Assignment 2 : Retrieve data using join with where clause

CREATE TABLE [salesman](

salesman\_id INT PRIMARY KEY IDENTITY(1,1),

name VARCHAR(20) NOT NULL,

city VARCHAR(20) NOT NULL,

commission float NOT NULL

);

CREATE TABLE [customer](

customer\_id INT PRIMARY KEY IDENTITY(1,1),

cust\_name VARCHAR(20) NOT NULL,

city VARCHAR(10) NOT NULL,

grade VARCHAR(2) NOT NULL,

salesman\_id INT FOREIGN KEY REFERENCES salesman(salesman\_id) NOT NULL

);

CREATE TABLE [orders](

ord\_no INT PRIMARY KEY IDENTITY(1,1),

purch\_amt FLOAT NOT NULL,

ord\_date DATE NOT NULL,

customer\_id INT FOREIGN KEY REFERENCES customer(customer\_id) NOT NULL,

salesman\_id INT FOREIGN KEY REFERENCES salesman(salesman\_id) NOT NULL

);

insert into salesman values

('jhon','ahmedabad',0.12),

('divak','mumbai',0.16),

('kartik','vadodara',0.15),

('aryan','banaras',0.11),

('diya','rajkot',0.19),

('raj','chennai',0.25)

select \* from salesman

insert into customer values

('Abhi','ahmedabad','A',1),

('yash','surat','B',2),

('ketul','vadodara','C',3),

('shyam','rajkot','A',4),

('meet','banaras','B',5),

('dhara','chennai',600,2),

('mansi','surat',600,3)

insert into orders values

(300,'2001-02-12',1,1),

(200,'2002-02-22',2,2),

(400,'2003-04-12',3,3),

(600,'2002-08-12',4,4),

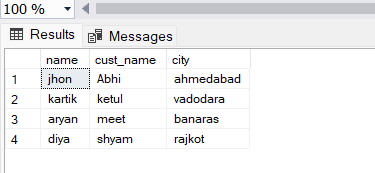
(750,'2004-02-12',5,5),

(5000,'2004-09-25',1,2)

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

Return Salesman, cust\_name and city

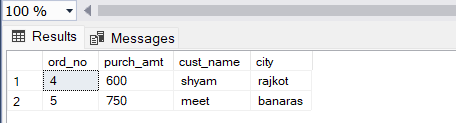
Sol:select s.name,c.cust\_name,c.city from salesman s,customer c where s.city=c.city



Query:2) write a SQL query to find those orders where the order amount exists between 500

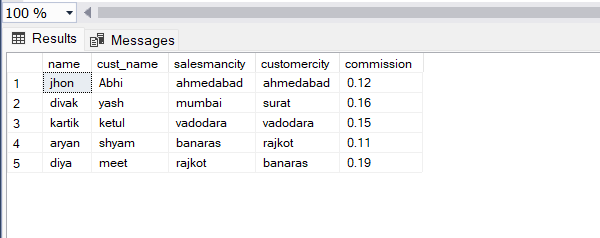
and 2000. Return ord\_no, purch\_amt, cust\_name, city

Sol: select o.ord\_no, o.purch\_amt, c.cust\_name, c.city from orders o inner join customer c on c.customer\_id=o.customer\_id where purch\_amt between 500 and 2000



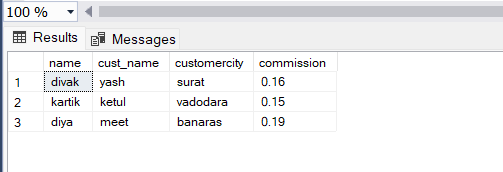
Query: 3) write a SQL query to find the salesperson(s) and the customer(s) he represents.Return Customer Name, city, Salesman, commission

Sol: select s.name,c.cust\_name,s.city as salesmancity,c.city as customercity,s.commission from salesman s inner join customer c on s.salesman\_id=c.salesman\_id



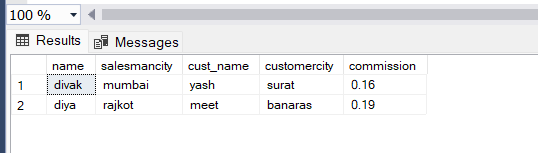
Query:4) write a SQL query to find salespeople who received commissions of more than 12percent from the company. Return Customer Name, customer city, Salesman,commission.

Sol: select s.name,c.cust\_name,c.city as customercity,s.commission from salesman s inner join customer c on s.salesman\_id=c.salesman\_id where s.commission>0.12



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 thecompany. Return Customer Name, customer city, Salesman, salesman city,commission .

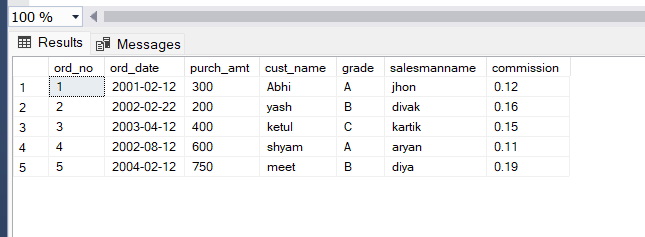
Sol: select s.name,s.city as salesmancity,c.cust\_name,c.city as customercity,s.commission from salesman s inner join customer c on s.salesman\_id=c.salesman\_id where s.city!=c.city and s.commission>0.12



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

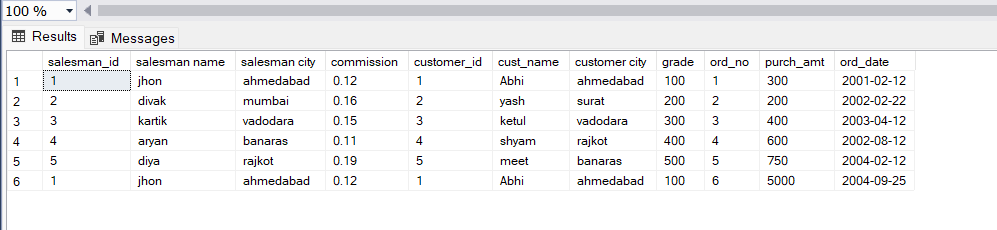
Sol: select o.ord\_no, o.ord\_date,o.purch\_amt, c.cust\_name, c.grade, s.name as salesmanname, s.commission from salesman s inner join customer c on s.salesman\_id=c.salesman\_id

inner join orders o on c.customer\_id=o.customer\_id



Query:7) Write a SQL statement to join the tables salesman, customer and orders so that thesame column of each table appears once and only the relational rows are returned.

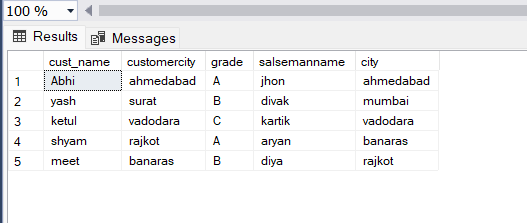
Sol: SELECT s.salesman\_id, s.name "salesman name", s.city "salesman city", s.commission, c.customer\_id, c.cust\_name, c.city "customer city", c.grade, o.ord\_no, o.purch\_amt, o.ord\_date FROM salesman s INNER JOIN customer c ON s.salesman\_id = c.salesman\_id INNER JOIN orders o ON c.customer\_id = o.customer\_id;



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.

Sol: select c.cust\_name,c.city as customercity,c.grade,s.name as salsemanname,s.city from salesman s inner join customer c on

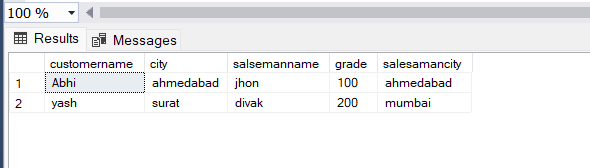
s.salesman\_id=c.salesman\_id order by c.customer\_id



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 beordered by ascending customer\_id.

Sol: select c.cust\_name as customername,c.city,s.name as salsemanname,c.grade,s.city as salesamancity from salesman s inner join customer c on

s.salesman\_id=c.salesman\_id where c.grade<300 order by c.customer\_id

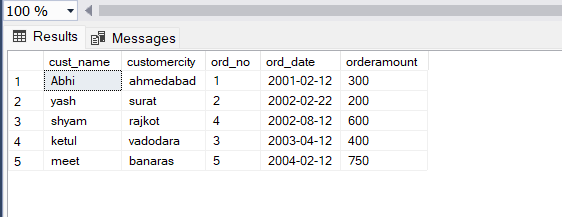


Query:10) Write a SQL statement to make a report with customer name, city, order number,order date, and order amount in ascending order according to the order date todetermine whether any of the existing customers have placed an order or not

Sol: SELECT c.cust\_name,c.city customercity, o.ord\_no,

o.ord\_date,o.purch\_amt AS orderamount FROM customer c LEFT JOIN orders o

ON c.customer\_id=o.customer\_id order by o.ord\_date;



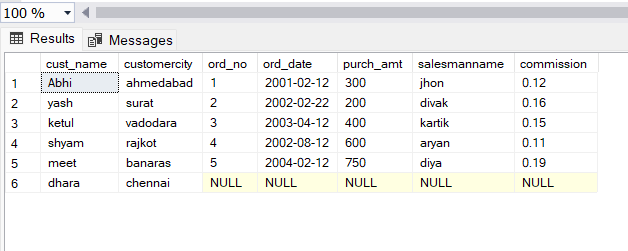
Query:11) Write a SQL statement to generate a report with customer name, city, order number,order date, order amount, salesperson name, and commission to determine if any of the existing customers have not placed orders or if they have placed orders through their salesman or by themselves

Sol: select c.cust\_name,c.city customercity, o.ord\_no,

o.ord\_date,o.purch\_amt,s.name as salesmanname,s.commission from customer c left join orders o

on c.customer\_id=o.customer\_id left join salesman s on

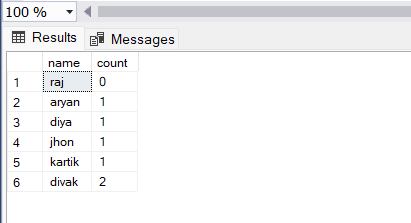
o.salesman\_id=s.salesman\_id



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

Sol: select s.name,count(c.customer\_id) as count from salesman s left join customer c on c.salesman\_id=s.salesman\_id

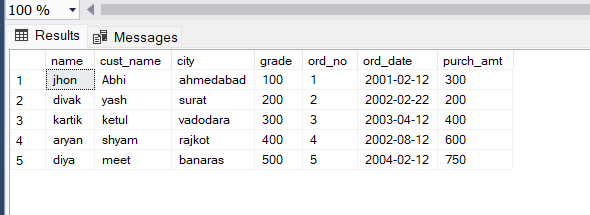
group by(s.name) order by count(c.customer\_id)



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

Sol: select s.name,c.cust\_name,c.city,c.grade,o.ord\_no,o.ord\_date,o.purch\_amt from salesman s inner join customer c

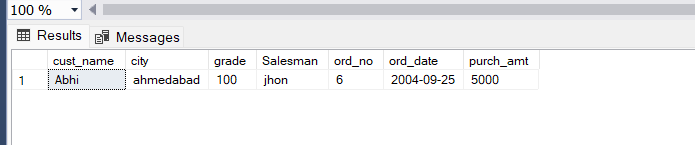
on s.salesman\_id=c.salesman\_id inner join orders o on o.customer\_id=c.customer\_id



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.

Sol: SELECT a.cust\_name,a.city,a.grade, b.name AS "Salesman", c.ord\_no, c.ord\_date, c.purch\_amt FROM customer a RIGHT OUTER JOIN salesman b

ON b.salesman\_id=a.salesman\_id LEFT OUTER JOIN orders c ON c.customer\_id=a.customer\_id WHERE c.purch\_amt>=2000 AND a.grade IS NOT NULL;

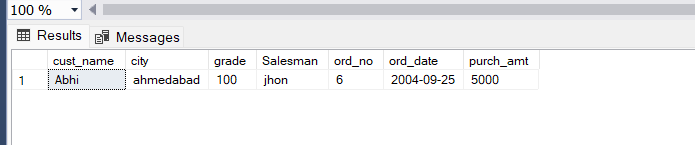


Query:15) Write a SQL statement to generate a list of all the salesmen who either work for oneor more customers or have yet to join any of them. The customer may have placedone or more orders at or above order amount 2000, and must have a grade, or hemay not have placed any orders to the associated supplier.

Sol: SELECT a.cust\_name,a.city,a.grade, b.name AS "Salesman", c.ord\_no, c.ord\_date, c.purch\_amt FROM customer a RIGHT OUTER JOIN salesman b ON b.salesman\_id=a.salesman\_id

LEFT OUTER JOIN orders c ON c.customer\_id=a.customer\_id WHERE c.purch\_amt>=2000

AND a.grade IS NOT NULL;

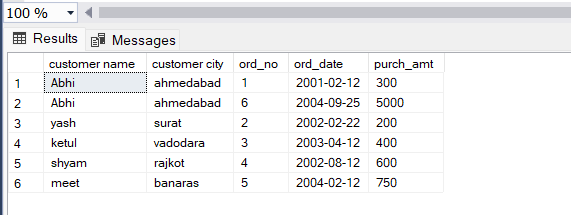


Query:16) Write a SQL statement to generate a report with the customer name, city, order no.order date, purchase amount for only those customers on the list who must have agrade and placed one or more orders or which order(s) have been placed by thecustomer who neither is on the list nor has a grade.

Sol: SELECT c.cust\_name "customer name", c.city "customer city", o.ord\_no, o.ord\_date, o.purch\_amt

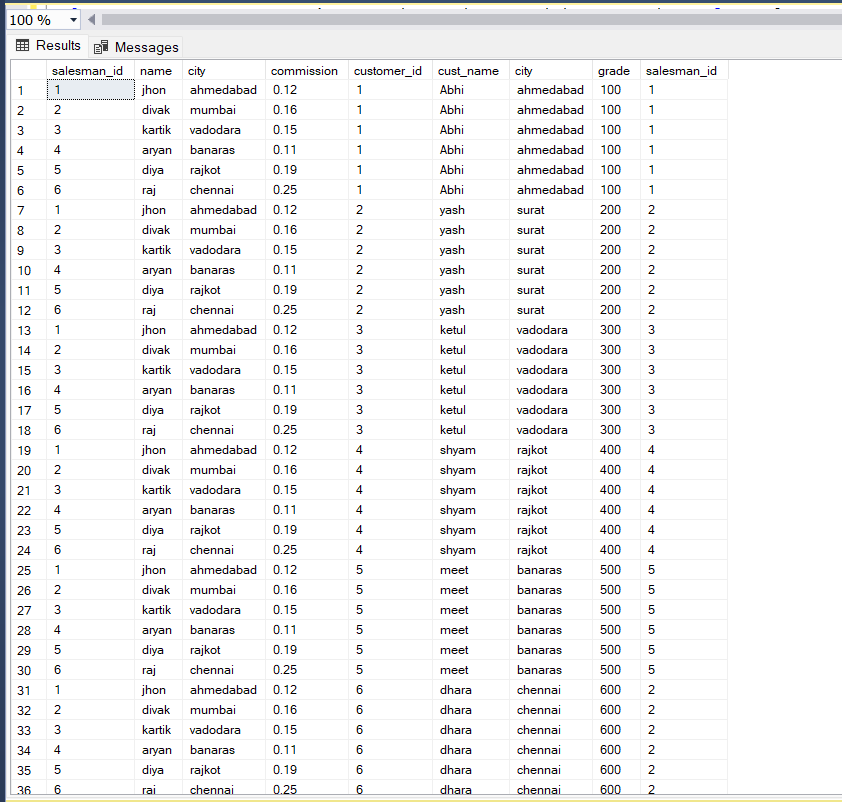
FROM customer c FULL OUTER JOIN orders o ON c.customer\_id = o.customer\_id WHERE (c.grade IS NOT NULL and o.ord\_no is not null)

or (c.grade IS NULL and o.ord\_no is null)

\_­­­­

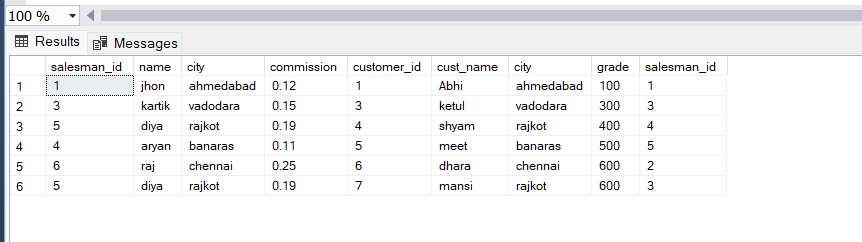
Query:17) Write a SQL query to combine each row of the salesman table with each row of thecustomer table

Sol: select \* from salesman s cross join customer c

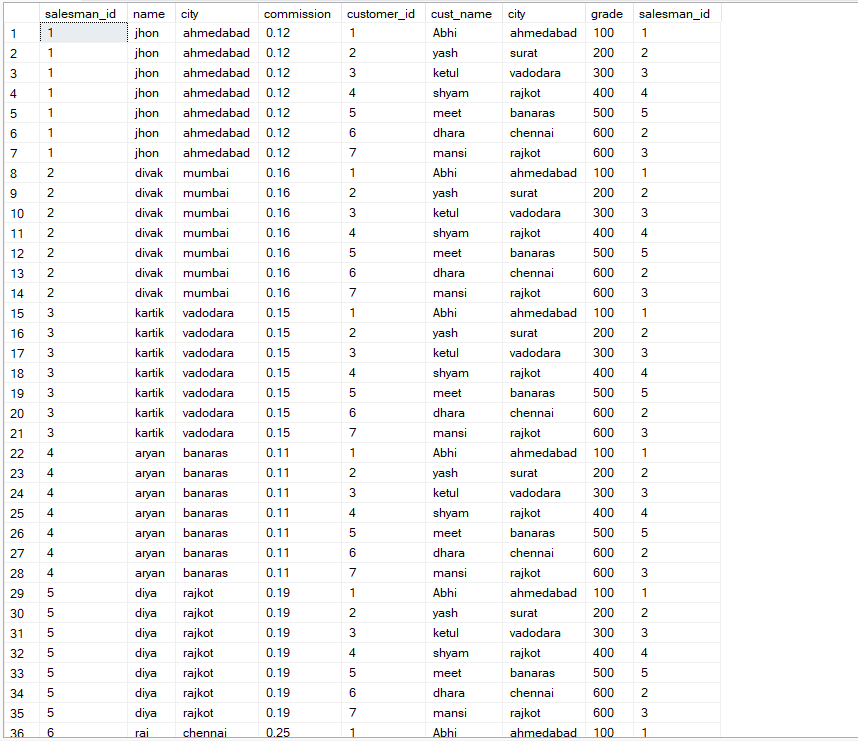


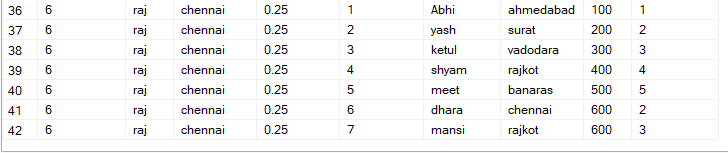
Query:18) Write a SQL statement to create a Cartesian product between salesperson andcustomer, i.e. each salesperson will appear for all customers and vice versa for thatsalesperson who belongs to that city

Sol: select \* from salesman s cross join customer c where s.city=c.city



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

Sol: select \* from salesman s cross join customer c where s.city is not null and c.grade is not null



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

Sol: select \* from salesman s cross join customer c where s.city!=c.city and c.grade is not null

