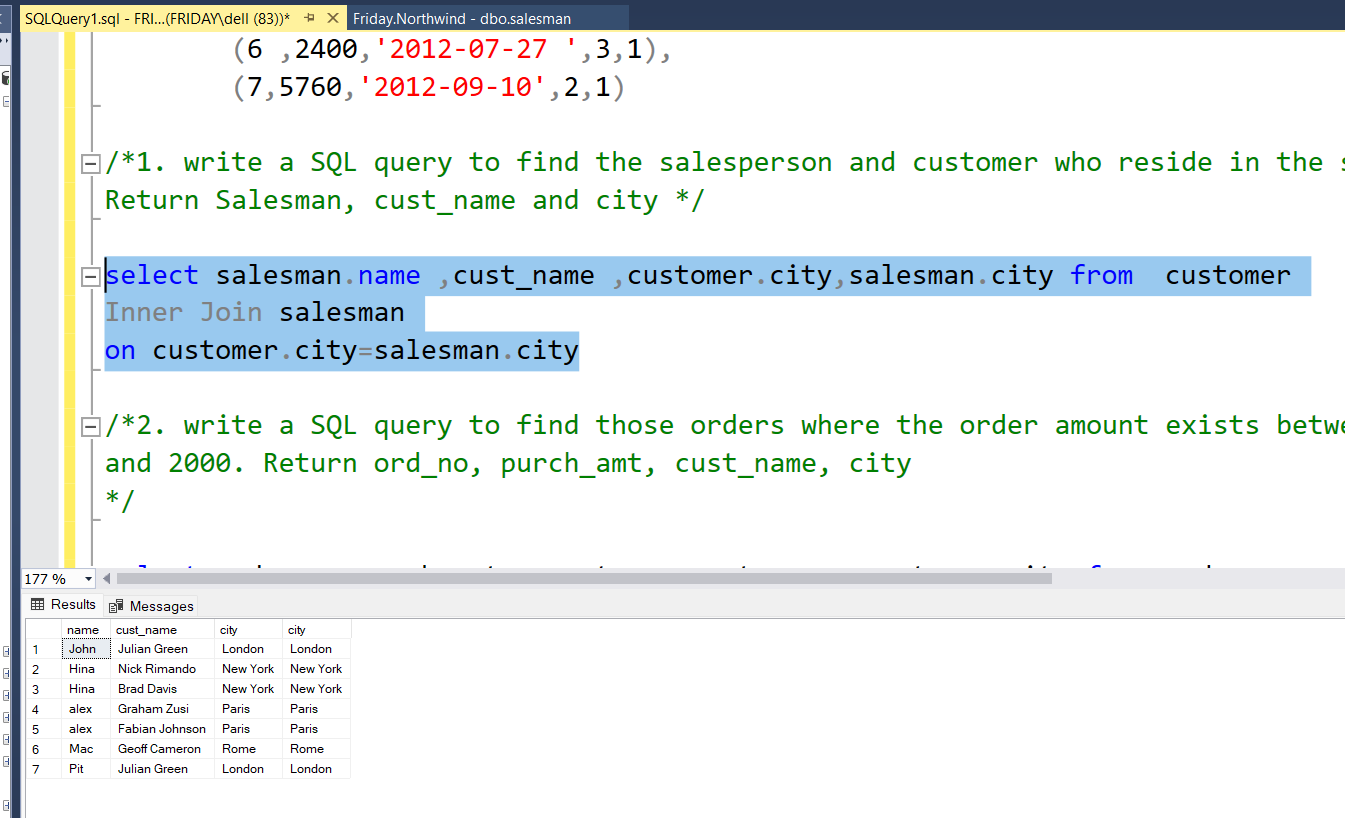
**SQL Assignment 2**

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 ,cust\_name ,customer.city,salesman.city from customer

Inner Join salesman

on customer.city=salesman.city



# 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 , punch\_amt , customer.cust\_name, customer.city from orders

INNER JOIN customer

on orders.customer\_id=customer.customer\_id

where punch\_amt BETWEEN 500 AND 2000

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

select customer.cust\_name ,customer.city ,name,commission from salesman

INNER JOIN customer

on salesman.salesman\_id=customer.salesman\_id

# 

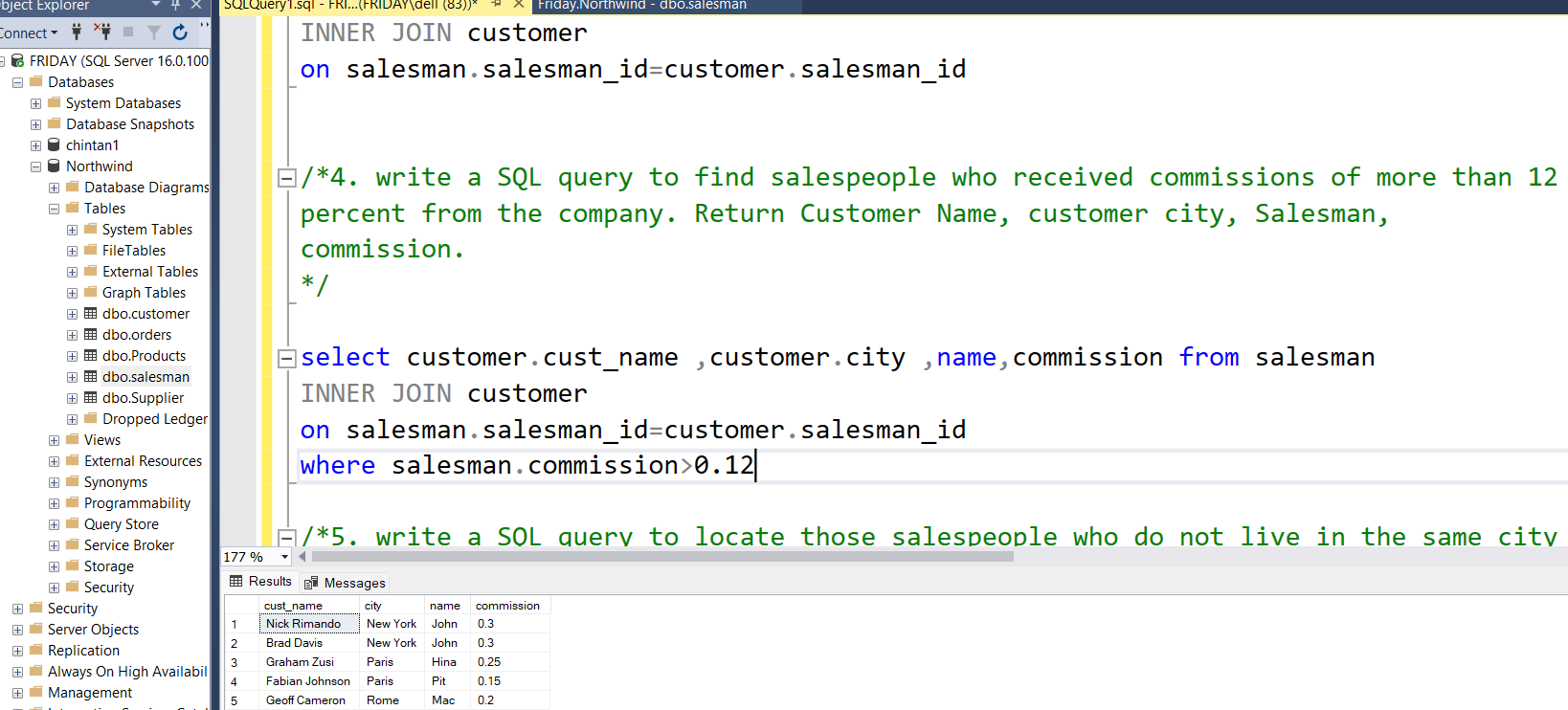
# write a SQL query to find salespeople who received commissions of more than 12 percent from the company. Return Customer Name, customer city, Salesman, commission.

select customer.cust\_name ,customer.city ,name,commission from salesman

INNER JOIN customer

on salesman.salesman\_id=customer.salesman\_id

where salesman.commission>0.12



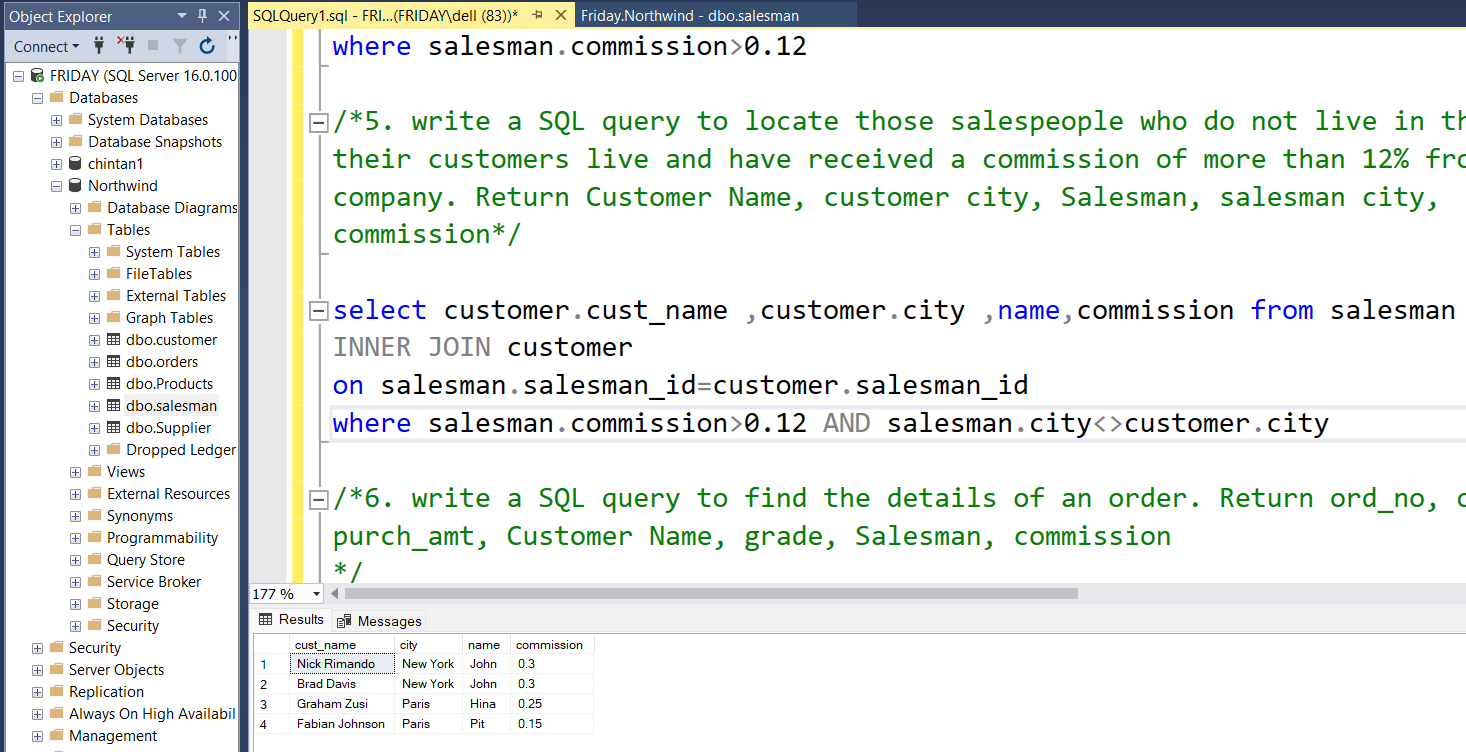
5.write a SQL query to locate those salespeople who do not live in the same city where their customers live and have received a commission of more than 12% from the company. Return Customer Name, customer city, Salesman, salesman city, commission

select customer.cust\_name ,customer.city ,name,commission from salesman

INNER JOIN customer

on salesman.salesman\_id=customer.salesman\_id

where salesman.commission>0.12 AND salesman.city<>customer.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

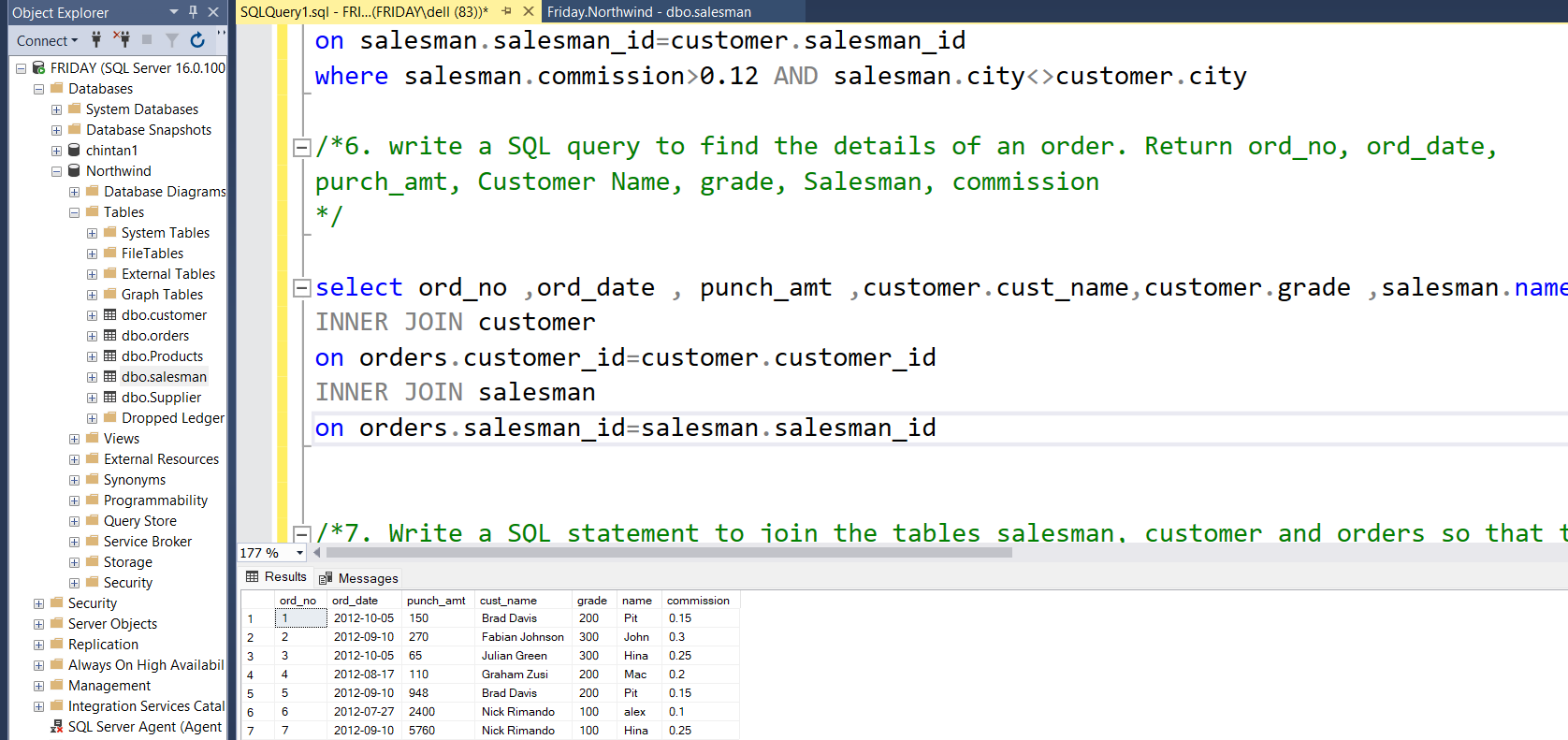
select ord\_no ,ord\_date , punch\_amt ,customer.cust\_name,customer.grade ,salesman.name ,salesman.commission from orders

INNER JOIN customer

on orders.customer\_id=customer.customer\_id

INNER JOIN salesman

on orders.salesman\_id=salesman.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.

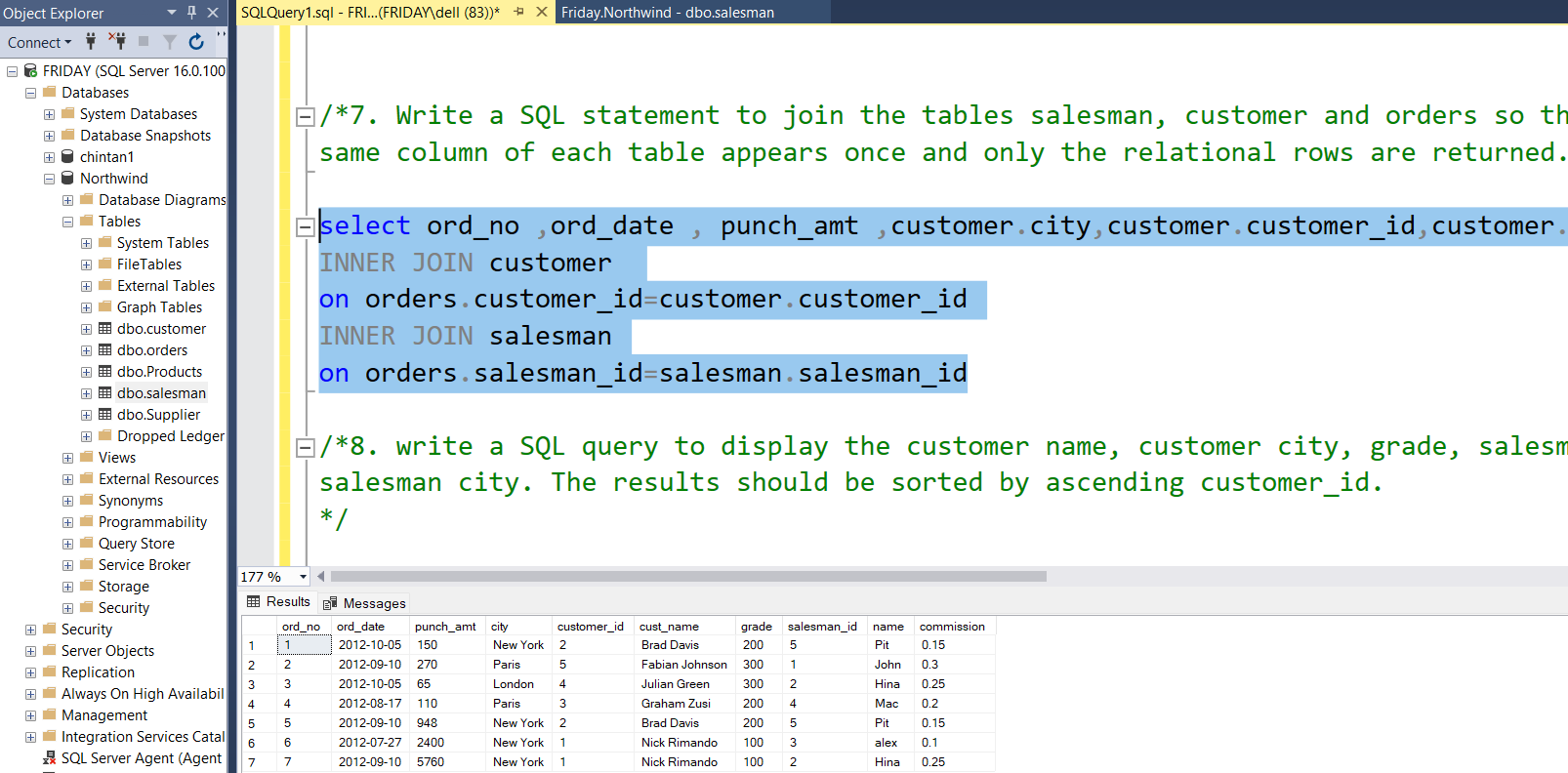
select ord\_no ,ord\_date , punch\_amt ,customer.city,customer.customer\_id,customer.cust\_name,customer.grade ,salesman.salesman\_id,salesman.name ,salesman.commission from orders

INNER JOIN customer

on orders.customer\_id=customer.customer\_id

INNER JOIN salesman

on orders.salesman\_id=salesman.salesman\_id



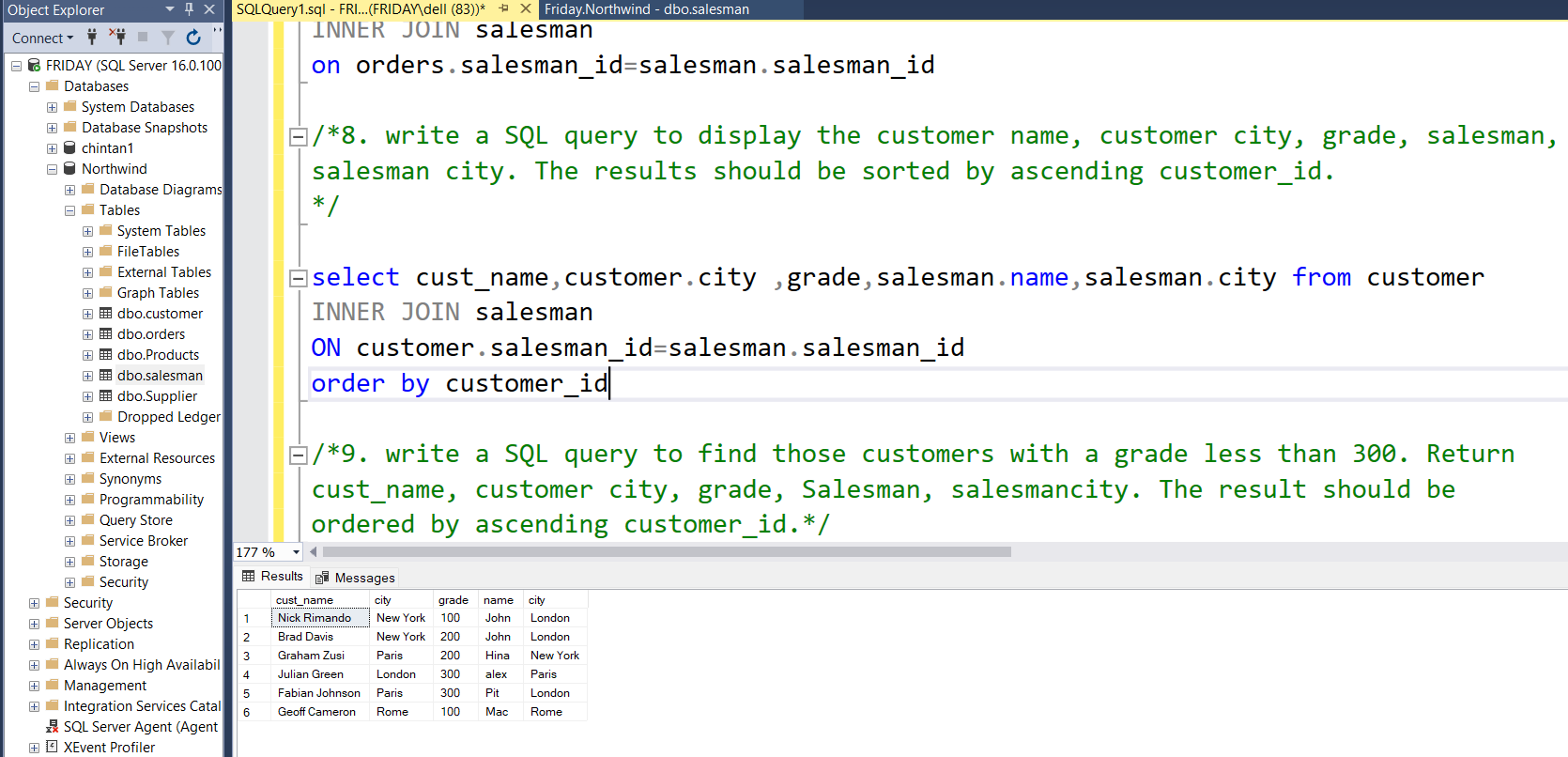
8. write a SQL query to display the customer name, customer city, grade, salesman, salesman city. The results should be sorted by ascending customer\_id.

select cust\_name,customer.city ,grade,salesman.name,salesman.city from customer

INNER JOIN salesman

ON customer.salesman\_id=salesman.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.

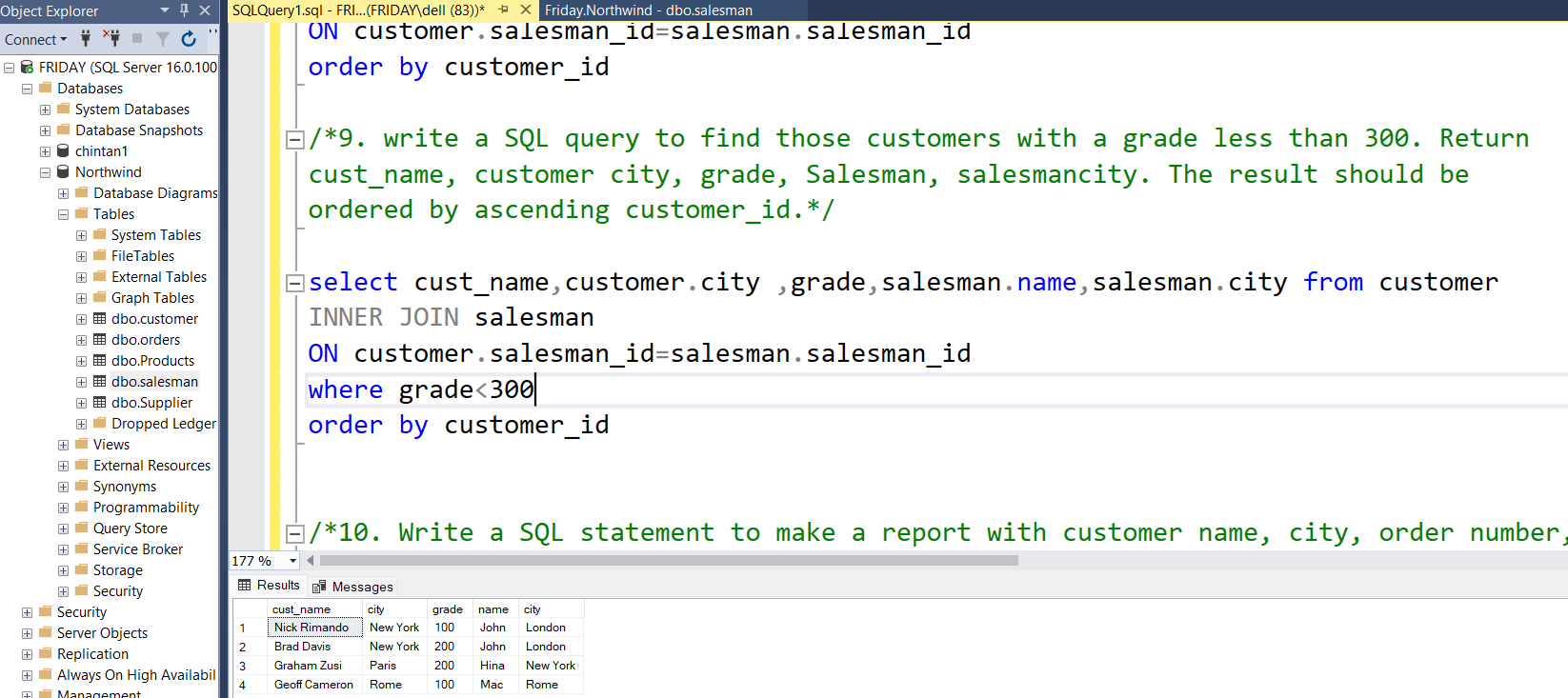
select cust\_name,customer.city ,grade,salesman.name,salesman.city from customer

INNER JOIN salesman

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

where grade<300

order by customer\_id



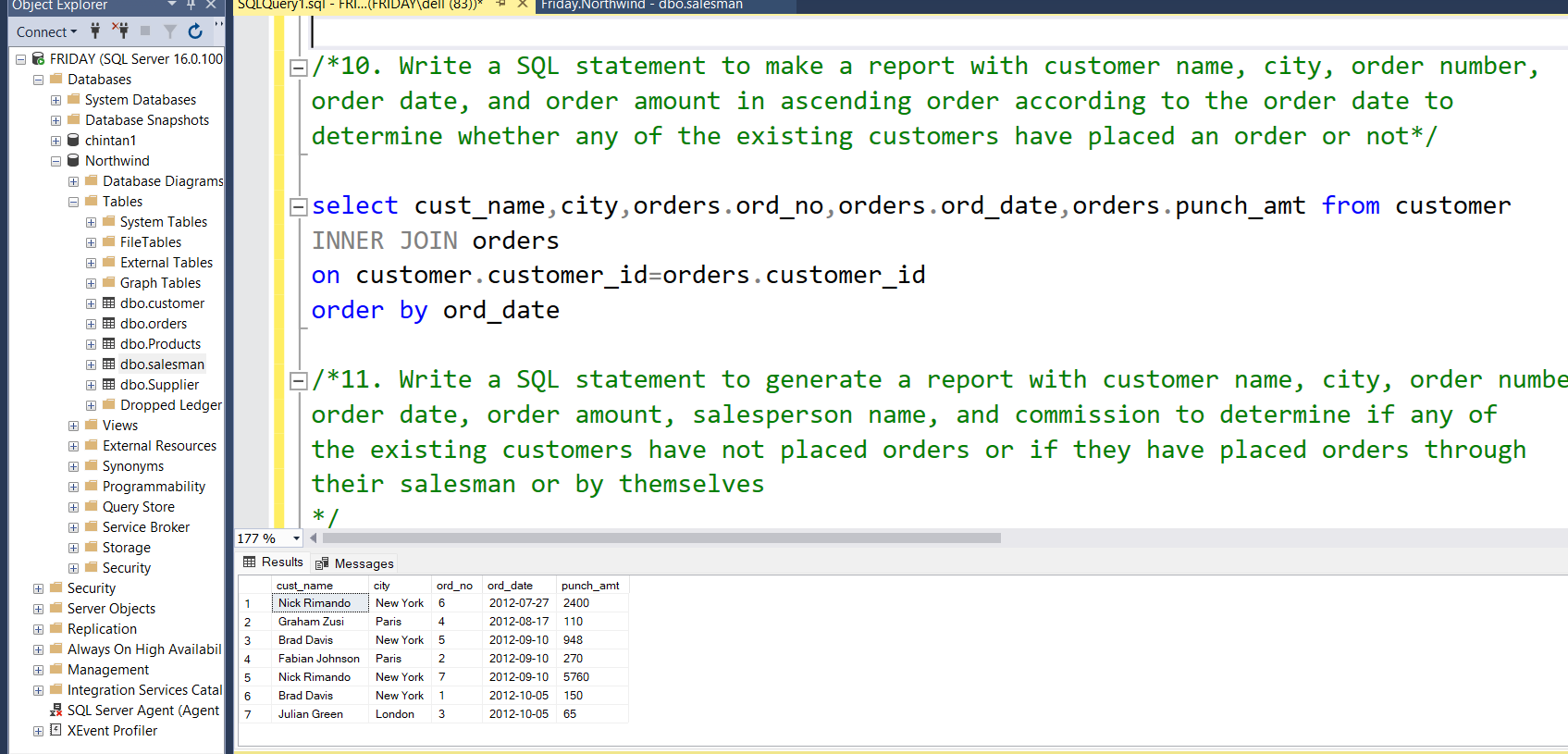
10. Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the order date to determine whether any of the existing customers have placed an order or not.

select cust\_name,city,orders.ord\_no,orders.ord\_date,orders.punch\_amt from customer

INNER JOIN orders

on customer.customer\_id=orders.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,customer.city,orders.ord\_no,orders.ord\_date,orders.punch\_amt, salesman.name,salesman.commission from customer

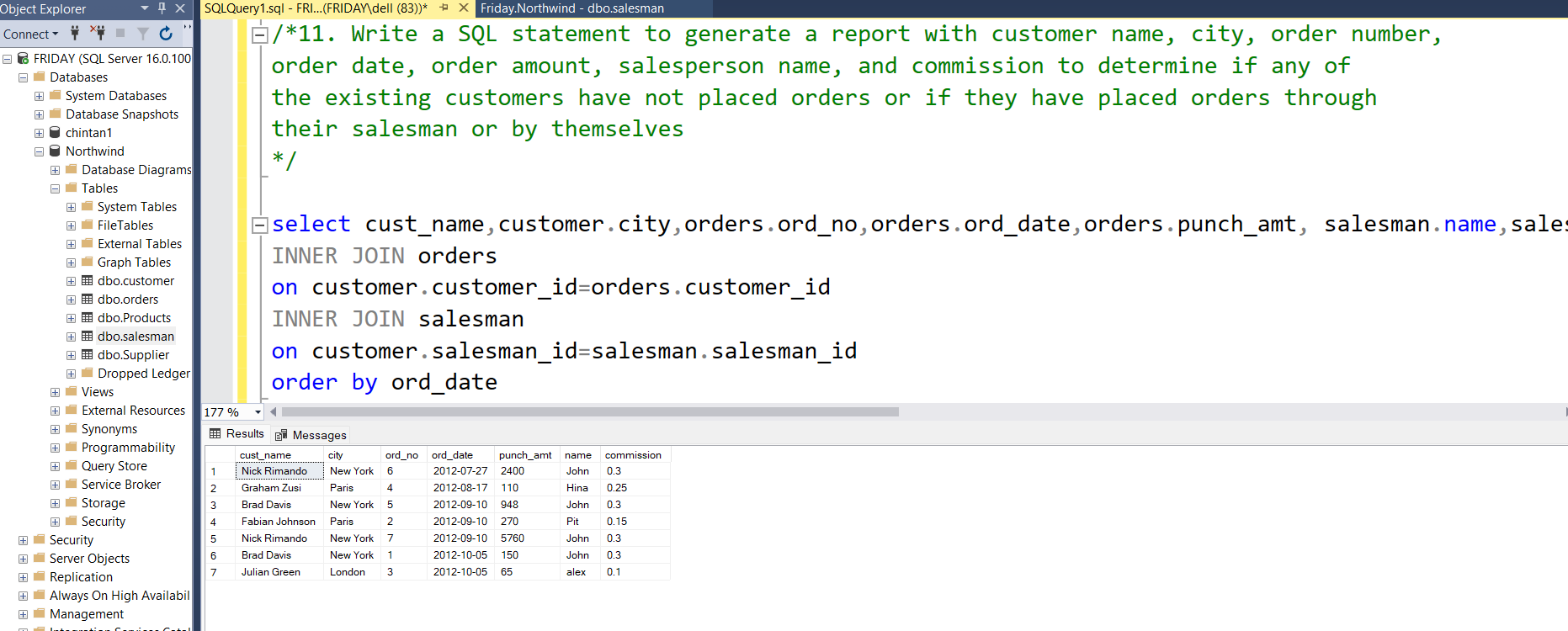
INNER JOIN orders

on customer.customer\_id=orders.customer\_id

INNER JOIN salesman

on customer.salesman\_id=salesman.salesman\_id

order by ord\_date



/\*12. Write a SQL statement to generate a list in ascending order of salespersons who

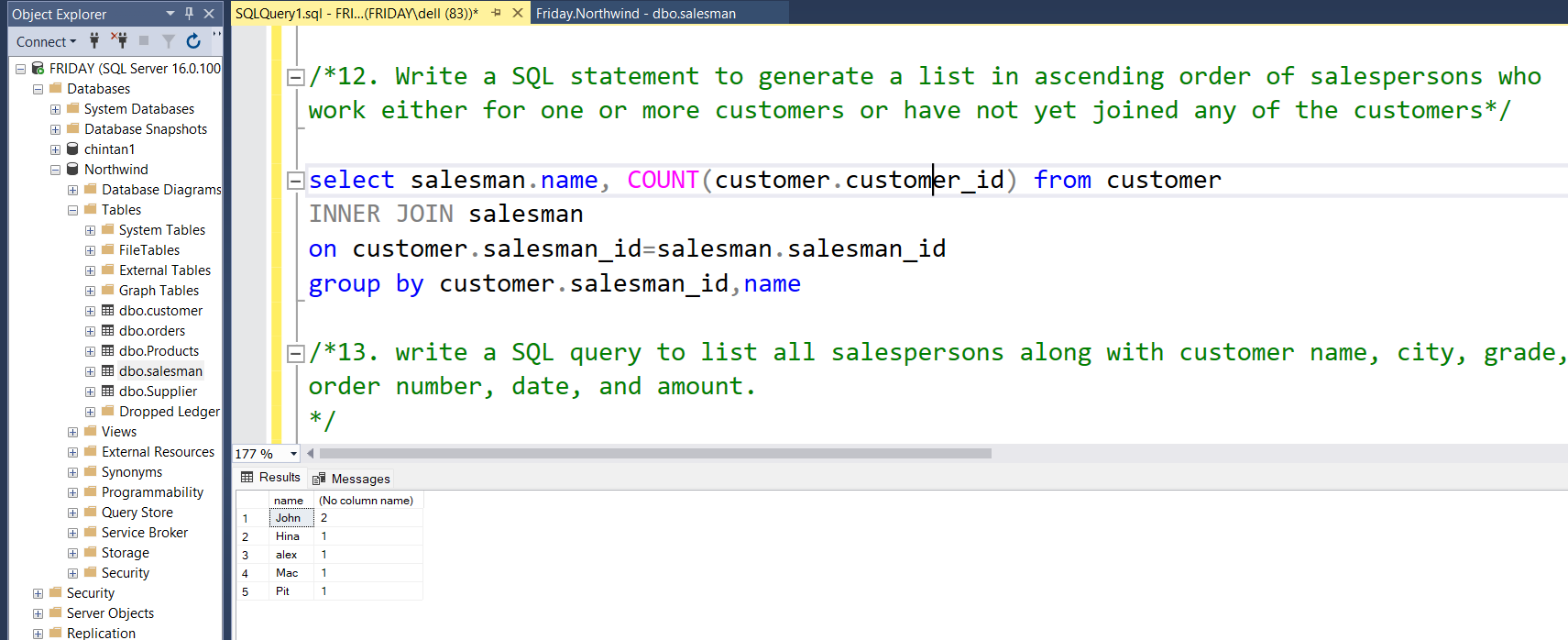
work either for one or more customers or have not yet joined any of the customers\*/

select salesman.name, COUNT(customer.customer\_id) from customer

INNER JOIN salesman

on customer.salesman\_id=salesman.salesman\_id

group by customer.salesman\_id,name



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

order number, date, and amount.

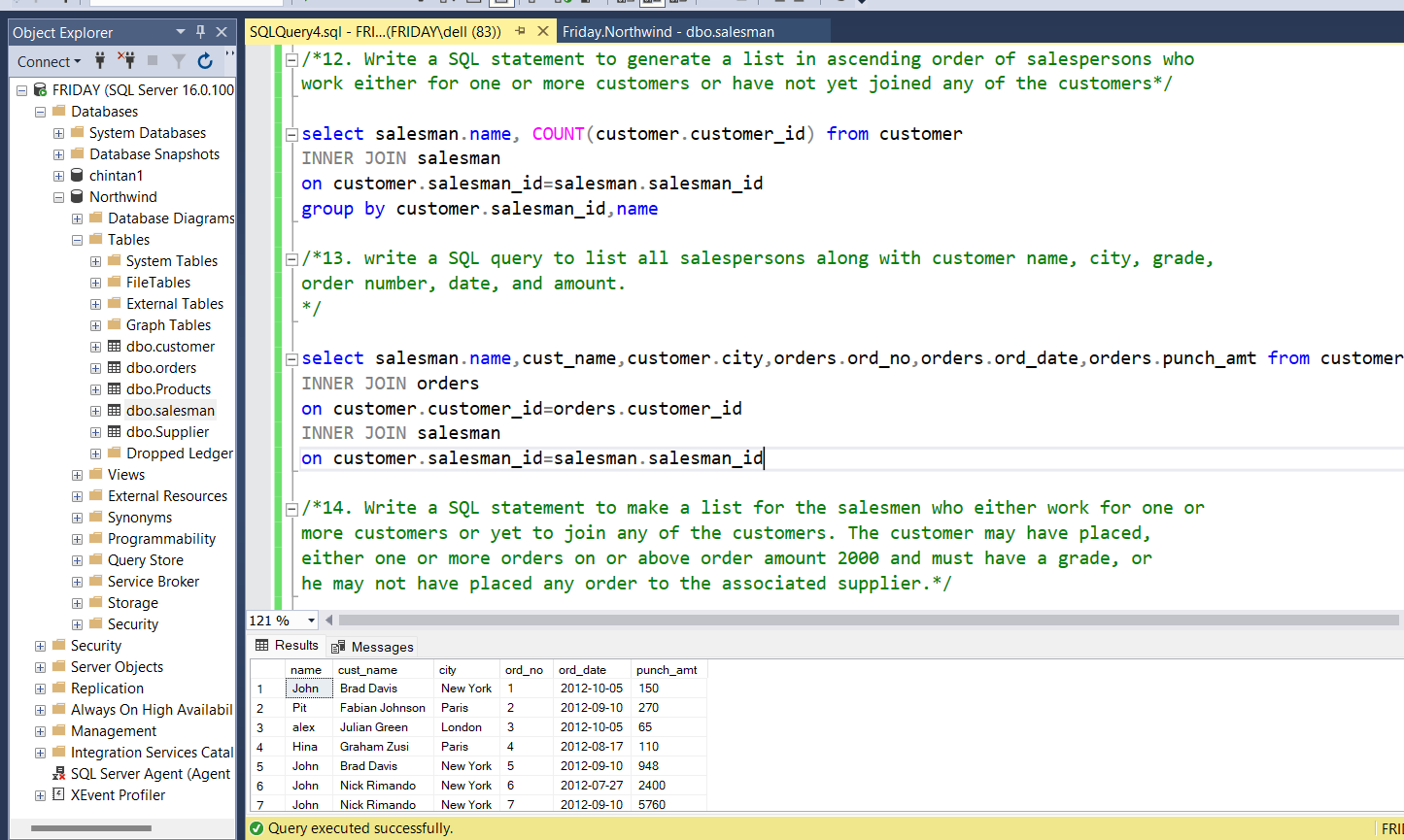
select salesman.name,cust\_name,customer.city,orders.ord\_no,orders.ord\_date,orders.punch\_amt from customer

INNER JOIN orders

on customer.customer\_id=orders.customer\_id

INNER JOIN salesman

on customer.salesman\_id=salesman.salesman\_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 salesman.name, COUNT(customer.customer\_id) from customer

INNER JOIN salesman

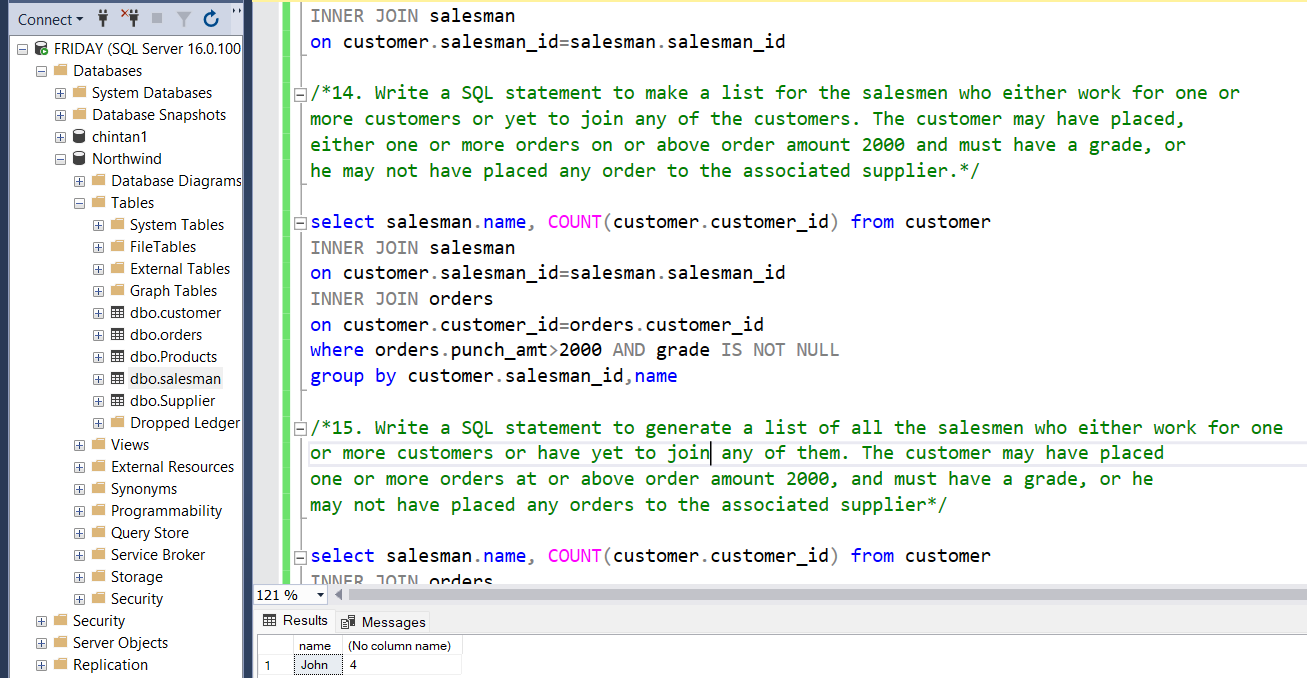
on customer.salesman\_id=salesman.salesman\_id

INNER JOIN orders

on customer.customer\_id=orders.customer\_id

where orders.punch\_amt>2000 AND grade IS NOT NULL

group by customer.salesman\_id,name



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 salesman.name, COUNT(customer.customer\_id) from customer

INNER JOIN orders

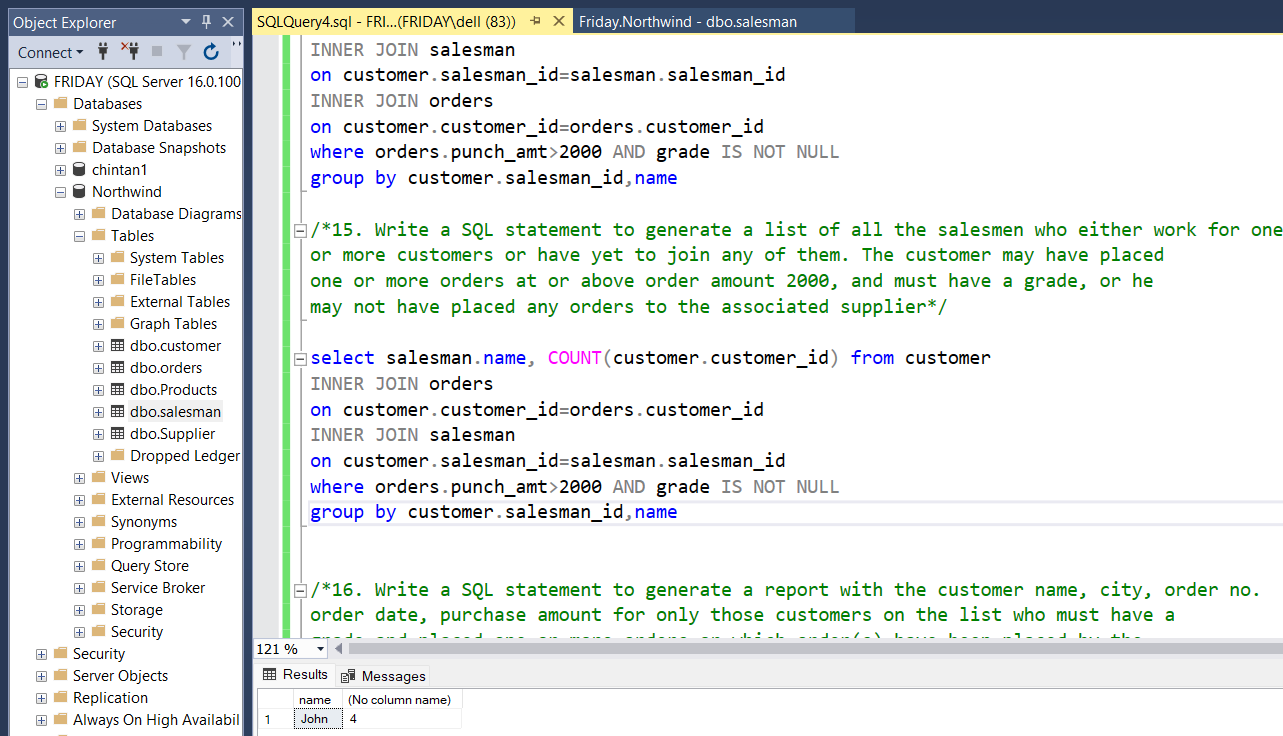
on customer.customer\_id=orders.customer\_id

INNER JOIN salesman

on customer.salesman\_id=salesman.salesman\_id

where orders.punch\_amt>2000 AND grade IS NOT NULL

group by customer.salesman\_id,name



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

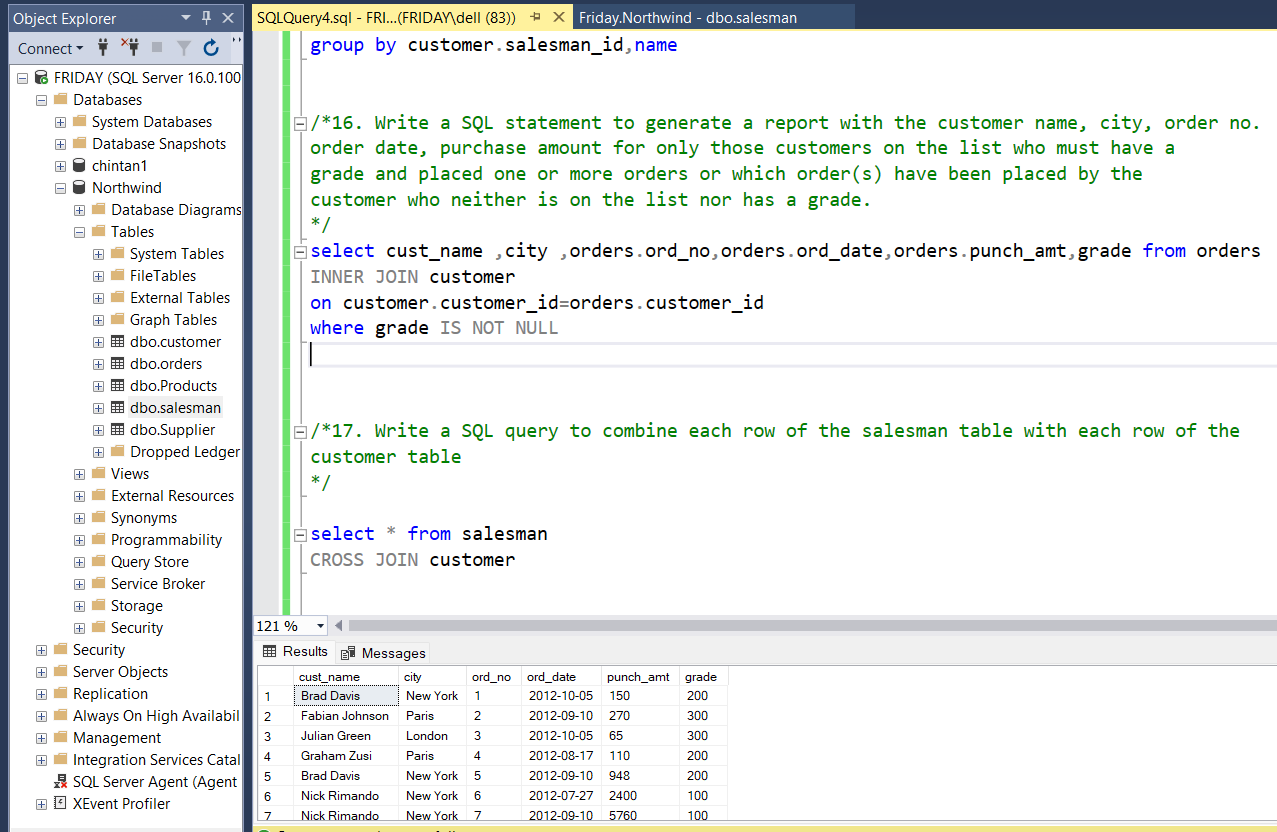
customer who neither is on the list nor has a grade.

select cust\_name ,city ,orders.ord\_no,orders.ord\_date,orders.punch\_amt,grade from orders

INNER JOIN customer

on customer.customer\_id=orders.customer\_id

where grade IS NOT NULL

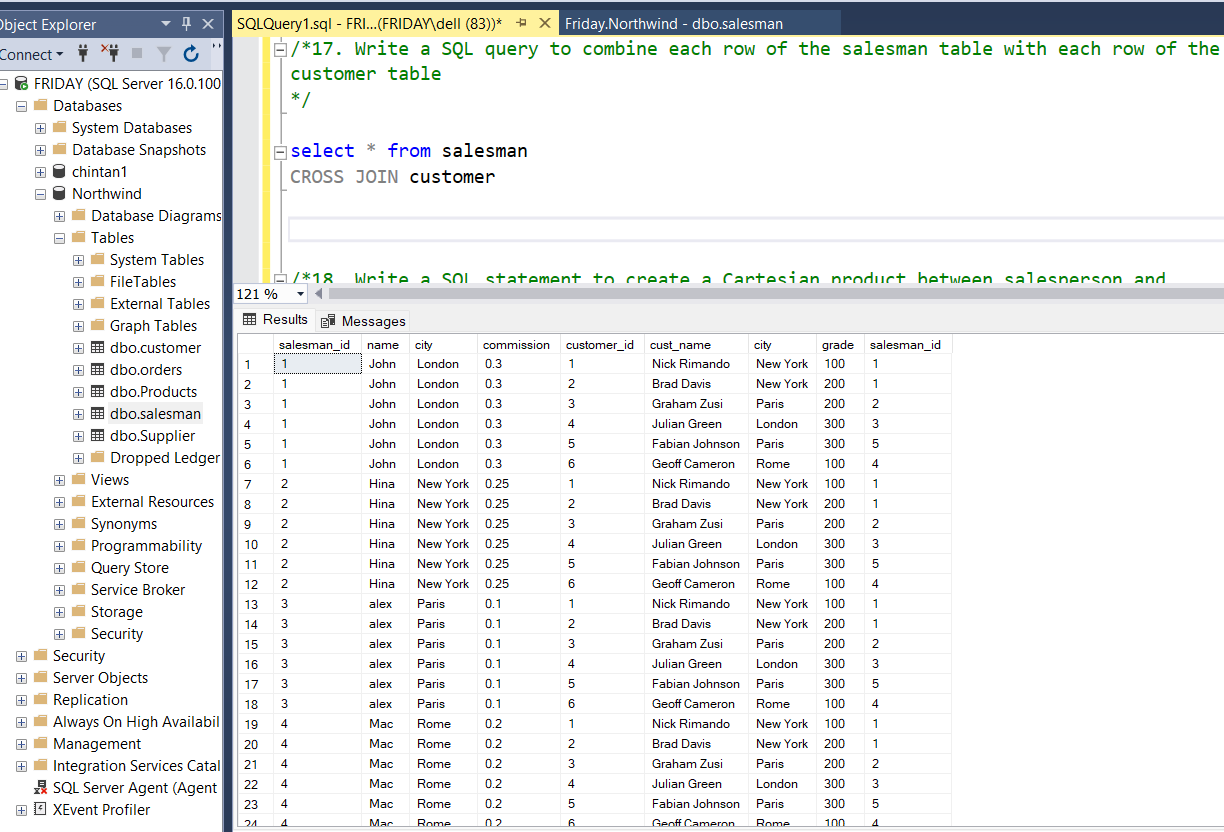


17. Write a SQL query to combine each row of the salesman table with each row of the

customer table

select \* from salesman

CROSS JOIN customer



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

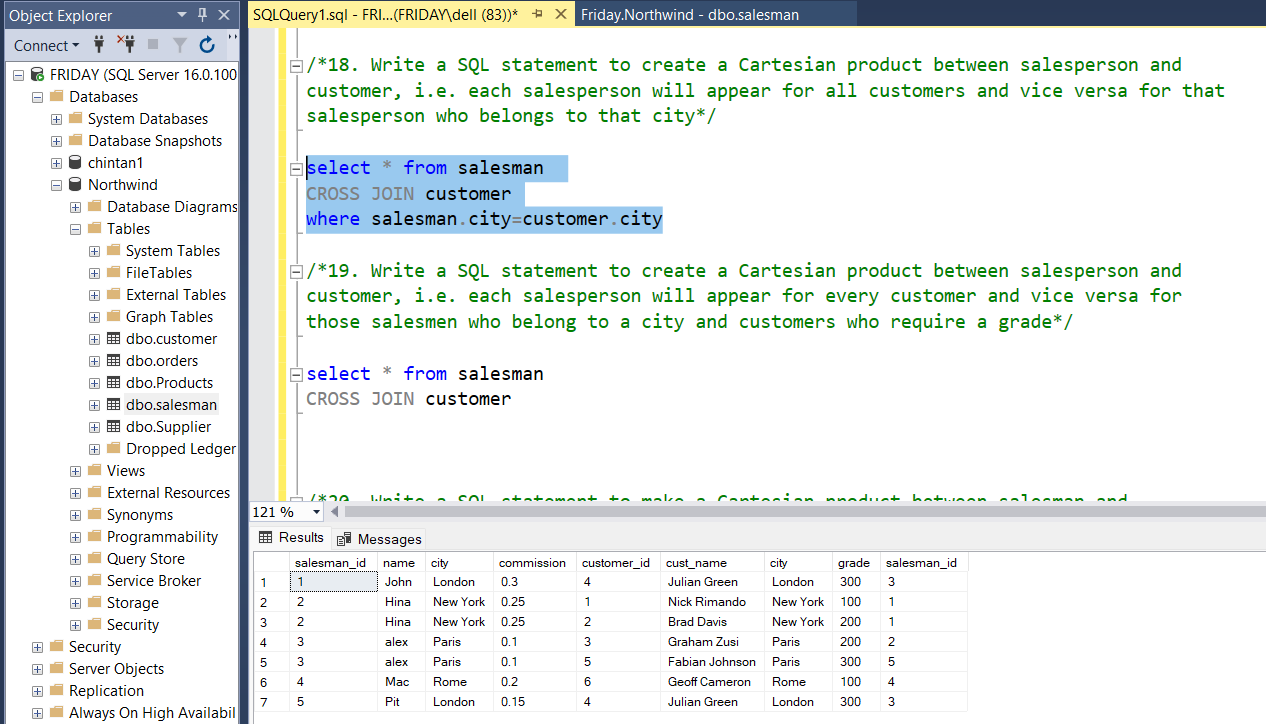
customer, i.e. each salesperson will appear for all customers and vice versa for that

salesperson who belongs to that city

select \* from salesman

CROSS JOIN customer

where salesman.city=customer.city



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

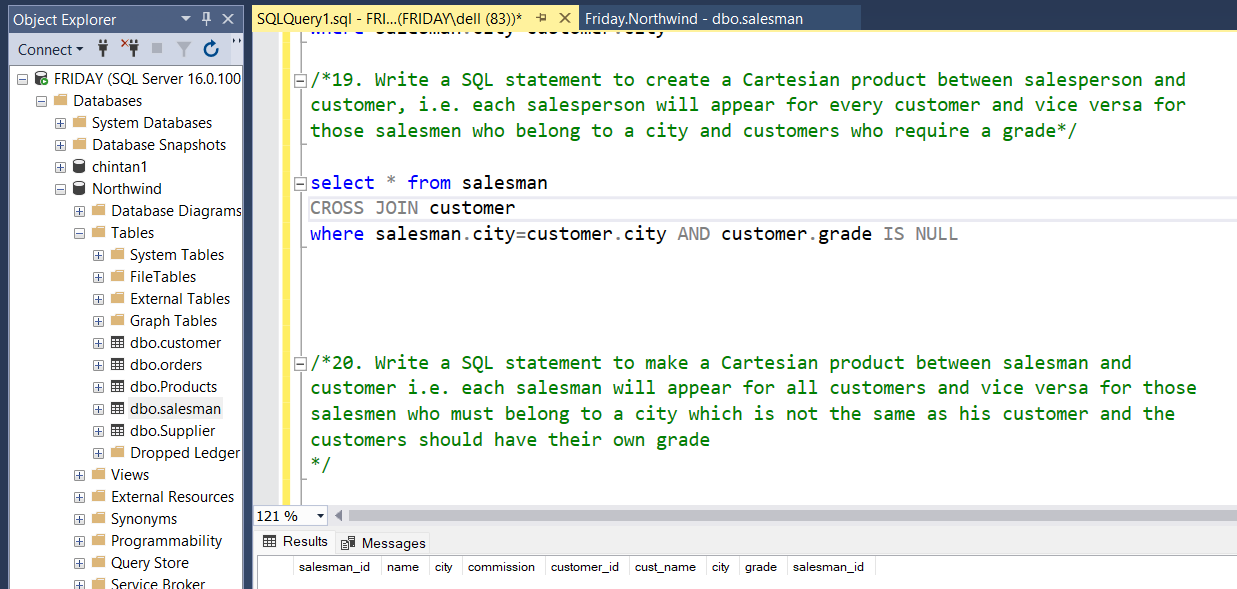
customer, i.e. each salesperson will appear for every customer and vice versa for

those salesmen who belong to a city and customers who require a grade\*/

select \* from salesman

CROSS JOIN customer

where salesman.city=customer.city AND customer.grade IS 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

CROSS JOIN customer

where salesman.city!=customer.city AND customer.grade IS NOT NULL

