SQL Assignments

Assignment 2

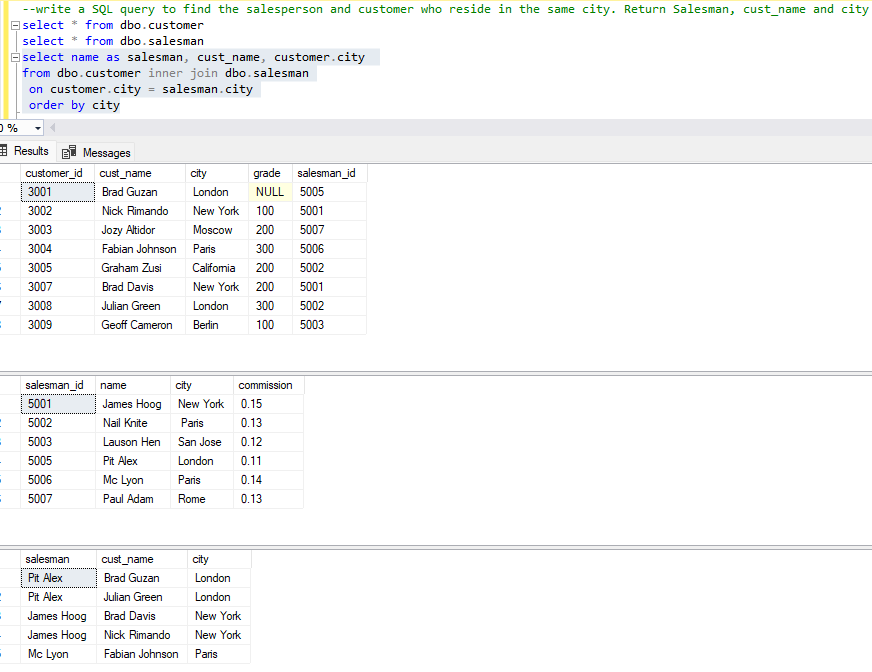
1. write a SQL query to find the salesperson and customer who reside in the same city. Return Salesman, cust\_name and city

select name as salesman, cust\_name, customer.city

from dbo.customer inner join dbo.salesman

on customer.city = salesman.city

order by city



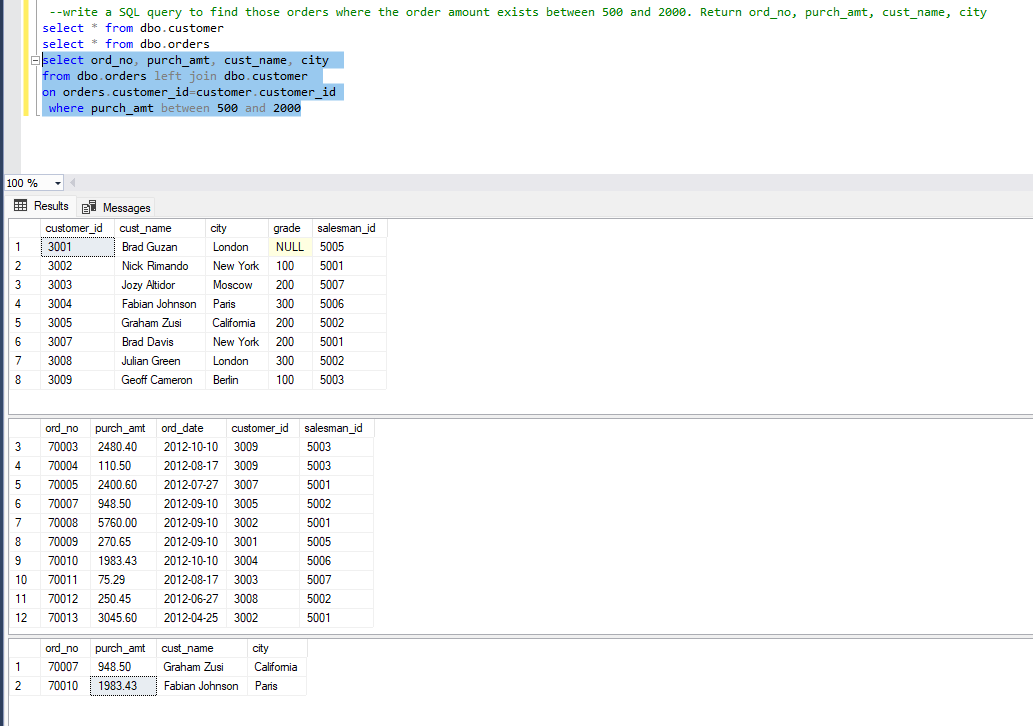
2. write a SQL query to find those orders where the order amount exists between 500 and 2000. Return ord\_no, purch\_amt, cust\_name, city

select ord\_no, purch\_amt, cust\_name, city

from dbo.orders left join dbo.customer

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

where purch\_amt between 500 and 2000

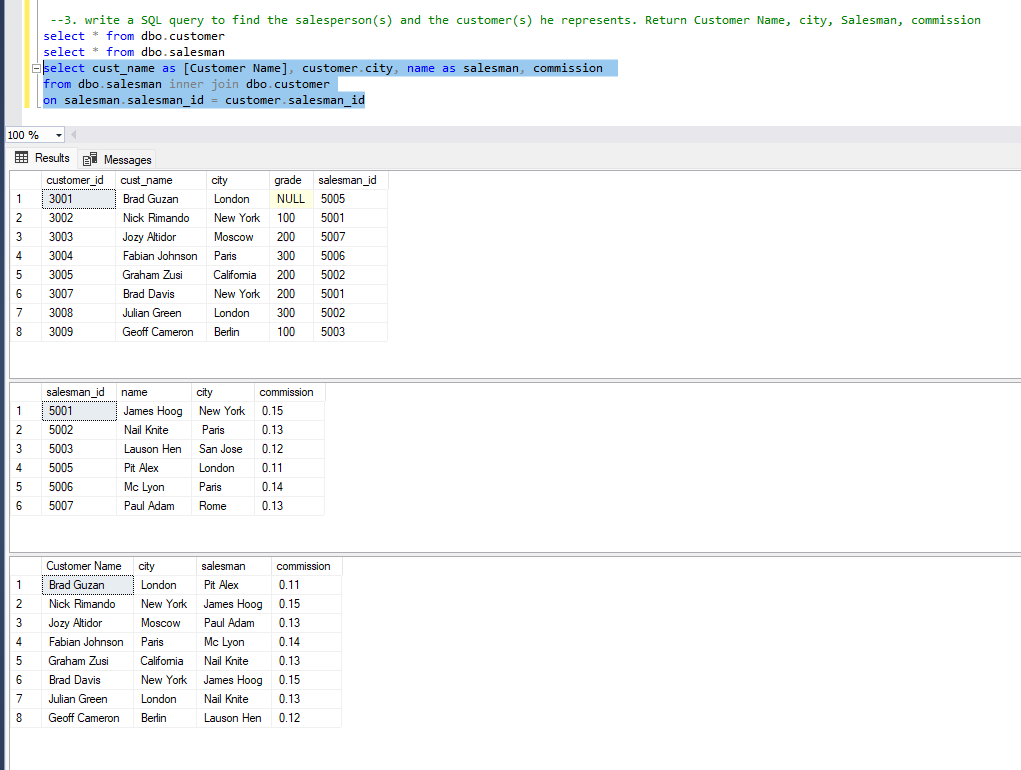


3. write a SQL query to find the salesperson(s) and the customer(s) he represents. Return Customer Name, city, Salesman, commission

select cust\_name as [Customer Name], customer.city, name as salesman, commission

from dbo.salesman inner join dbo.customer

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



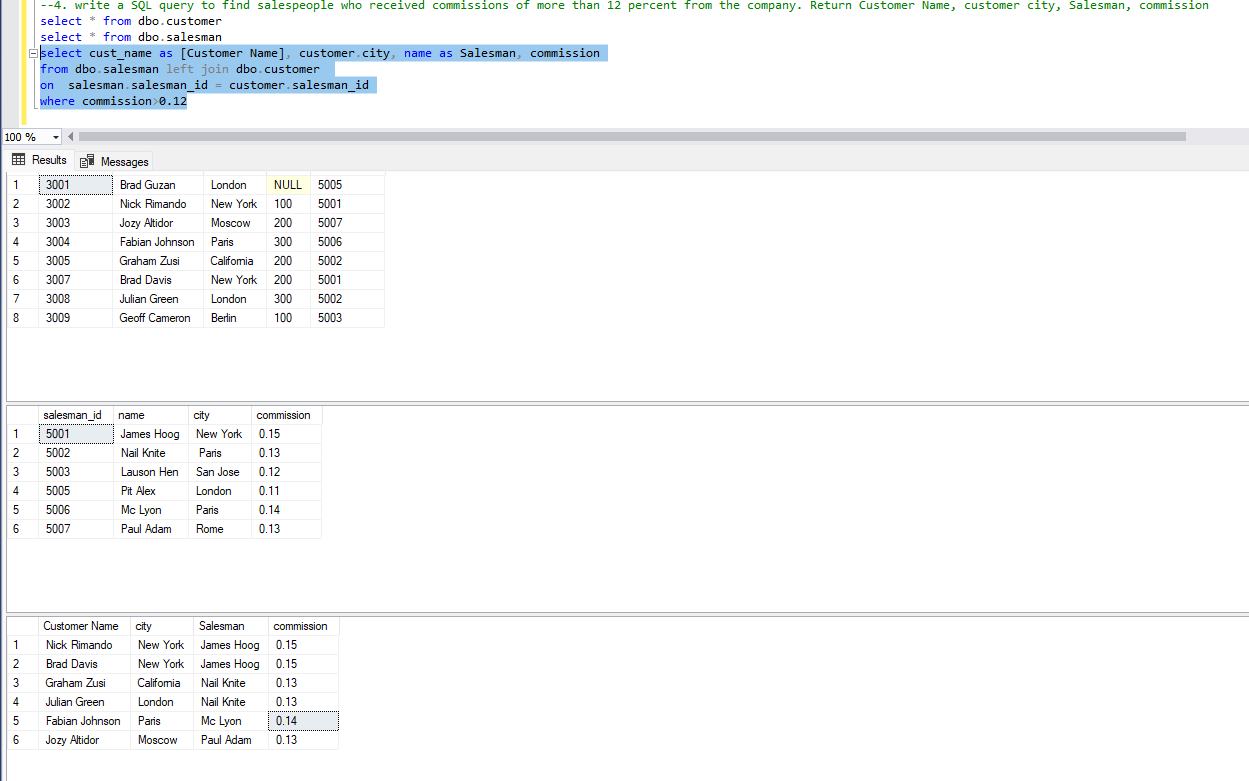
4. write a SQL query to find salespeople who received commissions of more than 12 percent from the company. Return Customer Name, customer city, Salesman, commission

select cust\_name as [Customer Name], customer.city, name as Salesman, commission

from dbo.salesman left join dbo.customer

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

where commission>0.12



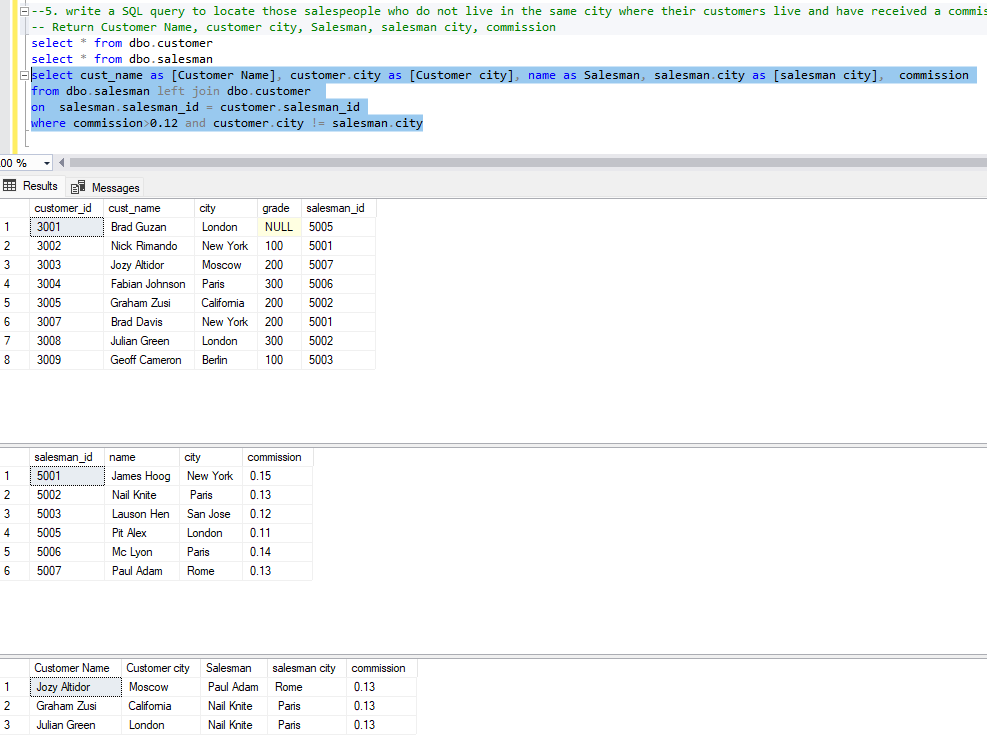
5. write a SQL query to locate those salespeople who do not live in the same city where their customers live and have received a commission of more than 12% from the company. Return Customer Name, customer city, Salesman, salesman city, commission

select cust\_name as [Customer Name], customer.city as [Customer city], name as Salesman, salesman.city as [salesman city], commission

from dbo.salesman left join dbo.customer

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

where commission>0.12 and customer.city != salesman.city

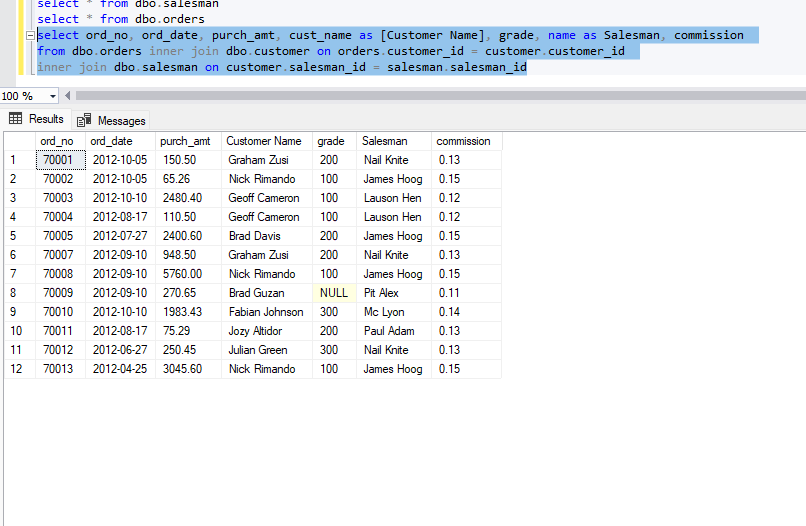


6. write a SQL query to find the details of an order. Return ord\_no, ord\_date, purch\_amt, Customer Name, grade, Salesman, commission

select ord\_no, ord\_date, purch\_amt, cust\_name as [Customer Name], grade, name as Salesman, commission

from dbo.orders inner join dbo.customer on orders.customer\_id = customer.customer\_id

inner join dbo.salesman on customer.salesman\_id = salesman.salesman\_id

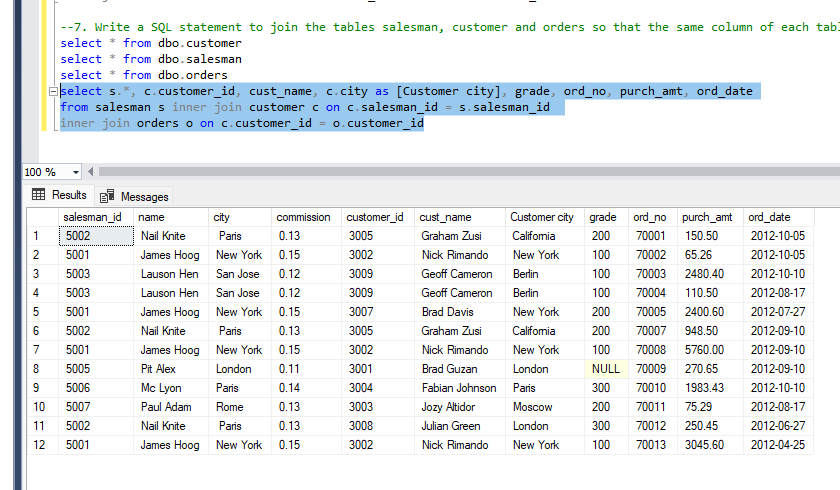


7. Write a SQL statement to join the tables salesman, customer and orders so that the same column of each table appears once and only the relational rows are returned.

select s.\*, c.customer\_id, cust\_name, c.city as [Customer city], grade, ord\_no, purch\_amt, ord\_date

from salesman s inner join customer c on c.salesman\_id = s.salesman\_id

inner join orders o on c.customer\_id = o.customer\_id



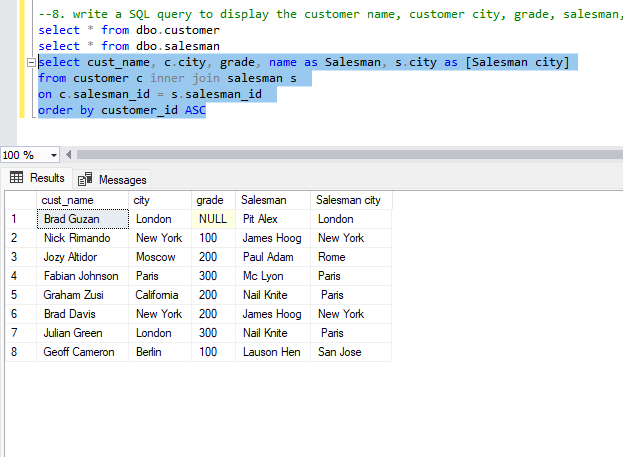
8. write a SQL query to display the customer name, customer city, grade, salesman, salesman city. The results should be sorted by ascending customer\_id.

select cust\_name, c.city, grade, name as Salesman, s.city as [Salesman city]

from customer c inner join salesman s

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

order by 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.

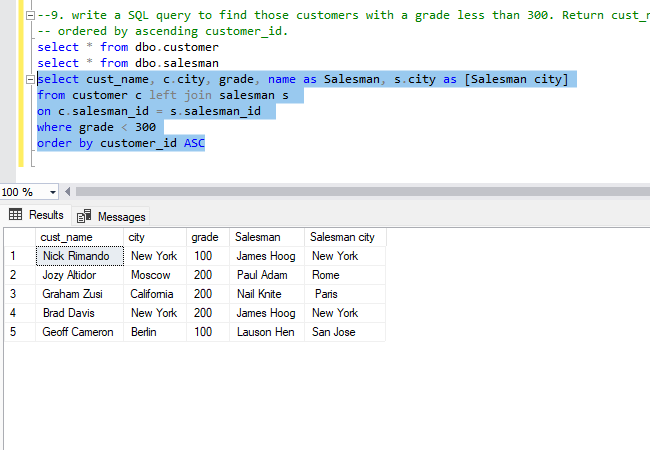
select cust\_name, c.city, grade, name as Salesman, s.city as [Salesman city]

from customer c left join salesman s

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

where grade < 300

order by customer\_id ASC



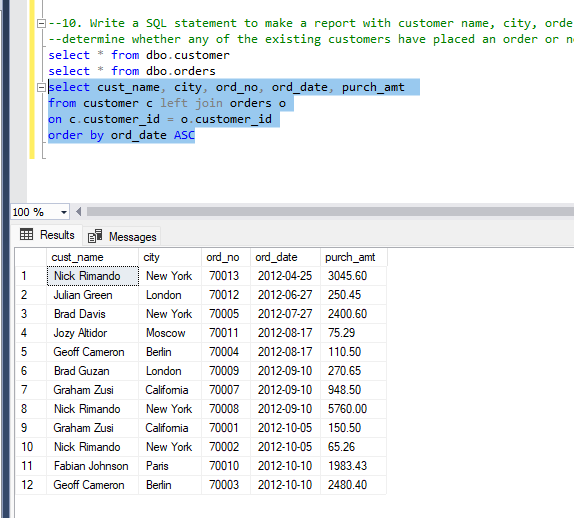
10. Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the order date to determine whether any of the existing customers have placed an order or not

select cust\_name, city, ord\_no, ord\_date, purch\_amt

from customer c left join orders o

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

order by ord\_date ASC

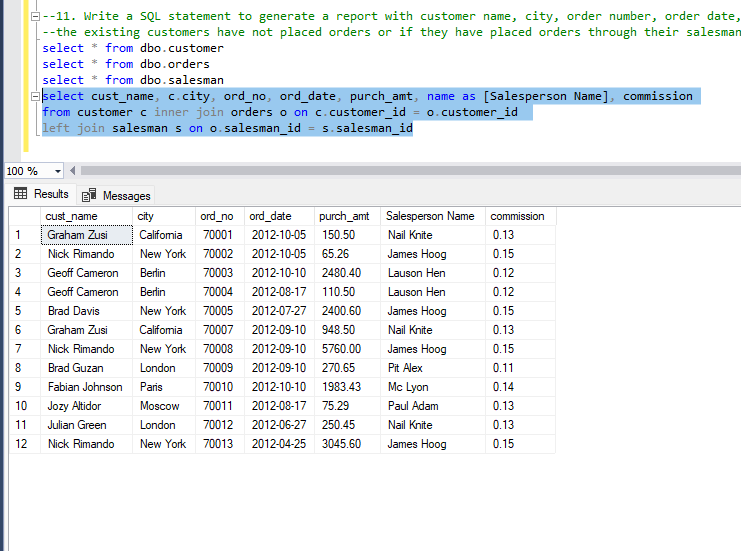


11. Write a SQL statement to generate a report with customer name, city, order number, order date, order amount, salesperson name, and commission to determine if any of the existing customers have not placed orders or if they have placed orders through their salesman or by themselves

select cust\_name, c.city, ord\_no, ord\_date, purch\_amt, name as [Salesperson Name], commission

from customer c inner join orders o on c.customer\_id = o.customer\_id

left join salesman s on o.salesman\_id = s.salesman\_id

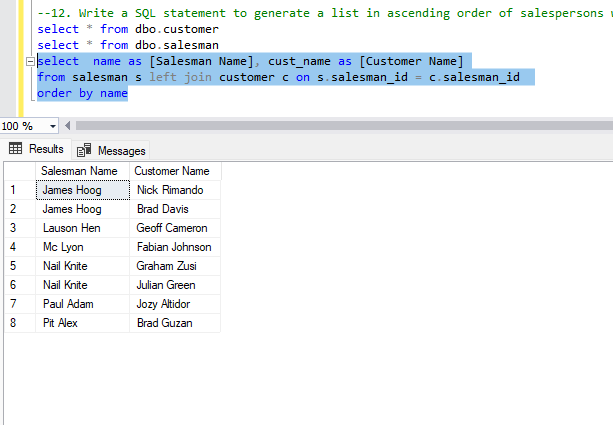


12. Write a SQL statement to generate a list in ascending order of salespersons who work either for one or more customers or have not yet joined any of the customers

select name as [Salesman Name], cust\_name as [Customer Name]

from salesman s left join customer c on s.salesman\_id = c.salesman\_id

order by name

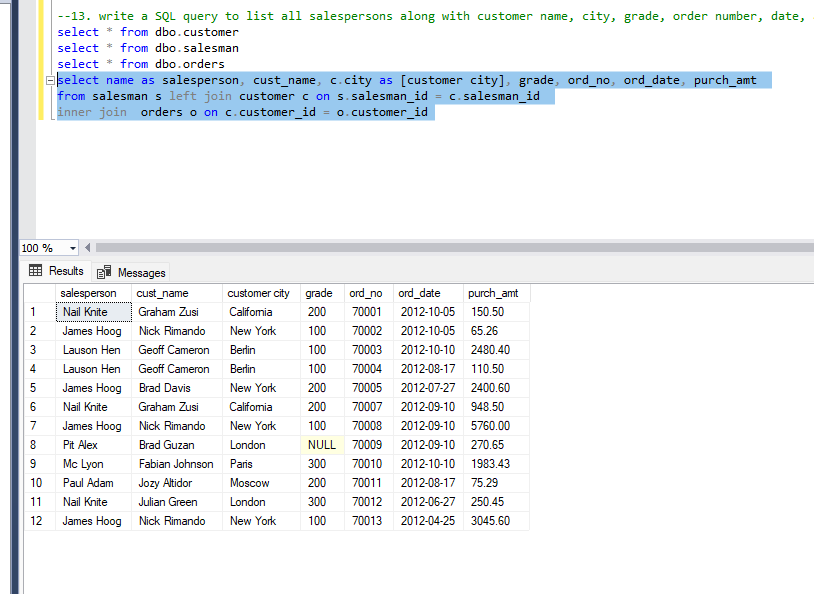


13. write a SQL query to list all salespersons along with customer name, city, grade, order number, date, and amount.

select name as salesperson, cust\_name, c.city as [customer city], grade, ord\_no, ord\_date, purch\_amt

from salesman s left join customer c on s.salesman\_id = c.salesman\_id

inner join orders o on c.customer\_id = o.customer\_id



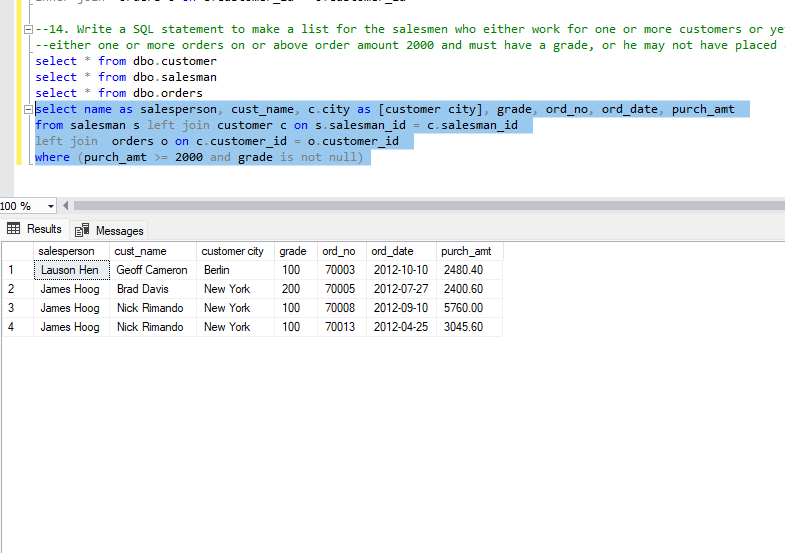
14. Write a SQL statement to make a list for the salesmen who either work for one or more customers or yet to join any of the customers. The customer may have placed, either one or more orders on or above order amount 2000 and must have a grade, or he may not have placed any order to the associated supplier.

select name as salesperson, cust\_name, c.city as [customer city], grade, ord\_no, ord\_date, purch\_amt

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

where (purch\_amt >= 2000 and grade is not null)



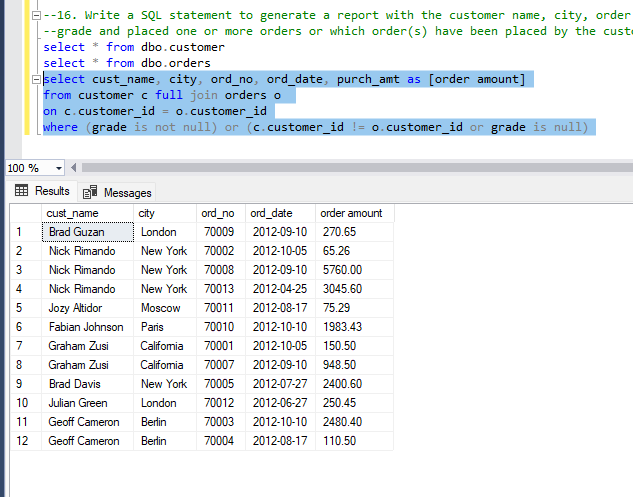
16. Write a SQL statement to generate a report with the customer name, city, order no. order date, purchase amount for only those customers on the list who must have a grade and placed one or more orders or which order(s) have been placed by the customer who neither is on the list nor has a grade

select cust\_name, city, ord\_no, ord\_date, purch\_amt as [order amount]

from customer c full join orders o

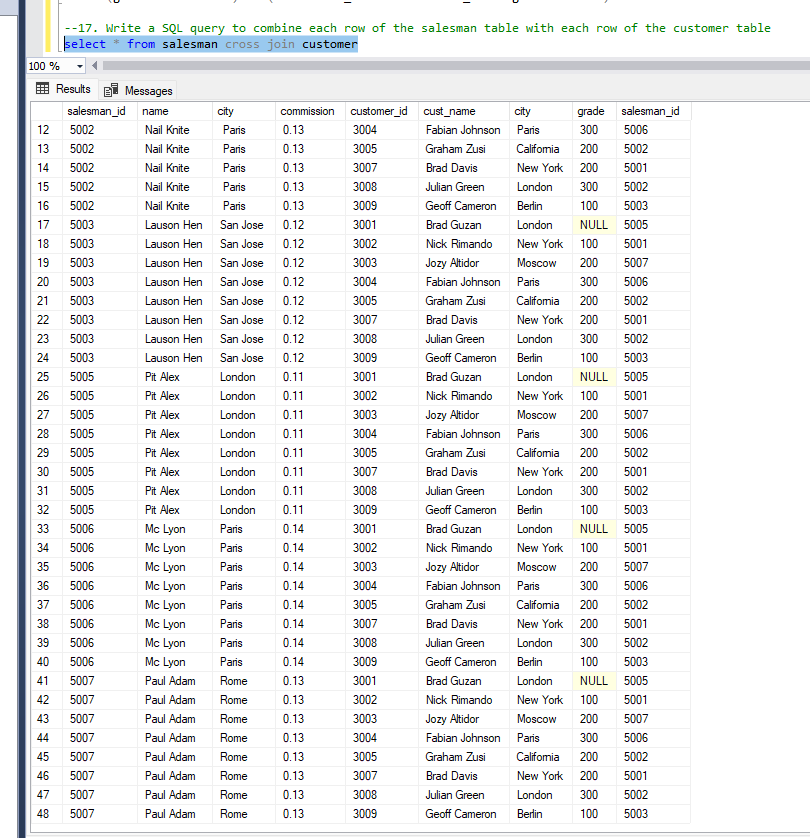
on c.customer\_id = o.customer\_id

where (grade is not null) or (c.customer\_id != o.customer\_id or grade is null)



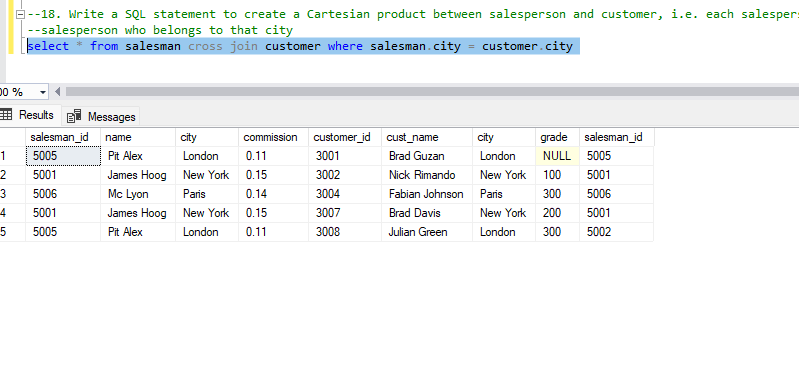
17. Write a SQL query to combine each row of the salesman table with each row of the customer table

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

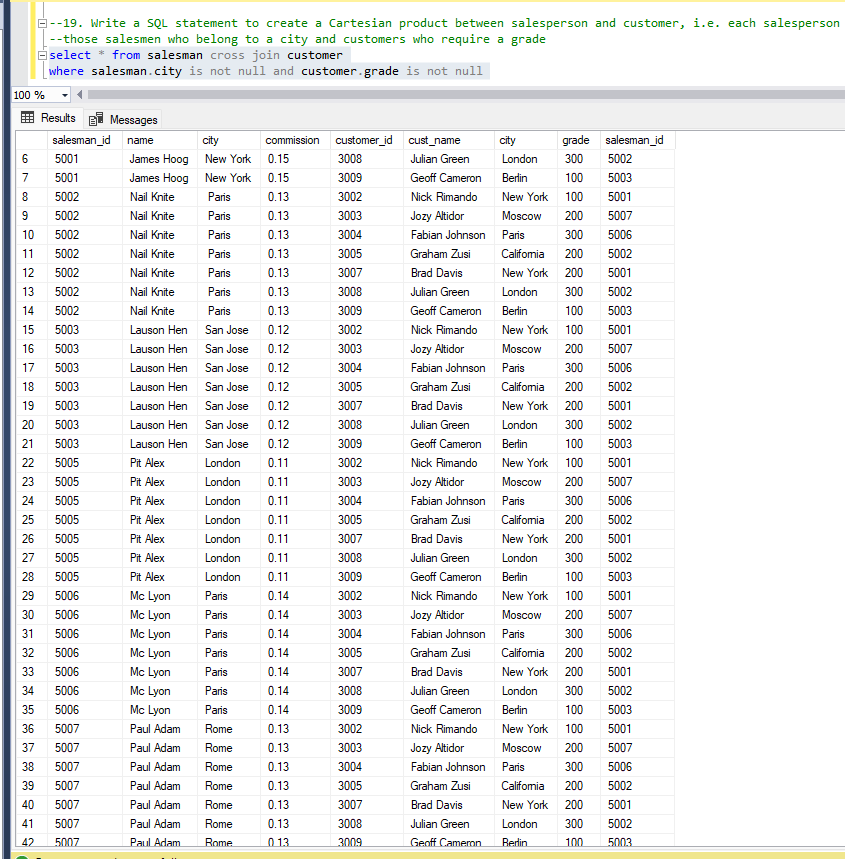
select \* from salesman cross join customer where salesman.city = customer.city



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 is not null and customer.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 cross join customer

where salesman.city != customer.city and grade is not null

