Assignment:-2

**Sample table1: salesman**

Create Table Salesman

(salesman\_id int Primary Key,

name nvarchar(20) not null,

city nvarchar(20) not null,

commission float not null)

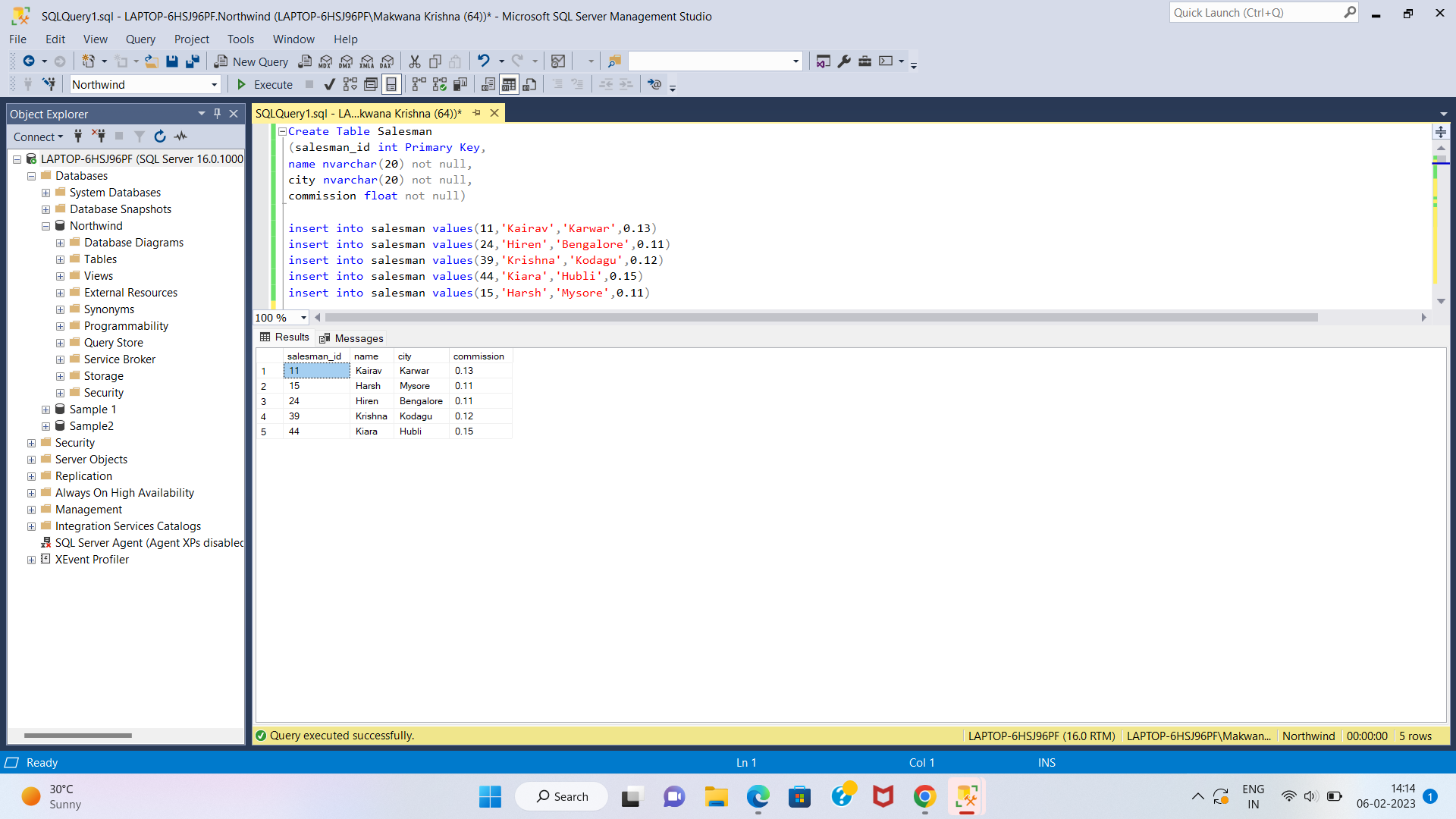
insert into salesman values(11,'Kairav','Karwar',0.13)

insert into salesman values(24,'Hiren','Bengalore',0.11)

insert into salesman values(39,'Krishna','Kodagu',0.12)

insert into salesman values(44,'Kiara','Hubli',0.15)

insert into salesman values(15,'Harsh','Mysore',0.11)



**Sample table2: customer**

create table customer

(customer\_id int primary key,

cust\_name varchar(20) not null,

city varchar(20) not null,

grade int not null,

salesman\_id int references salesman(salesman\_id) on delete set null);

insert into customer values(101,'Bhargav','Mysore',1,15);

insert into customer values(206,'Ramya','Bengalore',3,24);

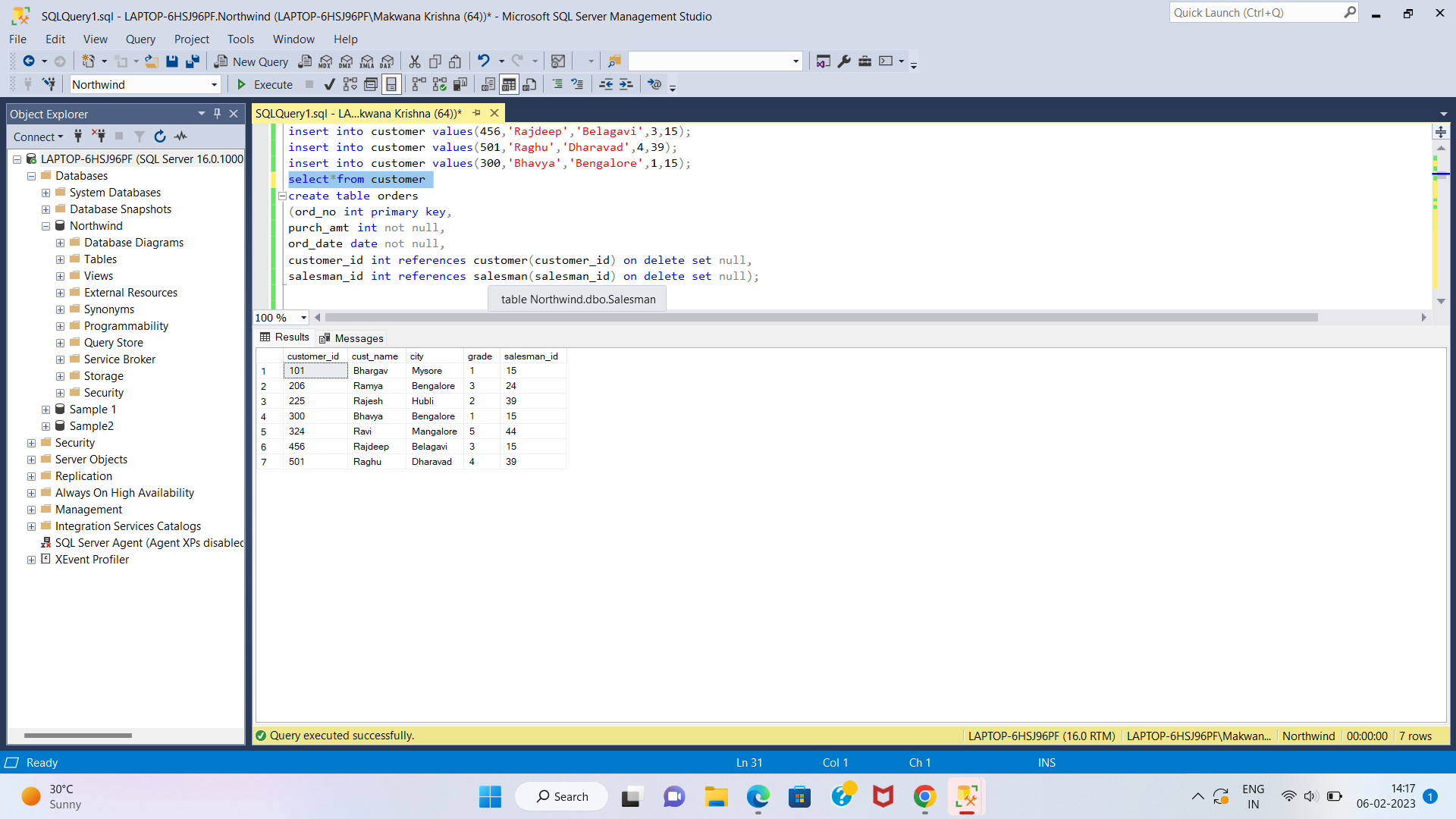
insert into customer values(225,'Rajesh','Hubli',2,39);

insert into customer values(324,'Ravi','Mangalore',5,44);

insert into customer values(456,'Rajdeep','Belagavi',3,15);

insert into customer values(501,'Raghu','Dharavad',4,39);

insert into customer values(300,'Bhavya','Bengalore',1,15);



**Sample table3: orders**

create table orders

(ord\_no int primary key,

purch\_amt int not null,

ord\_date date not null,

customer\_id int references customer(customer\_id) on delete set null,

salesman\_id int references salesman(salesman\_id) on delete set null);

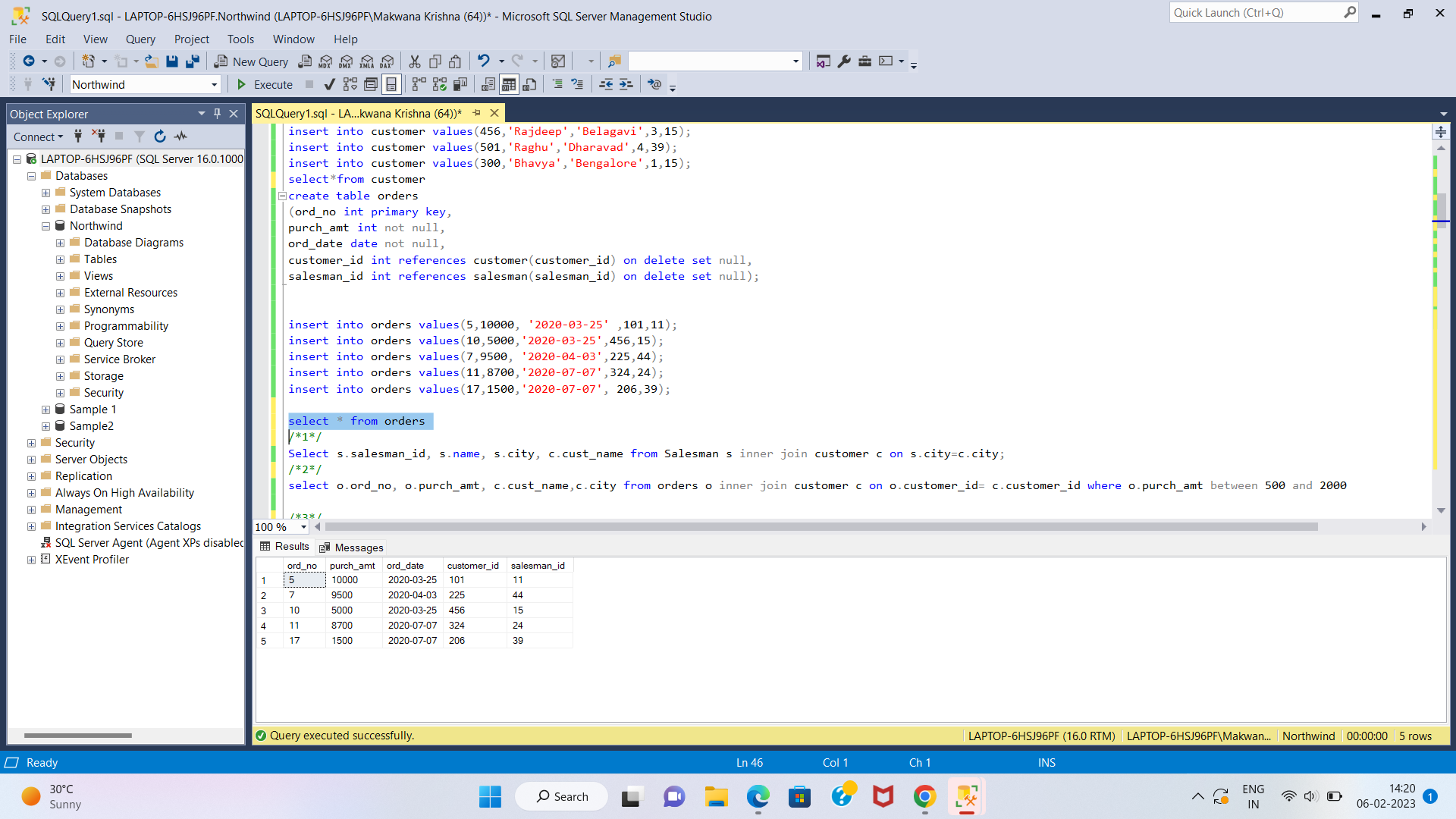
insert into orders values(5,10000, '2020-03-25' ,101,11);

insert into orders values(10,5000,'2020-03-25',456,15);

insert into orders values(7,9500, '2020-04-03',225,44);

insert into orders values(11,8700,'2020-07-07',324,24);

insert into orders values(17,1500,'2020-07-07', 206,39);



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

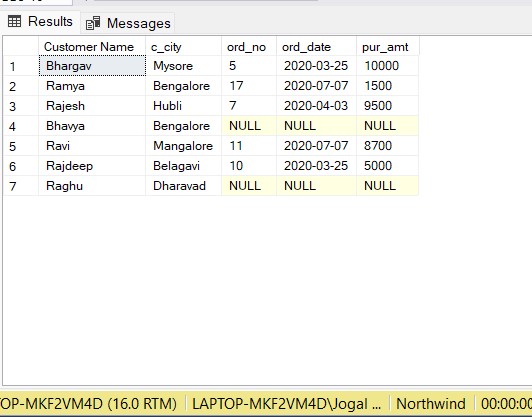
**Ans.** select name,cust\_name,c\_city

from salesman s

inner join customer c

on S.salesman\_id = C.salesman\_id

where s\_city = c\_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**

**Ans.**

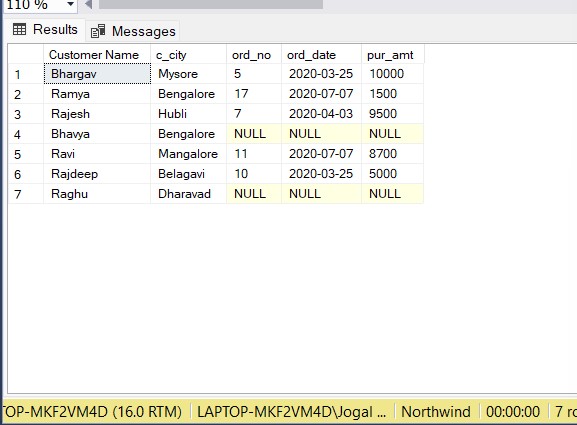
select ord\_no,pur\_amt,cust\_name,c\_city

from customer C

left JOIN orders O

on C.salesman\_id = O.salesman\_id

where O.pur\_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**

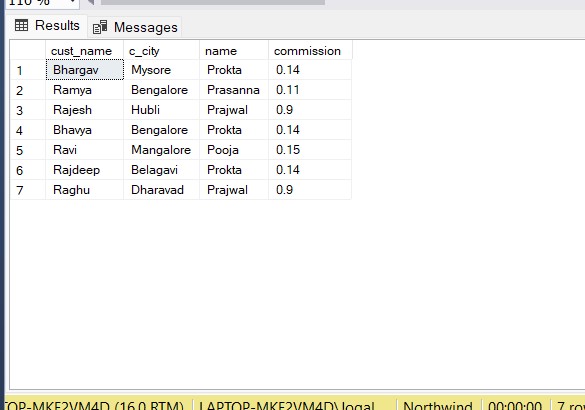
**Ans.**

SELECT cust\_name,c\_city,name,commission

FROM customer c

INNER JOIN salesman s

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

**Ans.**

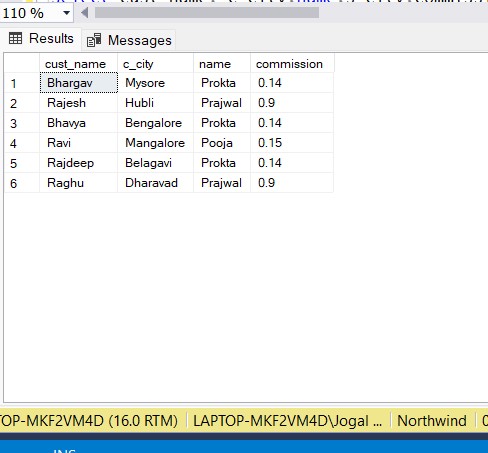
select cust\_name,c\_city,name,commission

from salesman S

inner join customer C

on S.salesman\_id = C.salesman\_id

where commission>0.12



**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**

**Ans.**

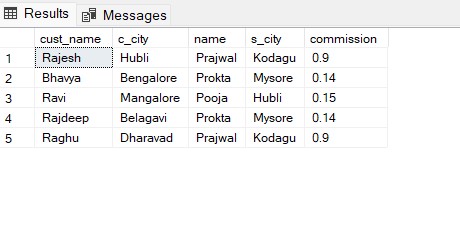
select cust\_name, c\_city,name,s\_city,commission

from salesman S

inner join customer C

on S.salesman\_id = C.salesman\_id

where s\_city != c\_city and commission>0.12



**6. write a SQL query to find the details of an order. Return ord\_no, ord\_date,**

**purch\_amt, Customer Name, grade, Salesman, commission**

**Ans**.

select ord\_no,ord\_date,pur\_amt,cust\_name,grade,name,commission

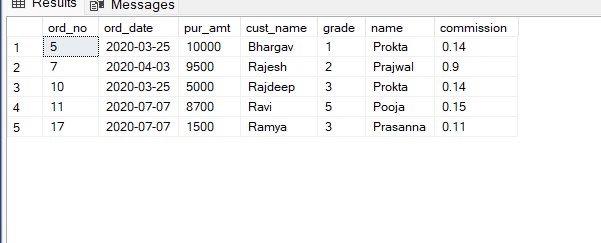
from customer C

inner join orders O

on C.customer\_id=O.customer\_id

inner join salesman S

on C.salesman\_id=S.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.**

**Ans.**

select \*

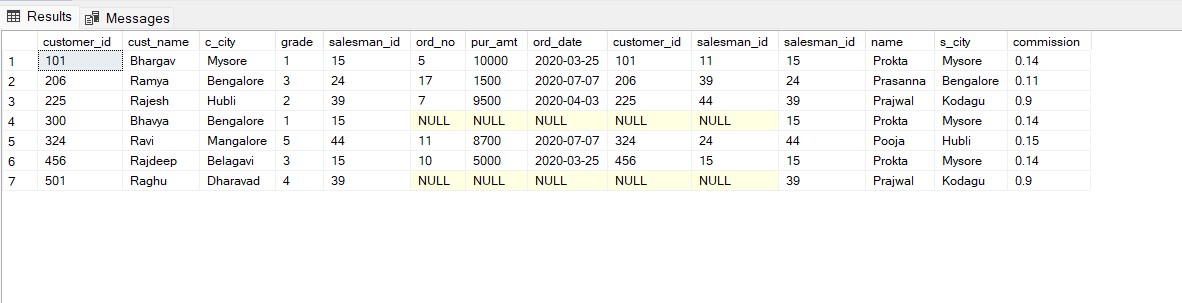
from customer C

left join orders O

on C.customer\_id=O.customer\_id

left join salesman S

on C.salesman\_id=S.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.**

**Ans.**

select cust\_name as customername,

c\_city as customercity,

grade,name as salesman,

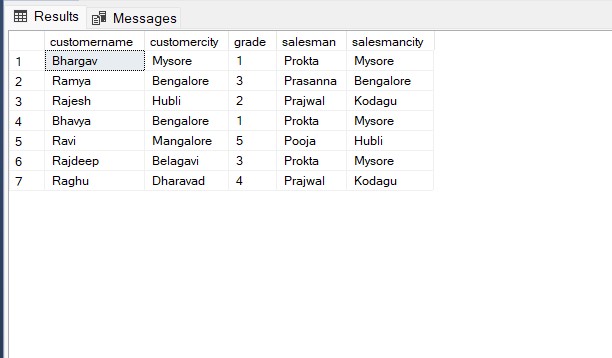
s\_city as salesmancity

from salesman S

inner join customer C

on S.salesman\_id = C.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.**

**Ans.**

select cust\_name as cust\_name,

c\_city as customercity,

grade,name as salesman,

s\_city as salesmancity

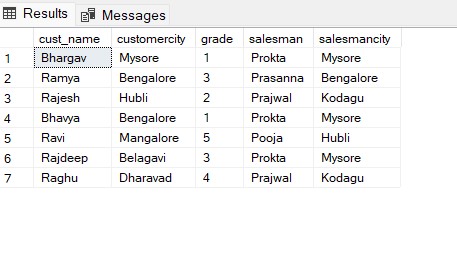
from salesman S

inner join customer C

on S.salesman\_id = C.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**

**Ans.**

select cust\_name as customer\_name,

c\_city as city,

ord\_no as order\_number,

ord\_date as order\_date,

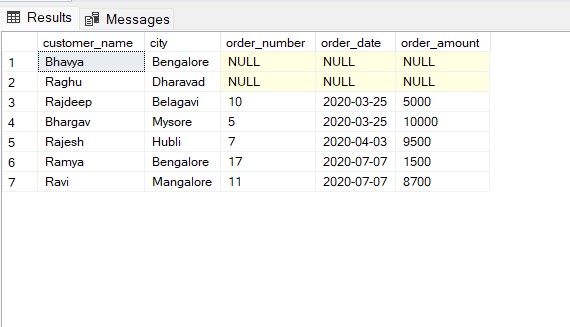
pur\_amt as order\_amount

from customer C

LEFT OUTER join orders O

on C.customer\_id = O.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**

**Ans.**

SELECT c.cust\_name AS "Customer Name",

c.c\_city,

o.ord\_no,

o.ord\_date,

o.pur\_amt,

s.name AS "Salesman",

s.commission

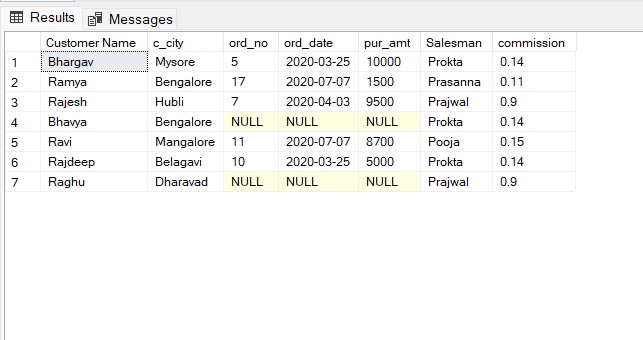
FROM customer c

LEFT OUTER JOIN orders o

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

LEFT OUTER JOIN salesman s

ON c.salesman\_id=s.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**

**Ans.**

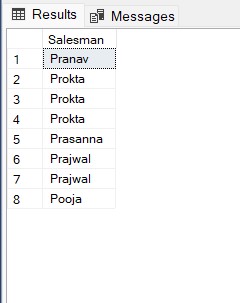
SELECT s.name AS "Salesman"

FROM salesman s

LEFT OUTER JOIN customer c

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

ORDER BY c.salesman\_id ASC;



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

**Ans.**

select s.name as "salespersons",

cust\_name as "customer name",

c\_city as "city",

grade,

ord\_no as "order number",

ord\_date as "date",

pur\_amt as "amount"

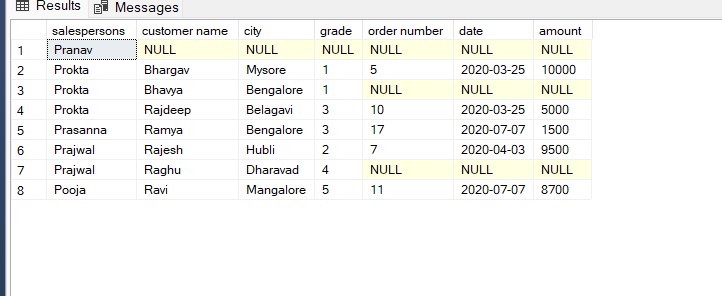
from salesman s

left join customer c

on s.salesman\_id = c.salesman\_id

left join orders o

on c.customer\_id = o.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.**

**Ans.**

select name as "salespermen"

from salesman s

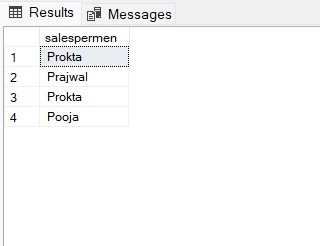
right join customer c

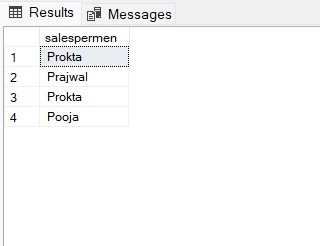
on s.salesman\_id = c.salesman\_id

right join orders o

on c.customer\_id = o.customer\_id

where pur\_amt>2000 and grade is not null or o.salesman\_id is 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.**



**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.**

**Ans.**

SELECT c.cust\_name AS "Customer Name",

c.c\_city,

o.ord\_no,

o.ord\_date,

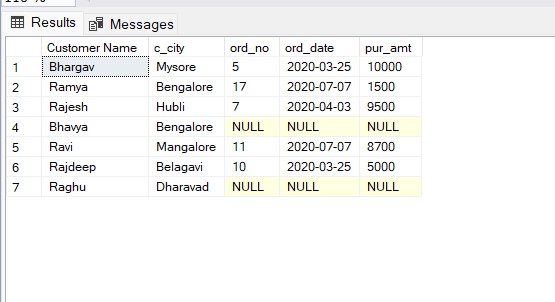
o.pur\_amt

FROM customer c

FULL OUTER JOIN orders o

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

AND c.grade IS NOT NULL;



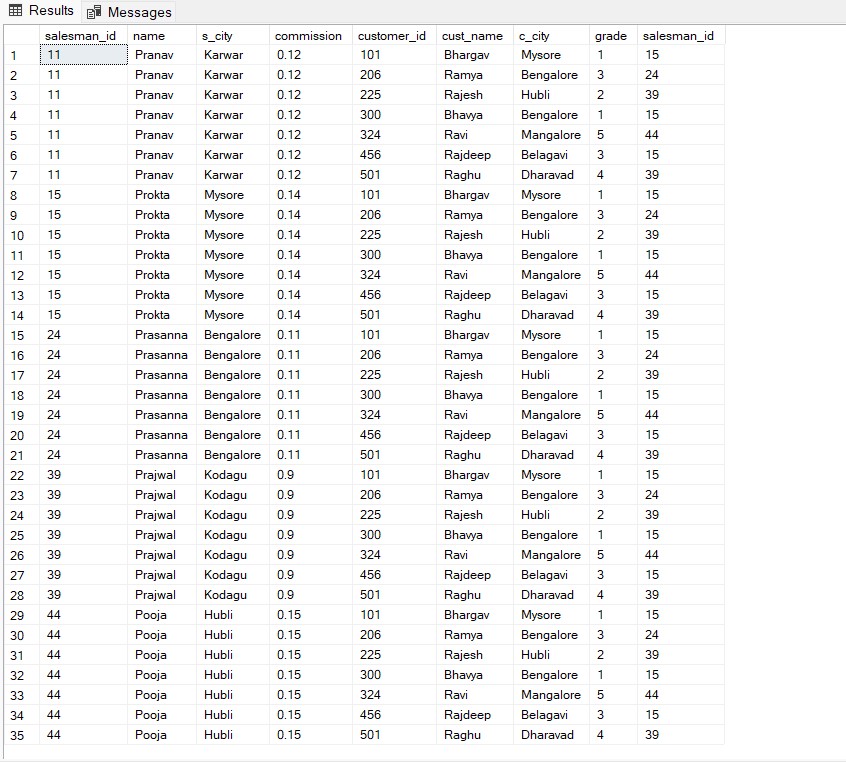
**17. Write a SQL query to combine each row of the salesman table with each row of the customer table**

**Ans.**

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**

**Ans.**

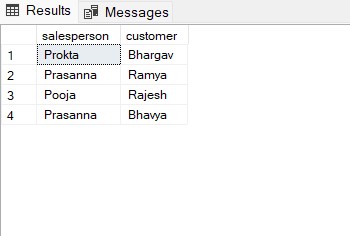
SELECT name as "salesperson",

cust\_name as "customer"

FROM salesman s

CROSS JOIN customer c

where s\_city = c\_city



**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**

**Ans.**

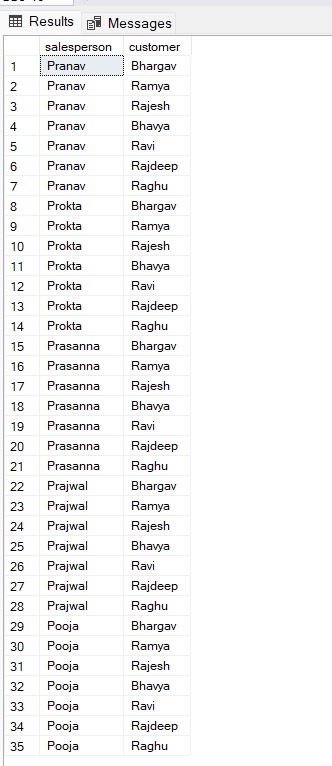
SELECT name as "salesperson",

cust\_name as "customer"

FROM salesman s

CROSS JOIN customer c

where s\_city is not null and 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**

Ans.

SELECT name as "salesperson",

cust\_name as "customer"

FROM salesman s

CROSS JOIN customer c

where s\_city != c\_city and grade is not null

