**SQL ASSIGNMENT – 2**

--> Creating Database Called **SalesDB**

**create** database SalesDB

USE SalesDB



-->Creating table named **Salesman**

**create** table Salesman

(

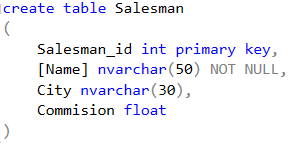
Salesman\_id int primary key,

[Name] nvarchar(50) NOT NULL,

City nvarchar(30),

Commision float

)



--> Creating table named **Customer**

**create** table Customer

(

Customer\_id int primary key,

Cust\_name nvarchar(50),

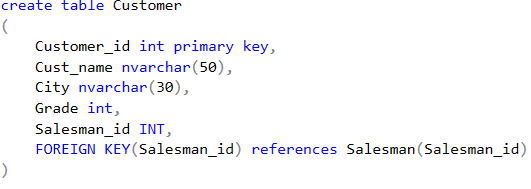
City nvarchar(30),

Grade int,

Salesman\_id INT,

FOREIGN KEY(Salesman\_id) references Salesman(Salesman\_id)

)



--> Creating table named **Orders**

create table Orders

(

Order\_no INT primary key,

Purch\_amt MONEY NOT NULL,

Order\_date date NOT NULL,

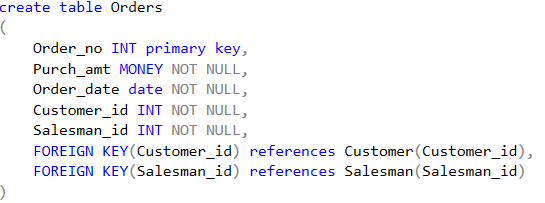
Customer\_id INT NOT NULL,

Salesman\_id INT NOT NULL,

FOREIGN KEY(Customer\_id) references Customer(Customer\_id),

FOREIGN KEY(Salesman\_id) references Salesman(Salesman\_id)

)



--> 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)

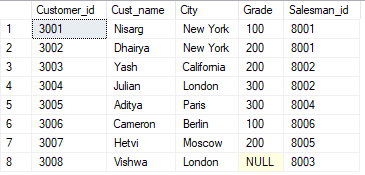
Insert into Orders values(6012, 3045.6, '2012-04-25', 3001, 8001)

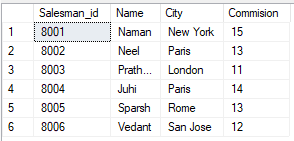
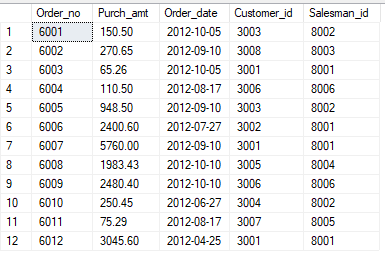
--> Viewing Inserted data:

select \* from Customer

select \* from Orders

select \* from Salesman



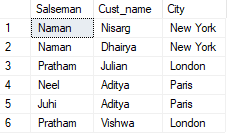


Query 1: write a SQL query to find the salesperson and customer who reside in the same city. Return Salesman, cust\_name and city.

SELECT Salesman.[Name] AS Salseman, Customer.Cust\_name, Customer.City

FROM Salesman,Customer

where Customer.City = Salesman.City



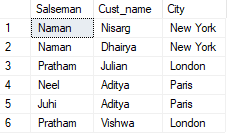
OR

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



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

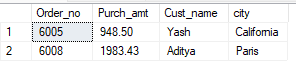
SELECT O.Order\_no, O.Purch\_amt, C.Cust\_name, C.city

FROM Orders O

Inner Join Customer C

ON C.Customer\_id = O.Customer\_id

WHERE O.Purch\_amt BETWEEN 500 AND 2000



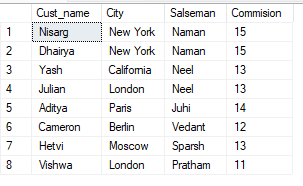
Query 3: write a SQL query to find the salesperson(s) and the customer(s) he represents. Return Customer Name, city, Salesman, commission

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



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.

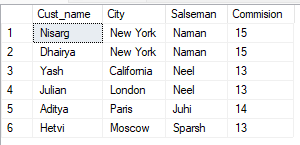
SELECT C.Cust\_name, C.City,S.[name] AS Salseman ,S.Commision

FROM Customer C

Inner Join Salesman S

ON S.Salesman\_id = C.Salesman\_id

WHERE Commision > 12



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,

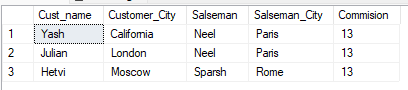
S.City As Salseman\_City, S.Commision

FROM Customer C

Inner Join Salesman S

ON S.Salesman\_id = C.Salesman\_id

WHERE S.city <> C.city AND Commision > 12



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

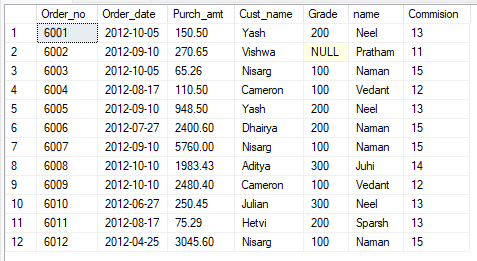
SELECT O.Order\_no, O.Order\_date, O.Purch\_amt,

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)



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,

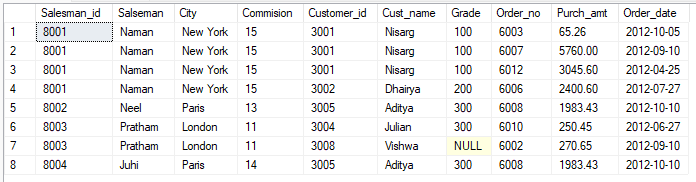
O.Order\_no, O.Purch\_amt, O.Order\_date

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



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,

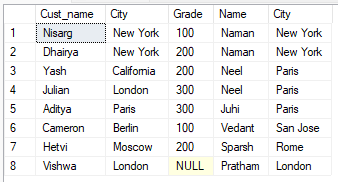
S.[Name], S.City

FROM Customer C

Inner Join Salesman S

ON S.Salesman\_id = C.Salesman\_id

Order By C.Customer\_id ASC



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

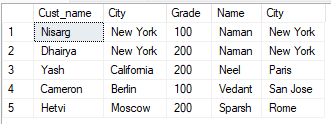
FROM Customer C

Inner Join Salesman S

ON S.Salesman\_id = C.Salesman\_id

where C.Grade < 300

Order By C.Customer\_id ASC



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,

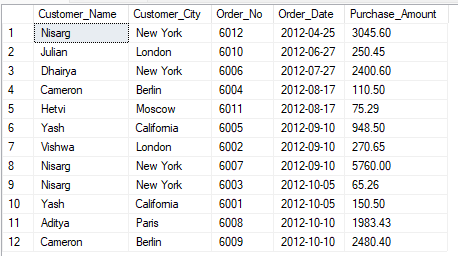
O.Order\_No, O.Order\_Date, O.Purch\_amt AS Purchase\_Amount

FROM Orders O

Left Outer Join Customer C

ON C.Customer\_id = O.Customer\_id

Order By O.Order\_date



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

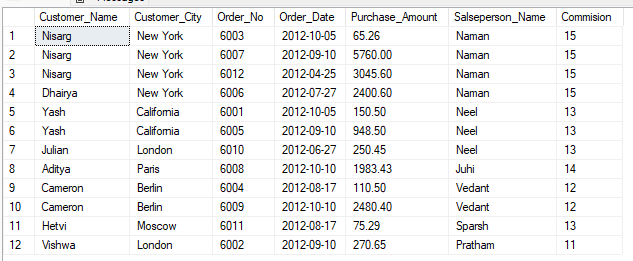
FROM Customer C

LEFT OUTER JOIN Orders O

ON O.Customer\_id = C.Customer\_id

LEFT OUTER JOIN Salesman S

ON S.Salesman\_id=O.Salesman\_id



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 Salseman\_City

FROM Customer C

RIGHT OUTER JOIN Salesman S

ON S.Salesman\_id = C.Salesman\_id

ORDER BY S.Name;



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,

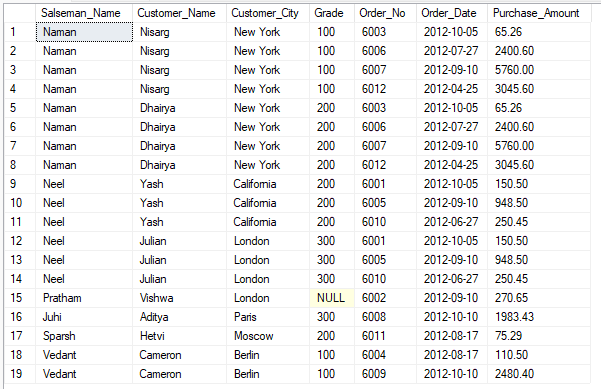
C.Cust\_name AS Customer\_Name, C.City AS Customer\_City,

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)



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.\* ,

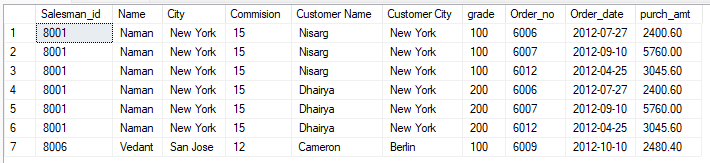
C.cust\_name as "Customer Name", C.city as "Customer City", C.grade,

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



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.\* ,

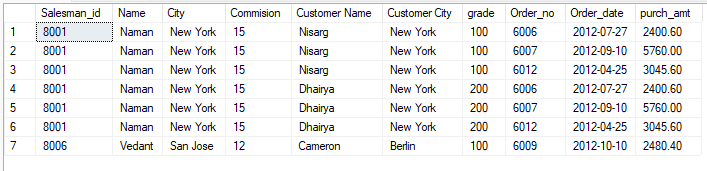
C.cust\_name as "Customer Name", C.city as "Customer City", C.grade,

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



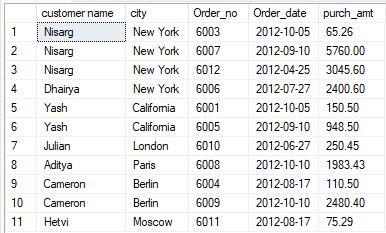
Query 16: Write a SQL statement to generate a report with the customer name, city, order no. order date, purchase amount for only those customers on the list who must have a grade and placed one or more orders or which order(s) have been placed by the customer who neither is on the list nor has a grade.

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



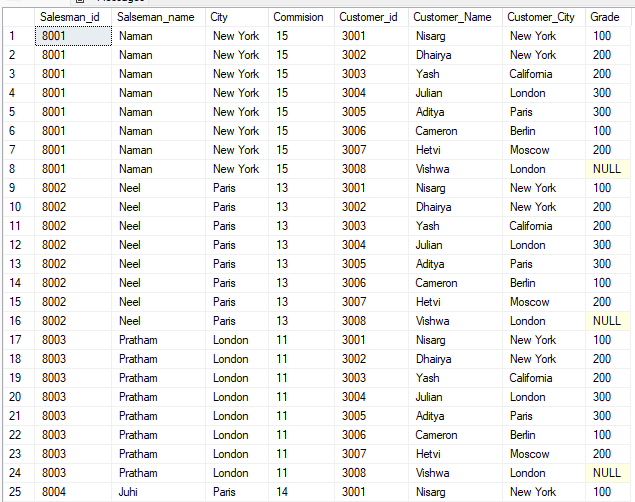
Query 17: Write a SQL query to combine each row of the salesman table with each row of the customer table

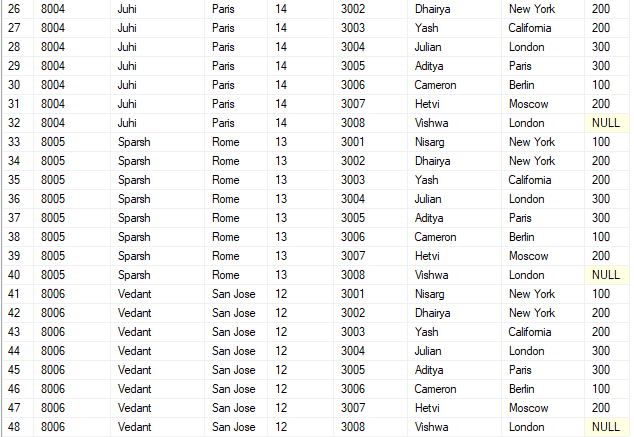
SELECT S.Salesman\_id, S.[Name] AS Salseman\_name, S.City, S.Commision,

C.Customer\_id, C.Cust\_name AS Customer\_Name,C.City AS Customer\_City, C.Grade

FROM Salesman s

CROSS JOIN Customer C





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

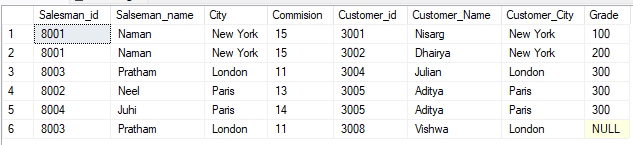
SELECT S.Salesman\_id, S.[Name] AS Salseman\_name, S.City, S.Commision,

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



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 Salseman\_name, S.City, S.Commision,

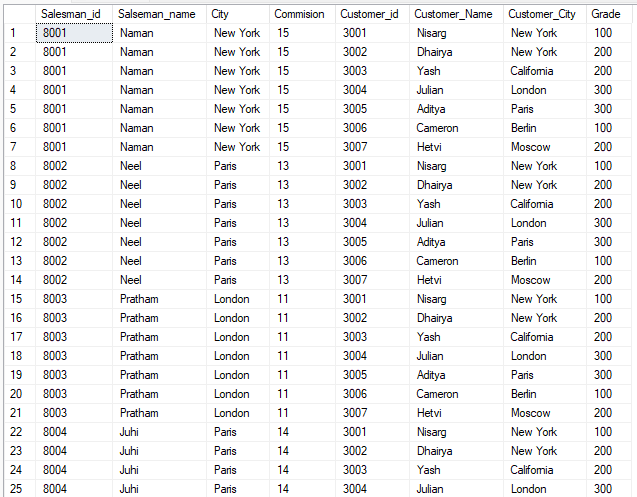
C.Customer\_id, C.Cust\_name AS Customer\_Name,C.City AS Customer\_City, C.Grade

FROM Salesman s

CROSS JOIN Customer C

WHERE C.City IS NOT NULL

AND C.Grade IS NOT NULL;





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;

