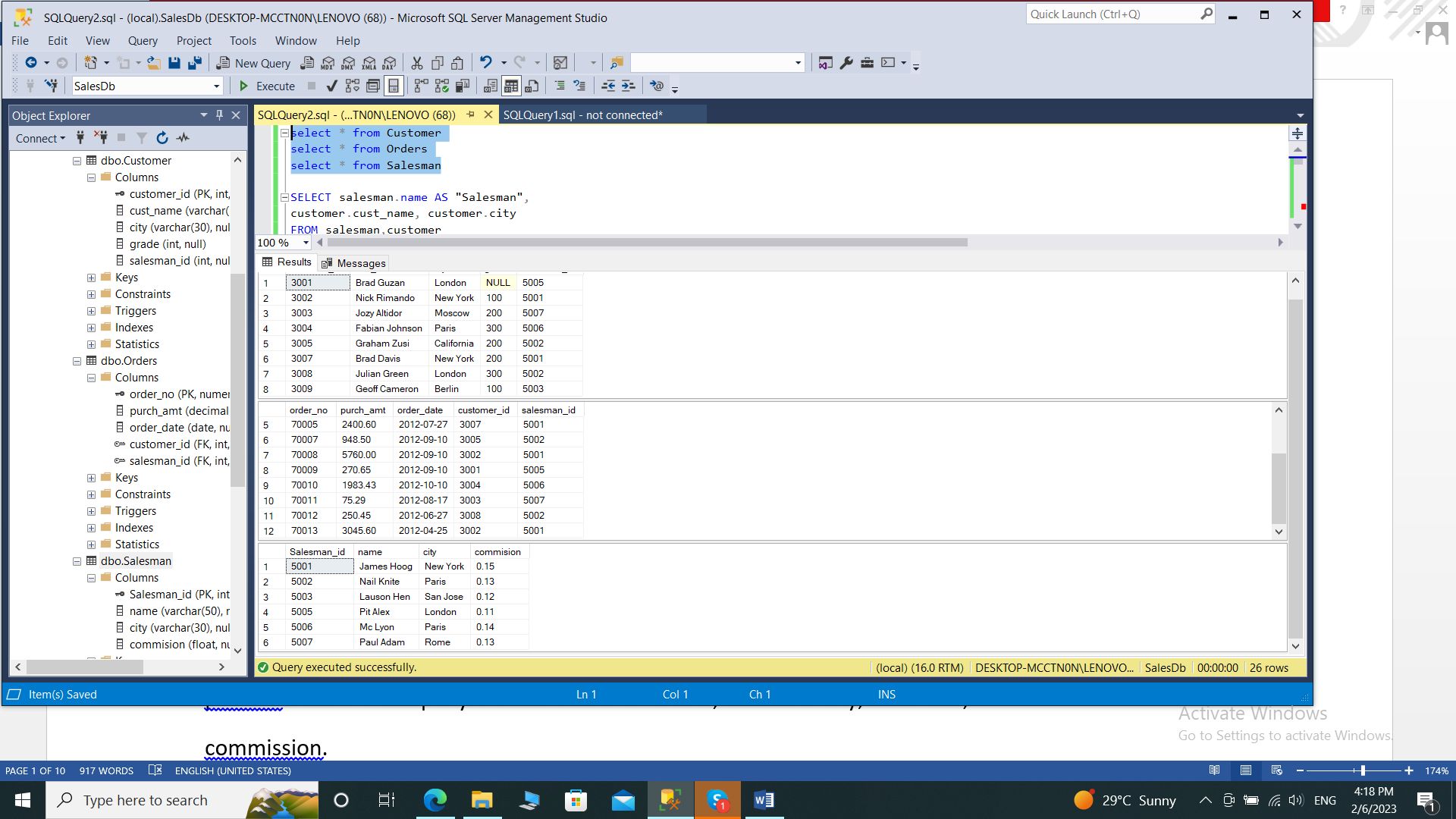
**Name:** Patel Aayushi Pravinbhai

**SQL Assignment – 2**

**Database:**



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

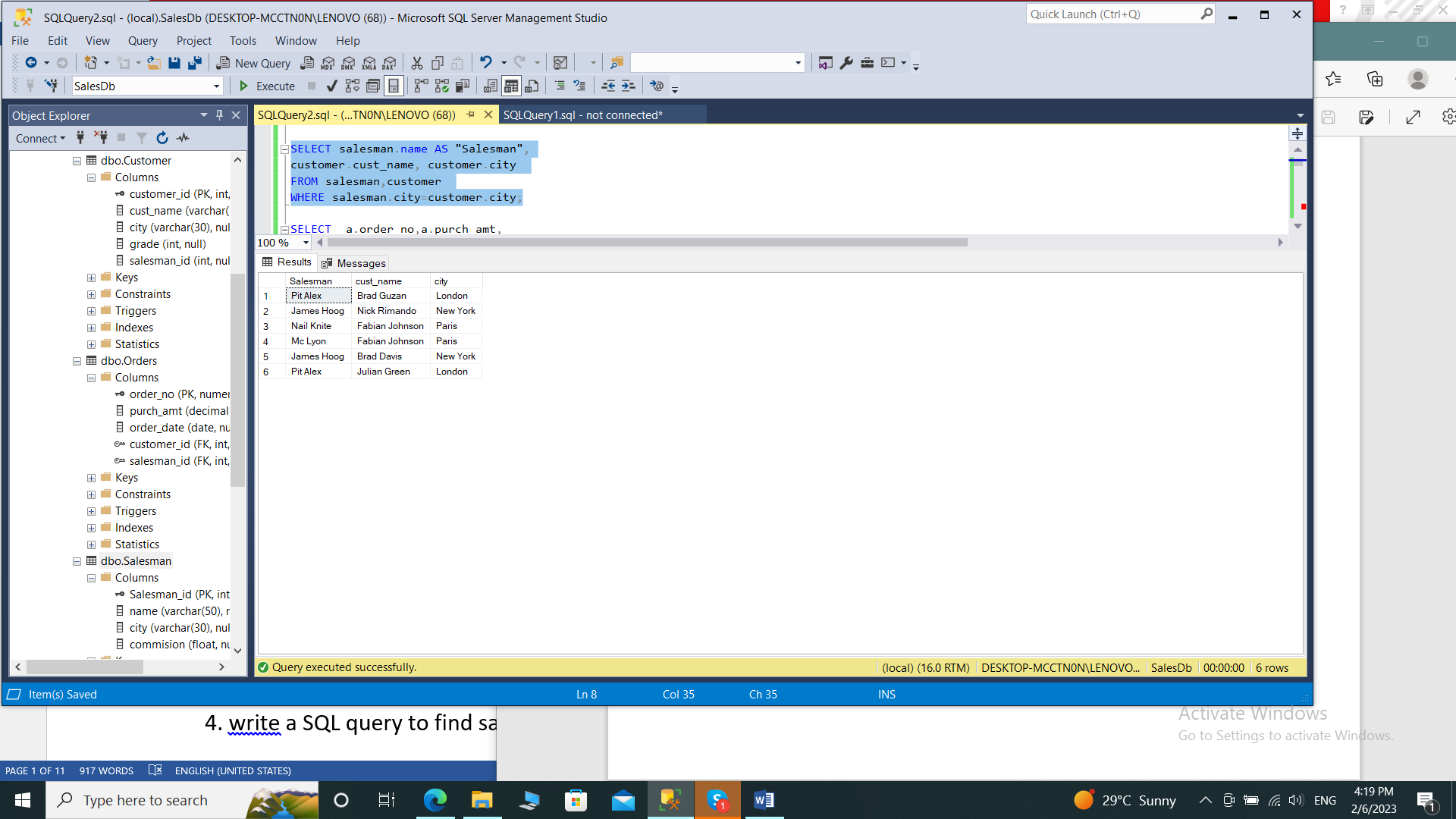
**Return Salesman, cust\_name and city**

SELECT salesman.name AS "Salesman",

customer.cust\_name, customer.city

FROM salesman,customer

WHERE 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 a.order\_no,a.purch\_amt,

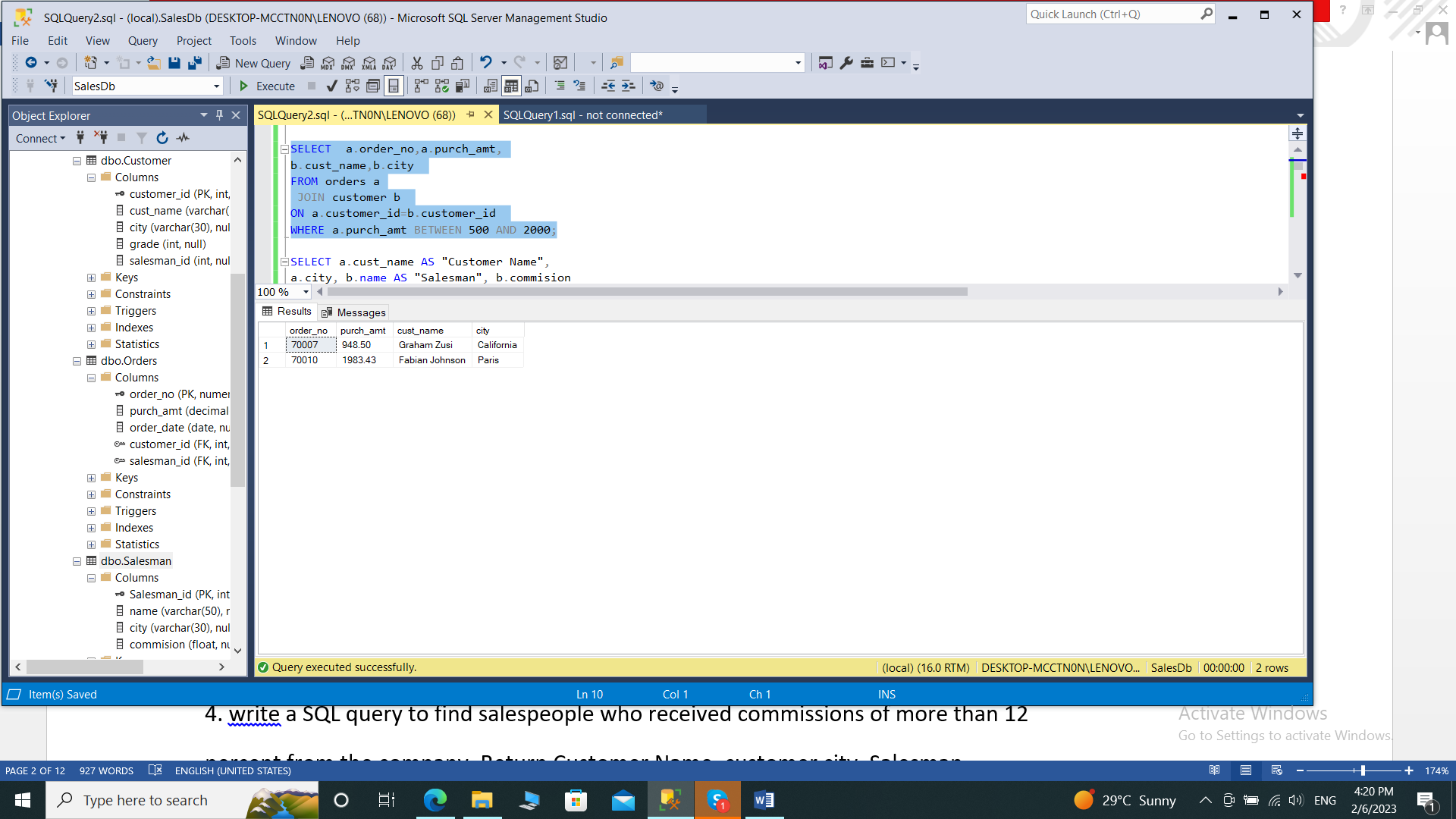
b.cust\_name,b.city

FROM orders a

JOIN customer b

ON a.customer\_id=b.customer\_id

WHERE a.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 a.cust\_name AS "Customer Name",

a.city, b.name AS "Salesman", b.commision

FROM customer a

RIGHT JOIN salesman b

ON a.salesman\_id=b.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 a.cust\_name AS "Customer Name",

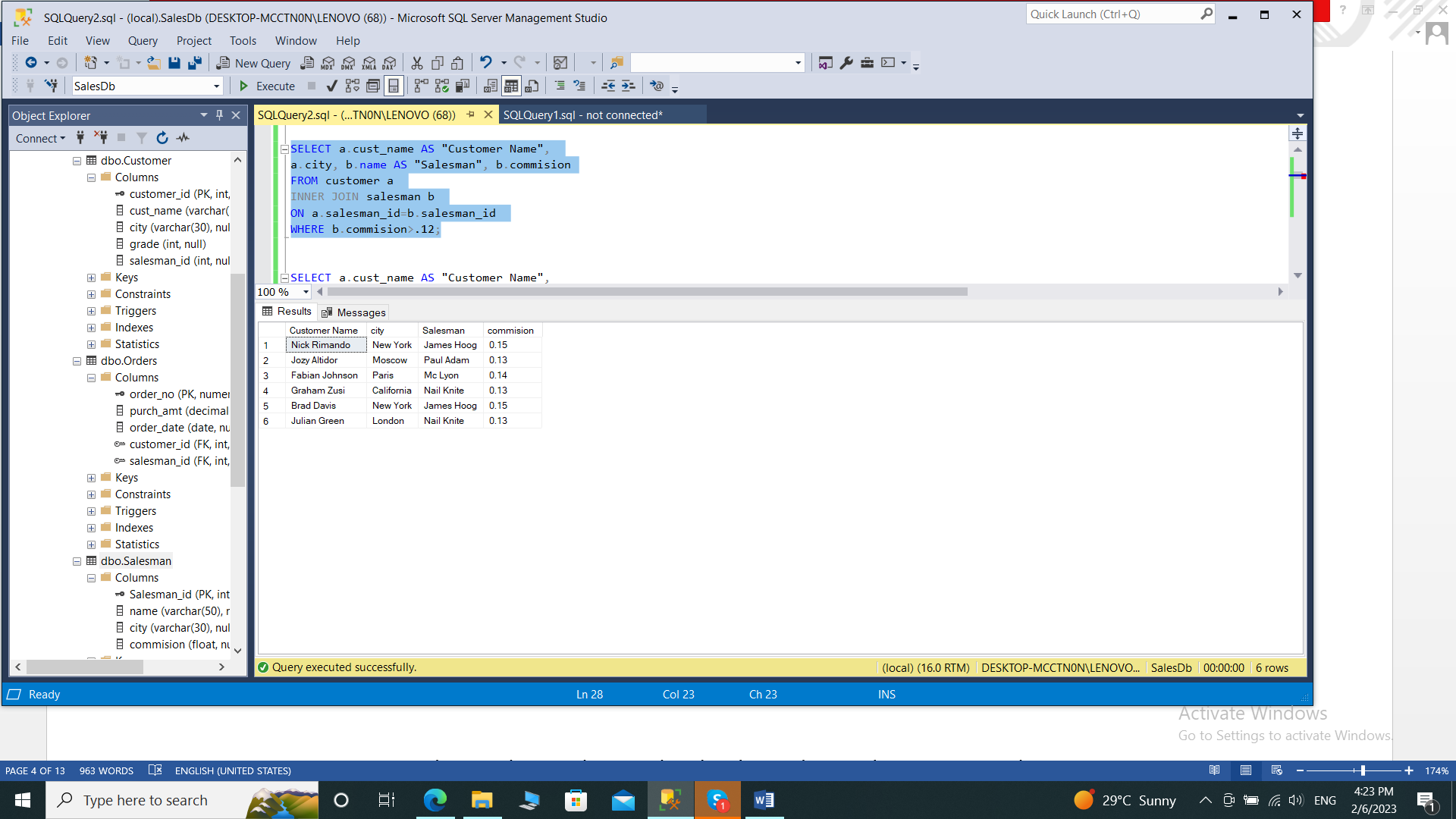
a.city, b.name AS "Salesman", b.commision

FROM customer a

INNER JOIN salesman b

ON a.salesman\_id=b.salesman\_id

WHERE b.commision>.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 a.cust\_name AS "Customer Name",

a.city, b.name AS "Salesman", b.city,b.commision

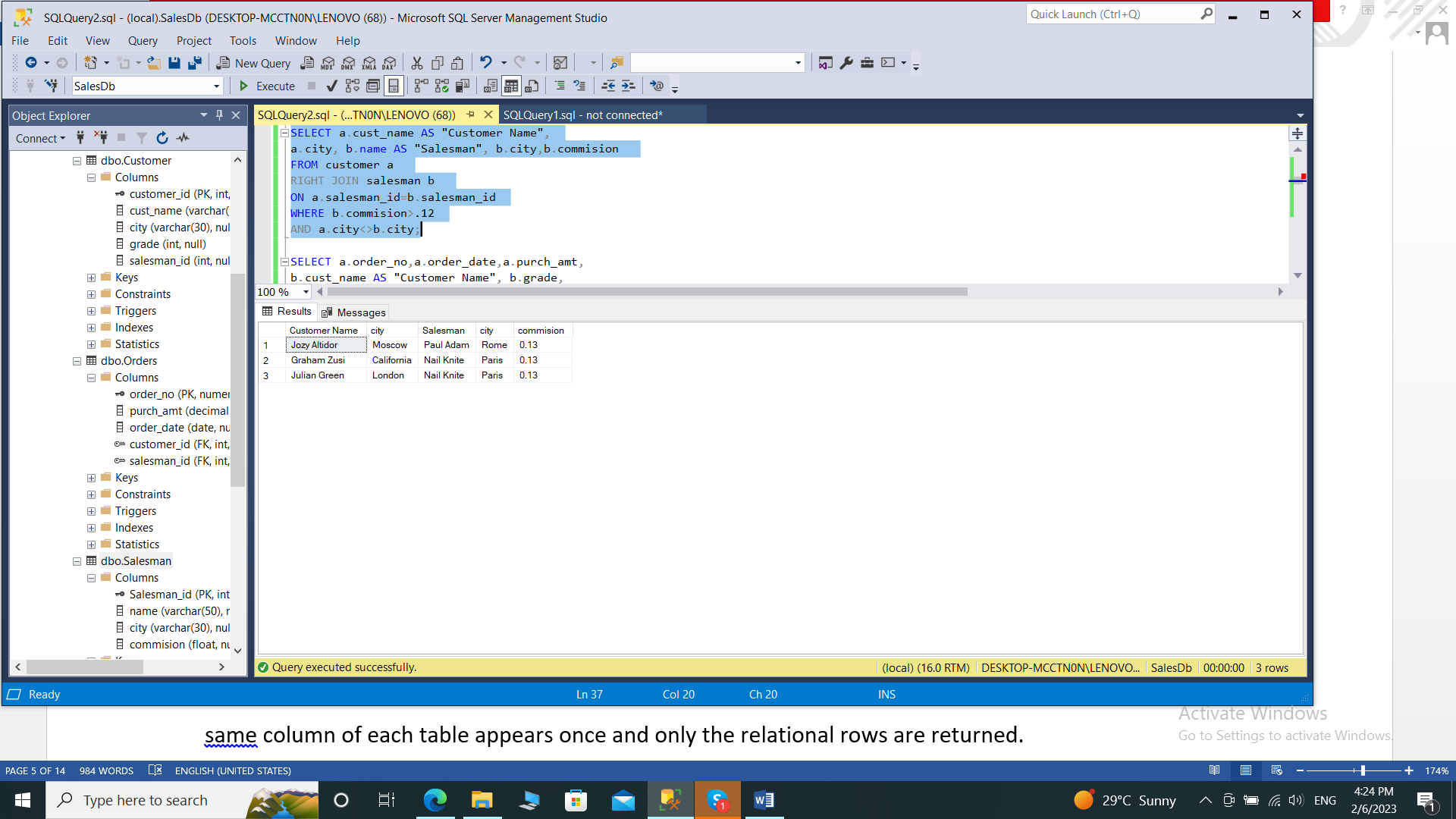
FROM customer a

RIGHT JOIN salesman b

ON a.salesman\_id=b.salesman\_id

WHERE b.commision>.12

AND a.city<>b.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 a.order\_no,a.order\_date,a.purch\_amt,

b.cust\_name AS "Customer Name", b.grade,

c.name AS "Salesman", c.commision

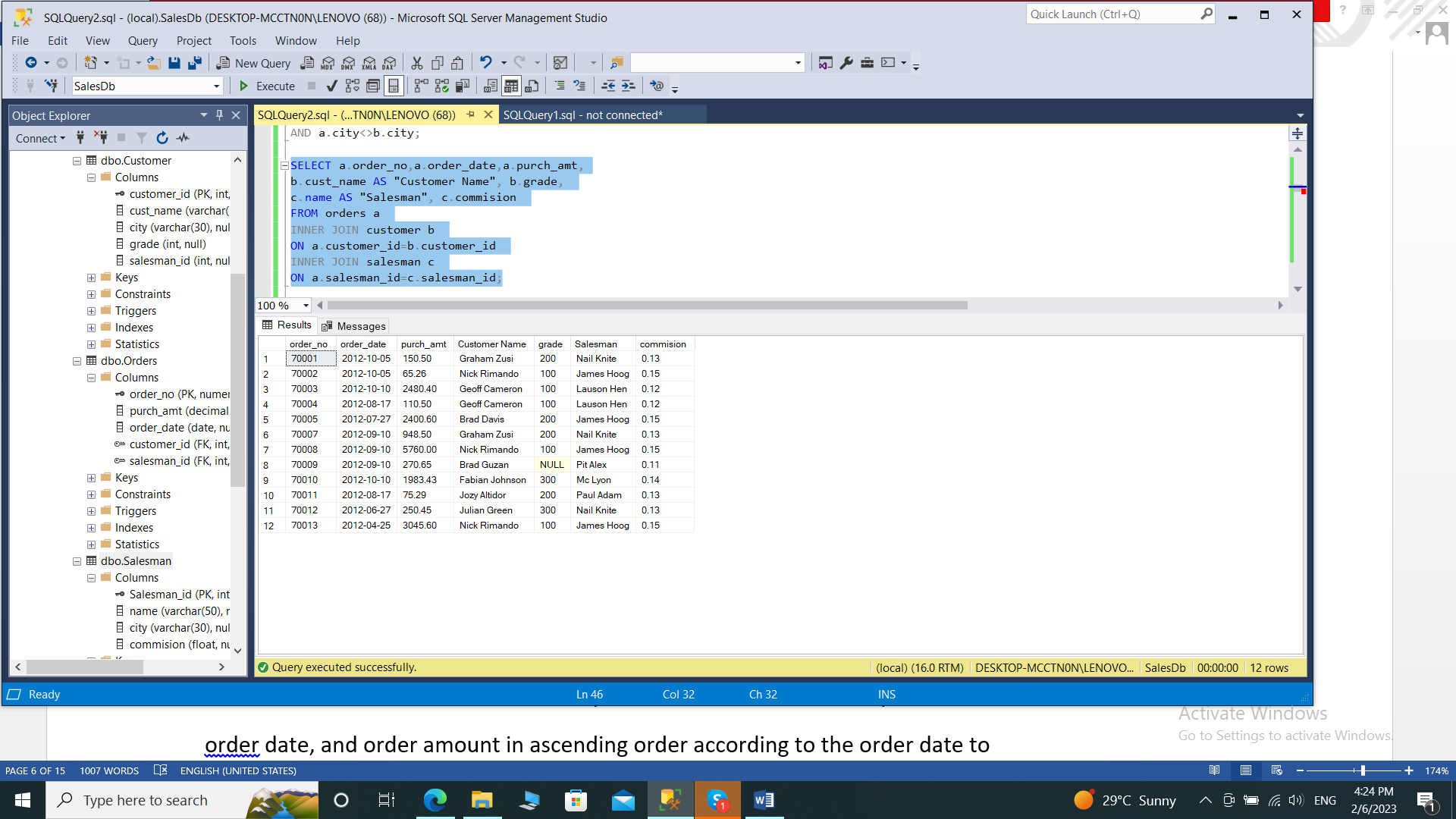
FROM orders a

INNER JOIN customer b

ON a.customer\_id=b.customer\_id

INNER JOIN salesman c

ON a.salesman\_id=c.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 b.customer\_id,b.cust\_name,b.city AS Customer\_City, b.grade,a.order\_no,a.purch\_amt,a.purch\_amt,a.order\_date,c.Salesman\_id,c.name AS Salesman\_name,c.city AS Salesman\_City,c.commision

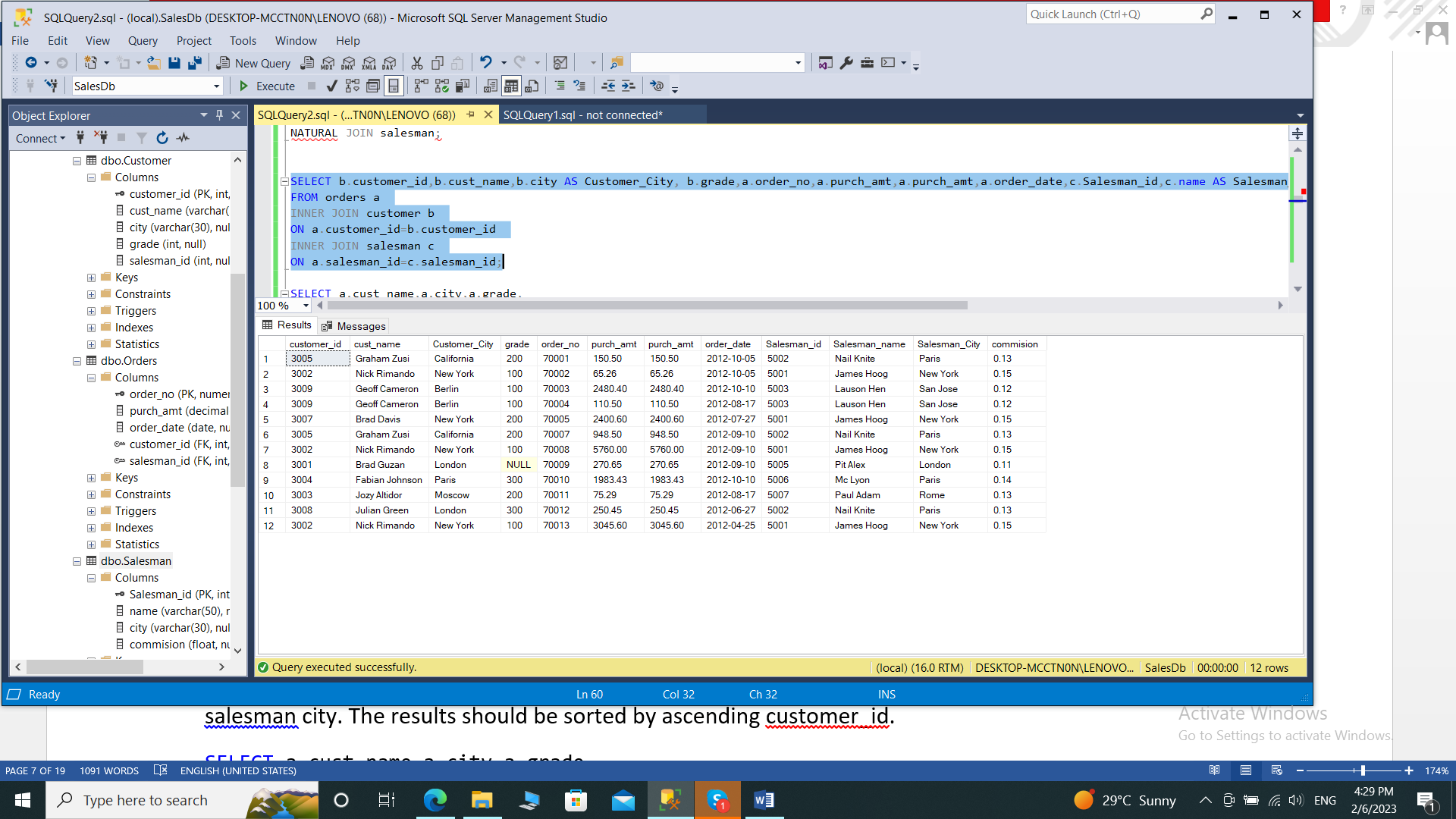
FROM orders a

INNER JOIN customer b

ON a.customer\_id=b.customer\_id

INNER JOIN salesman c

ON a.salesman\_id=c.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 a.cust\_name,a.city,a.grade,

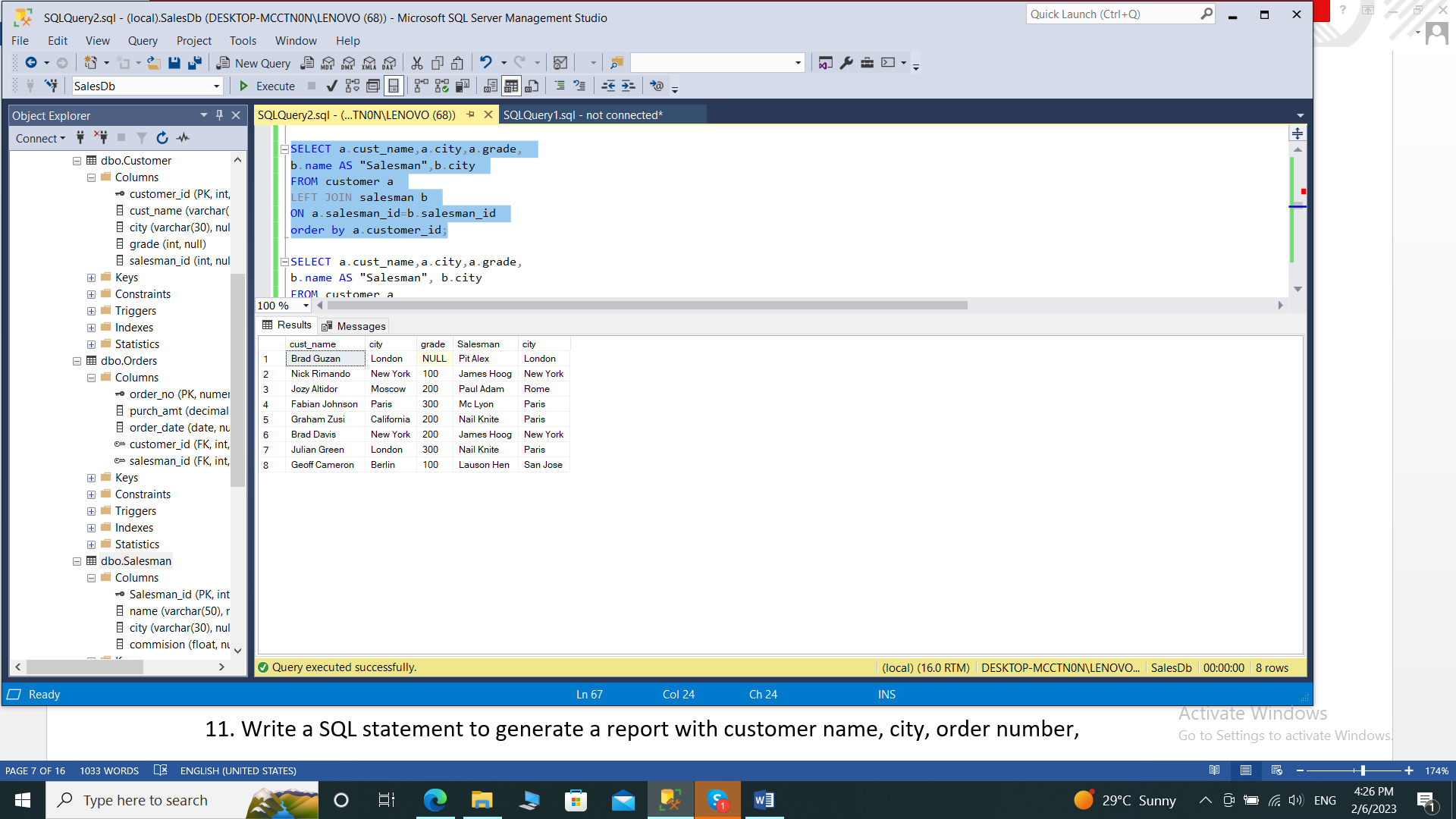
b.name AS "Salesman",b.city

FROM customer a

LEFT JOIN salesman b

ON a.salesman\_id=b.salesman\_id

order by a.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 a.cust\_name,a.city,a.grade,

b.name AS "Salesman", b.city

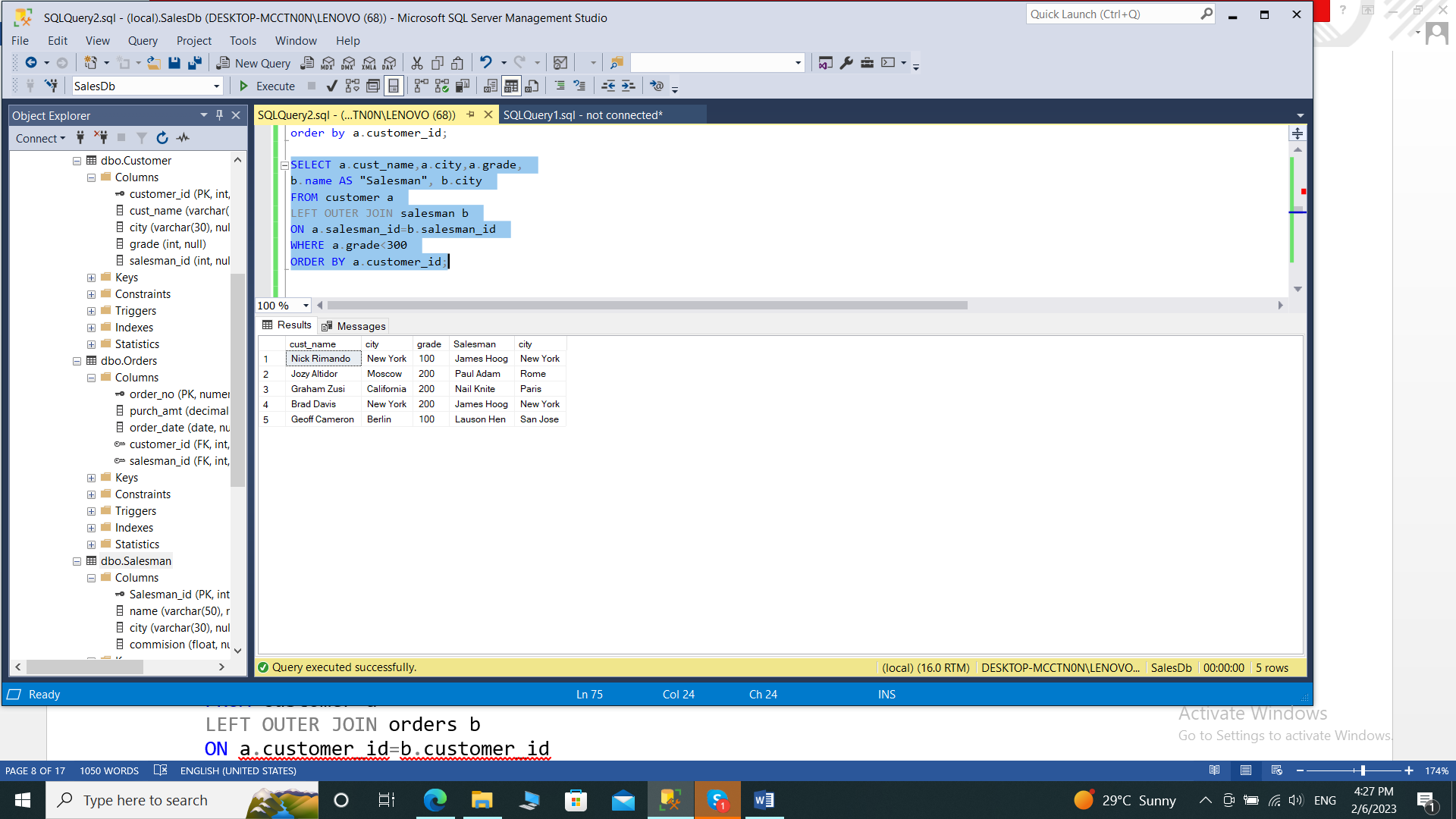
FROM customer a

LEFT OUTER JOIN salesman b

ON a.salesman\_id=b.salesman\_id

WHERE a.grade<300

ORDER BY a.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 a.cust\_name,a.city, b.order\_no,

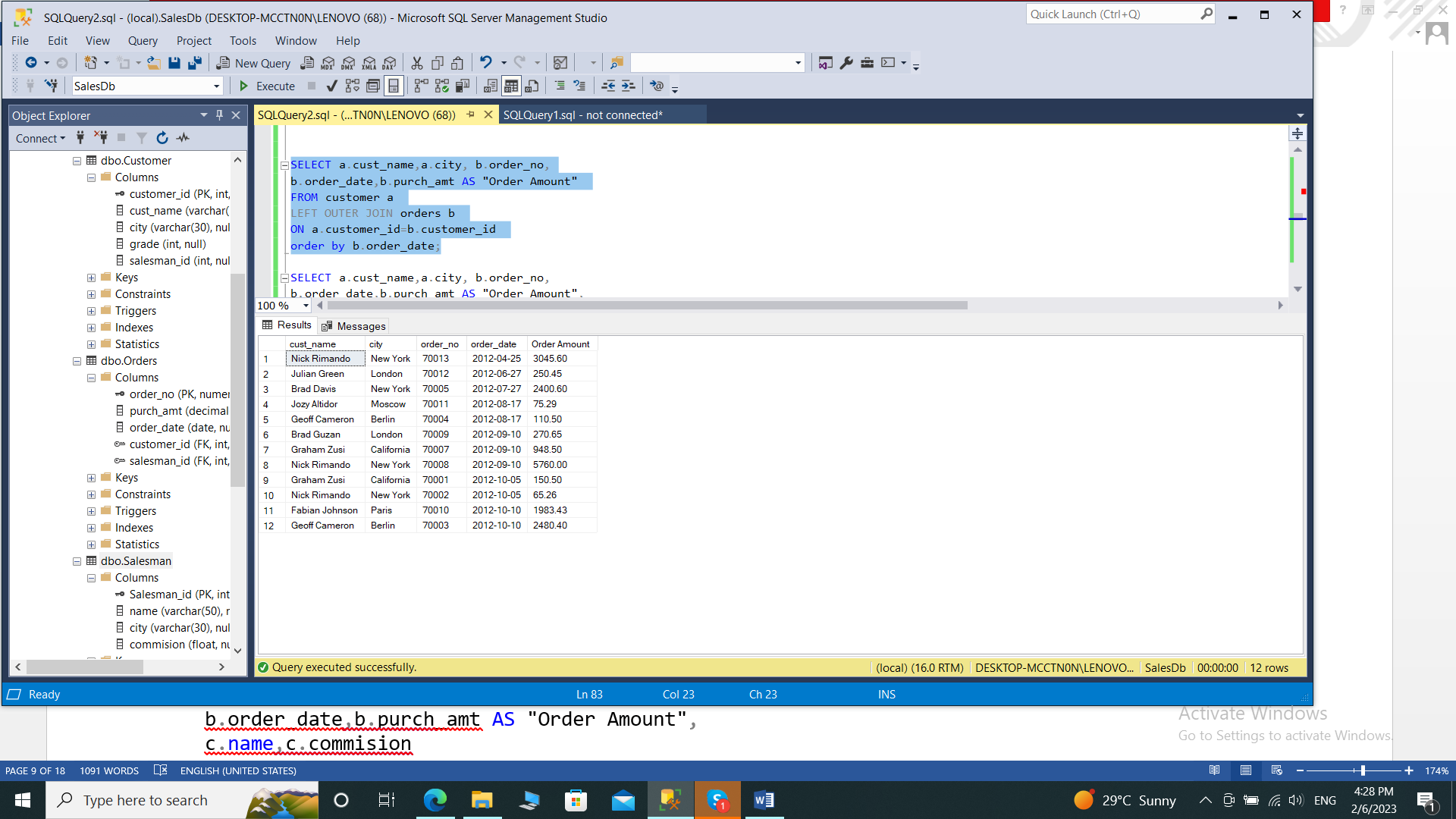
b.order\_date,b.purch\_amt AS "Order Amount"

FROM customer a

LEFT OUTER JOIN orders b

ON a.customer\_id=b.customer\_id

order by b.order\_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 a.cust\_name,a.city, b.order\_no,

b.order\_date,b.purch\_amt AS "Order Amount",

c.name,c.commision

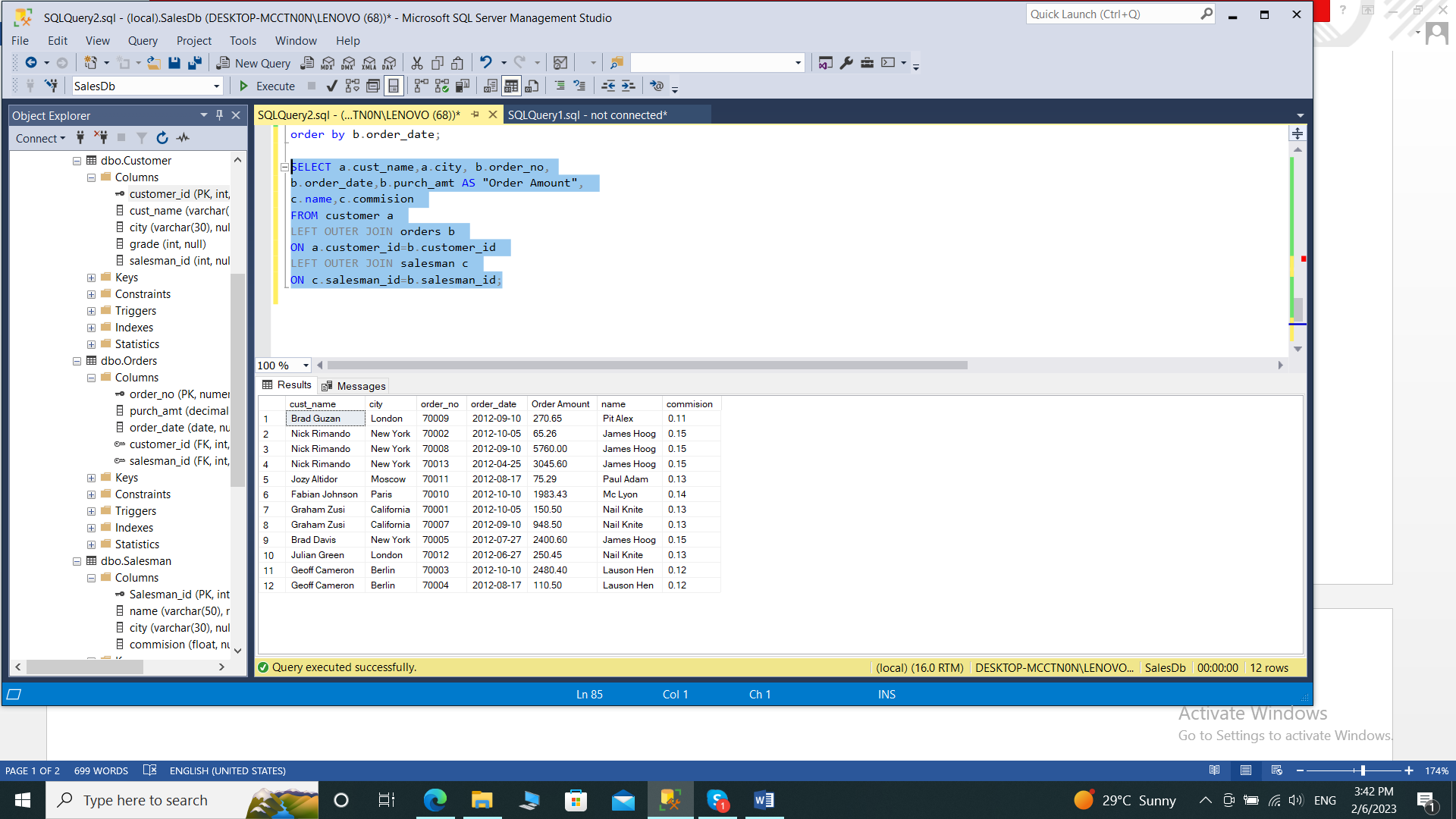
FROM customer a

LEFT OUTER JOIN orders b

ON a.customer\_id=b.customer\_id

LEFT OUTER JOIN salesman c

ON c.salesman\_id=b.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 b.name AS "Salesman", b.city,

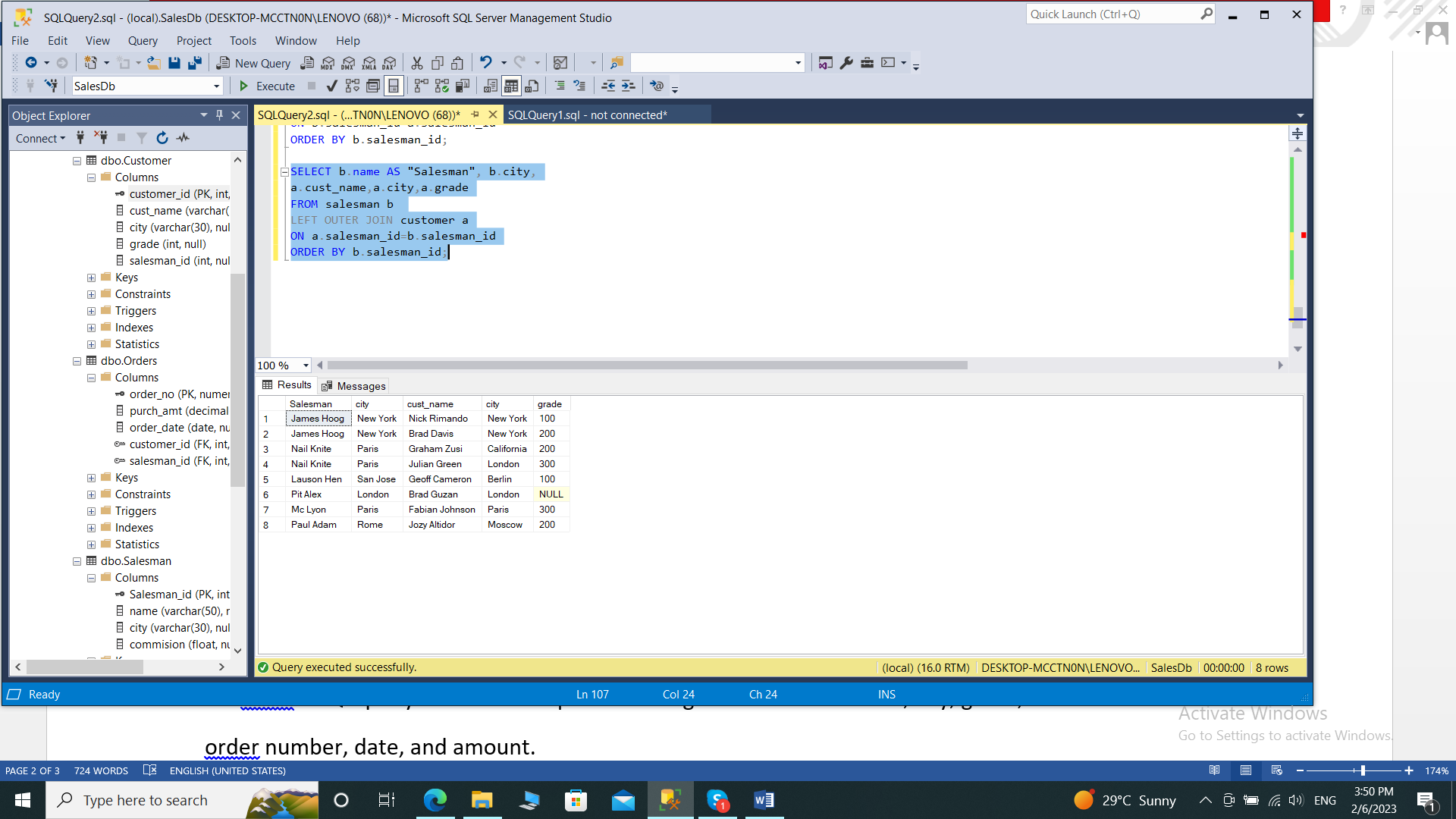
a.cust\_name,a.city,a.grade

FROM salesman b

LEFT OUTER JOIN customer a

ON a.salesman\_id=b.salesman\_id

ORDER BY b.salesman\_id;



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

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

SELECT

b.name AS "Salesman", a.cust\_name,a.city,a.grade,

c.order\_no, c.order\_date, c.purch\_amt

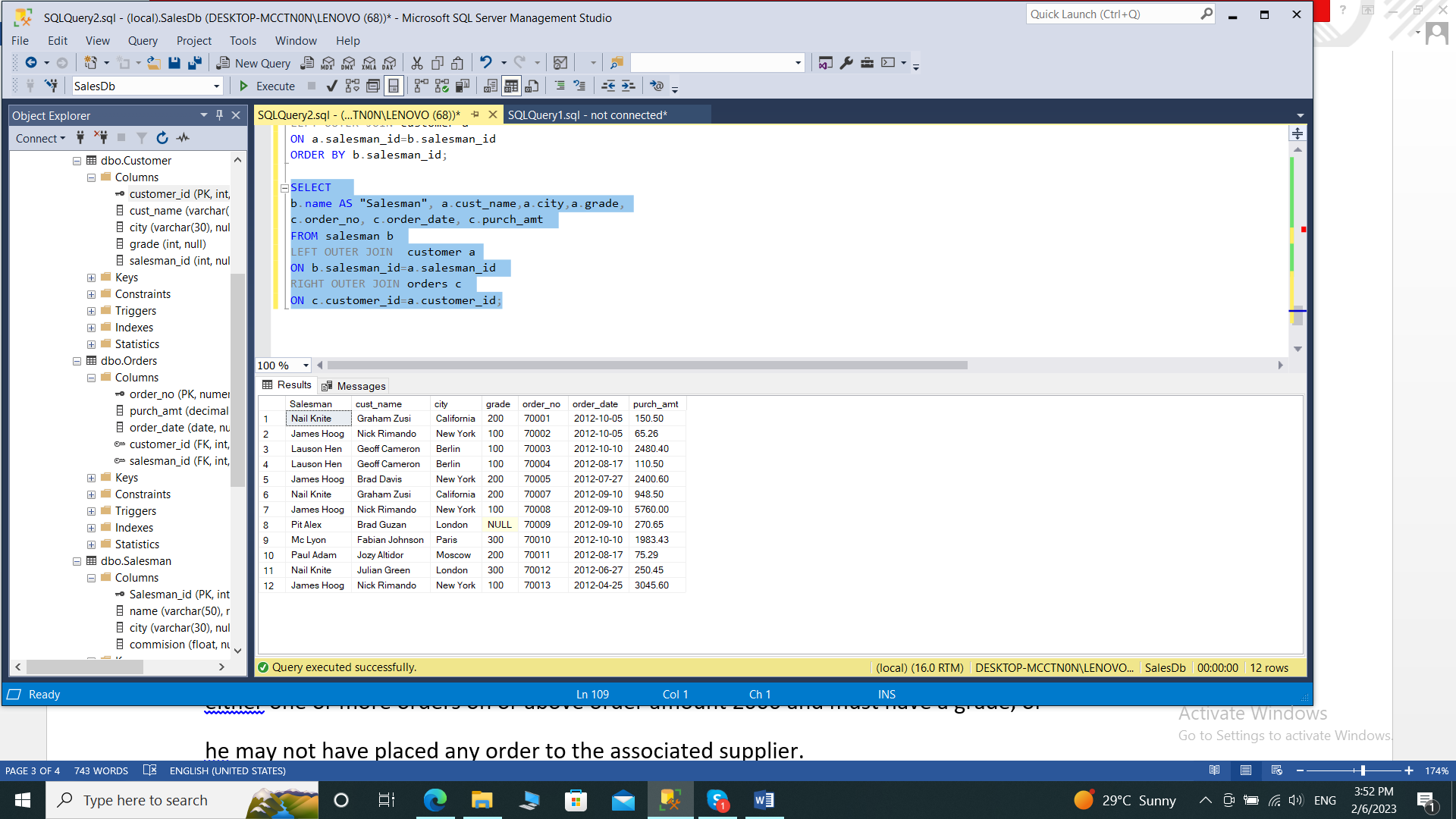
FROM salesman b

LEFT OUTER JOIN customer a

ON b.salesman\_id=a.salesman\_id

RIGHT OUTER JOIN orders c

ON c.customer\_id=a.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

b.name AS "Salesman", a.cust\_name,a.city,a.grade,

c.order\_no, c.order\_date, c.purch\_amt

FROM customer a

RIGHT OUTER JOIN salesman b

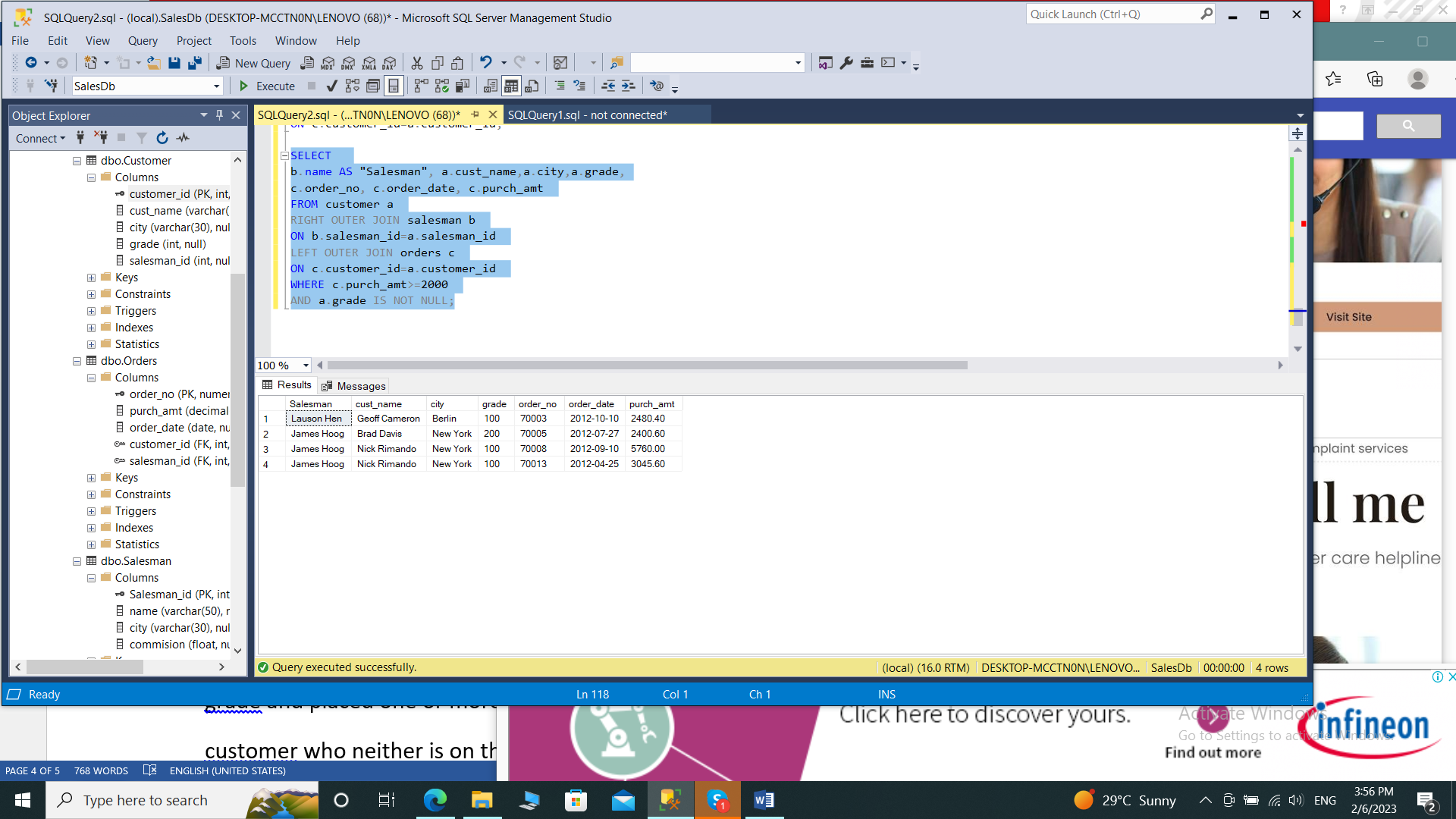
ON b.salesman\_id=a.salesman\_id

LEFT OUTER JOIN orders c

ON c.customer\_id=a.customer\_id

WHERE c.purch\_amt>=2000

AND a.grade IS NOT NULL;



**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

b.name AS "Salesman", a.cust\_name,a.city,a.grade,

c.order\_no, c.order\_date, c.purch\_amt

FROM customer a

RIGHT OUTER JOIN salesman b

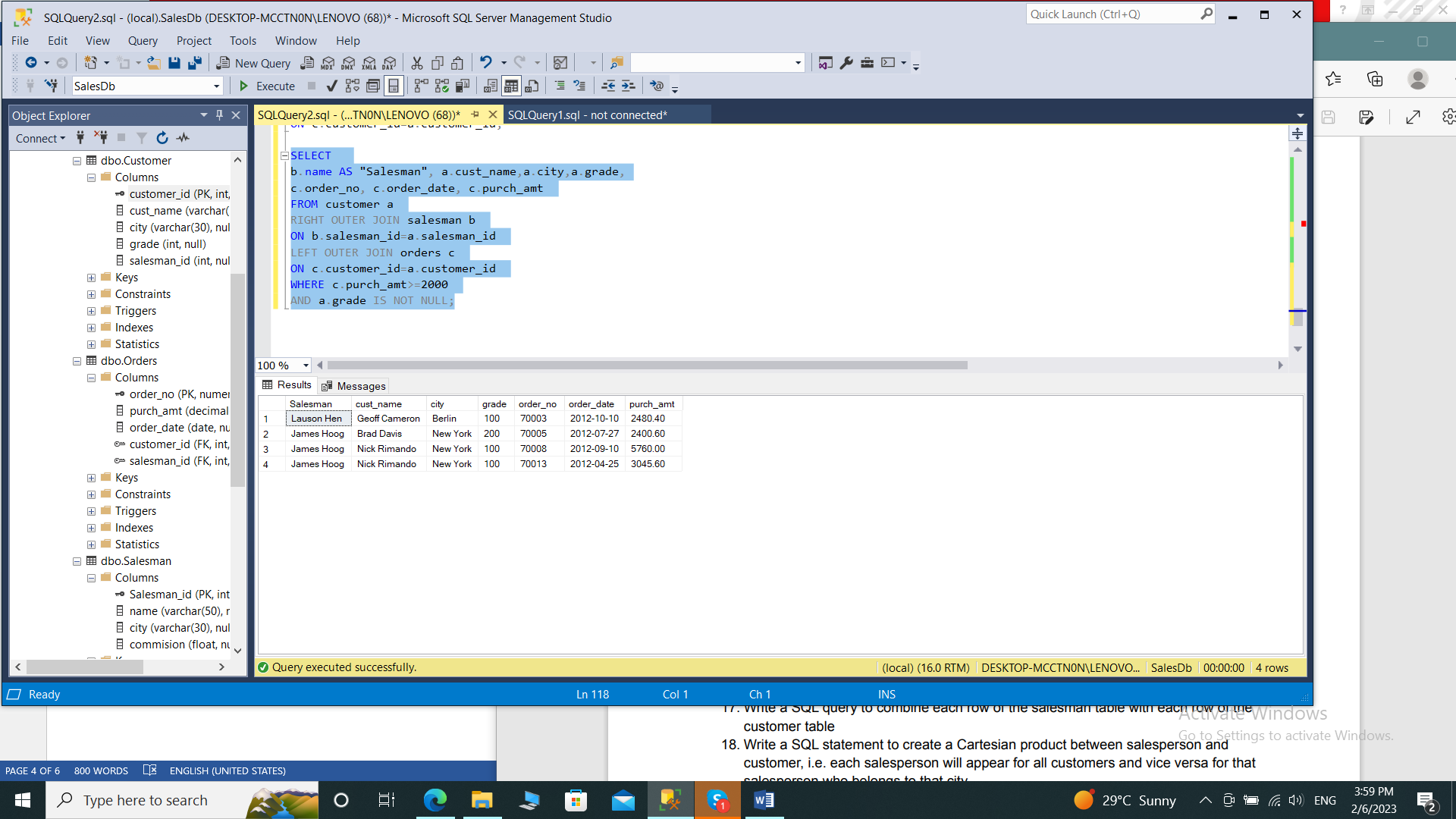
ON b.salesman\_id=a.salesman\_id

LEFT OUTER JOIN orders c

ON c.customer\_id=a.customer\_id

WHERE c.purch\_amt>=2000

AND a.grade IS NOT NULL;



**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 a.cust\_name,a.city, b.order\_no,

b.order\_date,b.purch\_amt AS "Order Amount"

FROM customer a

FULL OUTER JOIN orders b

ON a.customer\_id=b.customer\_id

WHERE a.grade IS NOT NULL;



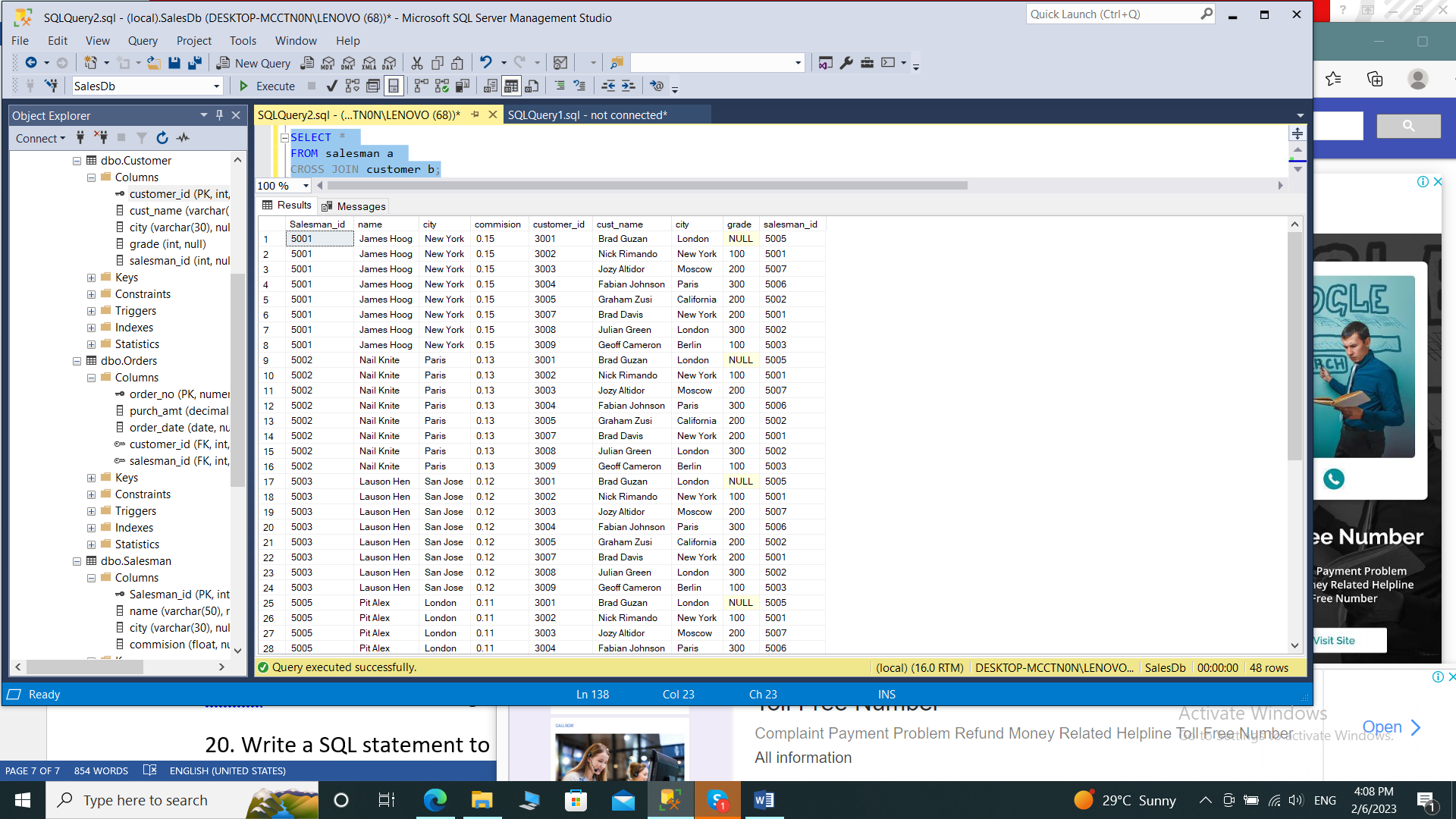
**17. Write a SQL query to combine each row of the salesman table with each row of the**

**customer table**

SELECT \*

FROM salesman a

CROSS JOIN customer b;



**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**

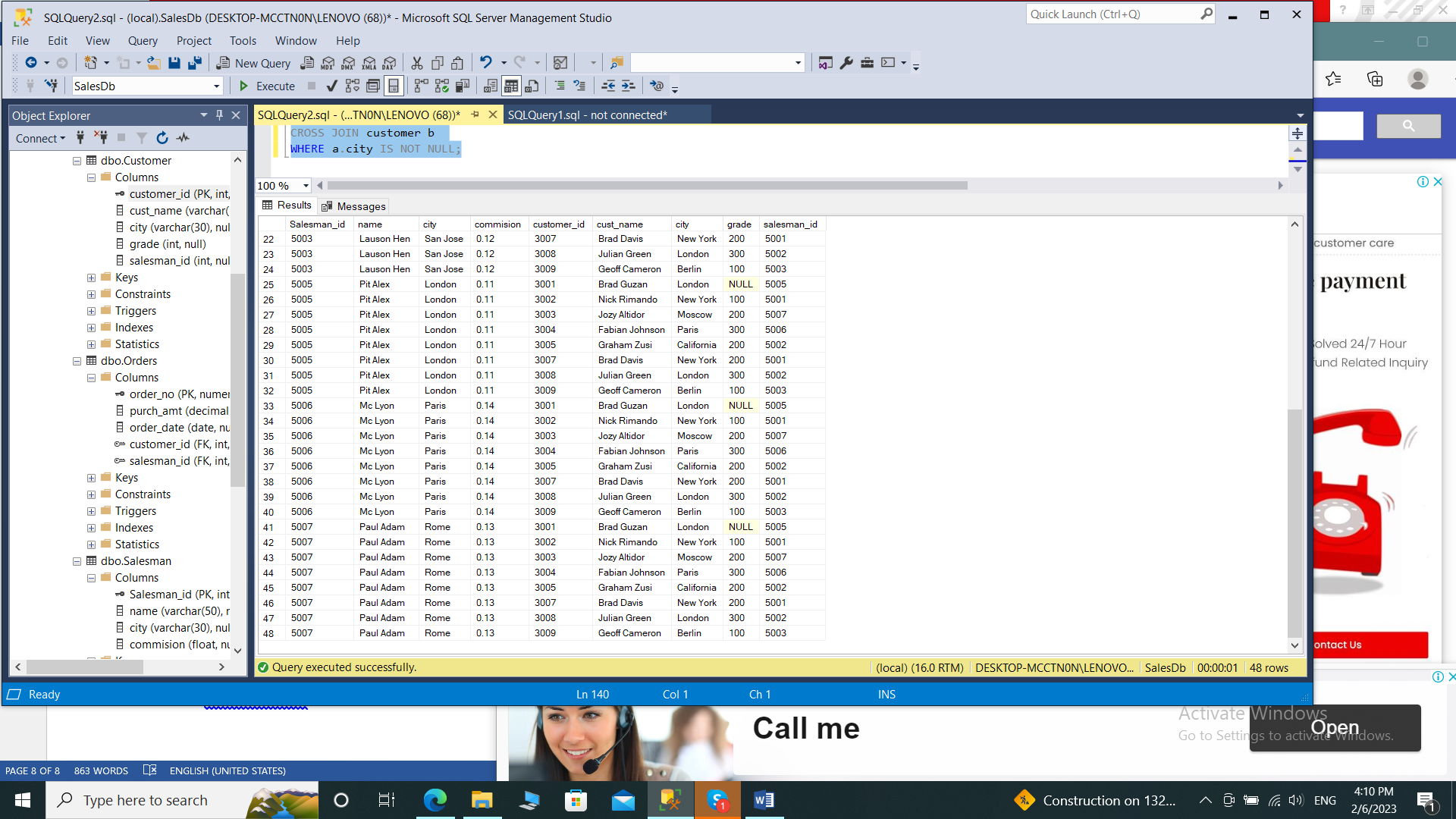
**salesperson who belongs to that city**

SELECT \*

FROM salesman a

CROSS JOIN customer b

WHERE a.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**

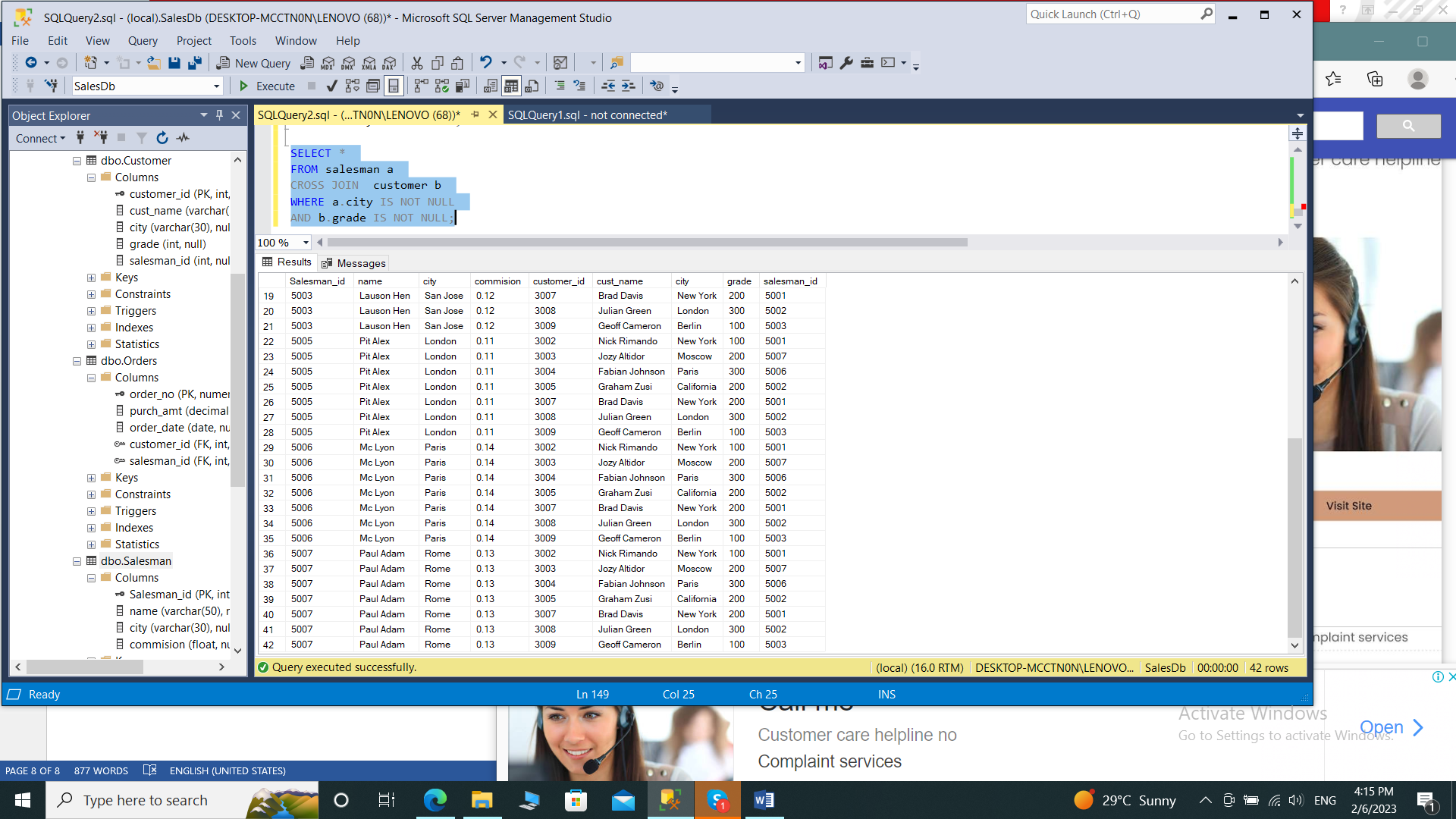
SELECT \*

FROM salesman a

CROSS JOIN customer b

WHERE a.city IS NOT NULL

AND b.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 a

CROSS JOIN customer b

WHERE a.city IS NOT NULL

AND b.grade IS NOT NULL

AND a.city<>b.city;

