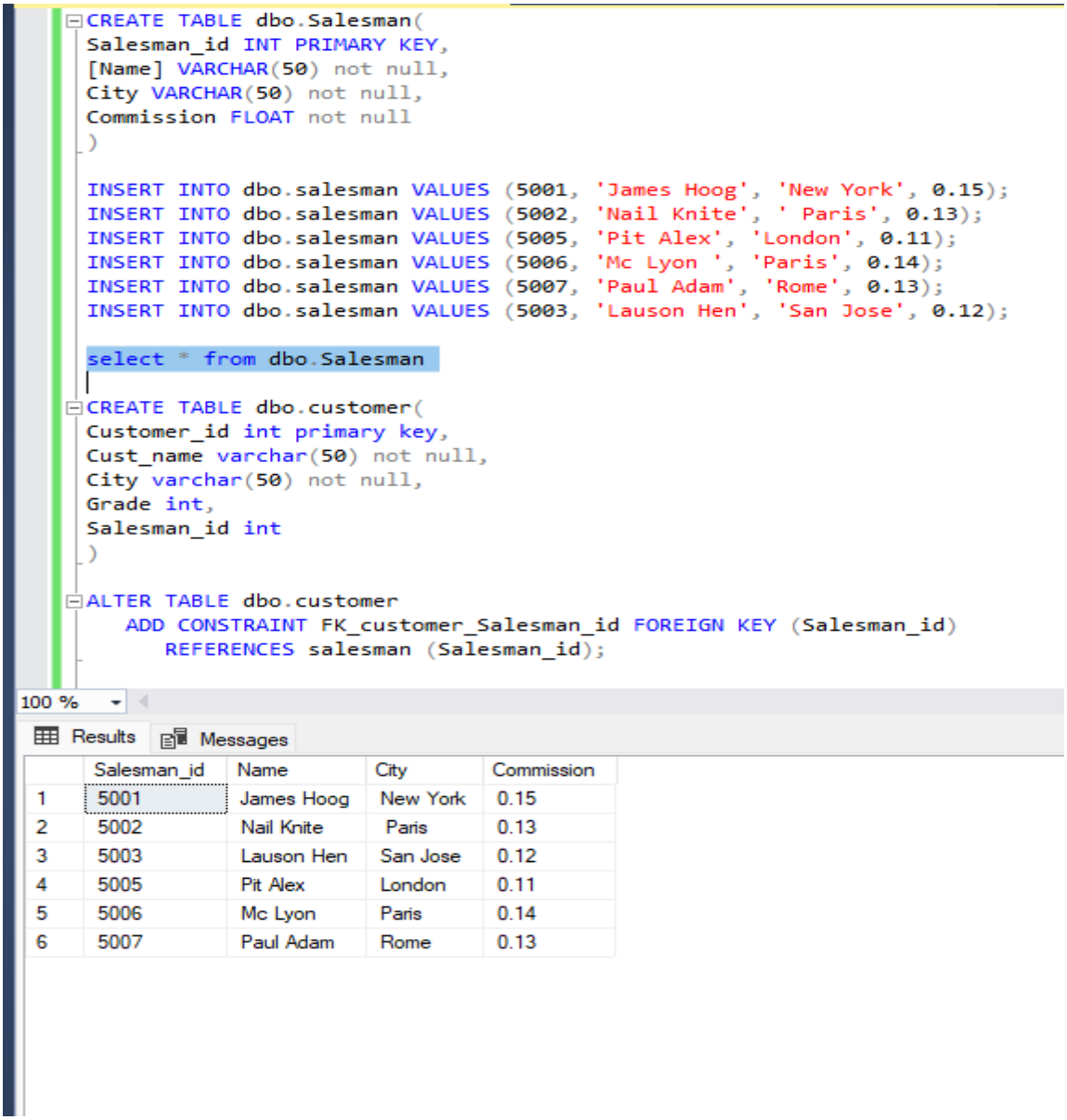
1) salesman table :

```sql
CREATE TABLE dbo.Salesman(
Salesman_id INT PRIMARY KEY,
[Name] VARCHAR(50) not null,
City VARCHAR(50) not null,
Commission FLOAT not null
)

INSERT INTO dbo.salesman VALUES (5001, 'James Hoog', 'New York', 0.15);
INSERT INTO dbo.salesman VALUES (5002, 'Nail Knite', ' Paris', 0.13);
INSERT INTO dbo.salesman VALUES (5005, 'Pit Alex', 'London', 0.11);
INSERT INTO dbo.salesman VALUES (5006, 'Mc Lyon ', 'Paris', 0.14);
INSERT INTO dbo.salesman VALUES (5007, 'Paul Adam', 'Rome', 0.13);
INSERT INTO dbo.salesman VALUES (5003, 'Lauson Hen', 'San Jose', 0.12);

select * from dbo.Salesman
```



	Salesman_id	Name	City	Commission
1	5001	James Hoog	New York	0.15
2	5002	Nail Knite	Paris	0.13
3	5003	Lauson Hen	San Jose	0.12
4	5005	Pit Alex	London	0.11
5	5006	Mc Lyon	Paris	0.14
6	5007	Paul Adam	Rome	0.13

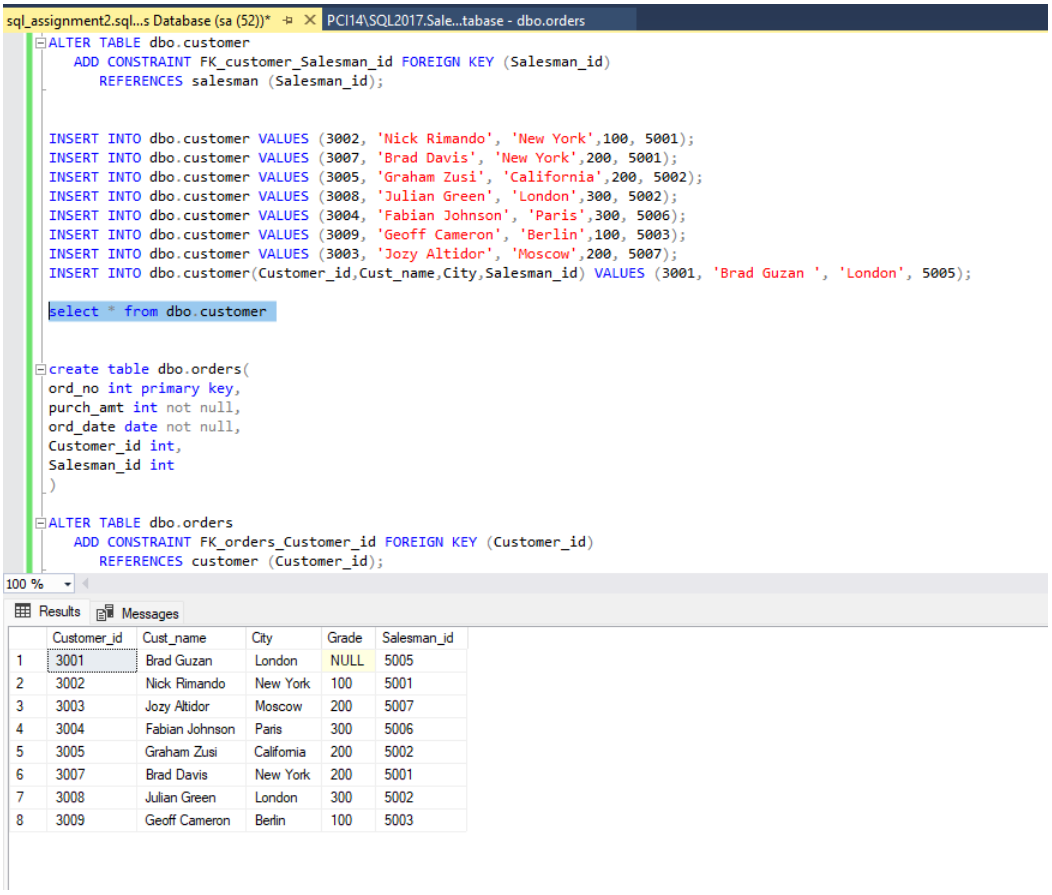
2) customer table :

```sql
CREATE TABLE dbo.customer(
Customer_id int primary key,
Cust_name varchar(50) not null,
City varchar(50) not null,
Grade int,
Salesman_id int
)

ALTER TABLE dbo.customer
   ADD CONSTRAINT FK_customer_Salesman_id FOREIGN KEY (Salesman_id)
      REFERENCES salesman (Salesman_id);


INSERT INTO dbo.customer VALUES (3002, 'Nick Rimando', 'New York',100, 5001);
INSERT INTO dbo.customer VALUES (3007, 'Brad Davis', 'New York',200, 5001);
INSERT INTO dbo.customer VALUES (3005, 'Graham Zusi', 'California',200, 5002);
INSERT INTO dbo.customer VALUES (3008, 'Julian Green', 'London',300, 5002);
INSERT INTO dbo.customer VALUES (3004, 'Fabian Johnson', 'Paris',300, 5006);
INSERT INTO dbo.customer VALUES (3009, 'Geoff Cameron', 'Berlin',100, 5003);
INSERT INTO dbo.customer VALUES (3003, 'Jozy Altidor', 'Moscow',200, 5007);
INSERT INTO dbo.customer(Customer_id,Cust_name,City,Salesman_id) VALUES (3001, 'Brad Guzan ',
'London', 5005);

select * from dbo.customer
```



	Customer_id	Cust_name	City	Grade	Salesman_id
1	3001	Brad Guzan	London	NULL	5005
2	3002	Nick Rimando	New York	100	5001
3	3003	Jozy Altidor	Moscow	200	5007
4	3004	Fabian Johnson	Paris	300	5006
5	3005	Graham Zusi	California	200	5002
6	3007	Brad Davis	New York	200	5001
7	3008	Julian Green	London	300	5002
8	3009	Geoff Cameron	Berlin	100	5003

3) order table:

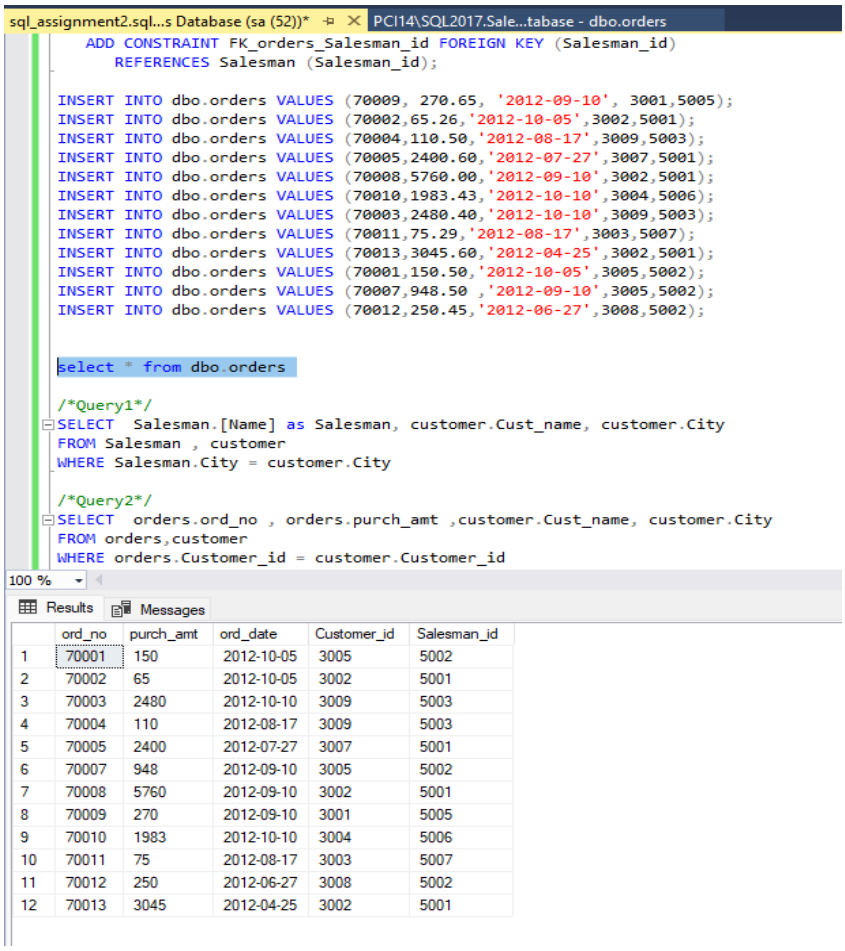
```sql
create table dbo.orders(
ord_no int primary key,
purch_amt int not null,
ord_date date not null,
Customer_id int,
Salesman_id int
)

ALTER TABLE dbo.orders
   ADD CONSTRAINT FK_orders_Customer_id FOREIGN KEY (Customer_id)
      REFERENCES customer (Customer_id);

ALTER TABLE dbo.orders
   ADD CONSTRAINT FK_orders_Salesman_id FOREIGN KEY (Salesman_id)
      REFERENCES Salesman (Salesman_id);

INSERT INTO dbo.orders VALUES (70009, 270.65, '2012-09-10', 3001,5005);
INSERT INTO dbo.orders VALUES (70002,65.26,'2012-10-05',3002,5001);
INSERT INTO dbo.orders VALUES (70004,110.50,'2012-08-17',3009,5003);
INSERT INTO dbo.orders VALUES (70005,2400.60,'2012-07-27',3007,5001);
INSERT INTO dbo.orders VALUES (70008,5760.00,'2012-09-10',3002,5001);
INSERT INTO dbo.orders VALUES (70010,1983.43,'2012-10-10',3004,5006);
INSERT INTO dbo.orders VALUES (70003,2480.40,'2012-10-10',3009,5003);
INSERT INTO dbo.orders VALUES (70011,75.29,'2012-08-17',3003,5007);
INSERT INTO dbo.orders VALUES (70013,3045.60,'2012-04-25',3002,5001);
INSERT INTO dbo.orders VALUES (70001,150.50,'2012-10-05',3005,5002);
INSERT INTO dbo.orders VALUES (70007,948.50      ,'2012-09-10',3005,5002);
INSERT INTO dbo.orders VALUES (70012,250.45,'2012-06-27',3008,5002);


select * from dbo.orders
```



	ord_no	purch_amt	ord_date	Customer_id	Salesman_id
1	70001	150	2012-10-05	3005	5002
2	70002	65	2012-10-05	3002	5001
3	70003	2480	2012-10-10	3009	5003
4	70004	110	2012-08-17	3009	5003
5	70005	2400	2012-07-27	3007	5001
6	70007	948	2012-09-10	3005	5002
7	70008	5760	2012-09-10	3002	5001
8	70009	270	2012-09-10	3001	5005
9	70010	1983	2012-10-10	3004	5006
10	70011	75	2012-08-17	3003	5007
11	70012	250	2012-06-27	3008	5002
12	70013	3045	2012-04-25	3002	5001

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

```sql
SELECT Salesman.[Name] as Salesman, customer.Cust_name, customer.City
FROM Salesman , customer
WHERE Salesman.City = customer.City
```

sql_assignment2.sql...s Database (sa (52))* | PCI14\SQL2017.Sale...tabase - dbo.orders

```sql
    ADD CONSTRAINT FK_orders_Salesman_id FOREIGN KEY (Salesman_id)
       REFERENCES Salesman (Salesman_id);

INSERT INTO dbo.orders VALUES (70009, 270.65, '2012-09-10', 3001,5005);
INSERT INTO dbo.orders VALUES (70002,65.26,'2012-10-05',3002,5001);
INSERT INTO dbo.orders VALUES (70004,110.50,'2012-08-17',3009,5003);
INSERT INTO dbo.orders VALUES (70005,2400.60,'2012-07-27',3007,5001);
INSERT INTO dbo.orders VALUES (70008,5760.00,'2012-09-10',3002,5001);
INSERT INTO dbo.orders VALUES (70010,1983.43,'2012-10-10',3004,5006);
INSERT INTO dbo.orders VALUES (70003,2480.40,'2012-10-10',3009,5003);
INSERT INTO dbo.orders VALUES (70011,75.29,'2012-08-17',3003,5007);
INSERT INTO dbo.orders VALUES (70013,3045.60,'2012-04-25',3002,5001);
INSERT INTO dbo.orders VALUES (70001,150.50,'2012-10-05',3005,5002);
INSERT INTO dbo.orders VALUES (70007,948.50 ,'2012-09-10',3005,5002);
INSERT INTO dbo.orders VALUES (70012,250.45,'2012-06-27',3008,5002);


select * from dbo.orders

/*Query1*/
SELECT  Salesman.[Name] as Salesman, customer.Cust_name, customer.City
FROM Salesman , customer
WHERE Salesman.City = customer.City

/*Query2*/
SELECT  orders.ord_no , orders.purch_amt ,customer.Cust_name, customer.City
FROM orders,customer
WHERE orders.Customer_id = customer.Customer_id
```

.00 %

Results | Messages

	Salesman	Cust_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
5	Pit Alex	Julian Green	London

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 orders.ord_no , orders.purch_amt ,customer.Cust_name, customer.City
FROM orders,customer
WHERE orders.Customer_id = customer.Customer_id
AND orders.purch_amt BETWEEN 500 AND 2000;
```

Execute Debug

sql_assignment2.sql...s Database (sa (52))* | PC114\SQL2017.Sale...tabase - dbo.orders

```
INSERT INTO dbo.orders VALUES (70003,2480.40,'2012-10-10',3009,5003);
INSERT INTO dbo.orders VALUES (70011,75.29,'2012-08-17',3003,5007);
INSERT INTO dbo.orders VALUES (70013,3045.60,'2012-04-25',3002,5001);
INSERT INTO dbo.orders VALUES (70001,150.50,'2012-10-05',3005,5002);
INSERT INTO dbo.orders VALUES (70007,948.50 ,'2012-09-10',3005,5002);
INSERT INTO dbo.orders VALUES (70012,250.45,'2012-06-27',3008,5002);


select * from dbo.orders

/*Query1*/
SELECT  Salesman.[Name] as Salesman, customer.Cust_name, customer.City
FROM Salesman , customer
WHERE Salesman.City = customer.City

/*Query2*/
SELECT  orders.ord_no , orders.purch_amt ,customer.Cust_name, customer.City
FROM orders,customer
WHERE orders.Customer_id = customer.Customer_id
AND orders.purch_amt BETWEEN 500 AND 2000;

/*Query3*/
SELECT customer.Cust_name, Salesman.City, Salesman.[Name] AS Salesman, Salesman.Commission
FROM customer
Inner JOIN Salesman
ON customer.Salesman_id = Salesman.Salesman_id

/*Query4*/
```

100 %

Results | Messages

	ord_no	purch_amt	Cust_name	City
1	70007	948	Graham Zusi	California
2	70010	1983	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 customer.Cust_name, Salesman.City, Salesman.[Name] AS Salesman, Salesman.Commission
FROM customer
Inner JOIN Salesman
ON customer.Salesman_id = Salesman.Salesman_id
```

sql_assignment2.sql...s Database (sa (52))* | PCI14\SQL2017.Sale...tabase - dbo.orders

```
INSERT INTO dbo.orders VALUES (70003,2480.40,'2012-10-10',3009,5003);
INSERT INTO dbo.orders VALUES (70011,75.29,'2012-08-17',3003,5007);
INSERT INTO dbo.orders VALUES (70013,3045.60,'2012-04-25',3002,5001);
INSERT INTO dbo.orders VALUES (70001,150.50,'2012-10-05',3005,5002);
INSERT INTO dbo.orders VALUES (70007,948.50 ,'2012-09-10',3005,5002);
INSERT INTO dbo.orders VALUES (70012,250.45,'2012-06-27',3008,5002);


select * from dbo.orders

/*Query1*/
SELECT  Salesman.[Name] as Salesman, customer.Cust_name, customer.City
FROM Salesman , customer
WHERE Salesman.City = customer.City

/*Query2*/
SELECT  orders.ord_no , orders.purch_amt ,customer.Cust_name, customer.City
FROM orders,customer
WHERE orders.Customer_id = customer.Customer_id
AND orders.purch_amt BETWEEN 500 AND 2000;

/*Query3*/
SELECT customer.Cust_name, Salesman.City, Salesman.[Name] AS Salesman, Salesman.Commission
FROM customer
Inner JOIN Salesman
ON customer.Salesman_id = Salesman.Salesman_id

/*Query4*/
```

100 %

Results | Messages

	Cust_name	City	Salesman	Commission
1	Brad Guzan	London	Pit Alex	0.11
2	Nick Rimando	New York	James Hoog	0.15
3	Jozy Altidor	Rome	Paul Adam	0.13
4	Fabian Johnson	Paris	Mc Lyon	0.14
5	Graham Zusi	Paris	Nail Knite	0.13
6	Brad Davis	New York	James Hoog	0.15
7	Julian Green	Paris	Nail Knite	0.13
8	Geoff Cameron	San Jose	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.

```sql
SELECT customer.Cust_name, Salesman.City, Salesman.[Name] AS Salesman, Salesman.Commission
FROM customer
Inner JOIN Salesman
ON  customer.Salesman_id = Salesman.Salesman_id
WHERE Salesman.Commission > 0.12
```

sql_assignment2.sql...s Database (sa (52))* | PCI14\SQL2017.Sale...tabase - dbo.orders

```sql
select * from dbo.orders

/*Query1*/
SELECT  Salesman.[Name] as Salesman, customer.Cust_name, customer.City
FROM Salesman , customer
WHERE Salesman.City = customer.City

/*Query2*/
SELECT  orders.ord_no , orders.purch_amt ,customer.Cust_name, customer.City
FROM orders,customer
WHERE orders.Customer_id = customer.Customer_id
AND orders.purch_amt BETWEEN 500 AND 2000;

/*Query3*/
SELECT customer.Cust_name, Salesman.City, Salesman.[Name] AS Salesman, Salesman.Commission
FROM customer
Inner JOIN Salesman
ON customer.Salesman_id = Salesman.Salesman_id

/*Query4*/
SELECT customer.Cust_name, Salesman.City, Salesman.[Name] AS Salesman, Salesman.Commission
FROM customer
Inner JOIN Salesman
ON  customer.Salesman_id = Salesman.Salesman_id
WHERE Salesman.Commission > 0.12

/*Query5*/
SELECT customer.Cust_name, Salesman.City, Salesman.[Name] AS Salesman, Salesman.Commission
```

100 %

Results | Messages

	Cust_name	City	Salesman	Commission
1	Nick Rimando	New York	James Hoog	0.15
2	Jozy Altidor	Rome	Paul Adam	0.13
3	Fabian Johnson	Paris	Mc Lyon	0.14
4	Graham Zusi	Paris	Nail Knite	0.13
5	Brad Davis	New York	James Hoog	0.15
6	Julian Green	Paris	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 customer.Cust_name, Salesman.City, Salesman.[Name] AS Salesman, Salesman.Commission
FROM customer
Inner JOIN Salesman
ON  customer.Salesman_id = Salesman.Salesman_id
WHERE Salesman.Commission > 0.12 AND customer.City<>Salesman.City
```

sql_assignment2.sql...s Database (sa (52))* | PCI14\SQL2017.Sale...tabase - dbo.orders

```
SELECT  orders.ord_no , orders.purch_amt ,customer.Cust_name, customer.City
FROM orders,customer
WHERE orders.Customer_id = customer.Customer_id
AND orders.purch_amt BETWEEN 500 AND 2000;

/*Query3*/
SELECT customer.Cust_name, Salesman.City, Salesman.[Name] AS Salesman, Salesman.Commission
FROM customer
Inner JOIN Salesman
ON customer.Salesman_id = Salesman.Salesman_id

/*Query4*/
SELECT customer.Cust_name, Salesman.City, Salesman.[Name] AS Salesman, Salesman.Commission
FROM customer
Inner JOIN Salesman
ON  customer.Salesman_id = Salesman.Salesman_id
WHERE Salesman.Commission > 0.12

/*Query5*/
SELECT customer.Cust_name, Salesman.City, Salesman.[Name] AS Salesman, Salesman.Commission
FROM customer
Inner JOIN Salesman
ON  customer.Salesman_id = Salesman.Salesman_id
WHERE Salesman.Commission > 0.12 AND customer.City<>Salesman.City

/*Query6*/
SELECT orders.ord_no,orders.ord_date,orders.purch_amt,customer.Cust_name,customer.Grade,Salesman.[Name]
FROM orders
```

100 %

Results | Messages

	Cust_name	City	Salesman	Commission
1	Jozy Altidor	Rome	Paul Adam	0.13
2	Graham Zusi	Paris	Nail Knite	0.13
3	Julian Green	Paris	Nail Knite	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.[N
ame] as Salesman,Salesman.Commission
FROM orders
Inner Join Salesman ON orders.Salesman_id = Salesman.Salesman_id
Inner Join customer ON orders.Customer_id = customer.Customer_id
```

```
/*Query6*/
SELECT orders.ord_no,orders.ord_date,orders.purch_amt,customer.Cust_name,customer
FROM orders
Inner Join Salesman ON orders.Salesman_id = Salesman.Salesman_id
Inner Join customer ON orders.Customer_id = customer.Customer_id

/*Query7*/
SELECT orders.ord_no,orders.ord_date,orders.purch_amt,customer.Customer_id,custom
,customer.Grade,Salesman.Salesman_id,Salesman.[Name] as Salesman_Name,Salesman.Co
```

100 %

Results | Messages

	ord_no	ord_date	purch_amt	Cust_name	Grade	Salesman	Commission
1	70001	2012-10-05	150	Graham Zusi	200	Nail Knite	0.13
2	70002	2012-10-05	65	Nick Rimando	100	James Hoog	0.15
3	70003	2012-10-10	2480	Geoff Cameron	100	Lauson Hen	0.12
4	70004	2012-08-17	110	Geoff Cameron	100	Lauson Hen	0.12
5	70005	2012-07-27	2400	Brad Davis	200	James Hoog	0.15
6	70007	2012-09-10	948	Graham Zusi	200	Nail Knite	0.13
7	70008	2012-09-10	5760	Nick Rimando	100	James Hoog	0.15
8	70009	2012-09-10	270	Brad Guzan	NULL	Pit Alex	0.11
9	70010	2012-10-10	1983	Fabian Johnson	300	Mc Lyon	0.14
10	70011	2012-08-17	75	Jozy Altidor	200	Paul Adam	0.13
11	70012	2012-06-27	250	Julian Green	300	Nail Knite	0.13
12	70013	2012-04-25	3045	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
orders.ord_no,orders.ord_date,orders.purch_amt,customer.Customer_id,customer.Cust_name,custo
mer.City
,customer.Grade,Salesman.Salesman_id,Salesman.[Name] as Salesman_Name,Salesman.Commission
FROM orders
Join Salesman ON orders.Salesman_id = Salesman.Salesman_id
Join customer ON orders.Customer_id = customer.Customer_id
```

```
/*Query7*/
SELECT orders.ord_no,orders.ord_date,orders.purch_amt,customer.Customer_id,customer.Cust_name,customer.City
,customer.Grade,Salesman.Salesman_id,Salesman.[Name] as Salesman_Name,Salesman.Commission
FROM orders
Join Salesman ON orders.Salesman_id = Salesman.Salesman_id
Join customer ON orders.Customer_id = customer.Customer_id

/*Query8*/
SELECT customer.Cust_name,customer.City as Customer_city,customer.Grade,Salesman.[Name] as Salesman_Name,Salesman.City as Sal
```

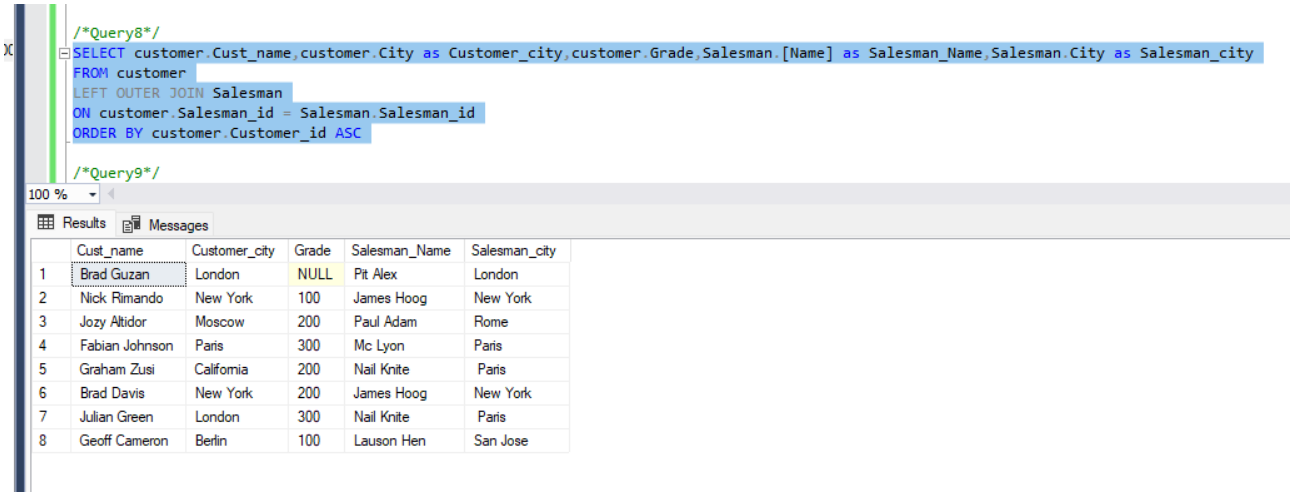
100 %

Results | Messages

	ord_no	ord_date	purch_amt	Customer_id	Cust_name	City	Grade	Salesman_id	Salesman_Name	Commission
1	70001	2012-10-05	150	3005	Graham Zusi	California	200	5002	Nail Knite	0.13
2	70002	2012-10-05	65	3002	Nick Rimando	New York	100	5001	James Hoog	0.15
3	70003	2012-10-10	2480	3009	Geoff Cameron	Berlin	100	5003	Lauson Hen	0.12
4	70004	2012-08-17	110	3009	Geoff Cameron	Berlin	100	5003	Lauson Hen	0.12
5	70005	2012-07-27	2400	3007	Brad Davis	New York	200	5001	James Hoog	0.15
6	70007	2012-09-10	948	3005	Graham Zusi	California	200	5002	Nail Knite	0.13
7	70008	2012-09-10	5760	3002	Nick Rimando	New York	100	5001	James Hoog	0.15
8	70009	2012-09-10	270	3001	Brad Guzan	London	NULL	5005	Pit Alex	0.11
9	70010	2012-10-10	1983	3004	Fabian Johnson	Paris	300	5006	Mc Lyon	0.14
10	70011	2012-08-17	75	3003	Jozy Altidor	Moscow	200	5007	Paul Adam	0.13
11	70012	2012-06-27	250	3008	Julian Green	London	300	5002	Nail Knite	0.13
12	70013	2012-04-25	3045	3002	Nick Rimando	New York	100	5001	James Hoog	0.15

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_Name,Salesman.City as Salesman_city
FROM customer
LEFT OUTER JOIN Salesman
ON customer.Salesman_id = Salesman.Salesman_id
ORDER BY customer.Customer_id ASC
```



```sql
/*Query8*/
SELECT customer.Cust_name,customer.City as Customer_city,customer.Grade,Salesman.[Name] as Salesman_Name,Salesman.City as Salesman_city
FROM customer
LEFT OUTER JOIN Salesman
ON customer.Salesman_id = Salesman.Salesman_id
ORDER BY customer.Customer_id ASC

/*Query9*/
```

100 %

Results | Messages

	Cust_name	Customer_city	Grade	Salesman_Name	Salesman_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 ordered by ascending customer_id.

```
SELECT customer.Cust_name,customer.City as Customer_city,customer.Grade,Salesman.[Name] as
Salesman_Name,Salesman.City as Salesman_city
FROM customer
LEFT OUTER JOIN Salesman
ON customer.Salesman_id = Salesman.Salesman_id
WHERE customer.Grade < 300
ORDER BY customer.Customer_id ASC
```

```
/*Query9*/
SELECT customer.Cust_name,customer.City as Customer_city,customer.Grade,Salesman.[Name] as Salesman_Name,Salesman.City as Salesman_city
FROM customer
LEFT OUTER JOIN Salesman
ON customer.Salesman_id = Salesman.Salesman_id
WHERE customer.Grade < 300
ORDER BY customer.Customer_id ASC

/*Query10*/
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
```

100 %

Results | Messages

	Cust_name	Customer_city	Grade	Salesman_Name	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
```

```
/*Query10*/
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

/*Query11*/
```

100 %

Results | Messages

	Cust_name	City	ord_no	ord_date	purch_amt
1	Nick Rimando	New York	70013	2012-04-25	3045
2	Julian Green	London	70012	2012-06-27	250
3	Brad Davis	New York	70005	2012-07-27	2400
4	Geoff Cameron	Berlin	70004	2012-08-17	110
5	Jozy Altidor	Moscow	70011	2012-08-17	75
6	Graham Zusi	California	70007	2012-09-10	948
7	Nick Rimando	New York	70008	2012-09-10	5760
8	Brad Guzan	London	70009	2012-09-10	270
9	Graham Zusi	California	70001	2012-10-05	150
10	Nick Rimando	New York	70002	2012-10-05	65
11	Geoff Cameron	Berlin	70003	2012-10-10	2480
12	Fabian Johnson	Paris	70010	2012-10-10	1983

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

```sql
SELECT
customer.Cust_name,customer.City,orders.ord_no,orders.ord_date,orders.purch_amt,Salesman.[Na
me] as Salesman_name,Salesman.Commission
FROM customer
JOIN orders ON customer.Customer_id = orders.Customer_id
JOIN Salesman ON customer.Salesman_id = Salesman.Salesman_id
```

```sql
/*Query11*/
SELECT customer.Cust_name,customer.City,orders.ord_no,orders.ord_date,orders.purch_amt,Salesman.[Name] as Salesman_name
FROM customer
JOIN orders ON customer.Customer_id = orders.Customer_id
JOIN Salesman ON customer.Salesman_id = Salesman.Salesman_id

/*Query12*/
```

100 %

Results | Messages

	Cust_name	City	ord_no	ord_date	purch_amt	Salesman_name	Commission
1	Graham Zusi	California	70001	2012-10-05	150	Nail Knite	0.13
2	Nick Rimando	New York	70002	2012-10-05	65	James Hoog	0.15
3	Geoff Cameron	Berlin	70003	2012-10-10	2480	Lauson Hen	0.12
4	Geoff Cameron	Berlin	70004	2012-08-17	110	Lauson Hen	0.12
5	Brad Davis	New York	70005	2012-07-27	2400	James Hoog	0.15
6	Graham Zusi	California	70007	2012-09-10	948	Nail Knite	0.13
7	Nick Rimando	New York	70008	2012-09-10	5760	James Hoog	0.15
8	Brad Guzan	London	70009	2012-09-10	270	Pit Alex	0.11
9	Fabian Johnson	Paris	70010	2012-10-10	1983	Mc Lyon	0.14
10	Jozy Altidor	Moscow	70011	2012-08-17	75	Paul Adam	0.13
11	Julian Green	London	70012	2012-06-27	250	Nail Knite	0.13
12	Nick Rimando	New York	70013	2012-04-25	3045	James Hoog	0.15

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 *
FROM customer
join Salesman
ON customer.Salesman_id = Salesman.Salesman_id
ORDER BY Salesman.Salesman_id
```

```
/*Query12*/
SELECT *
FROM customer
join Salesman
ON customer.Salesman_id = Salesman.Salesman_id
ORDER BY Salesman.Salesman_id

/*Query13*/
```

100 %

Results | Messages

	Customer_id	Cust_name	City	Grade	Salesman_id	Salesman_id	Name	City	Commission
1	3002	Nick Rimando	New York	100	5001	5001	James Hoog	New York	0.15
2	3007	Brad Davis	New York	200	5001	5001	James Hoog	New York	0.15
3	3008	Julian Green	London	300	5002	5002	Nail Knite	Paris	0.13
4	3005	Graham Zusi	California	200	5002	5002	Nail Knite	Paris	0.13
5	3009	Geoff Cameron	Berlin	100	5003	5003	Lauson Hen	San Jose	0.12
6	3001	Brad Guzan	London	NULL	5005	5005	Pit Alex	London	0.11
7	3004	Fabian Johnson	Paris	300	5006	5006	Mc Lyon	Paris	0.14
8	3003	Jozy Altidor	Moscow	200	5007	5007	Paul Adam	Rome	0.13

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_Name,
customer.Cust_name,customer.City,customer.Grade,orders.ord_no,orders.ord_date,orders.purch_a
mt
FROM customer
JOIN Salesman ON Salesman.Salesman_id = customer.Salesman_id
JOIN orders ON orders.Customer_id = customer.Customer_id
```

```
/*Query13*/
SELECT Salesman.[Name] as Salesman_Name, customer.Cust_name,custome
FROM customer
JOIN Salesman ON Salesman.Salesman_id = customer.Salesman_id
JOIN orders ON orders.Customer_id = customer.Customer_id

/*Query14 & Query15*/
```

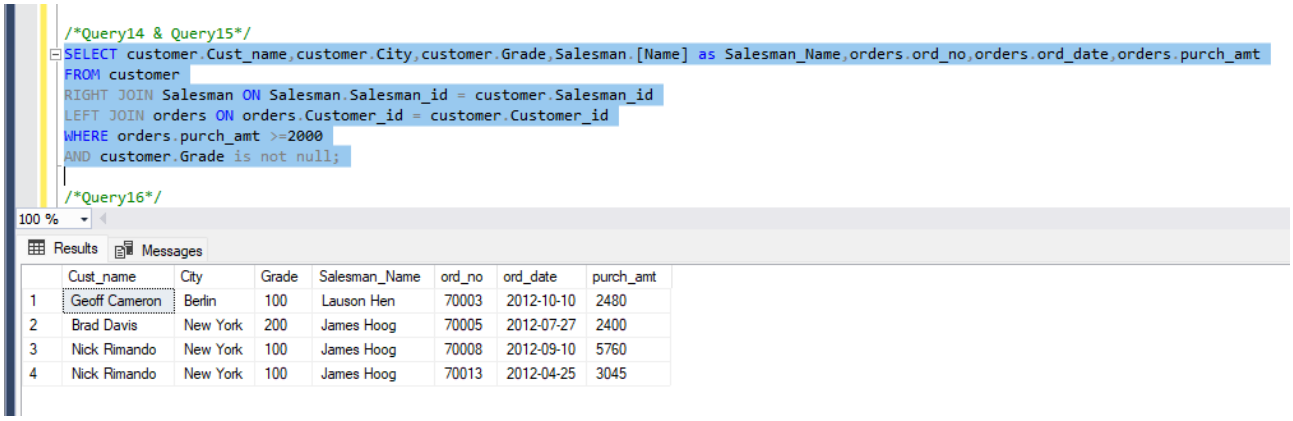
100 %

Results | Messages

	Salesman_Name	Cust_name	City	Grade	ord_no	ord_date	purch_am
1	Nail Knite	Graham Zusi	California	200	70001	2012-10-05	150
2	James Hoog	Nick Rimando	New York	100	70002	2012-10-05	65
3	Lauson Hen	Geoff Cameron	Berlin	100	70003	2012-10-10	2480
4	Lauson Hen	Geoff Cameron	Berlin	100	70004	2012-08-17	110
5	James Hoog	Brad Davis	New York	200	70005	2012-07-27	2400
6	Nail Knite	Graham Zusi	California	200	70007	2012-09-10	948
7	James Hoog	Nick Rimando	New York	100	70008	2012-09-10	5760
8	Pit Alex	Brad Guzan	London	NULL	70009	2012-09-10	270
9	Mc Lyon	Fabian Johnson	Paris	300	70010	2012-10-10	1983
10	Paul Adam	Jozy Altidor	Moscow	200	70011	2012-08-17	75
11	Nail Knite	Julian Green	London	300	70012	2012-06-27	250
12	James Hoog	Nick Rimando	New York	100	70013	2012-04-25	3045

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.
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.

```sql
SELECT customer.Cust_name,customer.City,customer.Grade,Salesman.[Name] as
Salesman_Name,orders.ord_no,orders.ord_date,orders.purch_amt
FROM customer
RIGHT JOIN Salesman ON Salesman.Salesman_id = customer.Salesman_id
LEFT JOIN orders ON orders.Customer_id = customer.Customer_id
WHERE orders.purch_amt >=2000
AND customer.Grade is not null;
```



```sql
/*Query14 & Query15*/
SELECT customer.Cust_name,customer.City,customer.Grade,Salesman.[Name] as Salesman_Name,orders.ord_no,orders.ord_date,orders.purch_amt
FROM customer
RIGHT JOIN Salesman ON Salesman.Salesman_id = customer.Salesman_id
LEFT JOIN orders ON orders.Customer_id = customer.Customer_id
WHERE orders.purch_amt >=2000
AND customer.Grade is not null;

/*Query16*/
```

100 %

Results | Messages

	Cust_name	City	Grade	Salesman_Name	ord_no	ord_date	purch_amt
1	Geoff Cameron	Berlin	100	Lauson Hen	70003	2012-10-10	2480
2	Brad Davis	New York	200	James Hoog	70005	2012-07-27	2400
3	Nick Rimando	New York	100	James Hoog	70008	2012-09-10	5760
4	Nick Rimando	New York	100	James Hoog	70013	2012-04-25	3045

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 customer.City, customer.City,orders.ord_no,orders.ord_date,orders.purch_amt
FROM customer
LEFT JOIN orders ON customer.Customer_id = orders.Customer_id
WHERE customer.Grade is not null
```

```
/*Query16*/
SELECT customer.City, customer.City,orders.ord_no,orders.ord_date,orders.purch_amt
FROM customer
LEFT JOIN orders ON customer.Customer_id = orders.Customer_id
WHERE customer.Grade is not null

/*Query17*/
```

100 %

Results | Messages

	City	City	ord_no	ord_date	purch_amt
1	New York	New York	70002	2012-10-05	65
2	New York	New York	70008	2012-09-10	5760
3	New York	New York	70013	2012-04-25	3045
4	Moscow	Moscow	70011	2012-08-17	75
5	Paris	Paris	70010	2012-10-10	1983
6	California	California	70001	2012-10-05	150
7	California	California	70007	2012-09-10	948
8	New York	New York	70005	2012-07-27	2400
9	London	London	70012	2012-06-27	250
10	Berlin	Berlin	70003	2012-10-10	2480
11	Berlin	Berlin	70004	2012-08-17	110

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

```
SELECT *
FROM Salesman
cross join customer
```

sql_assignment2.sql...s Database (sa (52))* | PCI14\SQL2017.Sale...tabase - dbo.orders

100 %

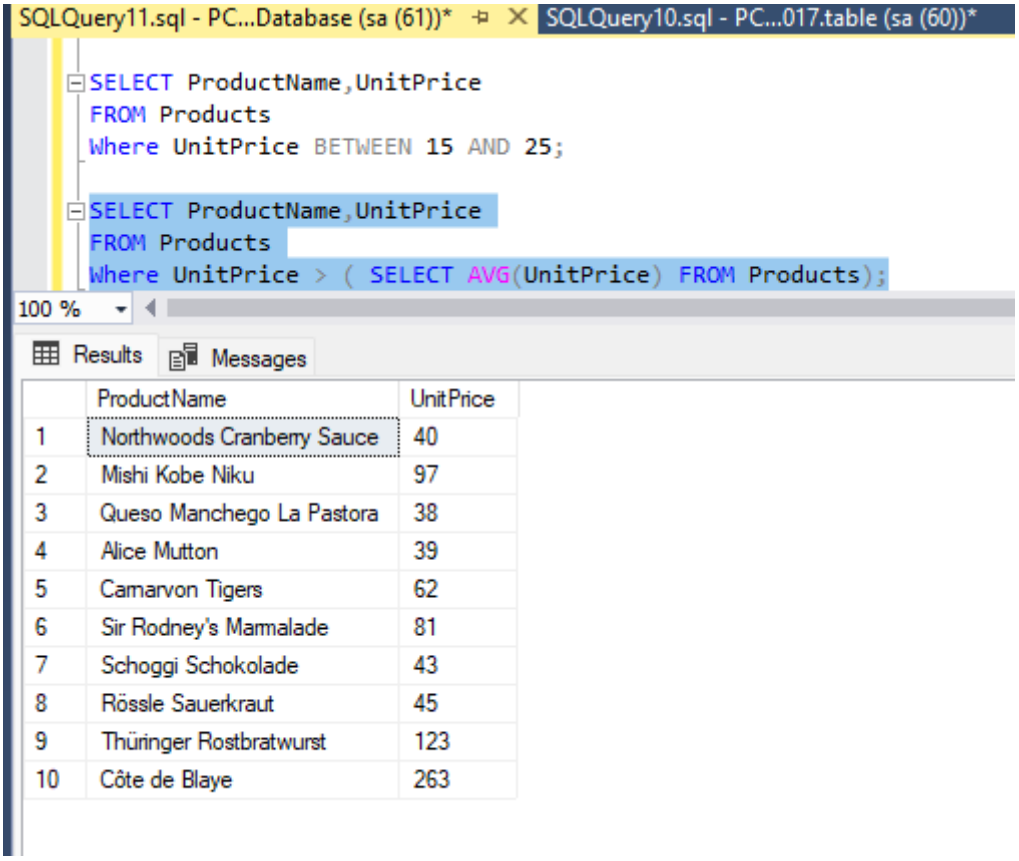
Results | Messages

	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

Query executed successfully.

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 *
FROM Salesman
cross join customer
WHERE Salesman.City= customer.City
```



	ProductName	UnitPrice
1	Northwoods Cranberry Sauce	40
2	Mishi Kobe Niku	97
3	Queso Manchego La Pastora	38
4	Alice Mutton	39
5	Camarvon Tigers	62
6	Sir Rodney's Marmalade	81
7	Schoggi Schokolade	43
8	Rössle Sauerkraut	45
9	Thüringer Rostbratwurst	123
10	Côte de Blaye	263

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 *
FROM Salesman
cross join customer
WHERE Salesman.City is not null
AND customer.Grade is not null
```

sql_assignment2.sql...s Database (sa (52))* | PCI14\SQL2017.Sale...tabase - dbo.orders

100 %

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

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 *
FROM Salesman
cross join customer
WHERE Salesman.City <> customer.City
AND customer.Grade is not null
```

sql_assignment2.sql...s Database (sa (52))* | PCI14\SQL2017.Sale...tabase - dbo.orders

```
/*Query19*/
```

100 %

Results | Messages

	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
37	5007	Paul Adam	Rome	0.13	3008	Julian Green	London	300	5002
38	5007	Paul Adam	Rome	0.13	3009	Geoff Cameron	Berlin	100	5003