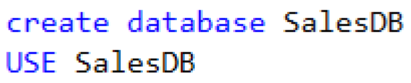
SQL ASSIGNMENT – 2

--> Creating Database Called **SalesDB**

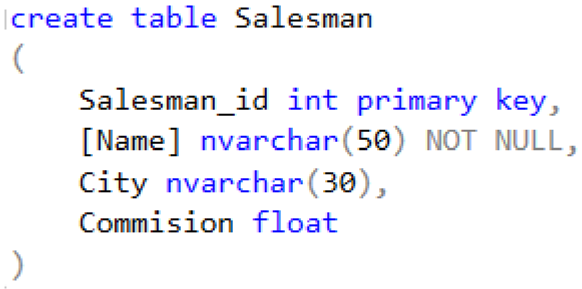
```
create database SalesDB
USE SalesDB
```



```
create database SalesDB
USE SalesDB
```

--> Creating table named **Salesman**

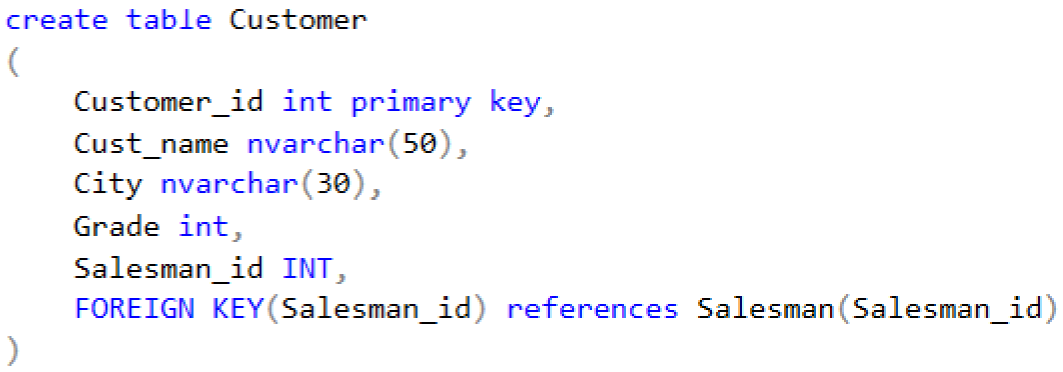
```
create table Salesman
(
        Salesman_id int primary key,
        [Name] nvarchar(50) NOT NULL,
        City nvarchar(30),
        Commision float
)
```



```
create table Salesman
(
    Salesman_id int primary key,
    [Name] nvarchar(50) NOT NULL,
    City nvarchar(30),
    Commision float
)
```

--> Creating table named **Customer**

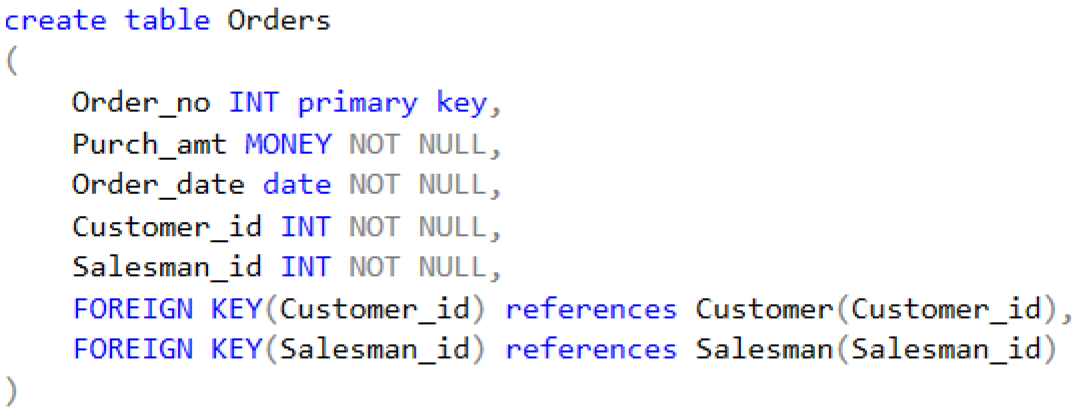
```
create table Customer

(
        Customer_id int primary key,
        Cust_name nvarchar(50),
        City nvarchar(30),
        Grade int,
        Salesman_id INT,
        FOREIGN KEY(Salesman_id) references Salesman(Salesman_id)
)
```



```
create table Customer
(
    Customer_id int primary key,
    Cust_name nvarchar(50),
    City nvarchar(30),
    Grade int,
    Salesman_id INT,
    FOREIGN KEY(Salesman_id) references Salesman(Salesman_id)
)
```

--> Creating table named **Orders**

```
create table Orders
(
        Order_no INT primary key,
        Purch_amt MONEY NOT NULL,
        Order_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)
)
```



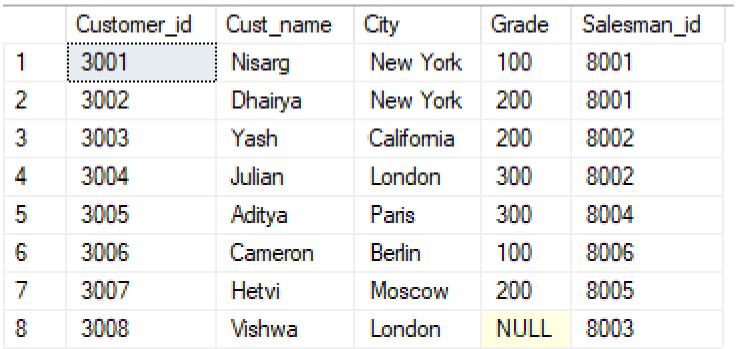
```
create table Orders
(
    Order_no INT primary key,
    Purch_amt MONEY NOT NULL,
    Order_date date NOT NULL,
    Customer_id INT NOT NULL,
    Salesman_id INT NOT NULL,
    FOREIGN KEY(Customer_id) references Customer(Customer_id),
    FOREIGN KEY(Salesman_id) references Salesman(Salesman_id)
)
```

--> Inserting data into Created Table

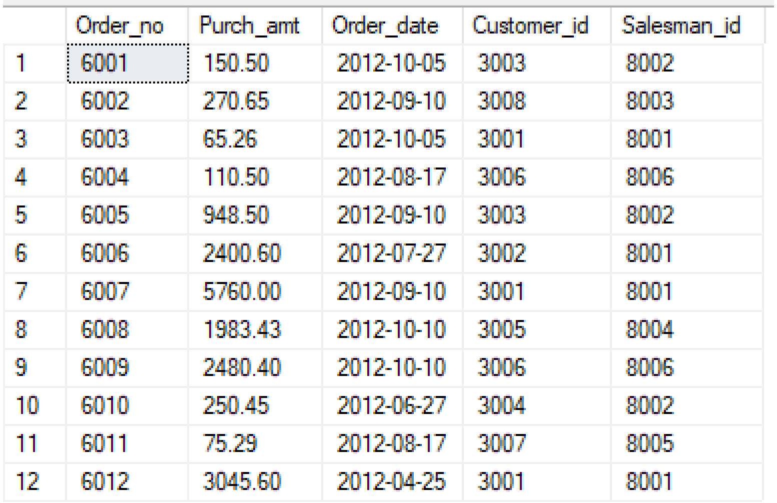
```
Insert into Salesman values(8001, 'Naman', 'New York', 15)
Insert into Salesman values(8002, 'Neel', 'Paris', 13)
Insert into Salesman values(8003, 'Pratham', 'London', 11)
Insert into Salesman values(8004, 'Juhi', 'Paris', 14)
Insert into Salesman values(8005, 'Sparsh', 'Rome', 13)
Insert into Salesman values(8006, 'Vedant', 'San Jose', 12)

Insert into Customer values(3001, 'Nisarg', 'New York', 100, 8001)
Insert into Customer values(3002, 'Dhairya', 'New York', 200, 8001)
Insert into Customer values(3003, 'Yash', 'California', 200, 8002)
Insert into Customer values(3004, 'Julian', 'London', 300, 8002)
Insert into Customer values(3005, 'Aditya', 'Paris', 300, 8004)
Insert into Customer values(3006, 'Cameron', 'Berlin', 100, 8006)
Insert into Customer values(3007, 'Hetvi', 'Moscow', 200, 8005)
Insert into Customer values(3008, 'Vishwa', 'London', NULL, 8003)

Insert into Orders values(6001, 150.5, '2012-10-05', 3003, 8002)
Insert into Orders values(6002, 270.65, '2012-09-10', 3008, 8003)
Insert into Orders values(6003, 65.26, '2012-10-05', 3001, 8001)
Insert into Orders values(6004, 110.5, '2012-08-17', 3006, 8006)
Insert into Orders values(6005, 948.5, '2012-09-10', 3003, 8002)
Insert into Orders values(6006, 2400.6, '2012-07-27', 3002, 8001)
Insert into Orders values(6007, 5760, '2012-09-10', 3001, 8001)
Insert into Orders values(6008, 1983.43, '2012-10-10', 3005, 8004)
Insert into Orders values(6009, 2480.4, '2012-10-10', 3006, 8006)
Insert into Orders values(6010, 250.45, '2012-06-27', 3004, 8002)
Insert into Orders values(6011, 75.29, '2012-08-17', 3007, 8005)
Insert into Orders values(6012, 3045.6, '2012-04-25', 3001, 8001)
```

--> Viewing Inserted data:

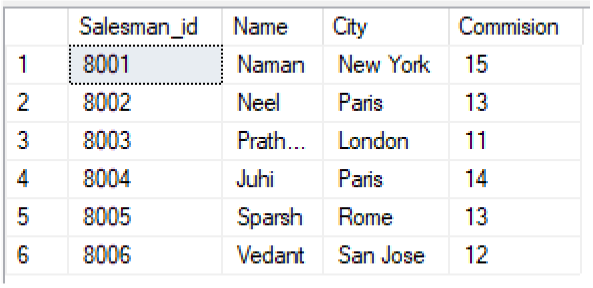
```
select * from Customer
select * from Orders
select * from Salesman
```



	Customer_id	Cust_name	City	Grade	Salesman_id
1	3001	Nisarg	New York	100	8001
2	3002	Dhairya	New York	200	8001
3	3003	Yash	California	200	8002
4	3004	Julian	London	300	8002
5	3005	Aditya	Paris	300	8004
6	3006	Cameron	Berlin	100	8006
7	3007	Hetvi	Moscow	200	8005
8	3008	Vishwa	London	NULL	8003



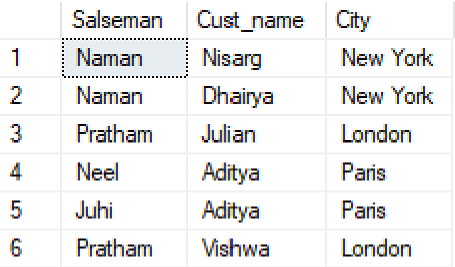
	Order_no	Purch_amt	Order_date	Customer_id	Salesman_id
1	6001	150.50	2012-10-05	3003	8002
2	6002	270.65	2012-09-10	3008	8003
3	6003	65.26	2012-10-05	3001	8001
4	6004	110.50	2012-08-17	3006	8006
5	6005	948.50	2012-09-10	3003	8002
6	6006	2400.60	2012-07-27	3002	8001
7	6007	5760.00	2012-09-10	3001	8001
8	6008	1983.43	2012-10-10	3005	8004
9	6009	2480.40	2012-10-10	3006	8006
10	6010	250.45	2012-06-27	3004	8002
11	6011	75.29	2012-08-17	3007	8005
12	6012	3045.60	2012-04-25	3001	8001



	Salesman_id	Name	City	Commision
1	8001	Naman	New York	15
2	8002	Neel	Paris	13
3	8003	Prath...	London	11
4	8004	Juhi	Paris	14
5	8005	Sparsh	Rome	13
6	8006	Vedant	San Jose	12

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

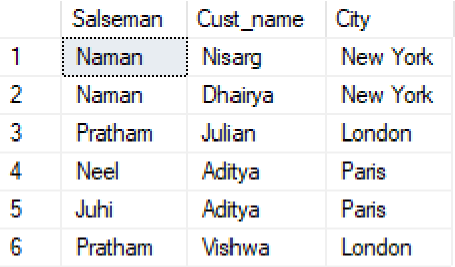
```
SELECT Salesman.[Name] AS Salseman, Customer.Cust_name, Customer.City
FROM Salesman,Customer
where  Customer.City = Salesman.City
```



	Salseman	Cust_name	City
1	Naman	Nisarg	New York
2	Naman	Dhairya	New York
3	Pratham	Julian	London
4	Neel	Aditya	Paris
5	Juhi	Aditya	Paris
6	Pratham	Vishwa	London

OR

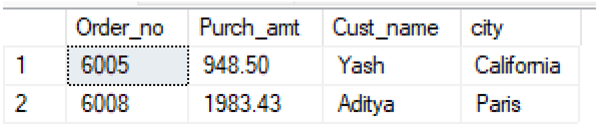
```
SELECT S.[Name] AS Salseman, C.Cust_name AS Customer_Name, C.City AS City
FROM Customer C
Inner Join Salesman S
ON S.City = C.City
```



	Salseman	Cust_name	City
1	Naman	Nisarg	New York
2	Naman	Dhairya	New York
3	Pratham	Julian	London
4	Neel	Aditya	Paris
5	Juhi	Aditya	Paris
6	Pratham	Vishwa	London

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

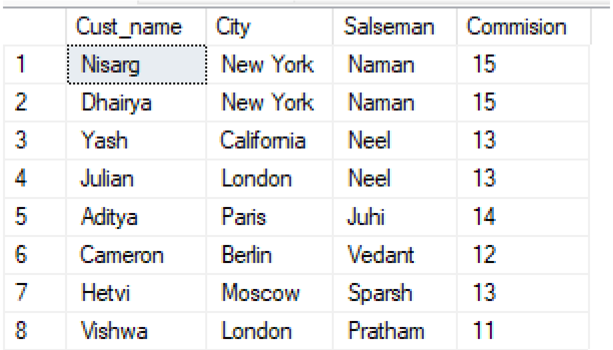
```
SELECT O.Order_no, O.Purch_amt, C.Cust_name, C.city
FROM Orders O
Inner Join Customer C
ON C.Customer_id = O.Customer_id
WHERE O.Purch_amt BETWEEN 500 AND 2000
```



	Order_no	Purch_amt	Cust_name	city
1	6005	948.50	Yash	California
2	6008	1983.43	Aditya	Paris

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

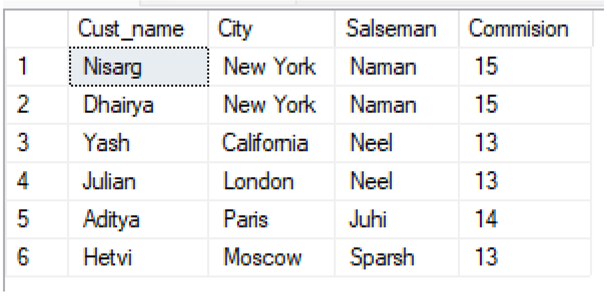
```
SELECT C.Cust_name, C.City,S.[name] AS Salseman,S.Commision
FROM Customer C
Inner Join Salesman S
ON S.Salesman_id = C.Salesman_id
```



	Cust_name	City	Salseman	Commision
1	Nisarg	New York	Naman	15
2	Dhairya	New York	Naman	15
3	Yash	California	Neel	13
4	Julian	London	Neel	13
5	Aditya	Paris	Juhi	14
6	Cameron	Berlin	Vedant	12
7	Hetvi	Moscow	Sparsh	13
8	Vishwa	London	Pratham	11

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

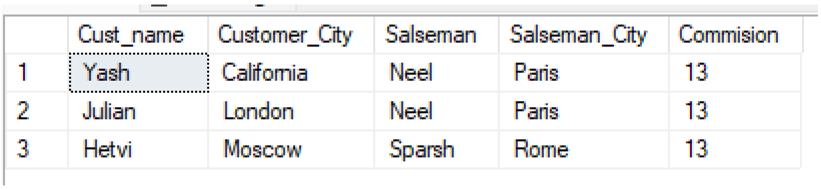
```
SELECT C.Cust_name, C.City,S.[name] AS Salseman ,S.Commision
FROM Customer C
Inner Join Salesman S
ON S.Salesman_id = C.Salesman_id
WHERE Commision > 12
```



	Cust_name	City	Salseman	Commision
1	Nisarg	New York	Naman	15
2	Dhairya	New York	Naman	15
3	Yash	California	Neel	13
4	Julian	London	Neel	13
5	Aditya	Paris	Juhi	14
6	Hetvi	Moscow	Sparsh	13

Query 5: write a SQL query to locate those salespeople who do not live in the same city where their customers live and have received a commission of more than 12% from the company. Return Customer Name, customer city, Salesman, salesman city, commission

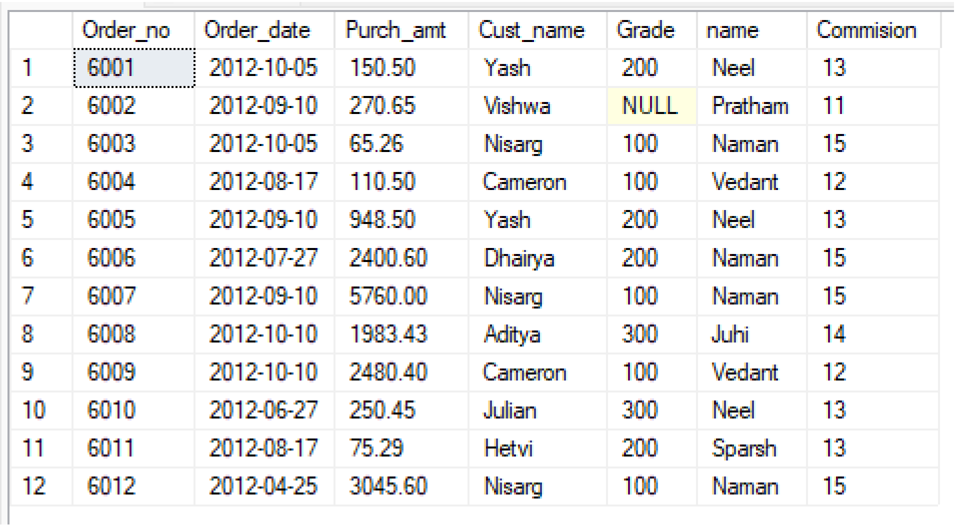
```
SELECT  C.Cust_name, C.City As Customer_City ,S.[name] AS Salseman,
S.City As Salseman_City, S.Commision
FROM Customer C
Inner Join Salesman S
ON S.Salesman_id = C.Salesman_id
WHERE S.city <> C.city AND Commision > 12
```



	Cust_name	Customer_City	Salseman	Salseman_City	Commision
1	Yash	California	Neel	Paris	13
2	Julian	London	Neel	Paris	13
3	Hetvi	Moscow	Sparsh	Rome	13

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

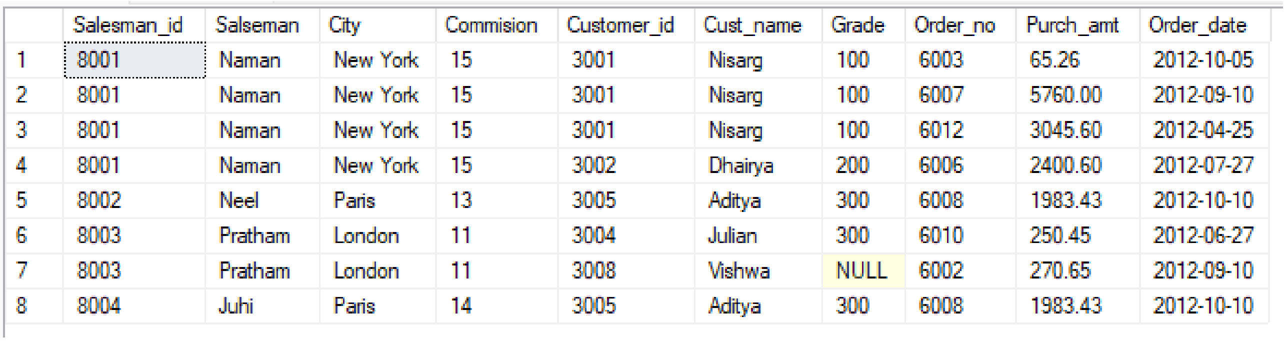
```
SELECT O.Order_no, O.Order_date, O.Purch_amt,
C.Cust_name, C.Grade, S.[name], S.Commision
FROM ((Orders O
Inner Join Customer C ON C.Customer_id = O.Customer_id )
Inner Join Salesman S ON S.Salesman_id = O.Salesman_id)
```



	Order_no	Order_date	Purch_amt	Cust_name	Grade	name	Commision
1	6001	2012-10-05	150.50	Yash	200	Neel	13
2	6002	2012-09-10	270.65	Vishwa	NULL	Pratham	11
3	6003	2012-10-05	65.26	Nisarg	100	Naman	15
4	6004	2012-08-17	110.50	Cameron	100	Vedant	12
5	6005	2012-09-10	948.50	Yash	200	Neel	13
6	6006	2012-07-27	2400.60	Dhairya	200	Naman	15
7	6007	2012-09-10	5760.00	Nisarg	100	Naman	15
8	6008	2012-10-10	1983.43	Aditya	300	Juhi	14
9	6009	2012-10-10	2480.40	Cameron	100	Vedant	12
10	6010	2012-06-27	250.45	Julian	300	Neel	13
11	6011	2012-08-17	75.29	Hetvi	200	Sparsh	13
12	6012	2012-04-25	3045.60	Nisarg	100	Naman	15

Query 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.

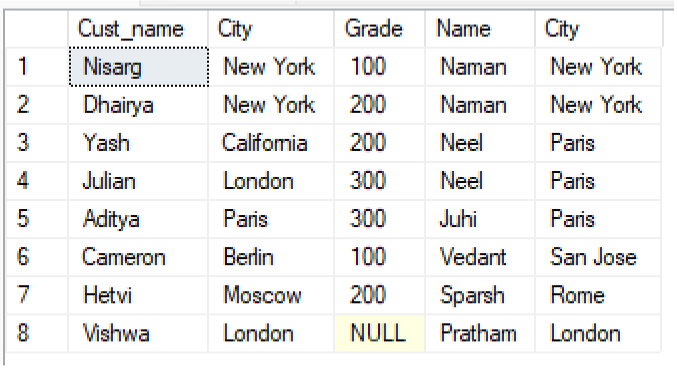
```
SELECT S.Salesman_id, S.[Name] AS Salseman, S.City, S.Commision,
C.Customer_id, C.Cust_name, C.Grade,
O.Order_no, O.Purch_amt, O.Order_date
FROM Orders O
Join Customer C ON C.Customer_id = O.Customer_id
Join Salesman S ON S.City = C.city
Order By S.Salesman_id ASC
```



	Salesman_id	Salseman	City	Commision	Customer_id	Cust_name	Grade	Order_no	Purch_amt	Order_date
1	8001	Naman	New York	15	3001	Nisarg	100	6003	65.26	2012-10-05
2	8001	Naman	New York	15	3001	Nisarg	100	6007	5760.00	2012-09-10
3	8001	Naman	New York	15	3001	Nisarg	100	6012	3045.60	2012-04-25
4	8001	Naman	New York	15	3002	Dhairya	200	6006	2400.60	2012-07-27
5	8002	Neel	Paris	13	3005	Aditya	300	6008	1983.43	2012-10-10
6	8003	Pratham	London	11	3004	Julian	300	6010	250.45	2012-06-27
7	8003	Pratham	London	11	3008	Vishwa	NULL	6002	270.65	2012-09-10
8	8004	Juhi	Paris	14	3005	Aditya	300	6008	1983.43	2012-10-10

Query 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.

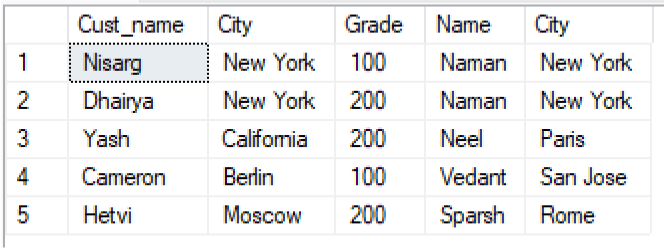
```
SELECT C.Cust_name, C.City, C.Grade,
S.[Name], S.City
FROM Customer C
Inner Join Salesman S
ON S.Salesman_id = C.Salesman_id
Order By C.Customer_id ASC
```



	Cust_name	City	Grade	Name	City
1	Nisarg	New York	100	Naman	New York
2	Dhairya	New York	200	Naman	New York
3	Yash	California	200	Neel	Paris
4	Julian	London	300	Neel	Paris
5	Aditya	Paris	300	Juhi	Paris
6	Cameron	Berlin	100	Vedant	San Jose
7	Hetvi	Moscow	200	Sparsh	Rome
8	Vishwa	London	NULL	Pratham	London

Query 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.

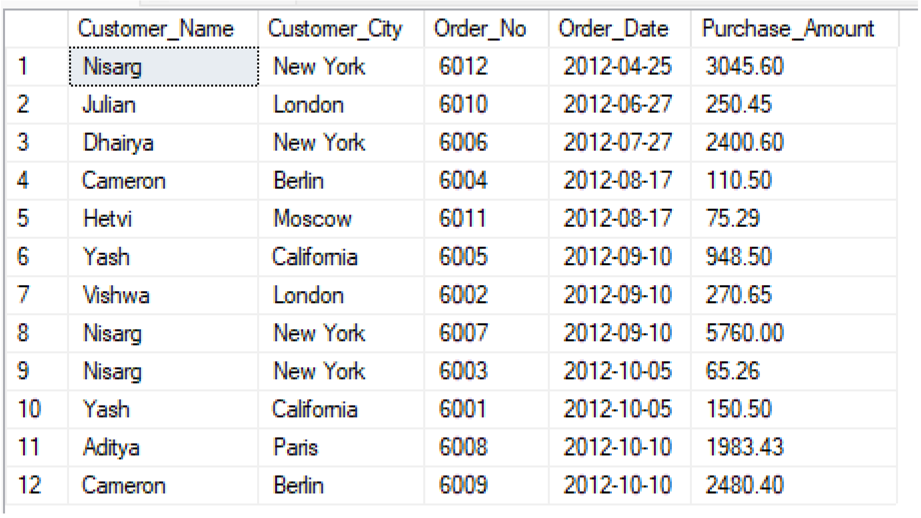
```
SELECT C.Cust_name, C.City, C.Grade,
S.[Name], S.City
FROM Customer C
Inner Join Salesman S
ON S.Salesman_id = C.Salesman_id
where C.Grade < 300
Order By C.Customer_id ASC
```



	Cust_name	City	Grade	Name	City
1	Nisarg	New York	100	Naman	New York
2	Dhairya	New York	200	Naman	New York
3	Yash	California	200	Neel	Paris
4	Cameron	Berlin	100	Vedant	San Jose
5	Hetvi	Moscow	200	Sparsh	Rome

Query 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

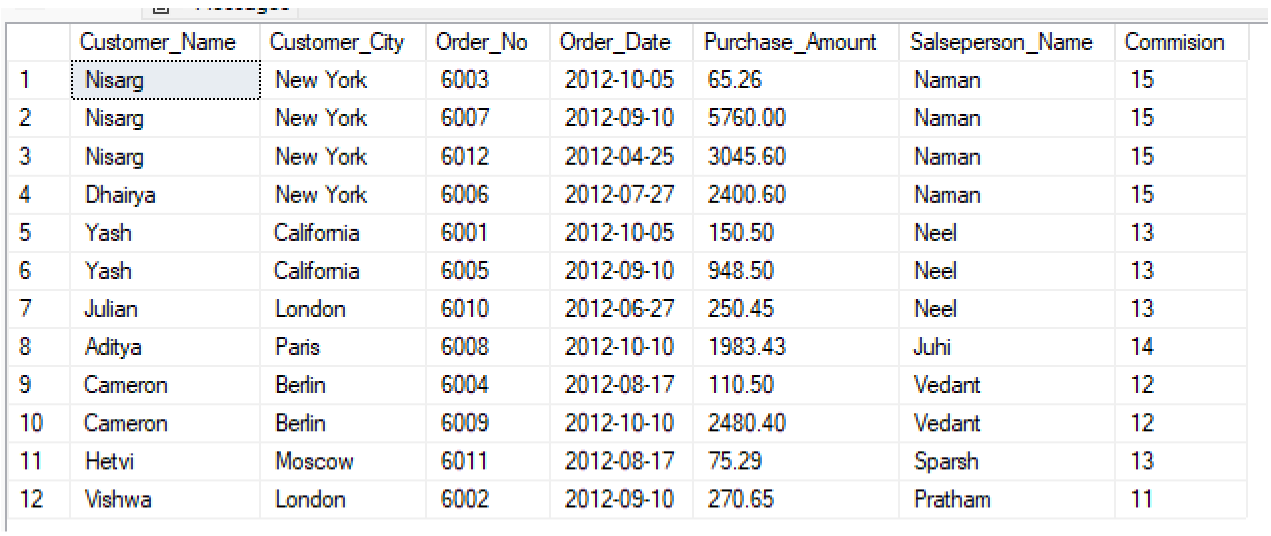
```
SELECT C.Cust_name AS Customer_Name, C.City AS Customer_City,
O.Order_No, O.Order_Date, O.Purch_amt AS Purchase_Amount
FROM Orders O
Left Outer Join Customer C
ON C.Customer_id = O.Customer_id
Order By O.Order_date
```



	Customer_Name	Customer_City	Order_No	Order_Date	Purchase_Amount
1	Nisarg	New York	6012	2012-04-25	3045.60
2	Julian	London	6010	2012-06-27	250.45
3	Dhairya	New York	6006	2012-07-27	2400.60
4	Cameron	Berlin	6004	2012-08-17	110.50
5	Hetvi	Moscow	6011	2012-08-17	75.29
6	Yash	California	6005	2012-09-10	948.50
7	Vishwa	London	6002	2012-09-10	270.65
8	Nisarg	New York	6007	2012-09-10	5760.00
9	Nisarg	New York	6003	2012-10-05	65.26
10	Yash	California	6001	2012-10-05	150.50
11	Aditya	Paris	6008	2012-10-10	1983.43
12	Cameron	Berlin	6009	2012-10-10	2480.40

Query 11: Write a SQL statement to generate a report with customer name, city, order number, order date, order amount, salesperson name, and commission to determine if any of the existing customers have not placed orders or if they have placed orders through their salesman or by themselves

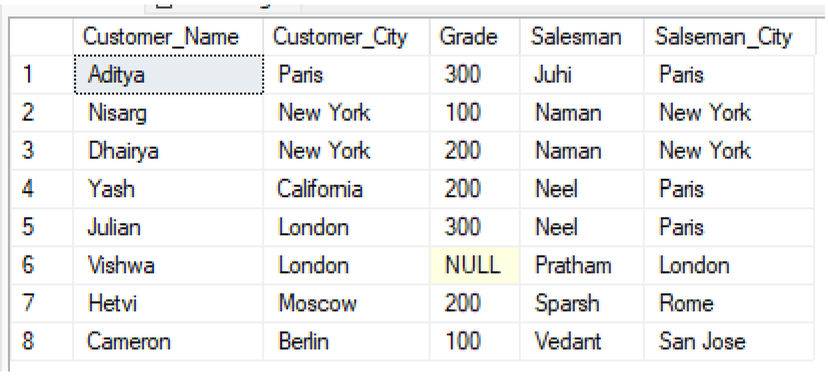
```
SELECT C.Cust_name AS Customer_Name, C.City AS Customer_City,
O.Order_No, O.Order_Date, O.Purch_amt AS Purchase_Amount,
S.[Name] AS Salseperson_Name,S.Commision
FROM Customer C
LEFT OUTER JOIN Orders O
ON O.Customer_id = C.Customer_id
LEFT OUTER JOIN Salesman S
ON S.Salesman_id=O.Salesman_id
```



	Customer_Name	Customer_City	Order_No	Order_Date	Purchase_Amount	Salseperson_Name	Commision
1	Nisarg	New York	6003	2012-10-05	65.26	Naman	15
2	Nisarg	New York	6007	2012-09-10	5760.00	Naman	15
3	Nisarg	New York	6012	2012-04-25	3045.60	Naman	15
4	Dhairya	New York	6006	2012-07-27	2400.60	Naman	15
5	Yash	California	6001	2012-10-05	150.50	Neel	13
6	Yash	California	6005	2012-09-10	948.50	Neel	13
7	Julian	London	6010	2012-06-27	250.45	Neel	13
8	Aditya	Paris	6008	2012-10-10	1983.43	Juhi	14
9	Cameron	Berlin	6004	2012-08-17	110.50	Vedant	12
10	Cameron	Berlin	6009	2012-10-10	2480.40	Vedant	12
11	Hetvi	Moscow	6011	2012-08-17	75.29	Sparsh	13
12	Vishwa	London	6002	2012-09-10	270.65	Pratham	11

Query 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

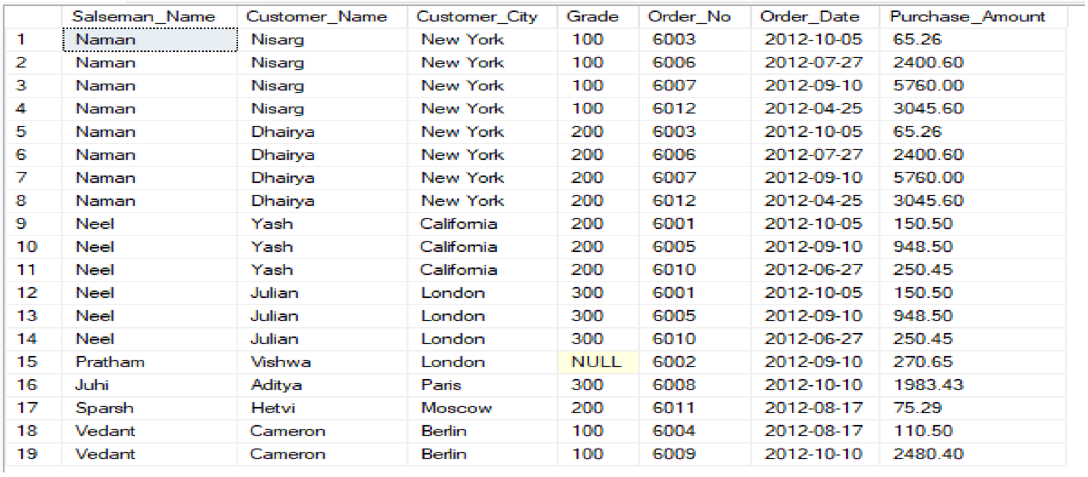
```
SELECT C.Cust_name AS Customer_Name,C.City AS Customer_City, C.Grade,
S.[Name] AS "Salesman", S.City AS Salesman_City
FROM Customer C
RIGHT OUTER JOIN Salesman S
ON S.Salesman_id = C.Salesman_id
ORDER BY S.Name;
```



	Customer_Name	Customer_City	Grade	Salesman	Salseman_City
1	Aditya	Paris	300	Juhi	Paris
2	Nisarg	New York	100	Naman	New York
3	Dhairya	New York	200	Naman	New York
4	Yash	California	200	Neel	Paris
5	Julian	London	300	Neel	Paris
6	Vishwa	London	NULL	Pratham	London
7	Hetvi	Moscow	200	Sparsh	Rome
8	Cameron	Berlin	100	Vedant	San Jose

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

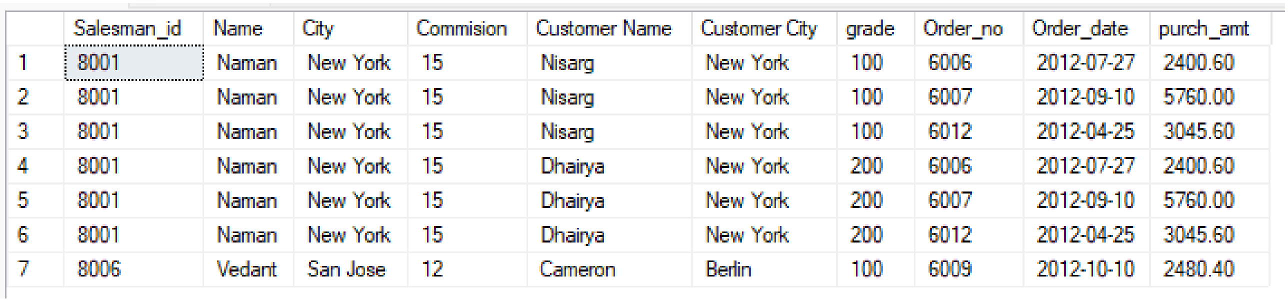
```
SELECT S.[Name] As Salseman_Name,
C.Cust_name AS Customer_Name, C.City AS Customer_City,
C.Grade, O.Order_No, O.Order_Date, O.Purch_amt AS Purchase_Amount
FROM ((Salesman S
lEFT OUTER JOIN Customer C ON C.Salesman_id = S.Salesman_id )
LEFT OUTER JOIN Orders O ON S.Salesman_id = O.Salesman_id)
```



	Salseman_Name	Customer_Name	Customer_City	Grade	Order_No	Order_Date	Purchase_Amount
1	Naman	Nisarg	New York	100	6003	2012-10-05	65.26
2	Naman	Nisarg	New York	100	6006	2012-07-27	2400.60
3	Naman	Nisarg	New York	100	6007	2012-09-10	5760.00
4	Naman	Nisarg	New York	100	6012	2012-04-25	3045.60
5	Naman	Dhairya	New York	200	6003	2012-10-05	65.26
6	Naman	Dhairya	New York	200	6006	2012-07-27	2400.60
7	Naman	Dhairya	New York	200	6007	2012-09-10	5760.00
8	Naman	Dhairya	New York	200	6012	2012-04-25	3045.60
9	Neel	Yash	California	200	6001	2012-10-05	150.50
10	Neel	Yash	California	200	6005	2012-09-10	948.50
11	Neel	Yash	California	200	6010	2012-06-27	250.45
12	Neel	Julian	London	300	6001	2012-10-05	150.50
13	Neel	Julian	London	300	6005	2012-09-10	948.50
14	Neel	Julian	London	300	6010	2012-06-27	250.45
15	Pratham	Vishwa	London	NULL	6002	2012-09-10	270.65
16	Juhi	Aditya	Paris	300	6008	2012-10-10	1983.43
17	Sparsh	Hetvi	Moscow	200	6011	2012-08-17	75.29
18	Vedant	Cameron	Berlin	100	6004	2012-08-17	110.50
19	Vedant	Cameron	Berlin	100	6009	2012-10-10	2480.40

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

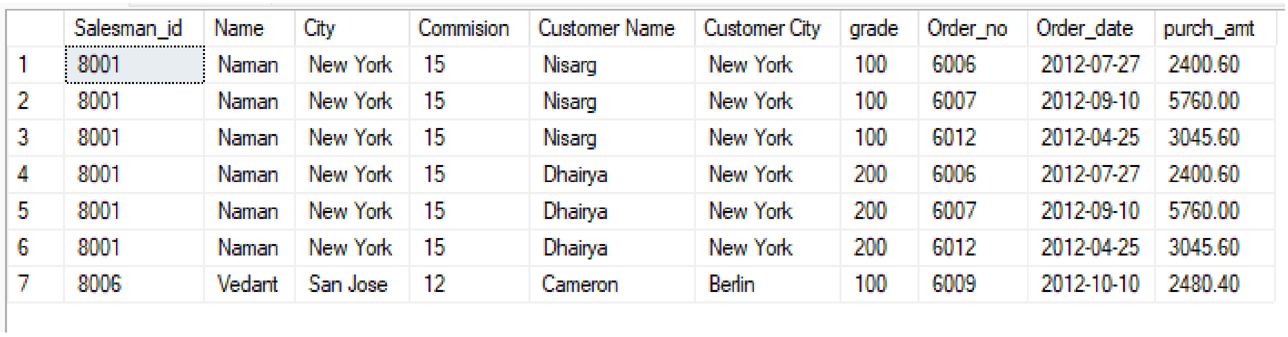
```
SELECT S.* ,
C.cust_name as "Customer Name", C.city as "Customer City", C.grade,
O.Order_no, O.Order_date, O.purch_amt
FROM ((salesman S LEFT OUTER JOIN customer C ON S.salesman_id =
C.salesman_id )
LEFT OUTER JOIN orders O ON S.salesman_id = O.salesman_id )
WHERE O.purch_amt > 2000 AND C.grade IS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



	Salesman_id	Name	City	Commision	Customer Name	Customer City	grade	Order_no	Order_date	purch_amt
1	8001	Naman	New York	15	Nisarg	New York	100	6006	2012-07-27	2400.60
2	8001	Naman	New York	15	Nisarg	New York	100	6007	2012-09-10	5760.00
3	8001	Naman	New York	15	Nisarg	New York	100	6012	2012-04-25	3045.60
4	8001	Naman	New York	15	Dhairya	New York	200	6006	2012-07-27	2400.60
5	8001	Naman	New York	15	Dhairya	New York	200	6007	2012-09-10	5760.00
6	8001	Naman	New York	15	Dhairya	New York	200	6012	2012-04-25	3045.60
7	8006	Vedant	San Jose	12	Cameron	Berlin	100	6009	2012-10-10	2480.40

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

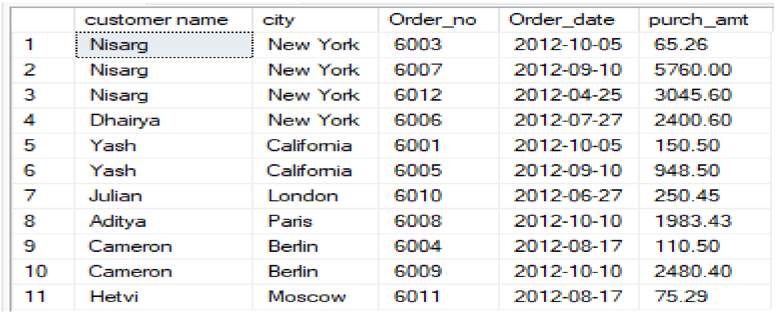
```
SELECT S.* ,
C.cust_name as "Customer Name", C.city as "Customer City", C.grade,
O.Order_no, O.Order_date, O.purch_amt
FROM ((salesman S LEFT OUTER JOIN customer C ON S.salesman_id =
C.salesman_id )
LEFT OUTER JOIN orders O ON S.salesman_id = O.salesman_id )
WHERE O.purch_amt > 2000 AND C.grade IS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



	Salesman_id	Name	City	Commision	Customer Name	Customer City	grade	Order_no	Order_date	purch_amt
1	8001	Naman	New York	15	Nisarg	New York	100	6006	2012-07-27	2400.60
2	8001	Naman	New York	15	Nisarg	New York	100	6007	2012-09-10	5760.00
3	8001	Naman	New York	15	Nisarg	New York	100	6012	2012-04-25	3045.60
4	8001	Naman	New York	15	Dhairya	New York	200	6006	2012-07-27	2400.60
5	8001	Naman	New York	15	Dhairya	New York	200	6007	2012-09-10	5760.00
6	8001	Naman	New York	15	Dhairya	New York	200	6012	2012-04-25	3045.60
7	8006	Vedant	San Jose	12	Cameron	Berlin	100	6009	2012-10-10	2480.40

Query 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.

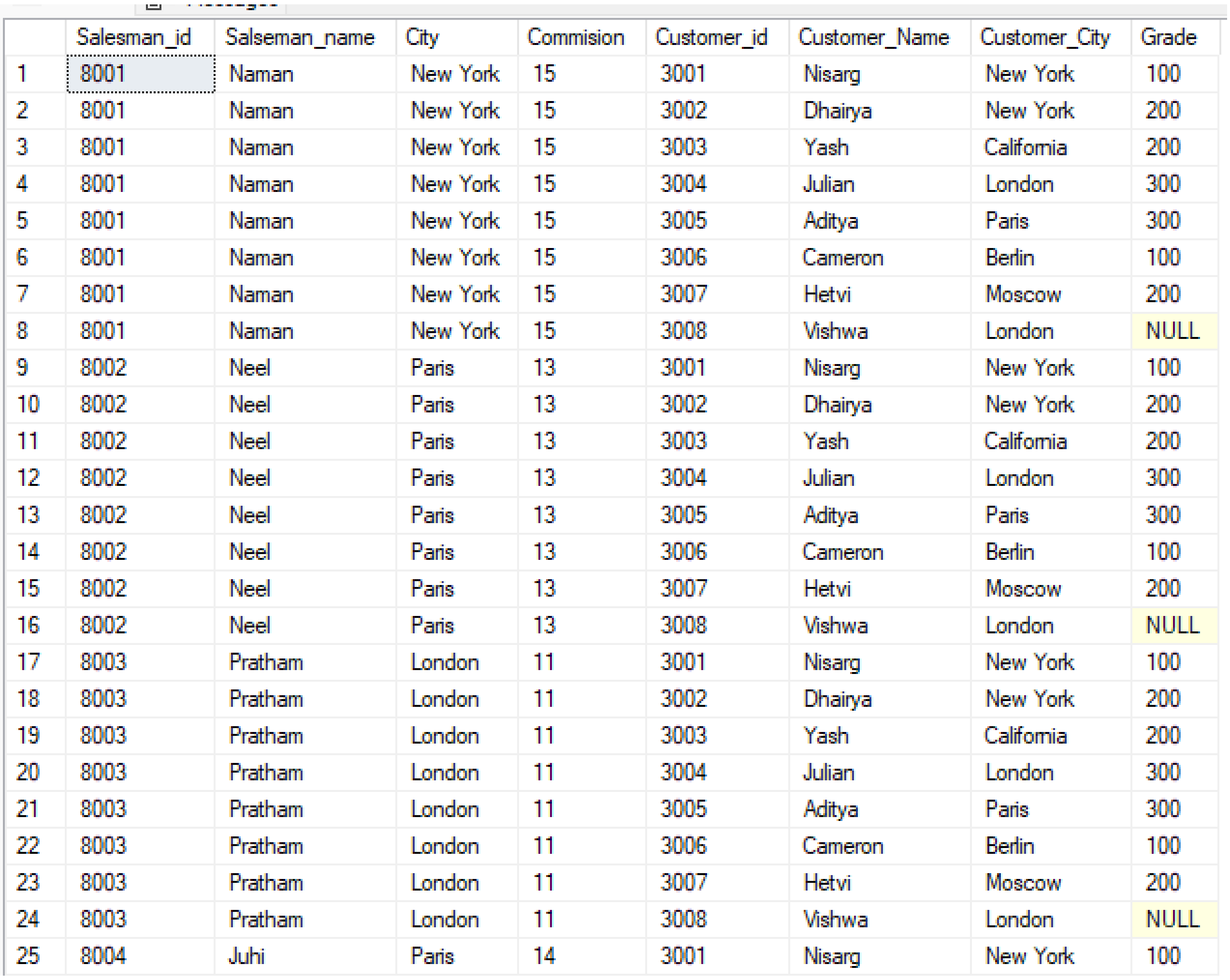
```
SELECT C.cust_name AS "customer name", C.city,
O.Order_no, O.Order_date, O.purch_amt
FROM customer C FULL OUTER JOIN orders O ON C.customer_id =
O.customer_id
WHERE C.grade IS NOT NULL
```



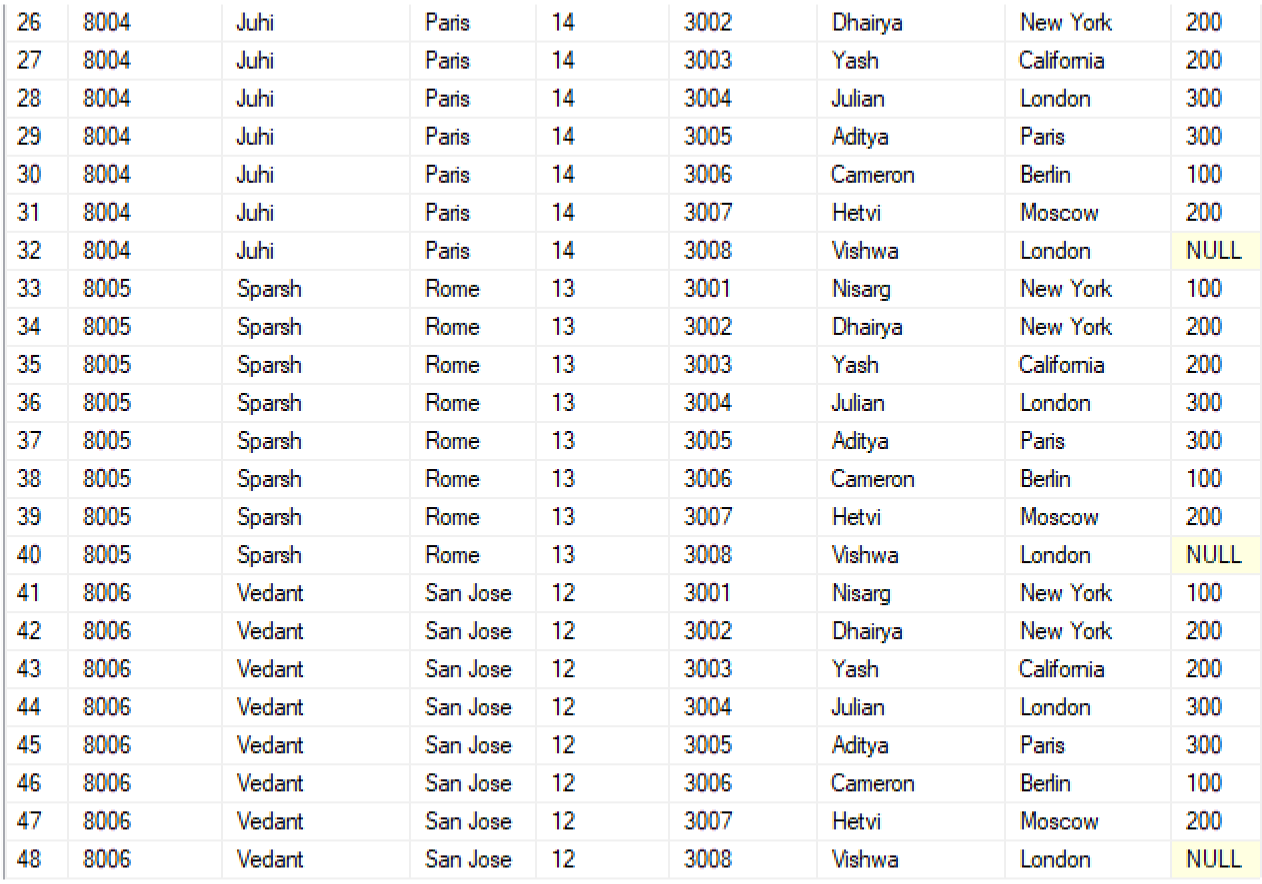
	customer name	city	Order_no	Order_date	purch_amt
1	Nisarg	New York	6003	2012-10-05	65.26
2	Nisarg	New York	6007	2012-09-10	5760.00
3	Nisarg	New York	6012	2012-04-25	3045.60
4	Dhairya	New York	6006	2012-07-27	2400.60
5	Yash	California	6001	2012-10-05	150.50
6	Yash	California	6005	2012-09-10	948.50
7	Julian	London	6010	2012-06-27	250.45
8	Aditya	Paris	6008	2012-10-10	1983.43
9	Cameron	Berlin	6004	2012-08-17	110.50
10	Cameron	Berlin	6009	2012-10-10	2480.40
11	Hetvi	Moscow	6011	2012-08-17	75.29

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

```
SELECT S.Salesman_id, S.[Name] AS Salseman_name, S.City, S.Commision,
C.Customer_id, C.Cust_name AS Customer_Name,C.City AS Customer_City,
C.Grade
FROM Salesman s
CROSS JOIN Customer C
```



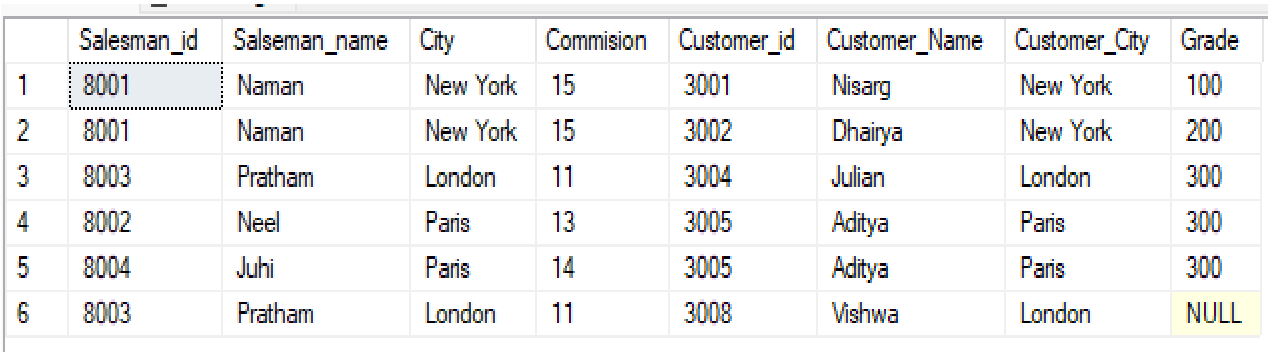
	Salesman_id	Salseman_name	City	Commision	Customer_id	Customer_Name	Customer_City	Grade
1	8001	Naman	New York	15	3001	Nisarg	New York	100
2	8001	Naman	New York	15	3002	Dhairya	New York	200
3	8001	Naman	New York	15	3003	Yash	California	200
4	8001	Naman	New York	15	3004	Julian	London	300
5	8001	Naman	New York	15	3005	Aditya	Paris	300
6	8001	Naman	New York	15	3006	Cameron	Berlin	100
7	8001	Naman	New York	15	3007	Hetvi	Moscow	200
8	8001	Naman	New York	15	3008	Vishwa	London	NULL
9	8002	Neel	Paris	13	3001	Nisarg	New York	100
10	8002	Neel	Paris	13	3002	Dhairya	New York	200
11	8002	Neel	Paris	13	3003	Yash	California	200
12	8002	Neel	Paris	13	3004	Julian	London	300
13	8002	Neel	Paris	13	3005	Aditya	Paris	300
14	8002	Neel	Paris	13	3006	Cameron	Berlin	100
15	8002	Neel	Paris	13	3007	Hetvi	Moscow	200
16	8002	Neel	Paris	13	3008	Vishwa	London	NULL
17	8003	Pratham	London	11	3001	Nisarg	New York	100
18	8003	Pratham	London	11	3002	Dhairya	New York	200
19	8003	Pratham	London	11	3003	Yash	California	200
20	8003	Pratham	London	11	3004	Julian	London	300
21	8003	Pratham	London	11	3005	Aditya	Paris	300
22	8003	Pratham	London	11	3006	Cameron	Berlin	100
23	8003	Pratham	London	11	3007	Hetvi	Moscow	200
24	8003	Pratham	London	11	3008	Vishwa	London	NULL
25	8004	Juhi	Paris	14	3001	Nisarg	New York	100



26	8004	Juhi	Paris	14	3002	Dhairya	New York	200
27	8004	Juhi	Paris	14	3003	Yash	California	200
28	8004	Juhi	Paris	14	3004	Julian	London	300
29	8004	Juhi	Paris	14	3005	Aditya	Paris	300
30	8004	Juhi	Paris	14	3006	Cameron	Berlin	100
31	8004	Juhi	Paris	14	3007	Hetvi	Moscow	200
32	8004	Juhi	Paris	14	3008	Vishwa	London	NULL
33	8005	Sparsh	Rome	13	3001	Nisarg	New York	100
34	8005	Sparsh	Rome	13	3002	Dhairya	New York	200
35	8005	Sparsh	Rome	13	3003	Yash	California	200
36	8005	Sparsh	Rome	13	3004	Julian	London	300
37	8005	Sparsh	Rome	13	3005	Aditya	Paris	300
38	8005	Sparsh	Rome	13	3006	Cameron	Berlin	100
39	8005	Sparsh	Rome	13	3007	Hetvi	Moscow	200
40	8005	Sparsh	Rome	13	3008	Vishwa	London	NULL
41	8006	Vedant	San Jose	12	3001	Nisarg	New York	100
42	8006	Vedant	San Jose	12	3002	Dhairya	New York	200
43	8006	Vedant	San Jose	12	3003	Yash	California	200
44	8006	Vedant	San Jose	12	3004	Julian	London	300
45	8006	Vedant	San Jose	12	3005	Aditya	Paris	300
46	8006	Vedant	San Jose	12	3006	Cameron	Berlin	100
47	8006	Vedant	San Jose	12	3007	Hetvi	Moscow	200
48	8006	Vedant	San Jose	12	3008	Vishwa	London	NULL

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

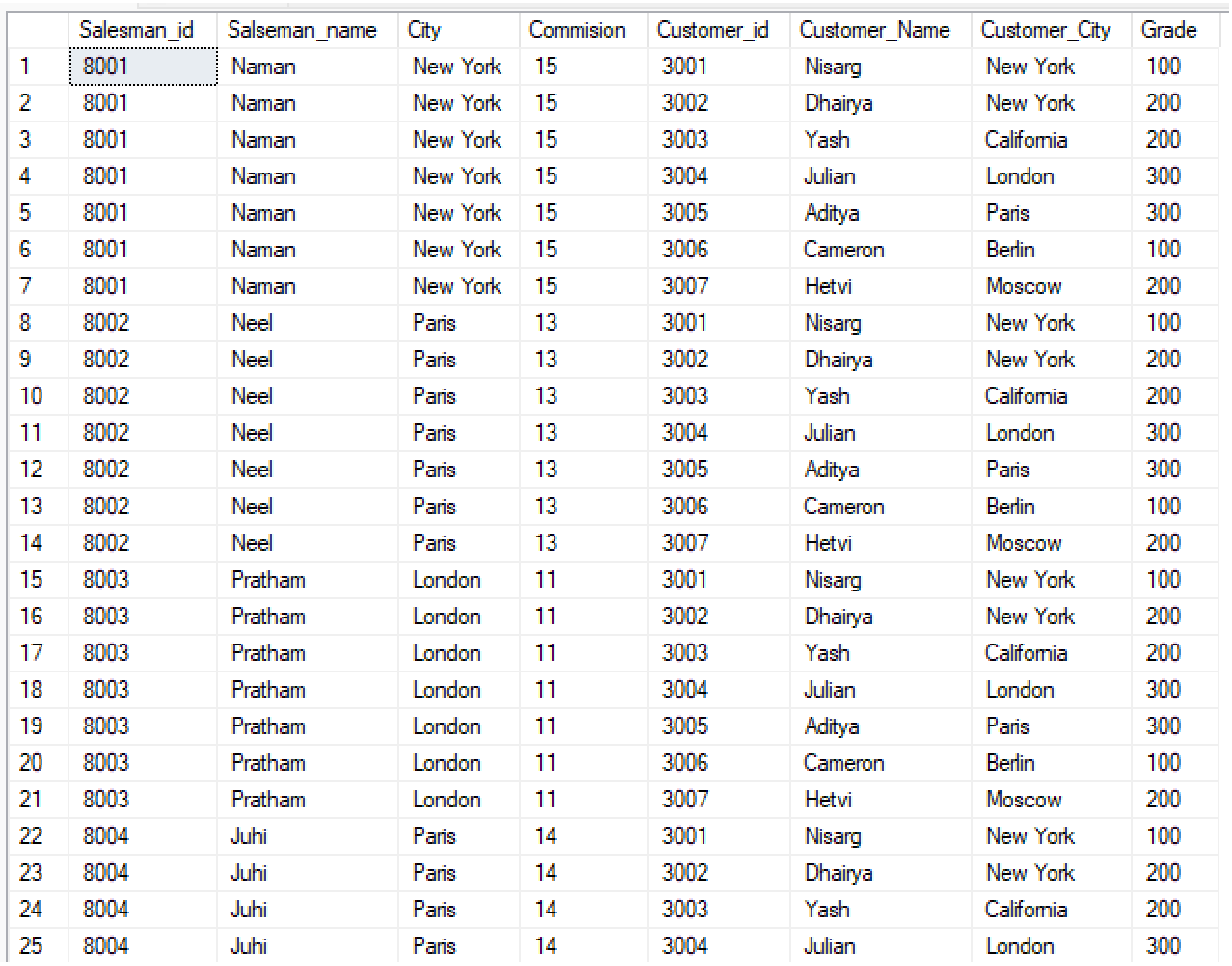
```
SELECT S.Salesman_id, S.[Name] AS Salesman_name, S.City, S.Commision,
C.Customer_id, C.Cust_name AS Customer_Name,C.City AS Customer_City,
C.Grade
FROM Salesman s
CROSS JOIN Customer C
WHERE C.City = S.City
```



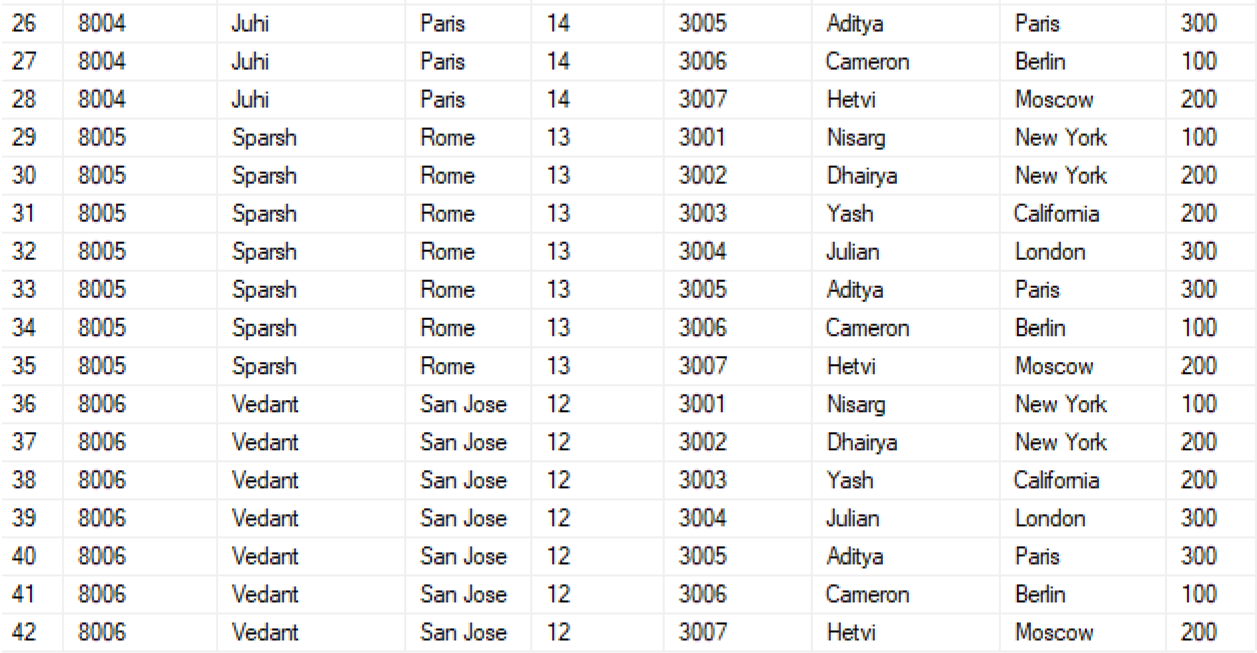
	Salesman_id	Salesman_name	City	Commision	Customer_id	Customer_Name	Customer_City	Grade
1	8001	Naman	New York	15	3001	Nisarg	New York	100
2	8001	Naman	New York	15	3002	Dhairya	New York	200
3	8003	Pratham	London	11	3004	Julian	London	300
4	8002	Neel	Paris	13	3005	Aditya	Paris	300
5	8004	Juhi	Paris	14	3005	Aditya	Paris	300
6	8003	Pratham	London	11	3008	Vishwa	London	NULL

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

```
SELECT S.Salesman_id, S.[Name] AS Salesman_name, S.City, S.Commision,
C.Customer_id, C.Cust_name AS Customer_Name,C.City AS Customer_City,
C.Grade
FROM Salesman s
CROSS JOIN Customer C
WHERE C.City IS NOT NULL
AND C.Grade IS NOT NULL;
```



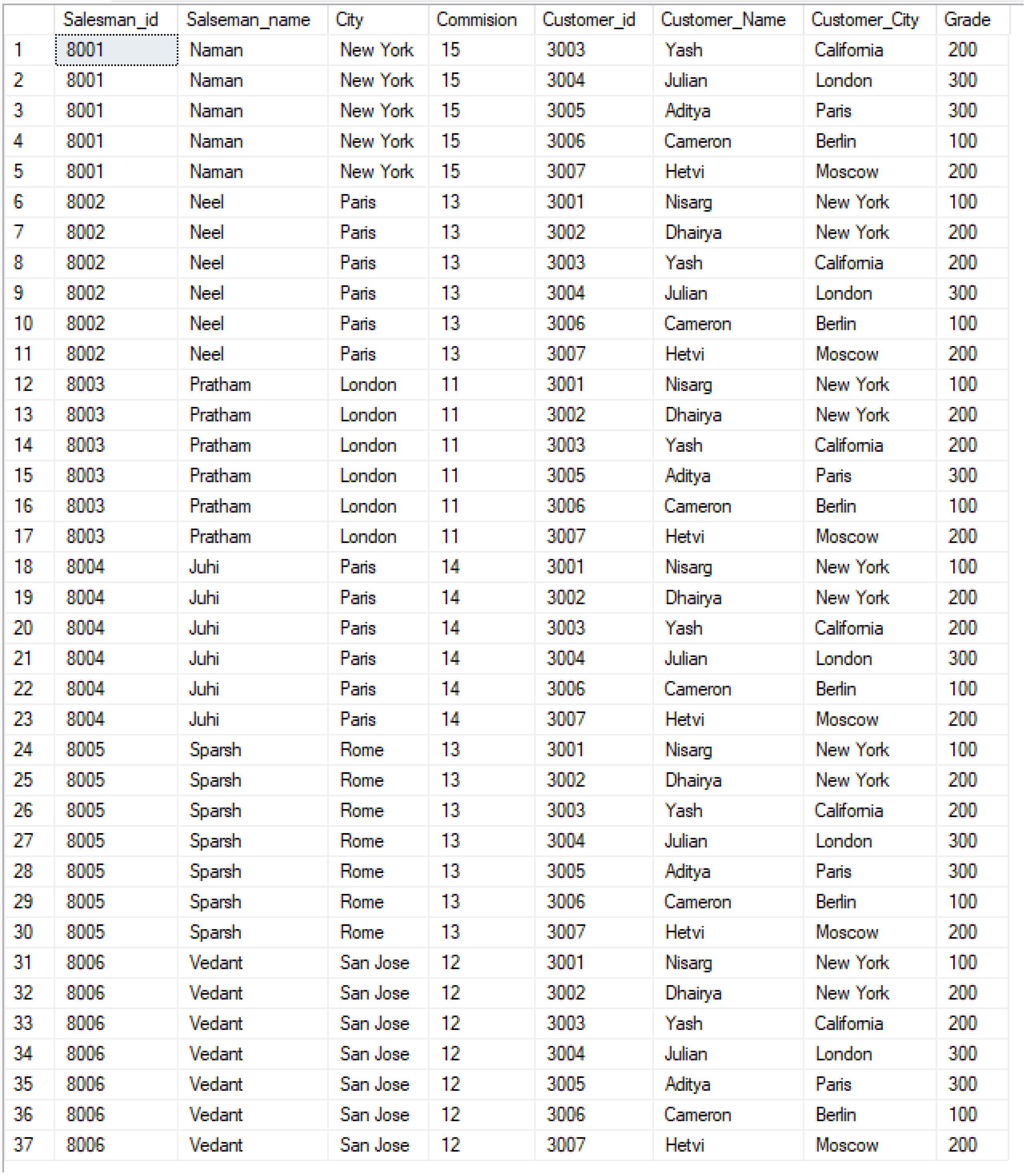
	Salesman_id	Salesman_name	City	Commision	Customer_id	Customer_Name	Customer_City	Grade
1	8001	Naman	New York	15	3001	Nisarg	New York	100
2	8001	Naman	New York	15	3002	Dhairya	New York	200
3	8001	Naman	New York	15	3003	Yash	California	200
4	8001	Naman	New York	15	3004	Julian	London	300
5	8001	Naman	New York	15	3005	Aditya	Paris	300
6	8001	Naman	New York	15	3006	Cameron	Berlin	100
7	8001	Naman	New York	15	3007	Hetvi	Moscow	200
8	8002	Neel	Paris	13	3001	Nisarg	New York	100
9	8002	Neel	Paris	13	3002	Dhairya	New York	200
10	8002	Neel	Paris	13	3003	Yash	California	200
11	8002	Neel	Paris	13	3004	Julian	London	300
12	8002	Neel	Paris	13	3005	Aditya	Paris	300
13	8002	Neel	Paris	13	3006	Cameron	Berlin	100
14	8002	Neel	Paris	13	3007	Hetvi	Moscow	200
15	8003	Pratham	London	11	3001	Nisarg	New York	100
16	8003	Pratham	London	11	3002	Dhairya	New York	200
17	8003	Pratham	London	11	3003	Yash	California	200
18	8003	Pratham	London	11	3004	Julian	London	300
19	8003	Pratham	London	11	3005	Aditya	Paris	300
20	8003	Pratham	London	11	3006	Cameron	Berlin	100
21	8003	Pratham	London	11	3007	Hetvi	Moscow	200
22	8004	Juhi	Paris	14	3001	Nisarg	New York	100
23	8004	Juhi	Paris	14	3002	Dhairya	New York	200
24	8004	Juhi	Paris	14	3003	Yash	California	200
25	8004	Juhi	Paris	14	3004	Julian	London	300



26	8004	Juhi	Paris	14	3005	Aditya	Paris	300
27	8004	Juhi	Paris	14	3006	Cameron	Berlin	100
28	8004	Juhi	Paris	14	3007	Hetvi	Moscow	200
29	8005	Sparsh	Rome	13	3001	Nisarg	New York	100
30	8005	Sparsh	Rome	13	3002	Dhairya	New York	200
31	8005	Sparsh	Rome	13	3003	Yash	California	200
32	8005	Sparsh	Rome	13	3004	Julian	London	300
33	8005	Sparsh	Rome	13	3005	Aditya	Paris	300
34	8005	Sparsh	Rome	13	3006	Cameron	Berlin	100
35	8005	Sparsh	Rome	13	3007	Hetvi	Moscow	200
36	8006	Vedant	San Jose	12	3001	Nisarg	New York	100
37	8006	Vedant	San Jose	12	3002	Dhairya	New York	200
38	8006	Vedant	San Jose	12	3003	Yash	California	200
39	8006	Vedant	San Jose	12	3004	Julian	London	300
40	8006	Vedant	San Jose	12	3005	Aditya	Paris	300
41	8006	Vedant	San Jose	12	3006	Cameron	Berlin	100
42	8006	Vedant	San Jose	12	3007	Hetvi	Moscow	200

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

```
SELECT S.Salesman_id, S.[Name] AS Salseman_name, S.City, S.Commision,
C.Customer_id, C.Cust_name AS Customer_Name,C.City AS Customer_City,
C.Grade
FROM Salesman s
CROSS JOIN Customer C
WHERE C.City <> S.City
AND C.Grade IS NOT NULL;
```



	Salesman_id	Salseman_name	City	Commision	Customer_id	Customer_Name	Customer_City	Grade
1	8001	Naman	New York	15	3003	Yash	California	200
2	8001	Naman	New York	15	3004	Julian	London	300
3	8001	Naman	New York	15	3005	Aditya	Paris	300
4	8001	Naman	New York	15	3006	Cameron	Berlin	100
5	8001	Naman	New York	15	3007	Hetvi	Moscow	200
6	8002	Neel	Paris	13	3001	Nisarg	New York	100
7	8002	Neel	Paris	13	3002	Dhairya	New York	200
8	8002	Neel	Paris	13	3003	Yash	California	200
9	8002	Neel	Paris	13	3004	Julian	London	300
10	8002	Neel	Paris	13	3006	Cameron	Berlin	100
11	8002	Neel	Paris	13	3007	Hetvi	Moscow	200
12	8003	Pratham	London	11	3001	Nisarg	New York	100
13	8003	Pratham	London	11	3002	Dhairya	New York	200
14	8003	Pratham	London	11	3003	Yash	California	200
15	8003	Pratham	London	11	3005	Aditya	Paris	300
16	8003	Pratham	London	11	3006	Cameron	Berlin	100
17	8003	Pratham	London	11	3007	Hetvi	Moscow	200
18	8004	Juhi	Paris	14	3001	Nisarg	New York	100
19	8004	Juhi	Paris	14	3002	Dhairya	New York	200
20	8004	Juhi	Paris	14	3003	Yash	California	200
21	8004	Juhi	Paris	14	3004	Julian	London	300
22	8004	Juhi	Paris	14	3006	Cameron	Berlin	100
23	8004	Juhi	Paris	14	3007	Hetvi	Moscow	200
24	8005	Sparsh	Rome	13	3001	Nisarg	New York	100
25	8005	Sparsh	Rome	13	3002	Dhairya	New York	200
26	8005	Sparsh	Rome	13	3003	Yash	California	200
27	8005	Sparsh	Rome	13	3004	Julian	London	300
28	8005	Sparsh	Rome	13	3005	Aditya	Paris	300
29	8005	Sparsh	Rome	13	3006	Cameron	Berlin	100
30	8005	Sparsh	Rome	13	3007	Hetvi	Moscow	200
31	8006	Vedant	San Jose	12	3001	Nisarg	New York	100
32	8006	Vedant	San Jose	12	3002	Dhairya	New York	200
33	8006	Vedant	San Jose	12	3003	Yash	California	200
34	8006	Vedant	San Jose	12	3004	Julian	London	300
35	8006	Vedant	San Jose	12	3005	Aditya	Paris	300
36	8006	Vedant	San Jose	12	3006	Cameron	Berlin	100
37	8006	Vedant	San Jose	12	3007	Hetvi	Moscow	200