--------------------------------------------------------------------------------------------------------------------------

**Assignment 2: Retrieve data using join with where clause**

**Sample table1: salesman**

-salesman\_id

-name

-city

-commission

**Sample table2: customer**

-customer\_id

cust\_name

-city

-grade

-salesman\_id

**Sample table3: orders**

-ord\_no

purch\_amt

-ord\_date

-customer\_id

-salesman\_id

Create Table salesman

(

salesman\_id int NOT NULL Primary key,

name nvarchar(50) NOT NULL,

city nvarchar(20) NOT NULL,

commission int NOT NULL,

)

Create Table customer

(

customer\_id int NOT NULL Primary key,

cust\_name nvarchar(50) NOT NULL,

city nvarchar(20) NOT NULL,

grade int,

salesman\_id int,

Foreign Key (salesman\_id) References salesman(salesman\_id)

)

Create Table orders

(

ord\_no int NOT NULL Primary Key,

purch\_amt money NOT NULL,

ord\_date date NOT NULL,

customer\_id int NOT NULL,

salesman\_id int NOT NULL,

Foreign Key (customer\_id) References customer(customer\_id),

Foreign Key (salesman\_id) References salesman(salesman\_id),

)

Inserting values to the column:

Insert Into salesman values

(5001, 'James', 'New York', 15),

(5002, 'Nail', 'Paris', 13),

(5003, 'Pit', 'London', 11),

(5004, 'Lyon', 'Paris', 14),

(5005, 'Paul', 'Rome', 13),

(5006, 'Lauson', 'San Fransisco', 12),

(5007, 'Harry', 'Amsterdam', 16),

(5008, 'Albus', 'Cape Town', 12);

Insert Into customer values

(3001, 'Nick', 'New York', 100, 5001),

(3002, 'Brad Davis', 'New York', 200, 5001),

(3003, 'Graham', 'California', 200, 5002),

(3004, 'Julian', 'London', 300, 5002),

(3005, 'Fabian', 'Paris', 300, 5004),

(3006, 'Cameron', 'Berlin', 100, 5006),

(3007, 'Jozy', 'Moscow', 200, 5005),

(3008, 'Brad Guzan', 'London', NULL , 5003),

(3009, 'Tony', 'Cape Town', 300, 5003),

(3010, 'Peter', 'San Fransisco', NULL , 5006);

Insert Into orders values

(7001, 150.5, '2012-10-05', 3003, 5002),

(7002, 270.65, '2012-09-10', 3008, 5003),

(7003, 65.26, '2012-10-05', 3001, 5001),

(7004, 110.5, '2012-08-17', 3006, 5006),

(7005, 948.5, '2012-09-10', 3003, 5002),

(7006, 2400.6, '2012-07-27', 3002, 5001),

(7007, 5760, '2012-09-10', 3001, 5001),

(7008, 1983.43, '2012-10-10', 3005, 5004),

(7009, 2480.4, '2012-10-10', 3006, 5006),

(7010, 250.45, '2012-06-27', 3004, 5002),

(7011, 75.29, '2012-08-17', 3007, 5005),

(7012, 3045.6, '2012-04-25', 3001, 5001);

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

Return Salesman, cust\_name and city

Select **name as Salesman,cust\_name,customer.city**

from salesman

INNER JOIN customer

**ON salesman.city=customer.city**

****

**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**

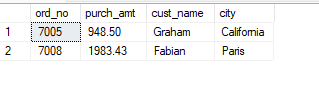
**Select ord\_no, purch\_amt, cust\_name, city**

from customer

INNER JOIN Orders

ON customer.customer\_id=orders.customer\_id

**where purch\_amt between 500 and 2000**

****

**3. write a SQL query to find the salesperson(s) and the customer(s) he represents.**

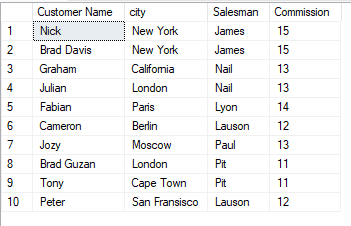
**Return Customer Name, city, Salesman, commission**

**Select cust\_name as [Customer Name],customer.city,name as Salesman,Commission**

from salesman

INNER JOIN customer

**ON salesman.salesman\_id= customer.salesman\_id**

****

**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.**

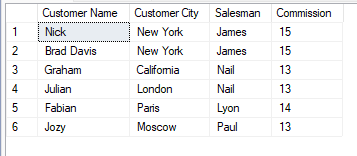
**Select cust\_name as [Customer Name],customer.city as [Customer City],name as Salesman,Commission**

from salesman

INNER JOIN customer

ON salesman.salesman\_id= customer.salesman\_id

**where commission>12;**

****

**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 cust\_name as [Customer Name],customer.city as [Customer City],name as Salesman,**

salesman.city as [Salesman City],Commission

from salesman

INNER JOIN customer

ON salesman.salesman\_id= customer.salesman\_id

**where salesman.city!=customer.city and commission>12;**

****

**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 ord\_no, ord\_date,purch\_amt, cust\_name as [Customer Name], Grade,**

name as Salesman , Commission

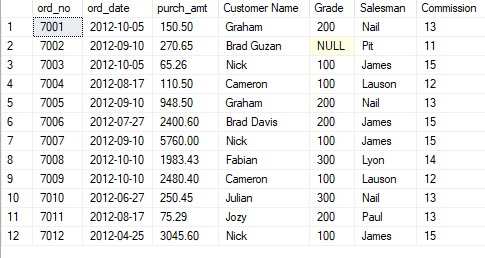
from Orders

INNER JOIN customer

ON customer.customer\_id=orders.customer\_id

INNER JOIN salesman

**ON salesman.salesman\_id= customer.salesman\_id**

****

**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 salesman.salesman\_id,Name,salesman.city,commission,customer.customer\_id,cust\_name,grade,**

ord\_no , purch\_amt,ord\_date

from salesman

INNER JOIN customer

ON salesman.salesman\_id= customer.salesman\_id

INNER JOIN Orders

ON customer.customer\_id=orders.customer\_id

**order by salesman\_id**

****

**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 cust\_name as [Customer Name],customer.city as [Customer City],Grade,Name as Salesman ,**

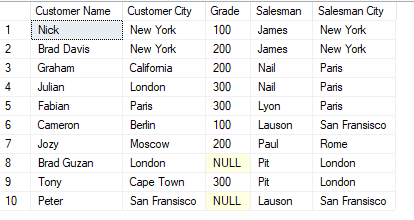
salesman.city as [Salesman City]

from customer

INNER JOIN salesman

ON salesman.salesman\_id=customer.salesman\_id

**order by customer\_id**

****

**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 cust\_name as [Customer Name],customer.city as [Customer City],Grade,Name as Salesman ,**

salesman.city as [Salesman City]

from customer

INNER JOIN salesman

ON salesman.salesman\_id=customer.salesman\_id

where grade<300

**order by customer\_id**

****

**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 cust\_name as [Customer Name],customer.city as [Customer City],ord\_no as [Order Number],**

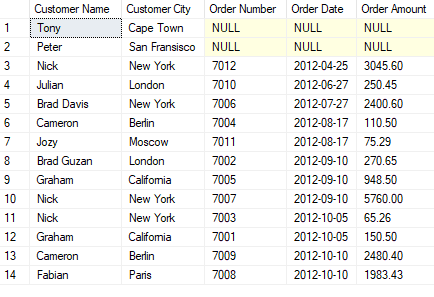
ord\_date as [Order Date],Purch\_amt as [Order Amount]

from customer

Left Outer JOIN orders

ON orders.customer\_id=customer.customer\_id

**order by ord\_date**

****

**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 cust\_name as [Customer Name],customer.city as [Customer City],ord\_no as [Order Number],**

ord\_date as [Order Date],Purch\_amt as [Order Amount],name as [Salesman Name],Commission

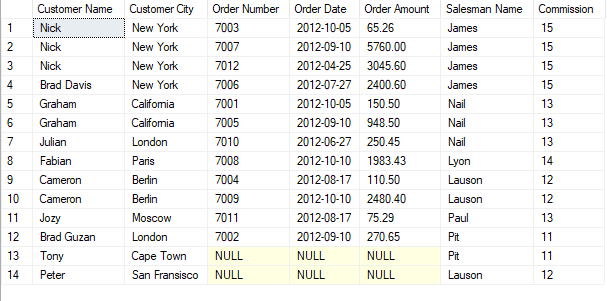
from customer

Left Outer JOIN orders

ON customer.customer\_id=orders.customer\_id

Left Outer JOIN salesman

ON customer.salesman\_id=salesman.salesman\_id



**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**

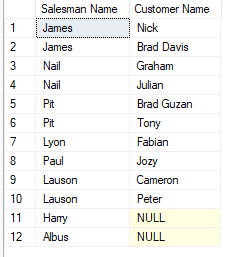
**SELECT name as [Salesman Name],cust\_name as [Customer Name]**

FROM salesman s

LEFT OUTER JOIN customer c

ON s.salesman\_id=c.salesman\_id

**ORDER BY s.salesman\_id ASC;**

****

**13. write a SQL query to list all salespersons along with customer name, city, grade,**

**order number, date, and amount.**

**SELECT name as [Salesman Name],cust\_name as [Customer Name],customer.city,Grade,ord\_no as [Order Number],**

ord\_date as [Date],purch\_amt as [Amount]

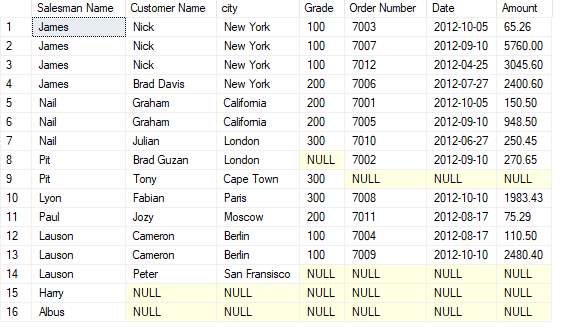
FROM salesman

Left Outer JOIN customer

ON salesman.salesman\_id=customer.salesman\_id

Left Outer JOIN orders

**ON customer.customer\_id=orders.customer\_id**

****

**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.salesman\_id,name as [Salesman Name],s.city,s.commission,cust\_name as [Customer Name],o.purch\_amt as [Order Amount],c.Grade**

FROM salesman s

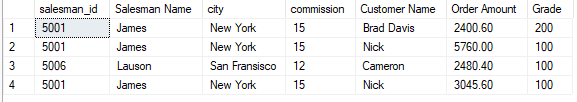
LEFT OUTER JOIN customer c

ON s.salesman\_id=c.salesman\_id

LEFT OUTER JOIN orders o

ON c.customer\_id=o.customer\_id

**where o.purch\_amt>=2000 and c.Grade is not null**

****

**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.salesman\_id,name as [Salesman Name],s.city,s.commission,cust\_name as [Customer Name],o.purch\_amt as [Order Amount],c.Grade**

FROM salesman s

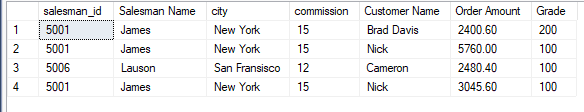
LEFT OUTER JOIN customer c

ON s.salesman\_id=c.salesman\_id

LEFT OUTER JOIN orders o

ON c.customer\_id=o.customer\_id

**where o.purch\_amt>=2000 and c.Grade is not null**

****

**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 cust\_name as [Customer Name],customer.city , ord\_no as [Order Number],ord\_date as [Order\_date],**

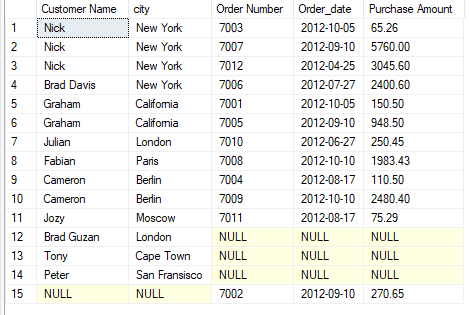
purch\_amt as [Purchase Amount]

FROM customer

FULL OUTER JOIN orders

ON customer.customer\_id= orders.customer\_id

**AND customer.grade IS NOT NULL;**

****

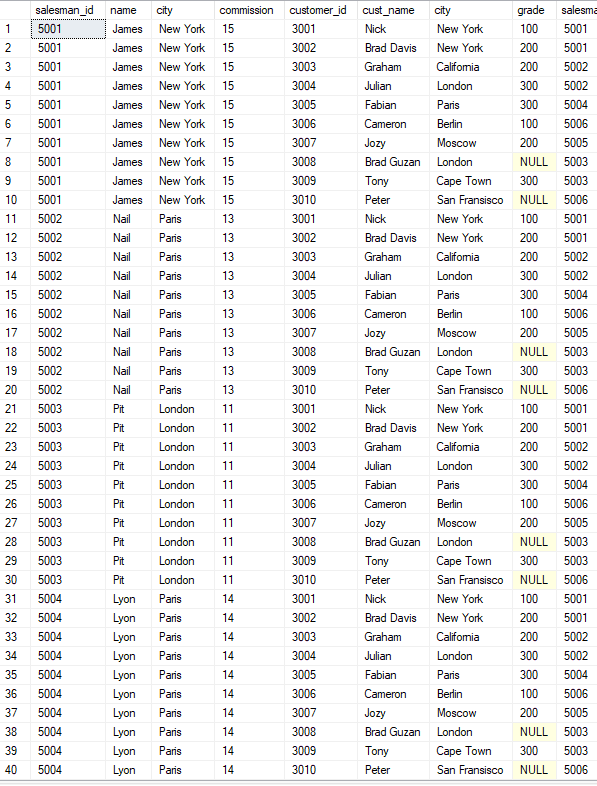
**17. Write a SQL query to combine each row of the salesman table with each row of the**

**customer table**

**SELECT \***

FROM salesman s

**CROSS JOIN customer c;**

****

**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.name AS Salesman,**

c.cust\_name AS [Customer Name]

FROM salesman s

CROSS JOIN customer c

**WHERE s.city IS NOT NULL;**

****

**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.name AS "Salesman", c.cust\_name AS "Customer Name"**

FROM salesman s

CROSS JOIN customer c

WHERE s.city IS NOT NULL

**AND c.grade IS NOT NULL;**

****

**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.name AS "Salesman",c.cust\_name AS "Customer Name"**

FROM salesman s

CROSS JOIN customer c

WHERE s.city IS NOT NULL

AND s.city != c.city

**AND c.grade IS NOT NULL;**

****