**Assignment 2 : Retrieve data using join with where clause**

**--------------------------------------------------------------------------------------------------------------------------**

**Sample table1: salesman**

**-salesman\_id**

**-name**

**-city**

**-commission**

**Sample table2: customer**

**-customer\_id**

**-cust\_name**

**-city**

**-grade**

**-salesman\_id**

**Sample table3: orders**

**-ord\_no**

**-purch\_amt**

**-ord\_date**

**-customer\_id**

**-salesman\_id**

**CREATING SALESMAN TABLE**

CREATE TABLE [SALESMAN]

(salesman\_id INT PRIMARY KEY IDENTITY(5001,1),

NAME varchar(50) NOT null,

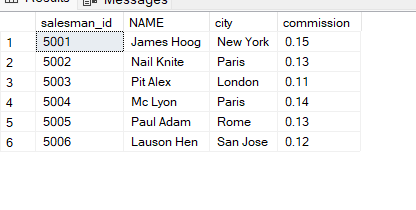
city varchar(50) Not null,

commission float Not null

)

INSERT INTO SALESMAN VALUES ('James Hoog','New York', 0.15),( 'Nail Knite','Paris', 0.13),('Pit Alex','London',0.11),('Mc Lyon','Paris', 0.14),('Paul Adam ','Rome', 0.13),('Lauson Hen','San Jose',0.12)

SELECT \* FROM SALESMAN



--**CREATING CUSTOMER TABLE**

CREATE TABLE [CUSTOMER]

( customer\_id INT PRIMARY KEY IDENTITY(3001,1),

cust\_name NCHAR(50) NOT NULL,

city NCHAR(50) NOT NULL,

grade INT NULL,

salesman\_id INT NOT NULL

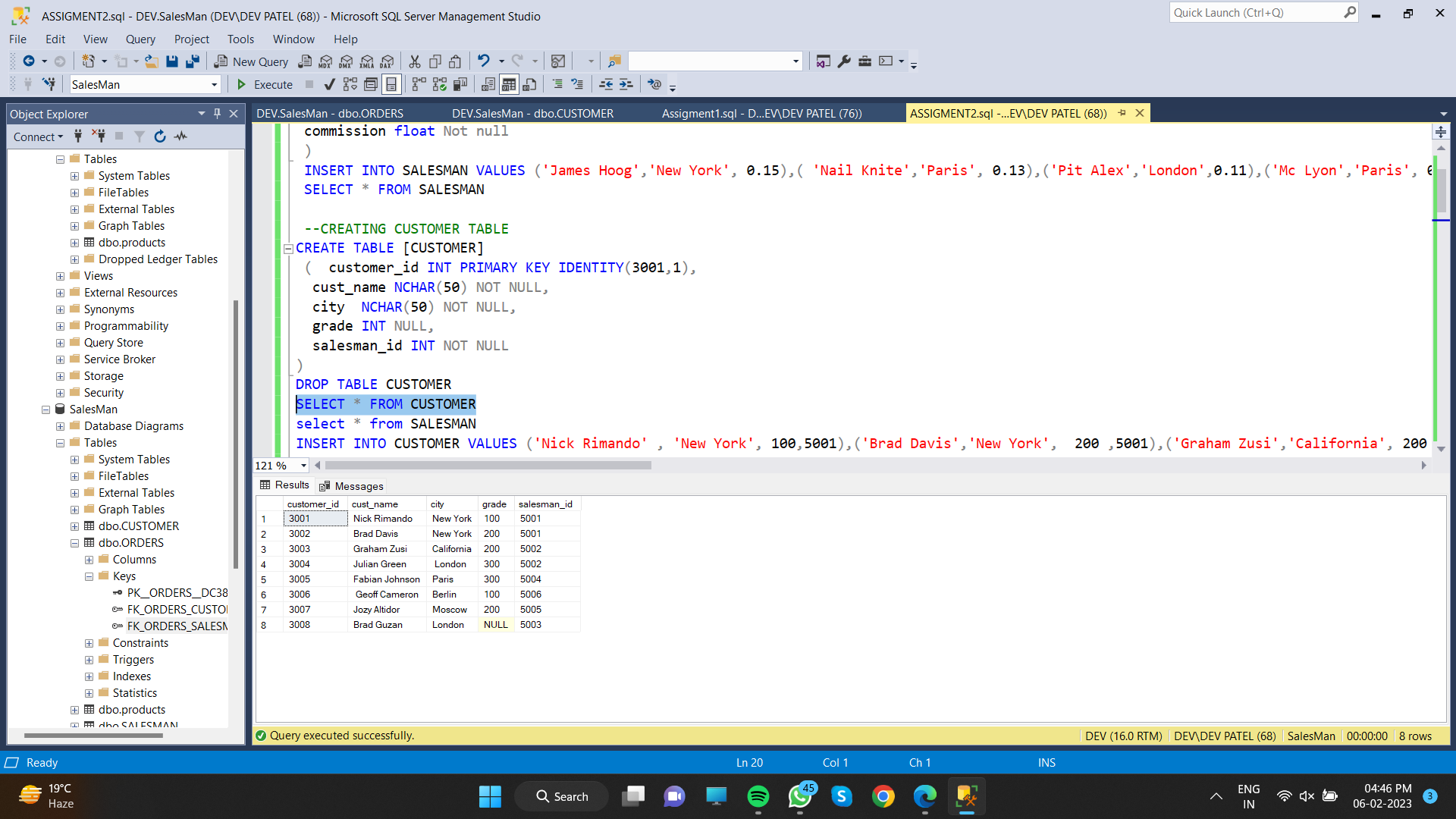
)

DROP TABLE CUSTOMER

SELECT \* FROM CUSTOMER

select \* from SALESMAN

INSERT INTO CUSTOMER VALUES ('Nick Rimando' , 'New York', 100,5001),('Brad Davis','New York', 200 ,5001),('Graham Zusi','California', 200 ,5002),('Julian Green',' London' , 300,5002),('Fabian Johnson','Paris' ,300, 5004),(' Geoff Cameron ', 'Berlin' ,100, 5006),('Jozy Altidor','Moscow ',200, 5005), ('Brad Guzan','London',NULL,5003)



**CREATING ORDER TABLE**

CREATE TABLE [ORDERS]

(ord\_no INT PRIMARY KEY IDENTITY(70001,1),

purch\_amt FLOAT NOT NULL,

ord\_date DATE NOT NULL,

customer\_id INT NOT NULL,

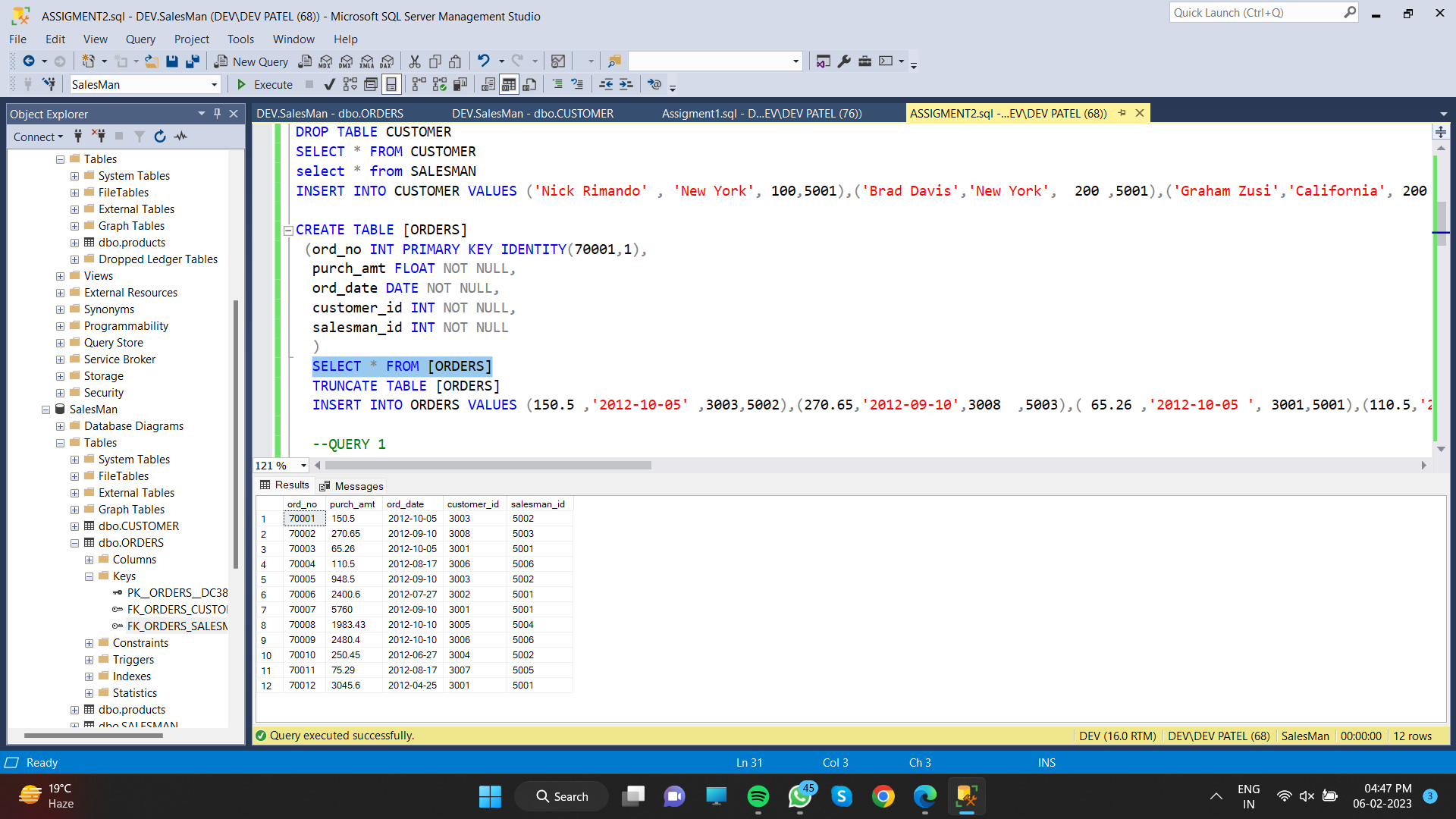
salesman\_id INT NOT NULL

)

SELECT \* FROM [ORDERS]

TRUNCATE TABLE [ORDERS]

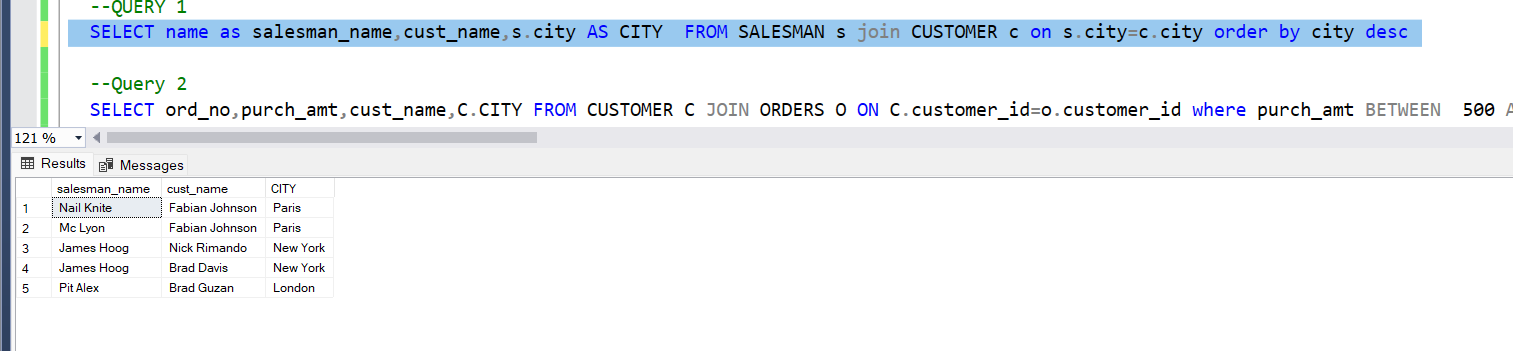
INSERT INTO ORDERS VALUES (150.5 ,'2012-10-05' ,3003,5002),(270.65,'2012-09-10',3008 ,5003),( 65.26 ,'2012-10-05 ', 3001,5001),(110.5,'2012-08-17 ' ,3006 , 5006),(948.5,'2012-09-10',3003,5002),(2400.6 ,'2012-07-27 ', 3002 , 5001),( 5760,'2012-09-10',3001, 5001),(1983.43 ,' 2012-10-10 ' ,3005, 5004),(2480.4 ,'2012-10-10',3006, 5006),(250.45 ,'2012-06-27',3004, 5002),(75.29,'2012-08-17',3007,5005),(3045.6 ,'2012-04-25',3001 ,5001)



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

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

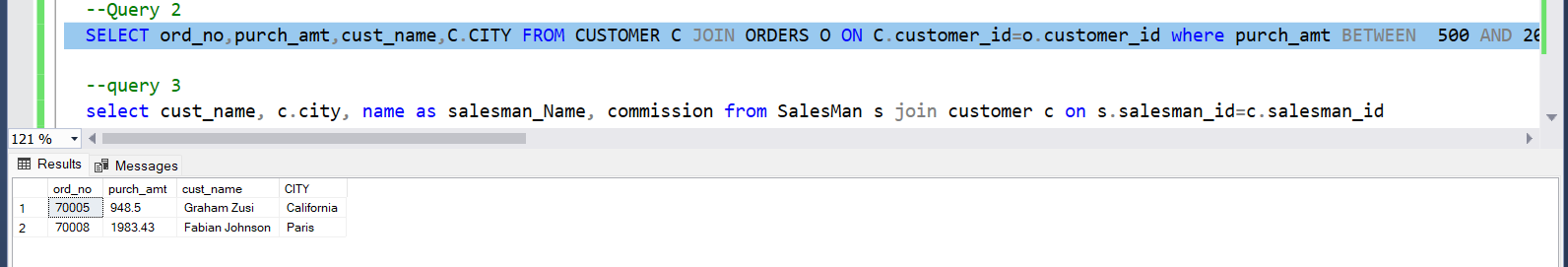
SELECT name as salesman\_name,cust\_name,s.city AS CITY FROM SALESMAN s join CUSTOMER c on s.city=c.city order by city desc



**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**

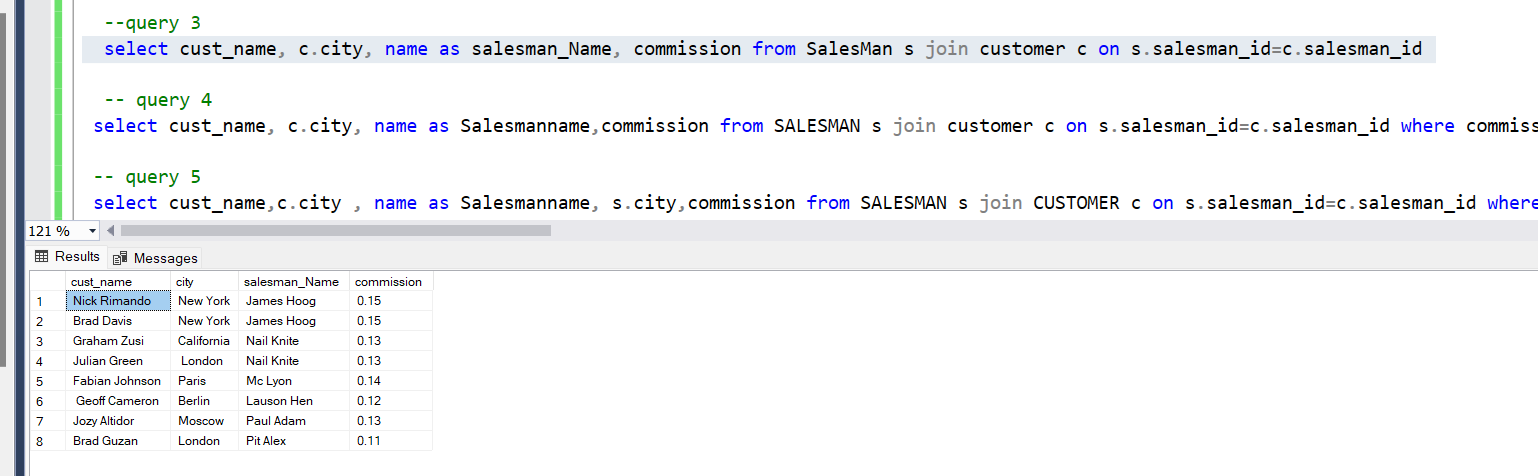
SELECT ord\_no,purch\_amt,cust\_name,C.CITY FROM CUSTOMER C JOIN ORDERS O ON C.customer\_id=o.customer\_id where purch\_amt BETWEEN 500 AND 2000

****

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

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

select cust\_name, c.city, name as salesman\_Name, commission from SalesMan s join customer c on s.salesman\_id=c.salesman\_id

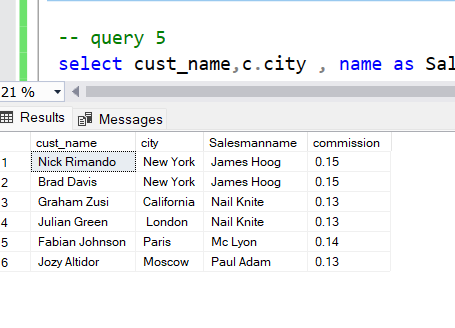


**4. write a SQL query to find salespeople who received commissions of more than 12**

**percent from the company. Return Customer Name, customer city, Salesman,**

**Commission.**

select cust\_name, c.city, name as Salesmanname,commission from SALESMAN s join customer c on s.salesman\_id=c.salesman\_id where commission > 0.12



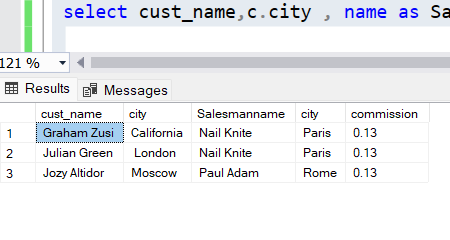
**5. write a SQL query to locate those salespeople who do not live in the same city where**

**their customers live and have received a commission of more than 12% from the**

**company. Return Customer Name, customer city, Salesman, salesman city,**

**Commission**

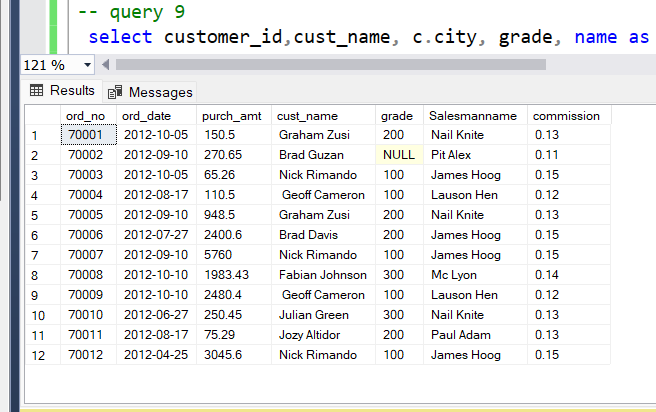
select cust\_name,c.city , name as Salesmanname, s.city,commission from SALESMAN s join CUSTOMER c on s.salesman\_id=c.salesman\_id where s.city!=c.city and commission > 0.12



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

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

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

****

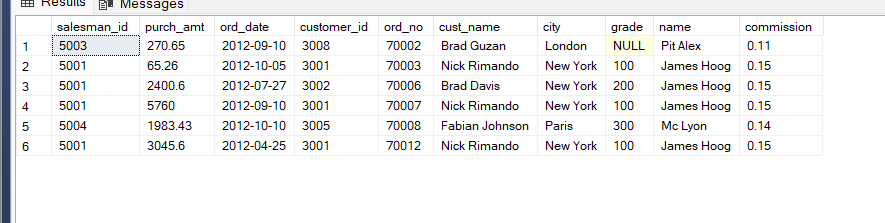
**7. Write a SQL statement to join the tables salesman, customer and orders so that the**

**same column of each table appears once and only the relational rows are returned.**

SELECT s.salesman\_id,o.purch\_amt,o.ord\_date, c.customer\_id, o.ord\_no,c.cust\_name,c.city,c.grade,s.name,s.commission

FROM orders AS o ,customer AS c, salesman as s

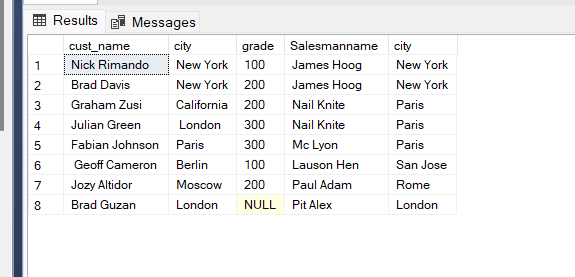
where o.customer\_id = c.customer\_id and o.salesman\_id = s.salesman\_id and c.city = s.city;



**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,c.city, grade, name as Salesmanname, s.city from SalesMan s join customer c on s.salesman\_id=c.salesman\_id order by c.customer\_id asc

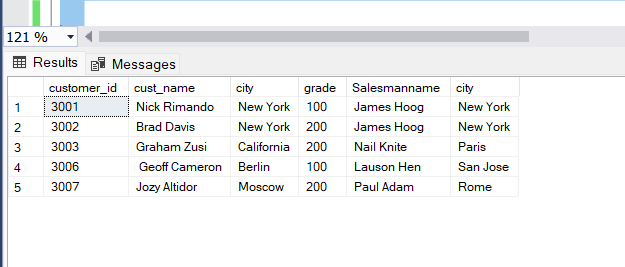


**9. write a SQL query to find those customers with a grade less than 300. Return**

**cust\_name, customer city, grade, Salesman, salesmancity. The result should be**

**ordered by ascending customer\_id.**

select customer\_id,cust\_name, c.city, grade, name as Salesmanname, s.city from salesman s join customer c on s.salesman\_id=c.salesman\_id where c.grade<300 order by customer\_id asc

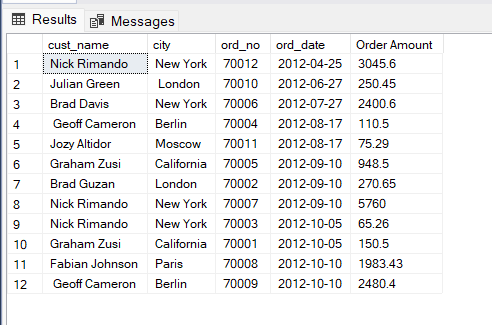
****

**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 a.cust\_name,a.city, b.ord\_no,b.ord\_date,b.purch\_amt AS "Order Amount" FROM customer a JOIN orders b ON a.customer\_id=b.customer\_id order by b.ord\_date;



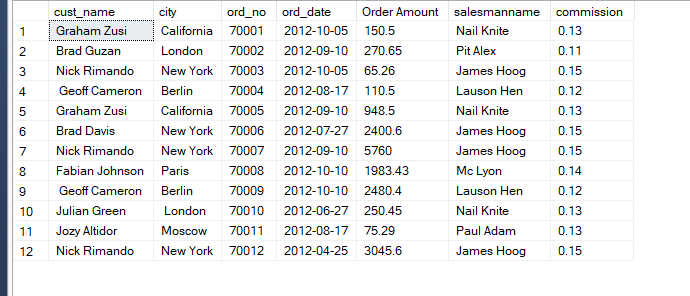
**11. Write a SQL statement to generate a report with customer name, city, order number,**

**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**

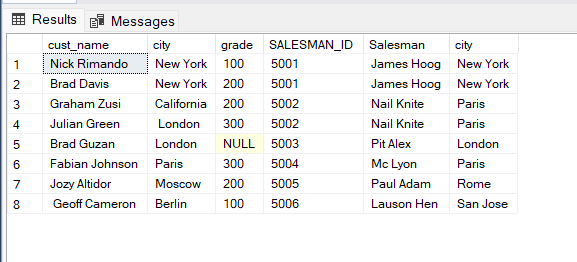
select a.cust\_name,a.city, b.ord\_no,b.ord\_date,b.purch\_amt AS "Order Amount", name as salesmanname,commission from orders b join customer a on b.customer\_id=a.customer\_id join salesman s on b.salesman\_id=s.salesman\_id

****

**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**

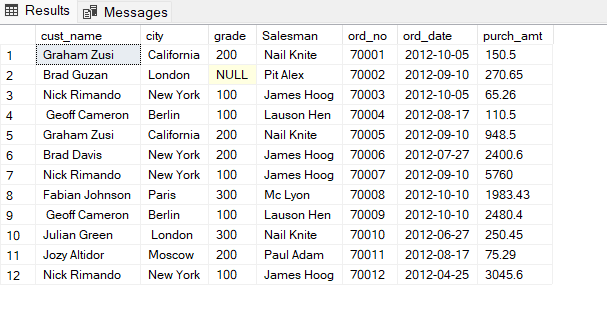
SELECT a.cust\_name,a.city,a.grade, B.SALESMAN\_ID,b.name AS "Salesman", b.city FROM customer a JOIN salesman b ON b.salesman\_id=a.salesman\_id ORDER BY b.salesman\_id;

****

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

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

SELECT 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 RIGHT OUTER JOIN orders c ON c.customer\_id=a.customer\_id;

****

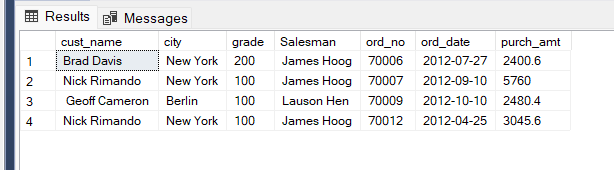
**14. Write a SQL statement to make a list for the salesmen who either work for one or**

**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 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>=2000AND a.grade IS NOT NULL;



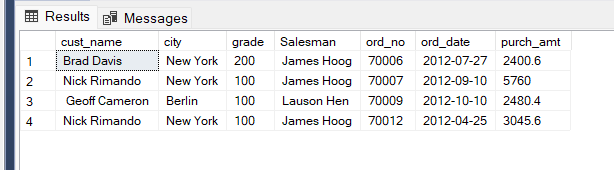
**15. Write a SQL statement to generate a list of all the salesmen who either work for one**

**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 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>=2000AND a.grade IS NOT NULL;



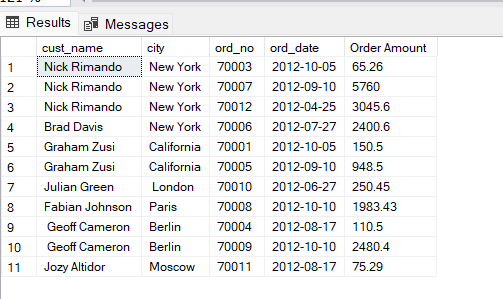
**16. Write a SQL statement to generate a report with the customer name, city, order no.**

**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.**

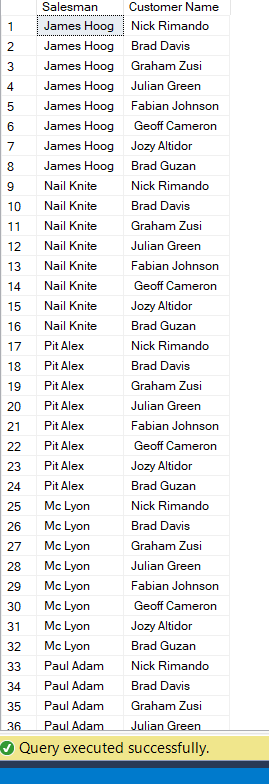
SELECT a.cust\_name,a.city, b.ord\_no,b.ord\_date,b.purch\_amt AS "Order Amount" FROM customer a FULL OUTER JOIN orders b ON a.customer\_id=b.customer\_id WHERE a.grade IS NOT NULL;



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

**customer table**

SELECT s.name AS "Salesman",c.cust\_name AS "Customer Name" 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**

SELECT s.name AS "Salesman",c.cust\_name AS "Customer Name" FROM salesman s CROSS JOIN customer c WHERE s.city IS NOT NULL;

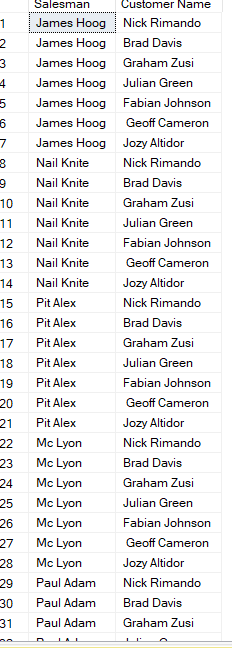


**19. Write a SQL statement to create a Cartesian product between salesperson and**

**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 s.name AS "Salesman", c.cust\_name AS "Customer Name" 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 s.name AS "Salesman",c.cust\_name AS "Customer Name" FROM salesman s CROSS JOIN customer c WHERE s.city IS NOT NULL AND s.city != c.city AND c.grade IS NOT NULL;

