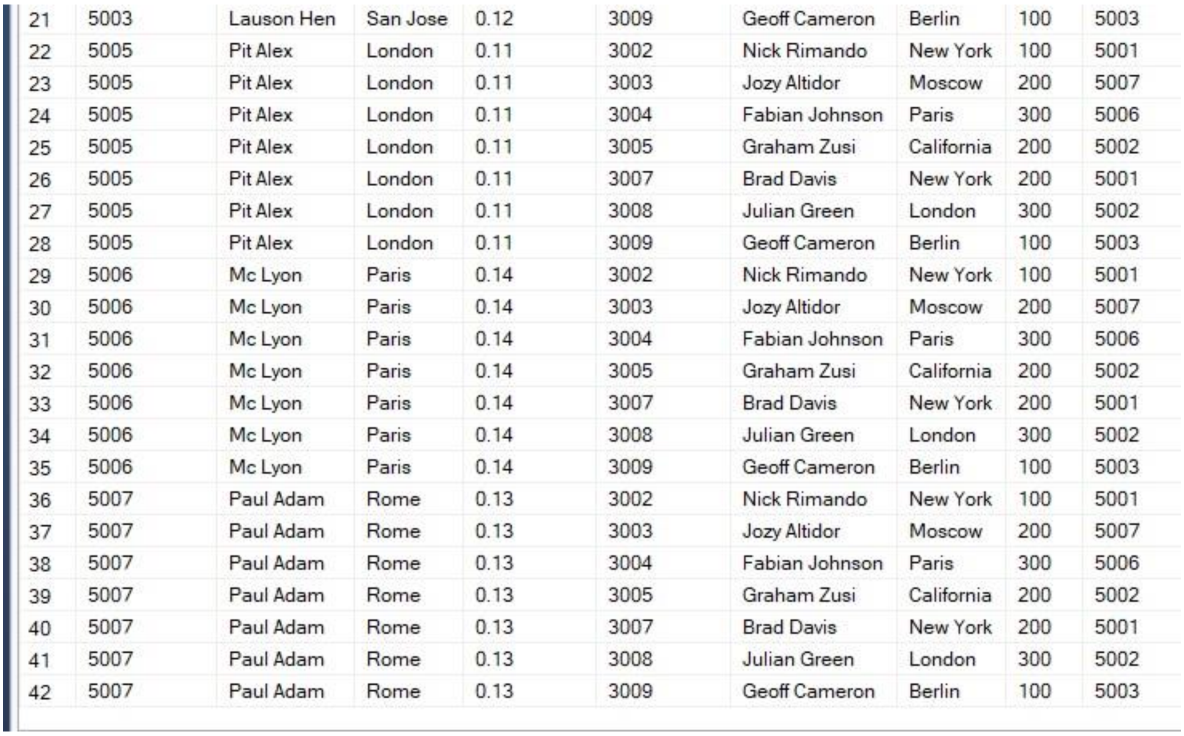
Ans:-
```
SELECT * FROM salesman a
CROSS JOIN  customer b
WHERE a.city IS NOT NULL
```

```
AND b.grade IS NOT NULL;
```



Results | Messages

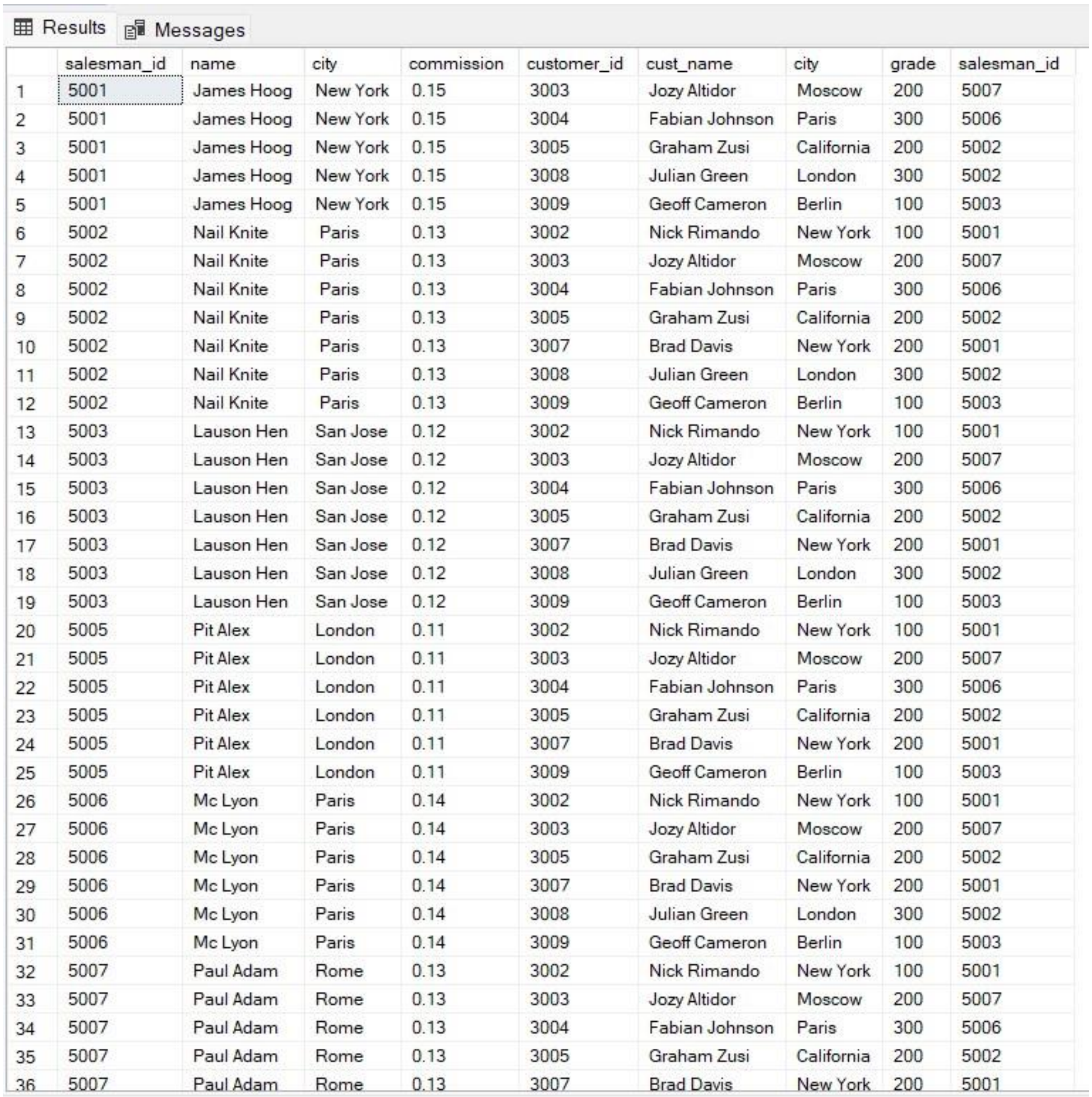
	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	5001	James Hoog	New York	0.15	3002	Nick Rimando	New York	100	5001
2	5001	James Hoog	New York	0.15	3003	Jozy Altidor	Moscow	200	5007
3	5001	James Hoog	New York	0.15	3004	Fabian Johnson	Paris	300	5006
4	5001	James Hoog	New York	0.15	3005	Graham Zusi	California	200	5002
5	5001	James Hoog	New York	0.15	3007	Brad Davis	New York	200	5001
6	5001	James Hoog	New York	0.15	3008	Julian Green	London	300	5002
7	5001	James Hoog	New York	0.15	3009	Geoff Cameron	Berlin	100	5003
8	5002	Nail Knite	Paris	0.13	3002	Nick Rimando	New York	100	5001
9	5002	Nail Knite	Paris	0.13	3003	Jozy Altidor	Moscow	200	5007
10	5002	Nail Knite	Paris	0.13	3004	Fabian Johnson	Paris	300	5006
11	5002	Nail Knite	Paris	0.13	3005	Graham Zusi	California	200	5002
12	5002	Nail Knite	Paris	0.13	3007	Brad Davis	New York	200	5001
13	5002	Nail Knite	Paris	0.13	3008	Julian Green	London	300	5002
14	5002	Nail Knite	Paris	0.13	3009	Geoff Cameron	Berlin	100	5003
15	5003	Lauson Hen	San Jose	0.12	3002	Nick Rimando	New York	100	5001
16	5003	Lauson Hen	San Jose	0.12	3003	Jozy Altidor	Moscow	200	5007
17	5003	Lauson Hen	San Jose	0.12	3004	Fabian Johnson	Paris	300	5006
18	5003	Lauson Hen	San Jose	0.12	3005	Graham Zusi	California	200	5002
19	5003	Lauson Hen	San Jose	0.12	3007	Brad Davis	New York	200	5001
20	5003	Lauson Hen	San Jose	0.12	3008	Julian Green	London	300	5002



21	5003	Lauson Hen	San Jose	0.12	3009	Geoff Cameron	Berlin	100	5003
22	5005	Pit Alex	London	0.11	3002	Nick Rimando	New York	100	5001
23	5005	Pit Alex	London	0.11	3003	Jozy Altidor	Moscow	200	5007
24	5005	Pit Alex	London	0.11	3004	Fabian Johnson	Paris	300	5006
25	5005	Pit Alex	London	0.11	3005	Graham Zusi	California	200	5002
26	5005	Pit Alex	London	0.11	3007	Brad Davis	New York	200	5001
27	5005	Pit Alex	London	0.11	3008	Julian Green	London	300	5002
28	5005	Pit Alex	London	0.11	3009	Geoff Cameron	Berlin	100	5003
29	5006	Mc Lyon	Paris	0.14	3002	Nick Rimando	New York	100	5001
30	5006	Mc Lyon	Paris	0.14	3003	Jozy Altidor	Moscow	200	5007
31	5006	Mc Lyon	Paris	0.14	3004	Fabian Johnson	Paris	300	5006
32	5006	Mc Lyon	Paris	0.14	3005	Graham Zusi	California	200	5002
33	5006	Mc Lyon	Paris	0.14	3007	Brad Davis	New York	200	5001
34	5006	Mc Lyon	Paris	0.14	3008	Julian Green	London	300	5002
35	5006	Mc Lyon	Paris	0.14	3009	Geoff Cameron	Berlin	100	5003
36	5007	Paul Adam	Rome	0.13	3002	Nick Rimando	New York	100	5001
37	5007	Paul Adam	Rome	0.13	3003	Jozy Altidor	Moscow	200	5007
38	5007	Paul Adam	Rome	0.13	3004	Fabian Johnson	Paris	300	5006
39	5007	Paul Adam	Rome	0.13	3005	Graham Zusi	California	200	5002
40	5007	Paul Adam	Rome	0.13	3007	Brad Davis	New York	200	5001
41	5007	Paul Adam	Rome	0.13	3008	Julian Green	London	300	5002
42	5007	Paul Adam	Rome	0.13	3009	Geoff Cameron	Berlin	100	5003

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 a
       CROSS JOIN customer b
       WHERE a.city IS NOT NULL
       AND b.grade IS NOT NULL
       AND  a.city<>b.city;
```



	salesman_id	name	city	commission	customer_id	cust_name	city	grade	salesman_id
1	5001	James Hoog	New York	0.15	3003	Jozy Altidor	Moscow	200	5007
2	5001	James Hoog	New York	0.15	3004	Fabian Johnson	Paris	300	5006
3	5001	James Hoog	New York	0.15	3005	Graham Zusi	California	200	5002
4	5001	James Hoog	New York	0.15	3008	Julian Green	London	300	5002
5	5001	James Hoog	New York	0.15	3009	Geoff Cameron	Berlin	100	5003
6	5002	Nail Knite	Paris	0.13	3002	Nick Rimando	New York	100	5001
7	5002	Nail Knite	Paris	0.13	3003	Jozy Altidor	Moscow	200	5007
8	5002	Nail Knite	Paris	0.13	3004	Fabian Johnson	Paris	300	5006
9	5002	Nail Knite	Paris	0.13	3005	Graham Zusi	California	200	5002
10	5002	Nail Knite	Paris	0.13	3007	Brad Davis	New York	200	5001
11	5002	Nail Knite	Paris	0.13	3008	Julian Green	London	300	5002
12	5002	Nail Knite	Paris	0.13	3009	Geoff Cameron	Berlin	100	5003
13	5003	Lauson Hen	San Jose	0.12	3002	Nick Rimando	New York	100	5001
14	5003	Lauson Hen	San Jose	0.12	3003	Jozy Altidor	Moscow	200	5007
15	5003	Lauson Hen	San Jose	0.12	3004	Fabian Johnson	Paris	300	5006
16	5003	Lauson Hen	San Jose	0.12	3005	Graham Zusi	California	200	5002
17	5003	Lauson Hen	San Jose	0.12	3007	Brad Davis	New York	200	5001
18	5003	Lauson Hen	San Jose	0.12	3008	Julian Green	London	300	5002
19	5003	Lauson Hen	San Jose	0.12	3009	Geoff Cameron	Berlin	100	5003
20	5005	Pit Alex	London	0.11	3002	Nick Rimando	New York	100	5001
21	5005	Pit Alex	London	0.11	3003	Jozy Altidor	Moscow	200	5007
22	5005	Pit Alex	London	0.11	3004	Fabian Johnson	Paris	300	5006
23	5005	Pit Alex	London	0.11	3005	Graham Zusi	California	200	5002
24	5005	Pit Alex	London	0.11	3007	Brad Davis	New York	200	5001
25	5005	Pit Alex	London	0.11	3009	Geoff Cameron	Berlin	100	5003
26	5006	Mc Lyon	Paris	0.14	3002	Nick Rimando	New York	100	5001
27	5006	Mc Lyon	Paris	0.14	3003	Jozy Altidor	Moscow	200	5007
28	5006	Mc Lyon	Paris	0.14	3005	Graham Zusi	California	200	5002
29	5006	Mc Lyon	Paris	0.14	3007	Brad Davis	New York	200	5001
30	5006	Mc Lyon	Paris	0.14	3008	Julian Green	London	300	5002
31	5006	Mc Lyon	Paris	0.14	3009	Geoff Cameron	Berlin	100	5003
32	5007	Paul Adam	Rome	0.13	3002	Nick Rimando	New York	100	5001
33	5007	Paul Adam	Rome	0.13	3003	Jozy Altidor	Moscow	200	5007
34	5007	Paul Adam	Rome	0.13	3004	Fabian Johnson	Paris	300	5006
35	5007	Paul Adam	Rome	0.13	3005	Graham Zusi	California	200	5002
36	5007	Paul Adam	Rome	0.13	3007	Brad Davis	New York	200	5001