**SQL**

**ASSIGNMENT 2**

use [Northwind]

go

create table salesman(

salesman\_id int primary key,

name varchar(50) not null,

city varchar(10) not null,

commission decimal(5,2)

);

create table customer(

customer\_id int primary key,

cust\_name varchar(10),

city varchar(10),

grade int,

salesman\_id int

foreign key (salesman\_id) references salesman (salesman\_id)

);

create table orders (

ord\_no int primary key,

purch\_amt decimal(5,2),

ord\_date date,

customer\_id int,

salesman\_id int,

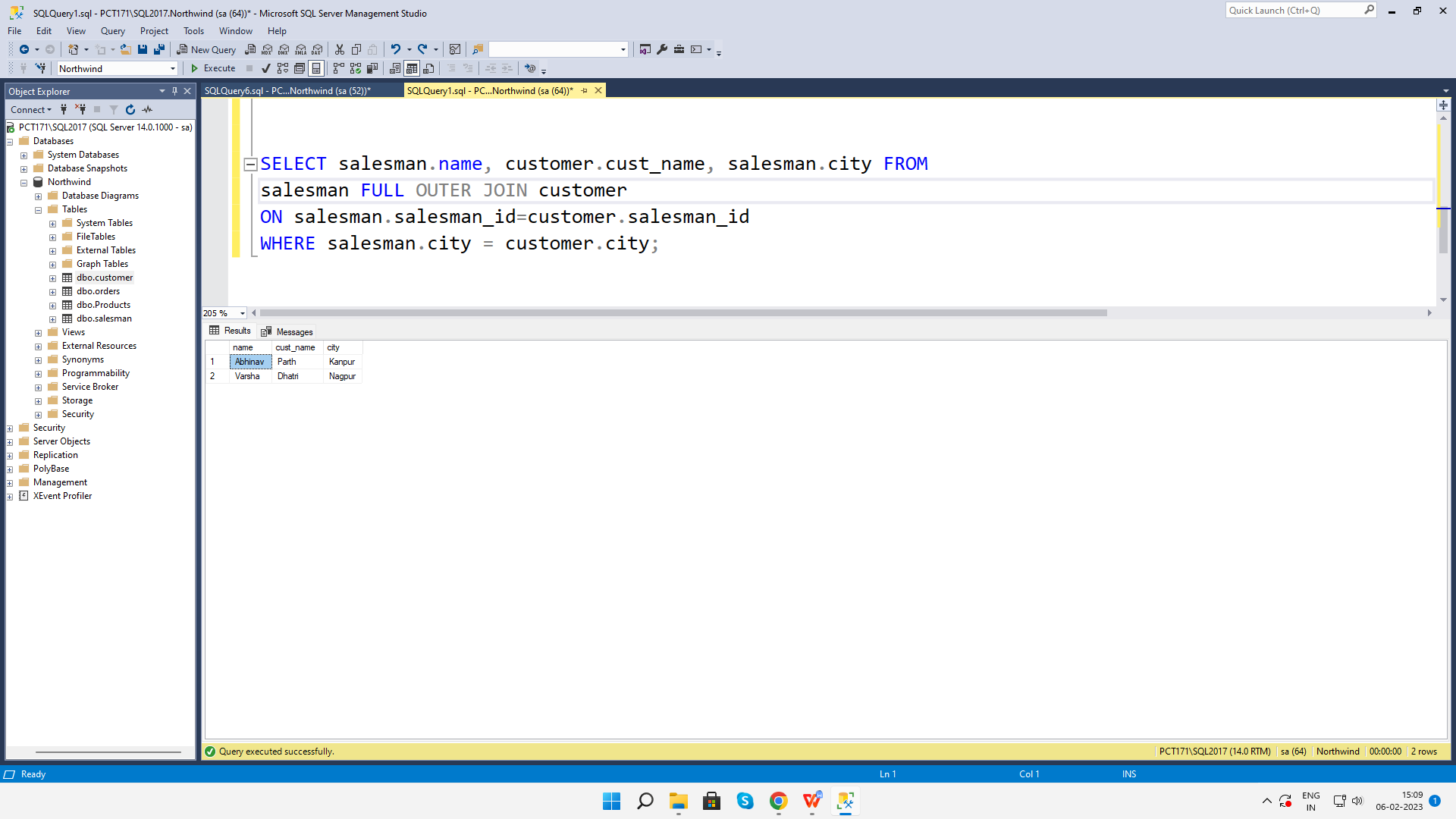
foreign key (customer\_id) references customer (customer\_id),

foreign key (salesman\_id) references salesman (salesman\_id)

);

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, customer.cust\_name, salesman.city FROM salesman FULL OUTER JOIN customer ON salesman.salesman\_id=customer.salesman\_id WHERE salesman.city = customer.city;

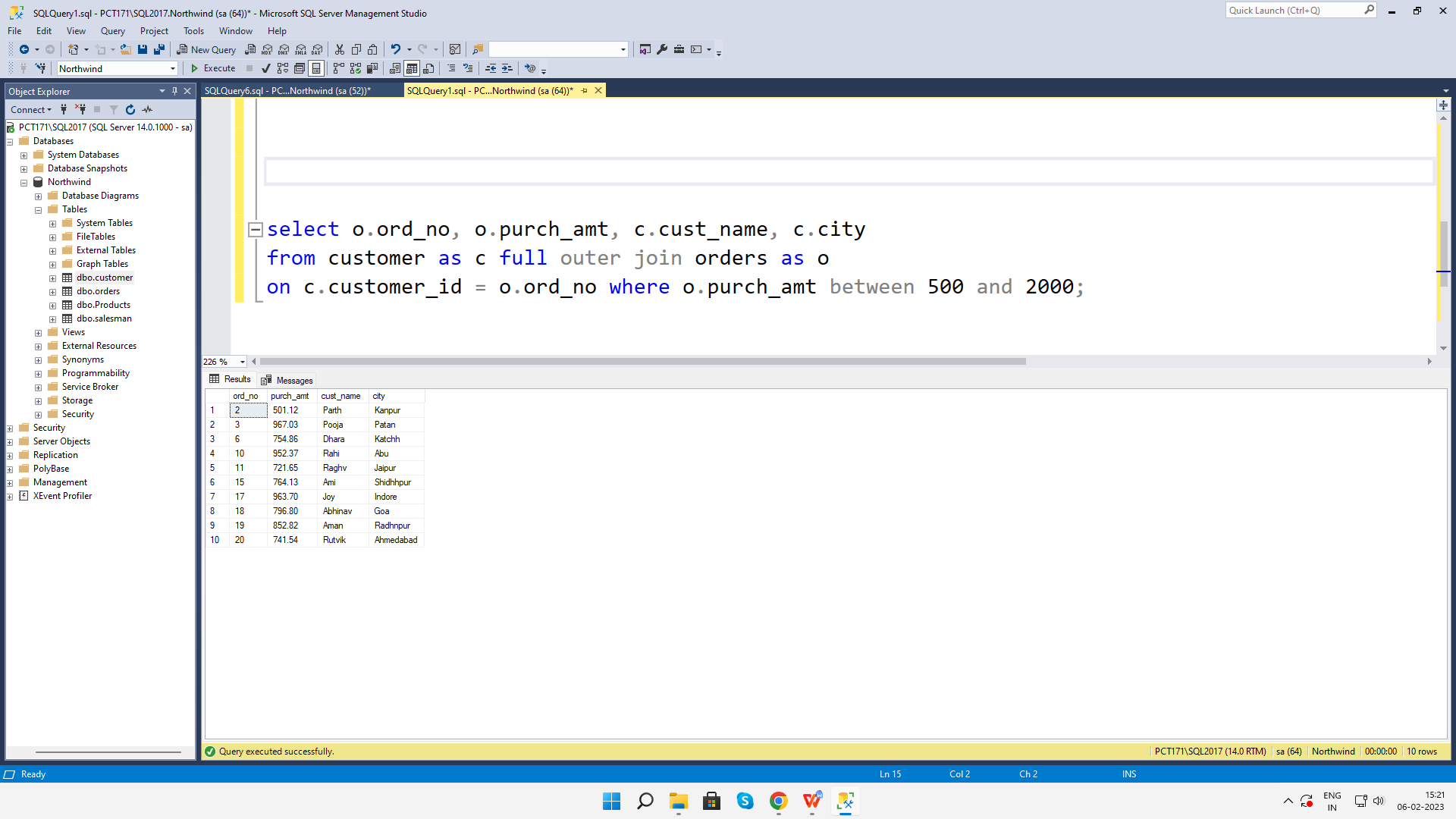


1. **write a SQL query to find those orders where the order amount exists between 500 and 2000. Return ord\_no, purch\_amt, cust\_name, city**

select o.ord\_no, o.purch\_amt, c.cust\_name, c.city

from customer as c full outer join orders as o

on c.customer\_id = o.ord\_no where o.purch\_amt between 500 and 2000;



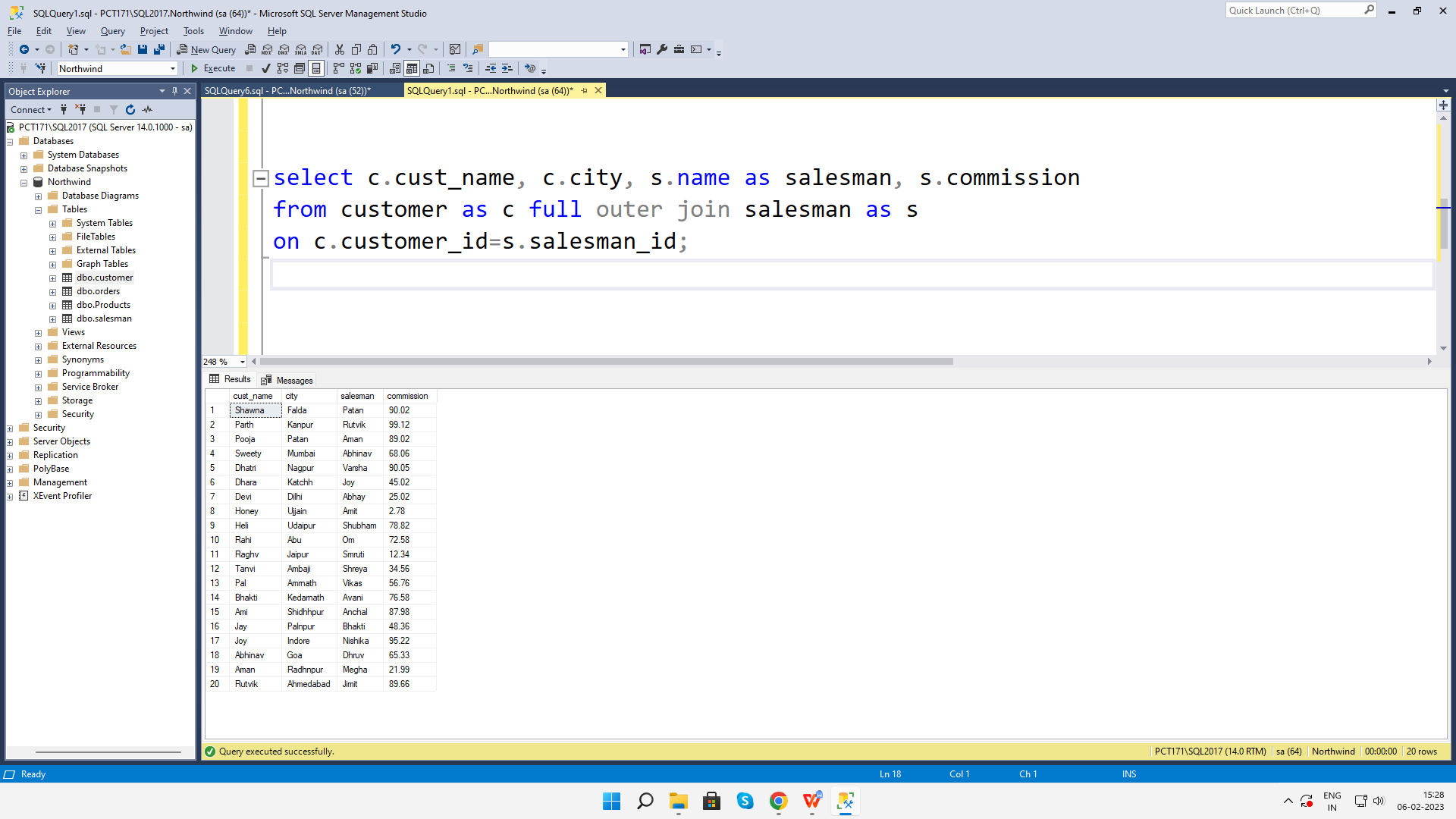
**3. write a SQL query to find the salesperson(s) and the customer(s) he represents.**

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

select c.cust\_name, c.city, s.name as salesman, s.commission

from customer as c full outer join salesman as s

on c.customer\_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,**

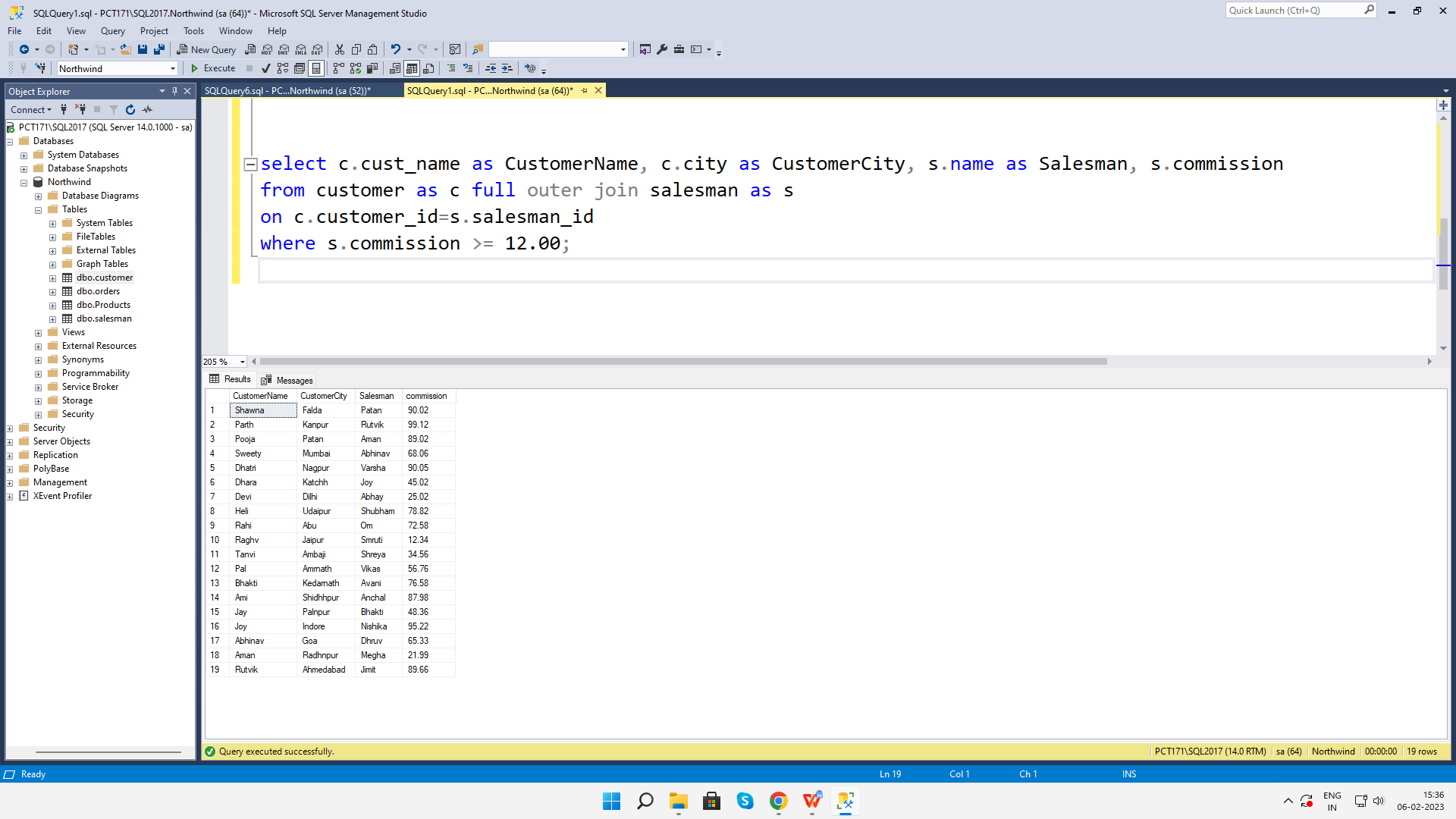
**commission.**

select c.cust\_name as CustomerName, c.city as CustomerCity, s.name as Salesman, s.commission

from customer as c full outer join salesman as s

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

where s.commission >= 12.00;



1. **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**

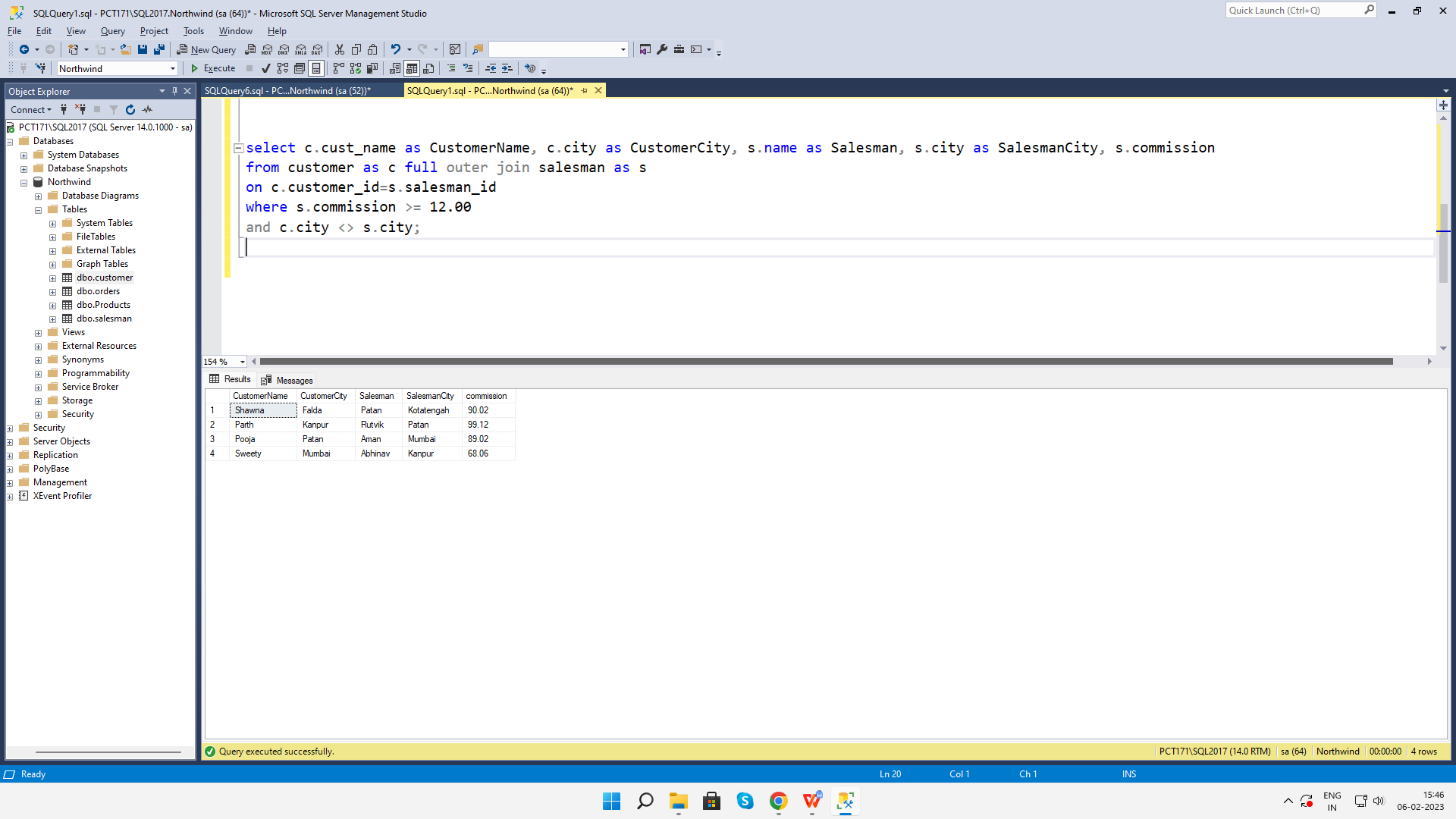
select c.cust\_name as CustomerName, c.city as CustomerCity, s.name as Salesman, s.city as SalesmanCity, s.commission

from customer as c full outer join salesman as s

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

where s.commission >= 12.00

and c.city <> s.city;

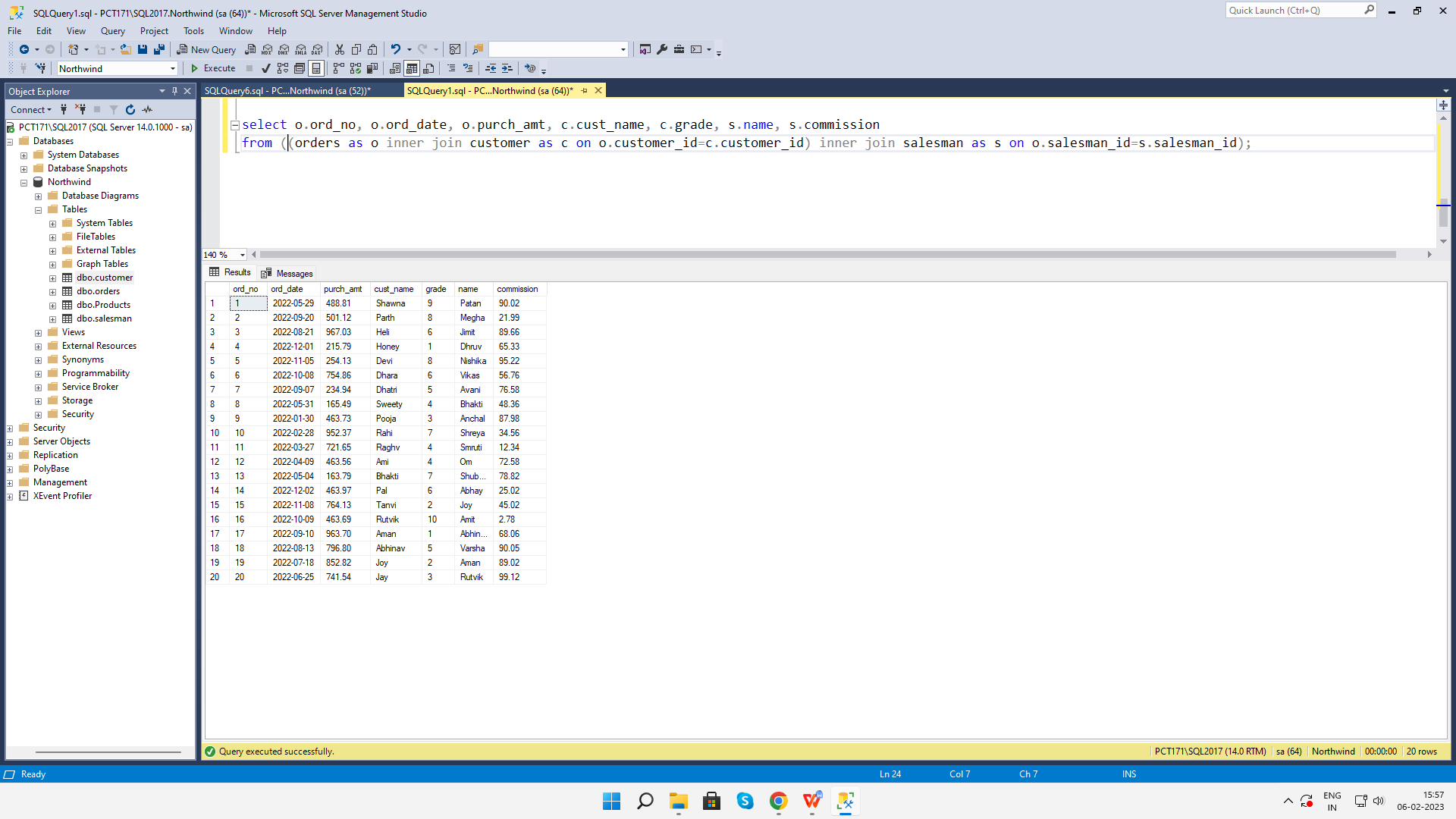


**6. write a SQL query to find the details of an order. Return ord\_no, ord\_date,**

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

select o.ord\_no, o.ord\_date, o.purch\_amt, c.cust\_name, c.grade, s.name, s.commission

from ((orders as o inner join customer as c on o.customer\_id=c.customer\_id) inner join salesman as s on o.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.**

SELECT \* FROM orders NATURAL JOIN customer NATURAL JOIN salesman;

**8. write a SQL query to display the customer name, customer city, grade, salesman,**

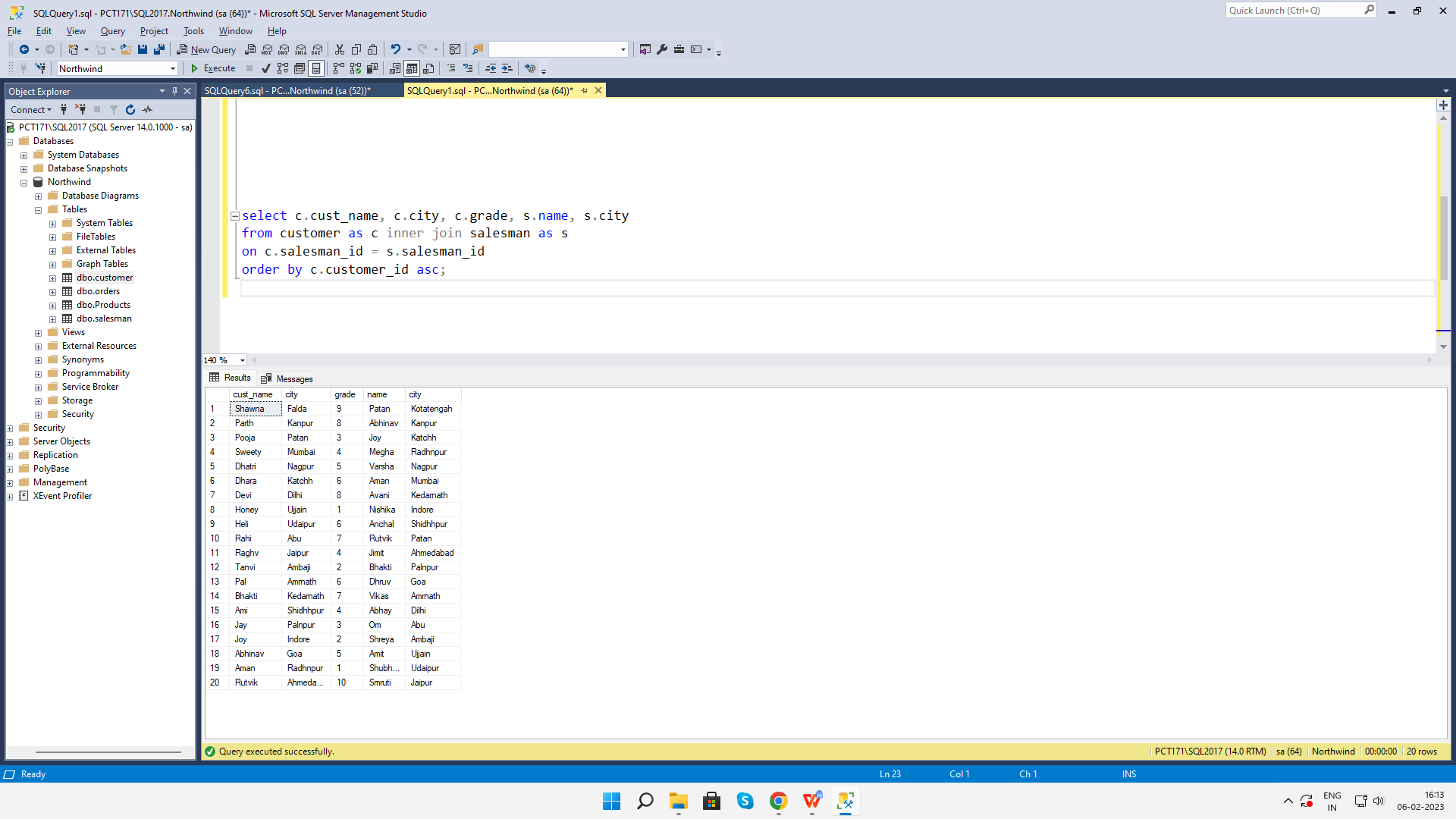
**salesman city. The results should be sorted by ascending customer\_id.**

select c.cust\_name, c.city, c.grade, s.name, s.city

from customer as c inner join salesman as s

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

order by c.customer\_id asc;



**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.**

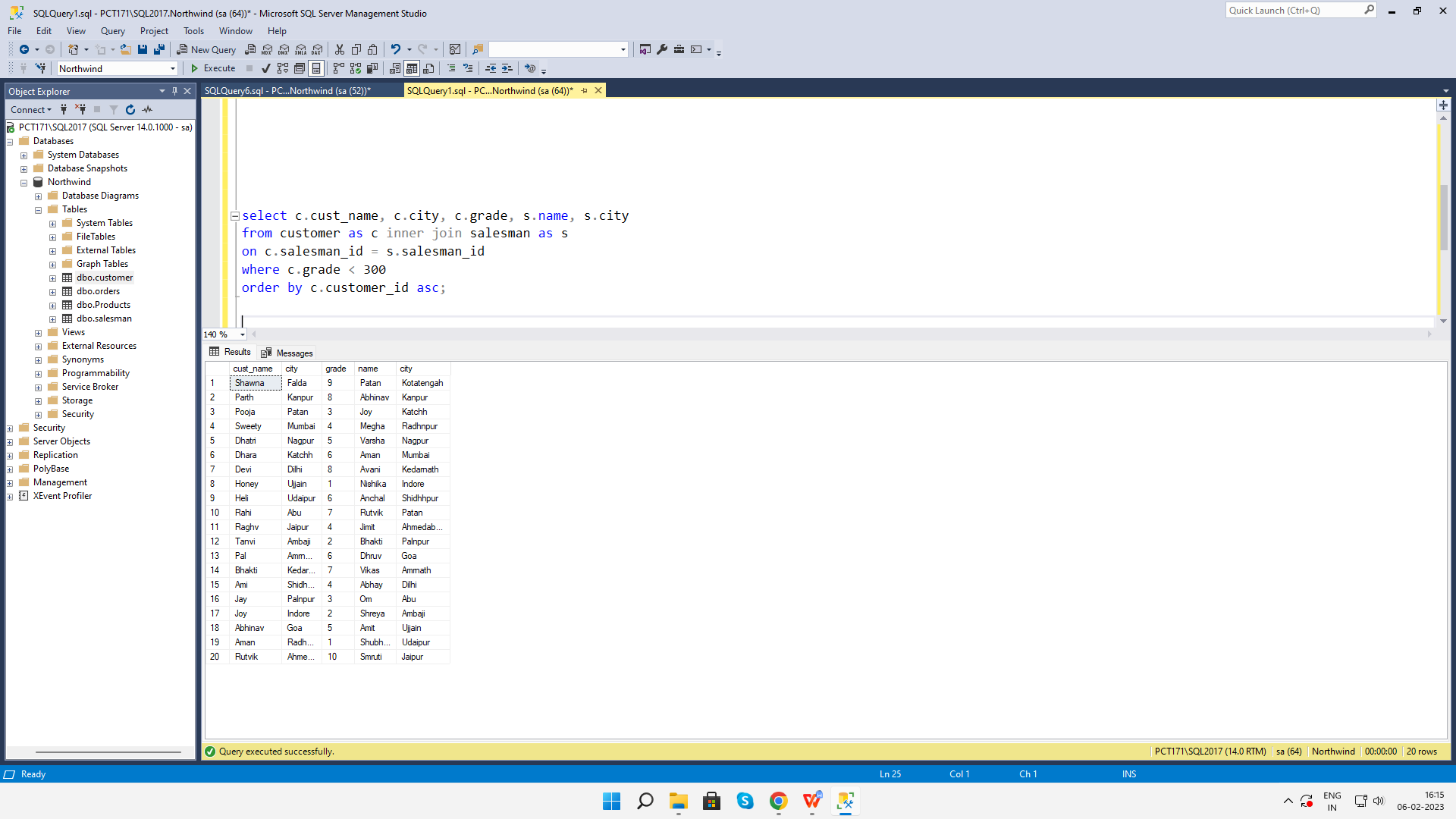
select c.cust\_name, c.city, c.grade, s.name, s.city

from customer as c inner join salesman as s

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

where c.grade < 300

order by c.customer\_id asc;



**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**

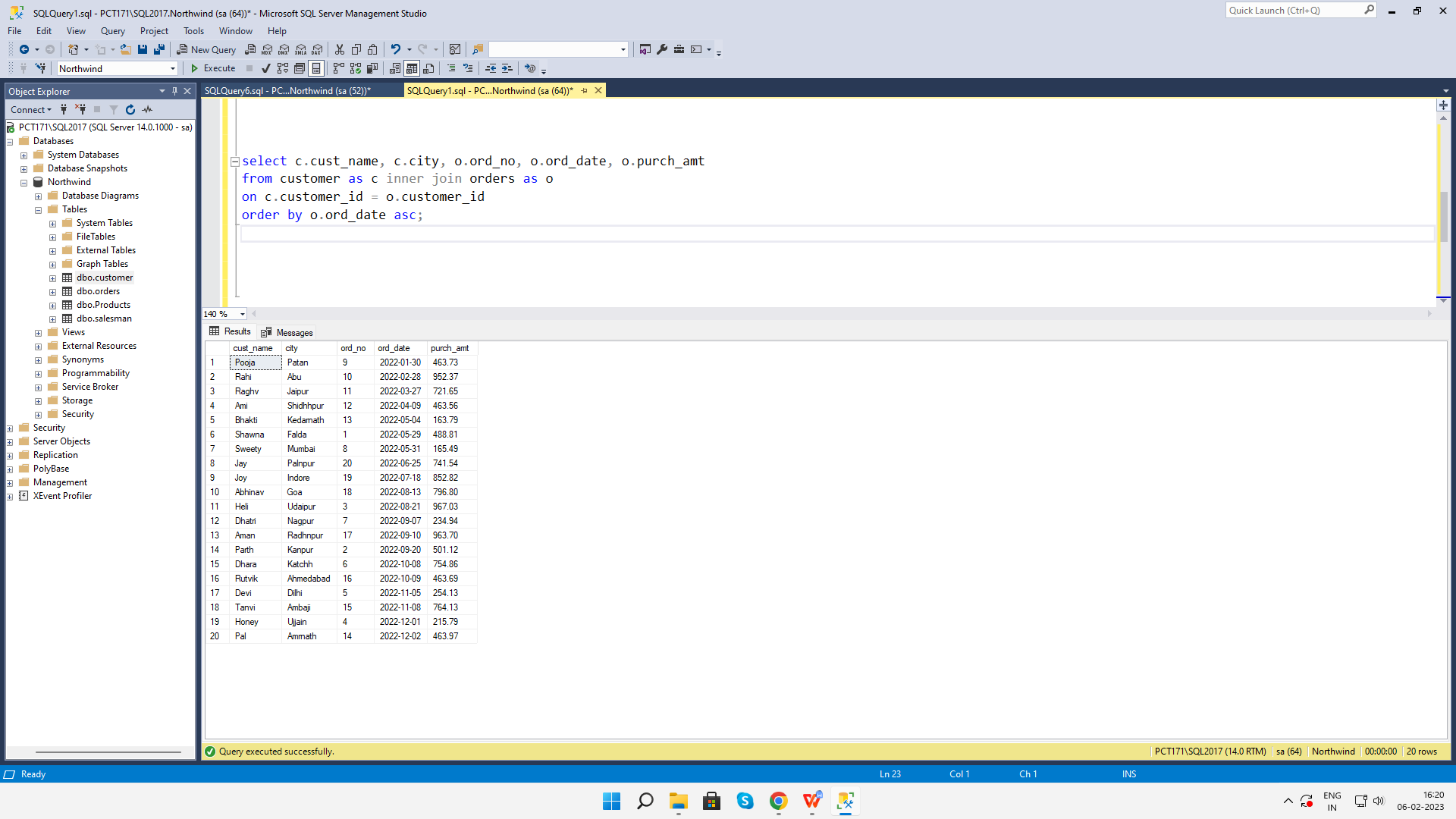
**determine whether any of the existing customers have placed an order or not**

select c.cust\_name, c.city, o.ord\_no, o.ord\_date, o.purch\_amt

from customer as c inner join orders as o

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

order by o.ord\_date asc;



1. **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**

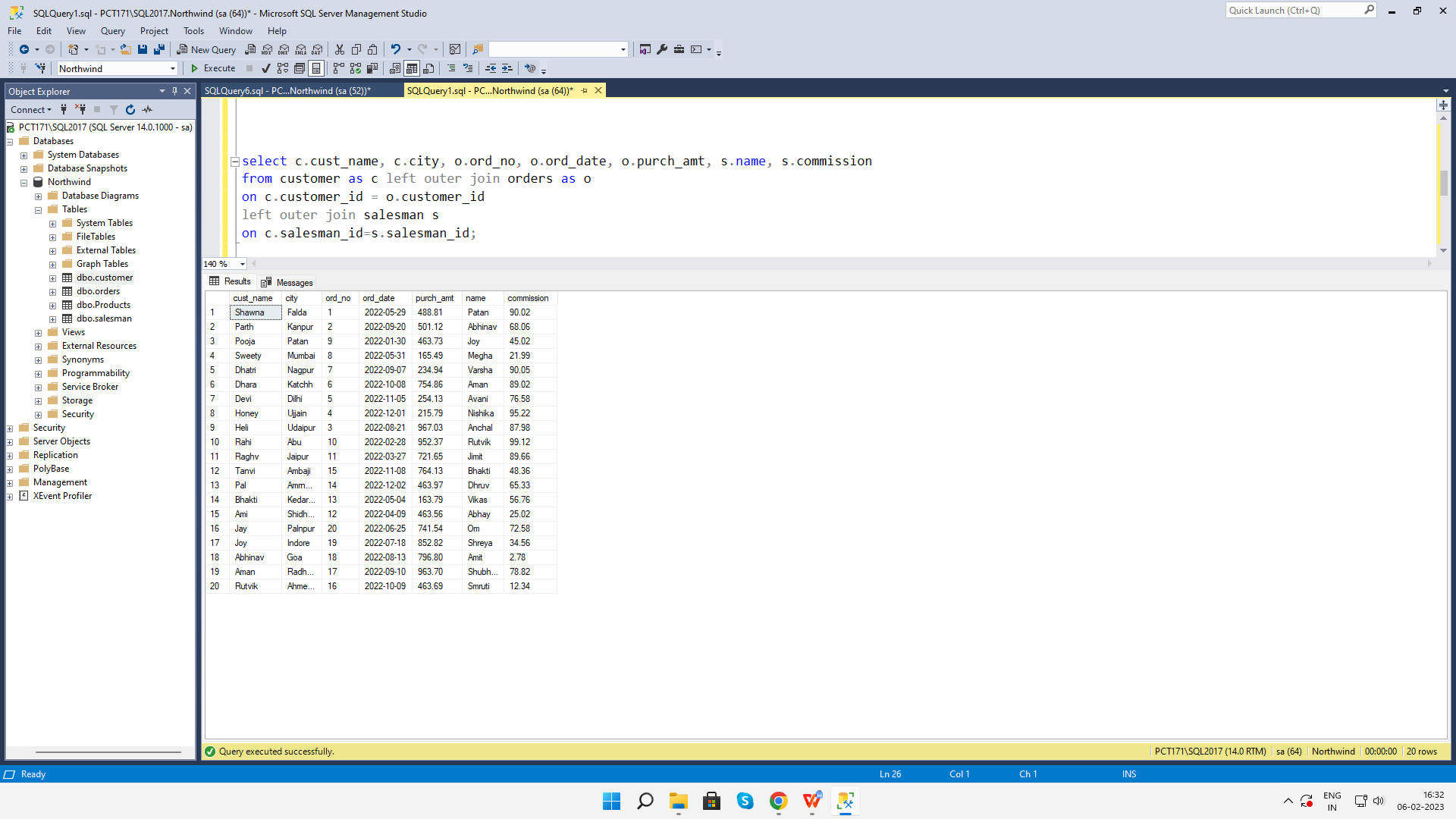
select c.cust\_name, c.city, o.ord\_no, o.ord\_date, o.purch\_amt, s.name, s.commission

from customer as c left outer join orders as o

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

left outer join salesman s

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



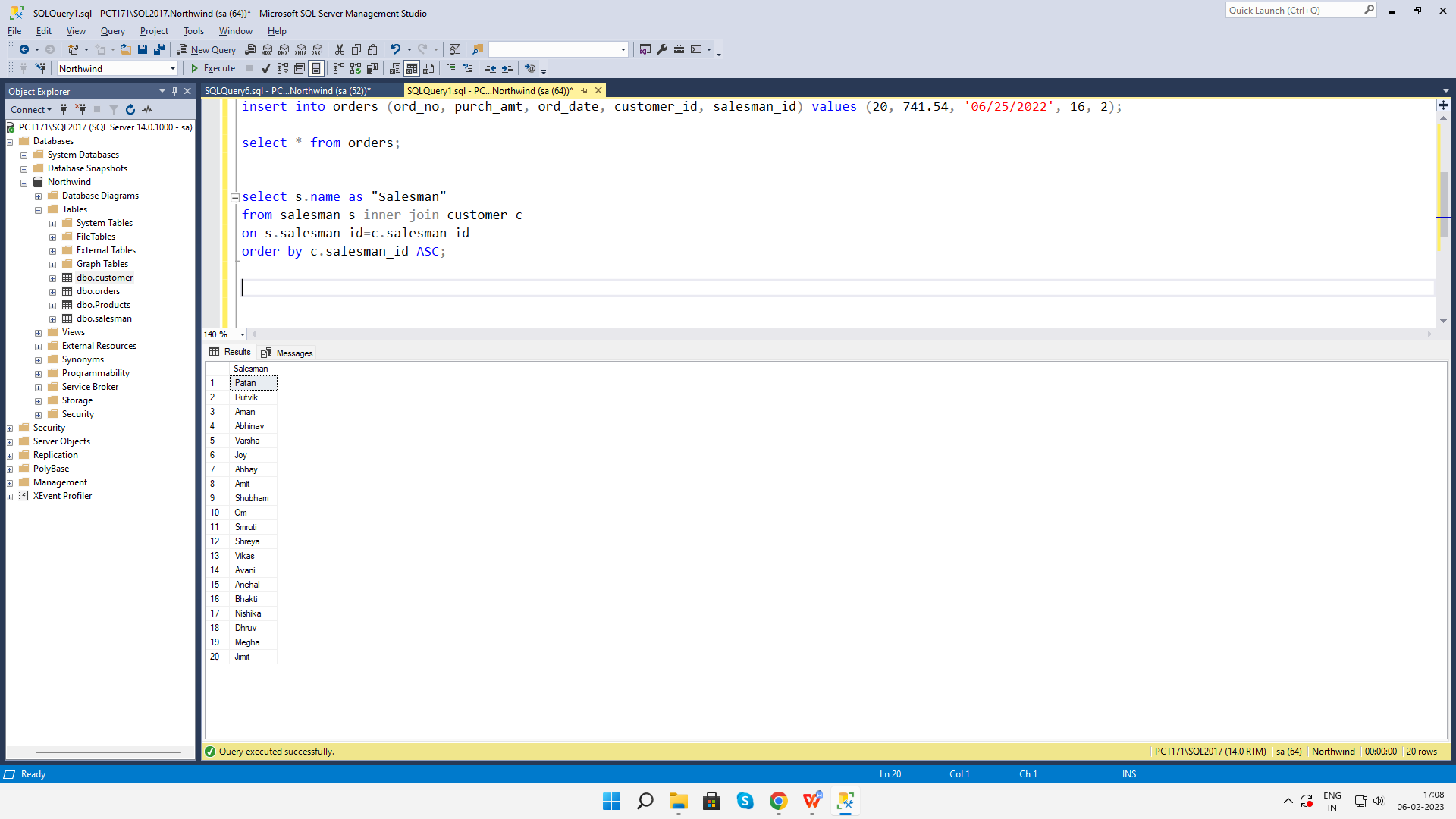
1. **Write a SQL statement to generate a list in ascending order of salespersons who work either for one or more customers or have not yet joined any of the customers**

select s.name as "Salesman"

from salesman s inner join customer c

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

order by c.salesman\_id ASC;



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

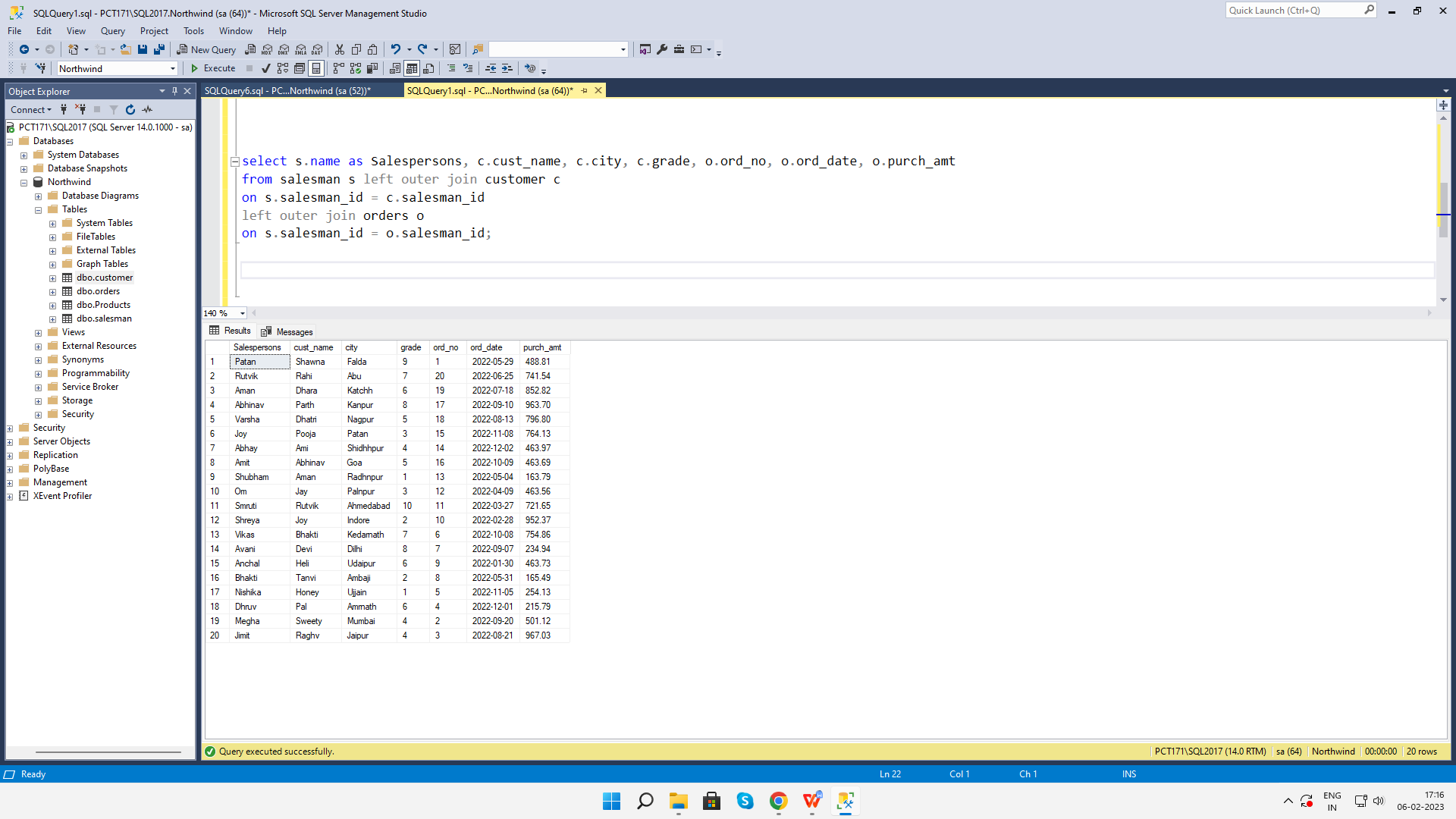
select s.name as Salespersons, c.cust\_name, c.city, c.grade, o.ord\_no, o.ord\_date, o.purch\_amt

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;



1. **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.name

from salesman s left outer join customer c

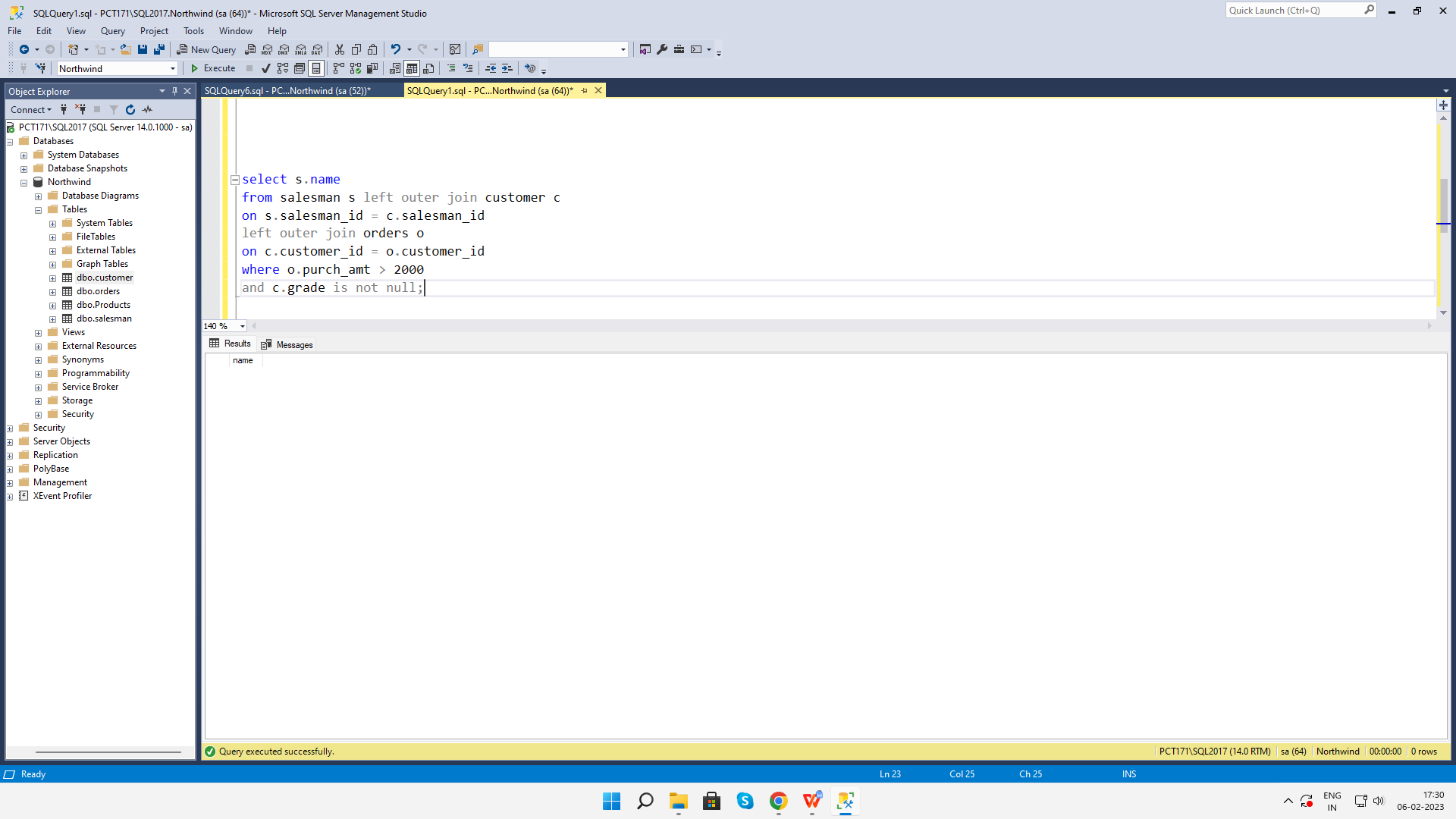
on s.salesman\_id = c.salesman\_id

left outer join orders o

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

where o.purch\_amt >= 2000

and c.grade is not null;



1. **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.name

from salesman s left outer join customer c

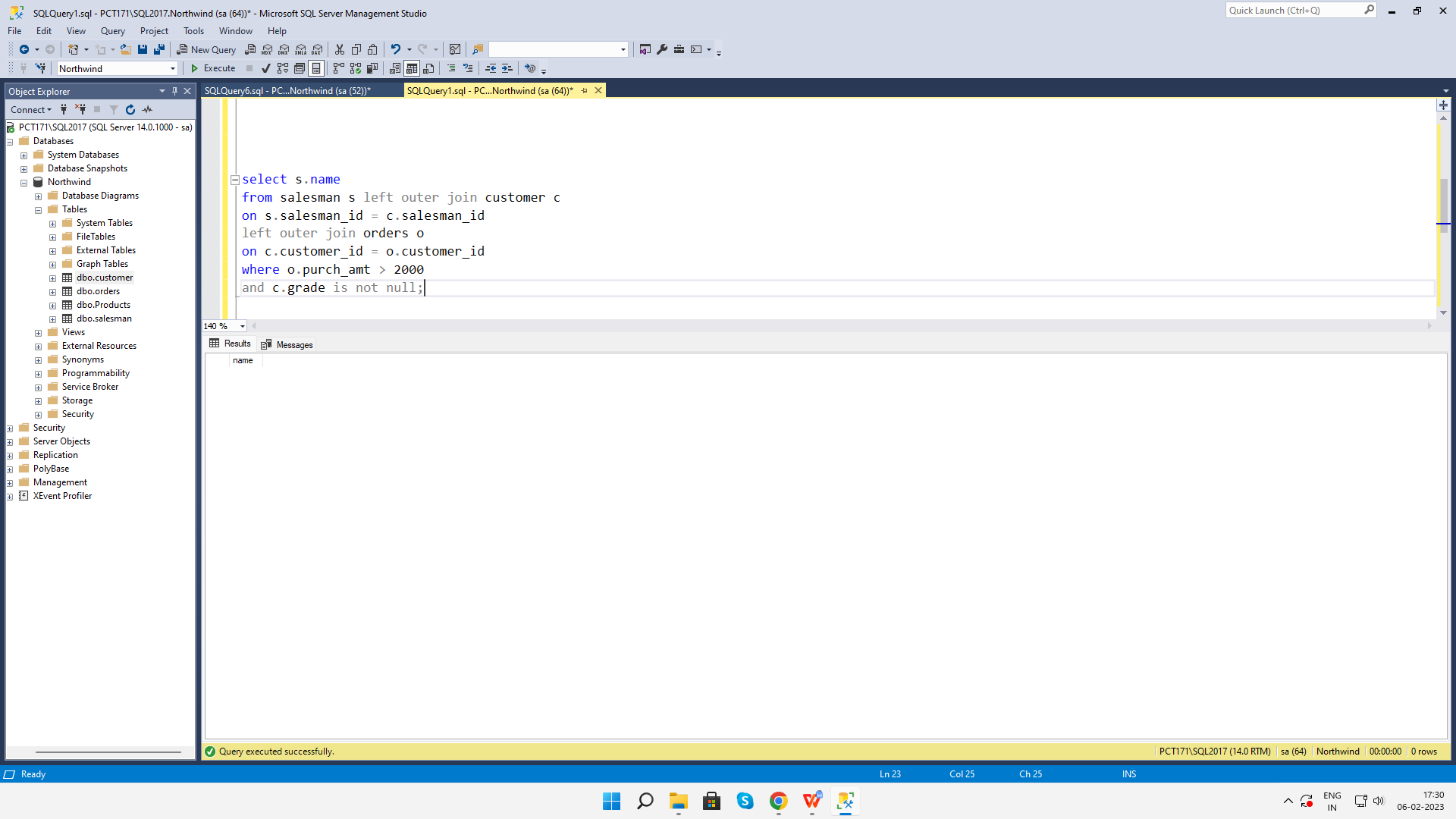
on s.salesman\_id = c.salesman\_id

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**

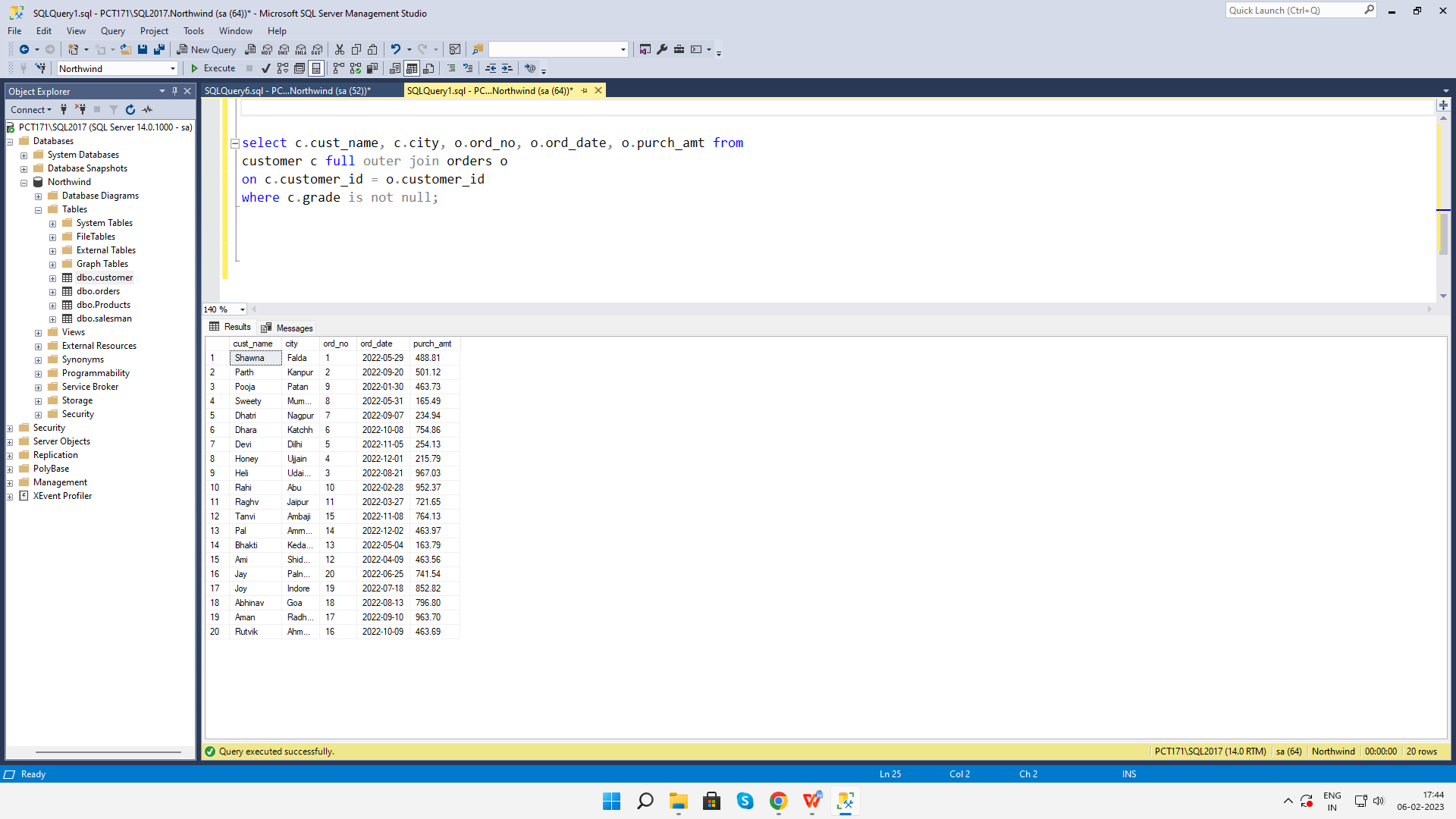
**customer who neither is on the list nor has a grade.**

select c.cust\_name, c.city, o.ord\_no, o.ord\_date, o.purch\_amt from

customer c full outer join orders o

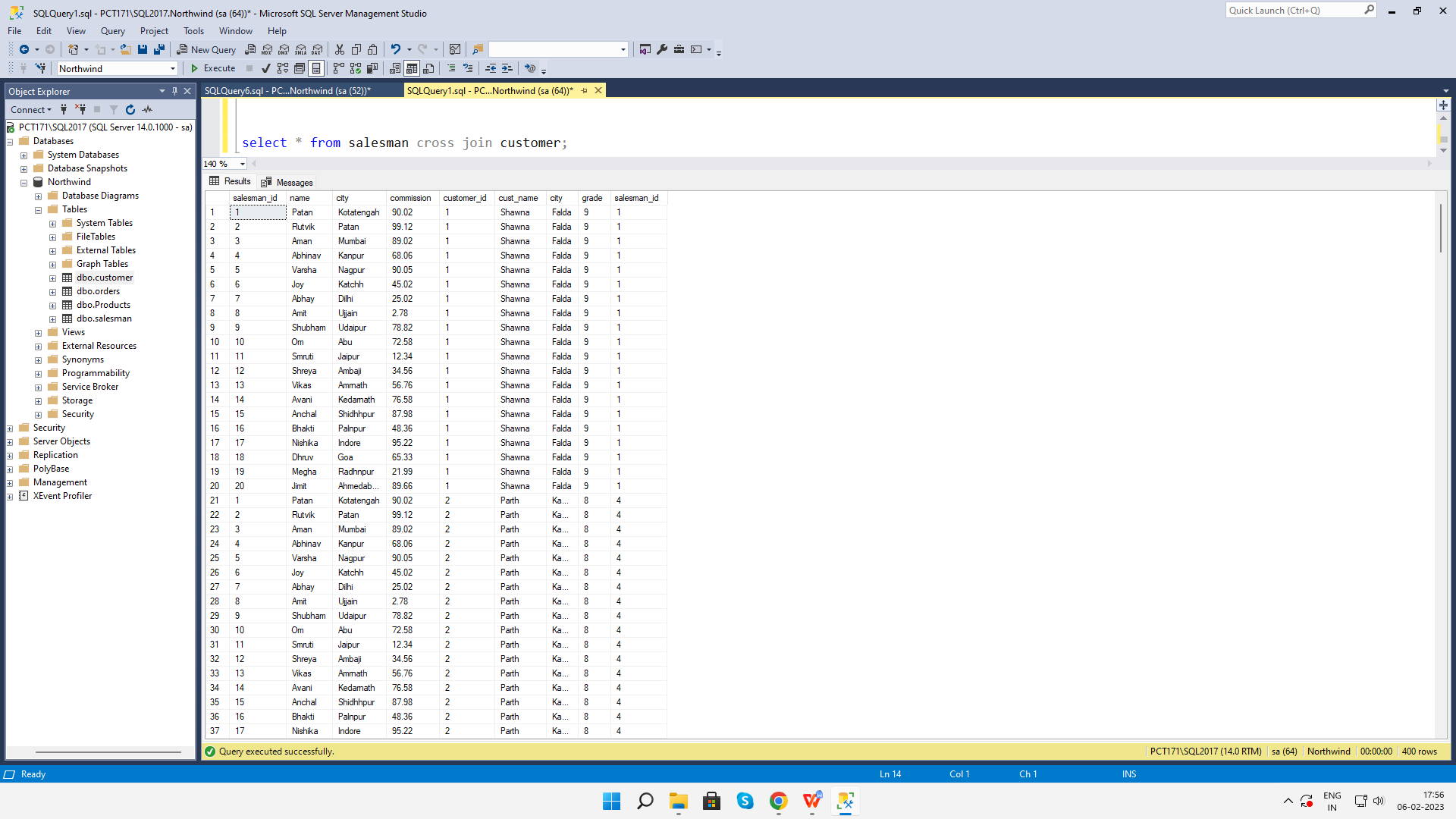
on c.customer\_id = o.customer\_id

where c.grade is not null;



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

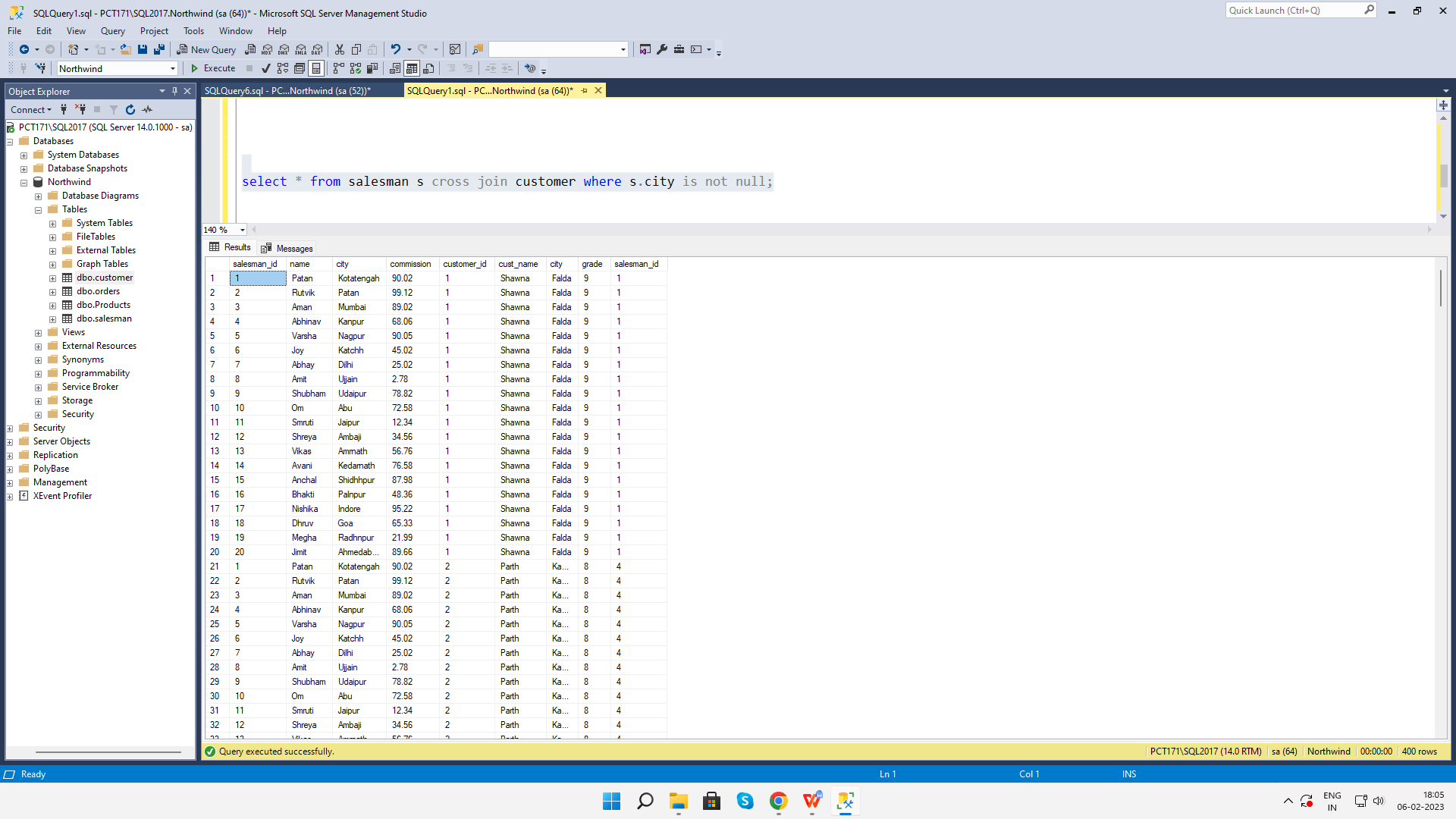
select \* from salesman cross join customer;



**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 \* from salesman s cross join customer where s.city is not null;

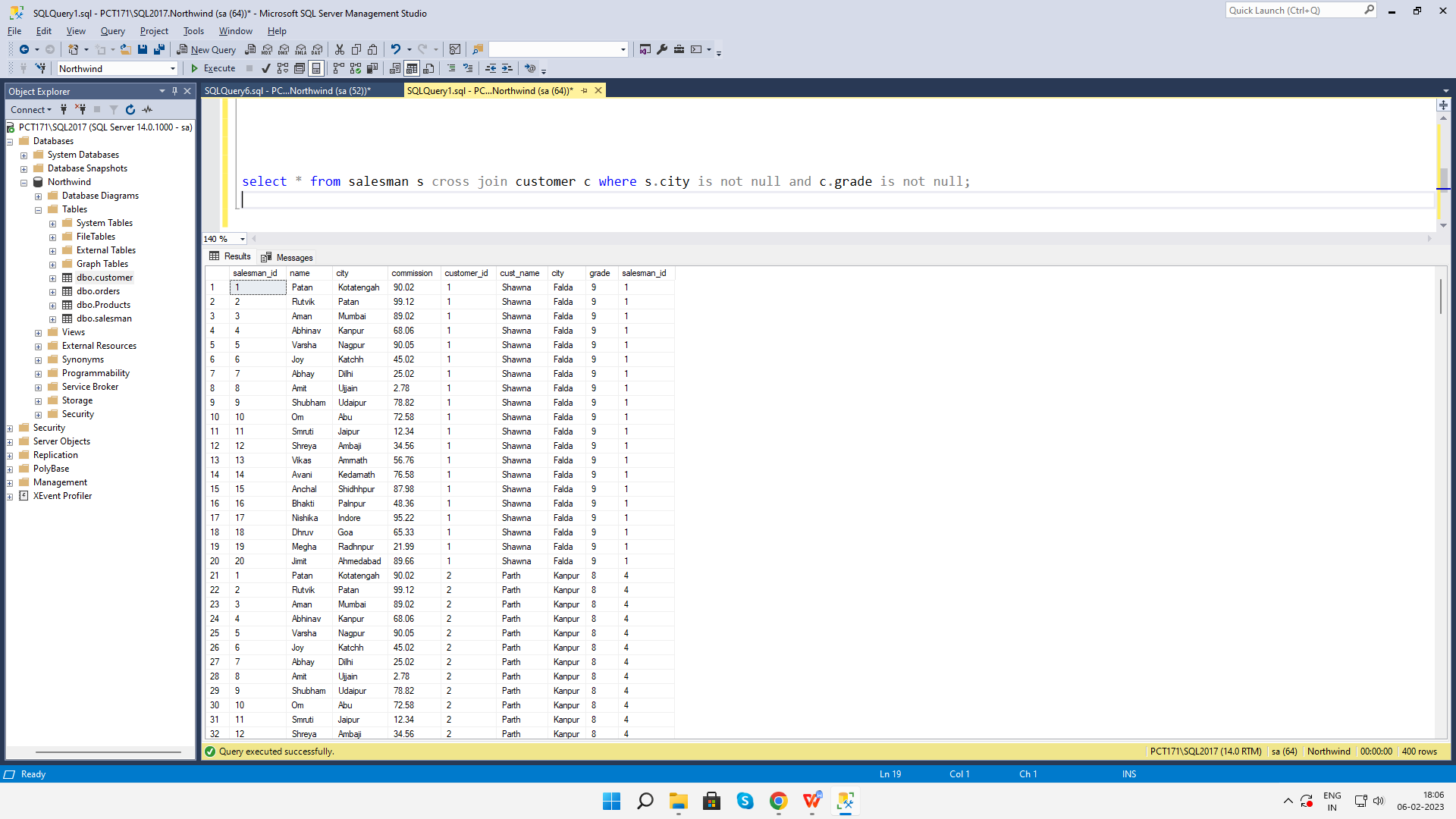


**19. Write a SQL statement to create a Cartesian product between salesperson and**

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

