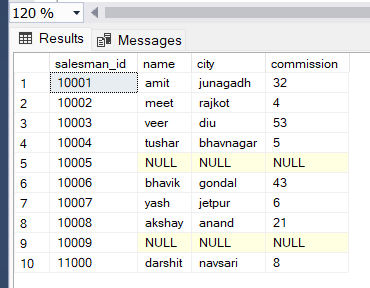
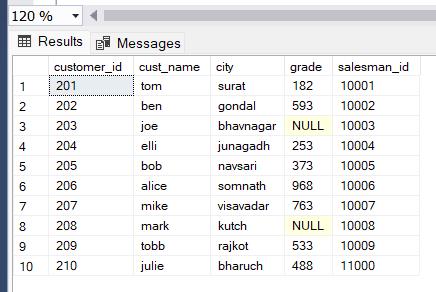
Output Screen shot of Assignment2

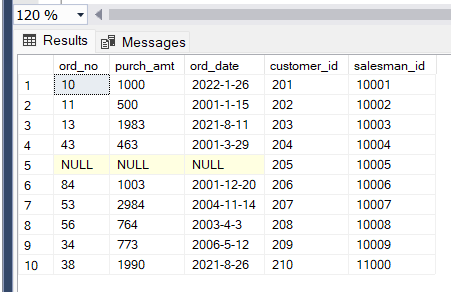
select \* from salesman;



select \* from customer;



select \* from orders;



Output of query1:

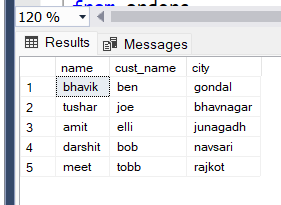
write a SQL query to find the salesperson and customer who reside in the same city. Return Salesman, cust\_name and city

select name,cust\_name,salesman.city

from salesman

inner join customer

on salesman.city=customer.city



Output of query2:

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

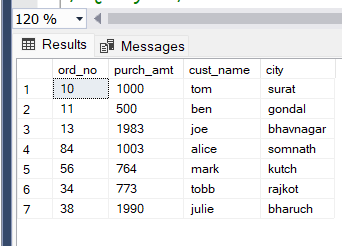
select ord\_no,purch\_amt,customer.cust\_name,customer.city

from orders

inner join customer

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

where purch\_amt between 500 and 2000



Output of query3:

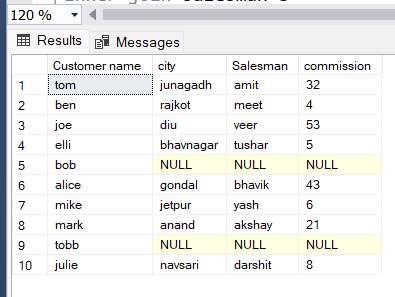
write a SQL query to find the salesperson(s) and the customer(s) he represents. Return Customer Name, city, Salesman, commission

select C.cust\_name as "Customer name",S.city,S.name as "Salesman",S.commission

from salesman S

inner join customer C

on S.salesman\_id = C.salesman\_id



Output of query4:

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.

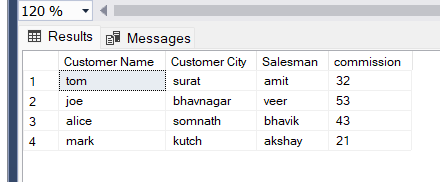
select C.cust\_name as "Customer Name",C.city as "Customer City", S.name as "Salesman",S.commission

from customer C

inner join salesman S

on S.salesman\_id=C.salesman\_id

where S.commission>12



Output of query5:

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 "Customer Name",C.city as "Customer City", S.name as "Salesman",S.city as "Salesman city",S.commission

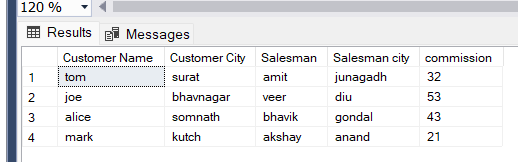
from customer C

inner join salesman S

on S.salesman\_id=C.salesman\_id

where S.commission>12

and S.city!=C.city



Output of query6:

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 as "Customer Name",C.grade,S.name as "Salesman",S.commission

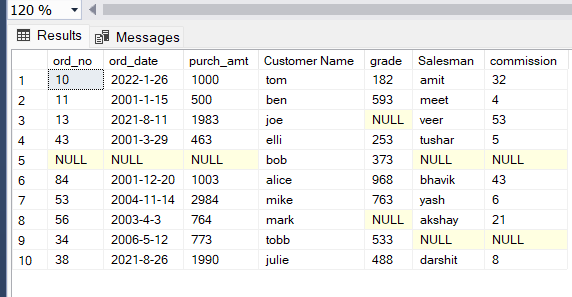
from orders O

join customer C

join salesman S

on S.salesman\_id=C.salesman\_id

on S.salesman\_id=O.salesman\_id



Output of query7:

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,S.name as "Salesman",S.city as "Salesmancity",S.commission,C.customer\_id,C.cust\_name,C.city as "Customer city",C.grade,

O.ord\_no,O.purch\_amt,O.ord\_date

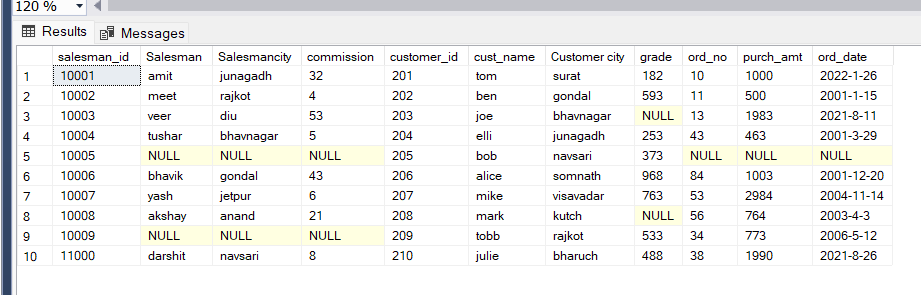
from salesman S

join customer C

join orders O

on O.salesman\_id=C.salesman\_id

on O.salesman\_id=S.salesman\_id



Output of query8:

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

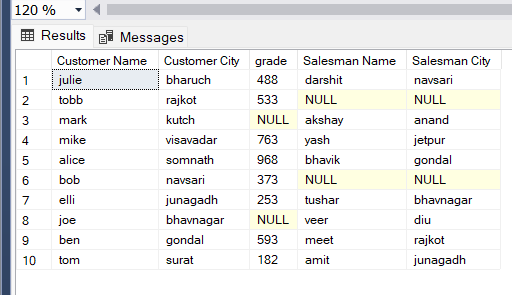
select C.cust\_name as "Customer Name",C.city as "Customer City",C.grade,S.name as "Salesman Name",S.city as "Salesman City"

from salesman S

join customer C

on S.salesman\_id=C.salesman\_id

order by C.customer\_id desc



Output of query9:

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 as "Customer City",C.grade,S.name as "Salesman Name",S.city as "Salesman City"

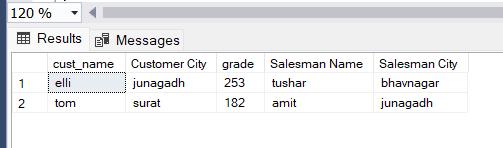
from salesman S

join customer C

on S.salesman\_id=C.salesman\_id

where C.grade<300

order by C.customer\_id desc



Output of query10:

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

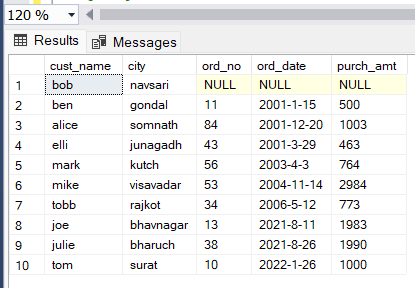
select C.cust\_name,C.city,O.ord\_no,O.ord\_date,O.purch\_amt

from customer C

join orders O

on C.customer\_id=O.customer\_id

order by O.ord\_date asc



Output of query11:

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 as "Customer City",O.ord\_no,O.ord\_date,O.purch\_amt,S.name as "Salesman Name",S.commission

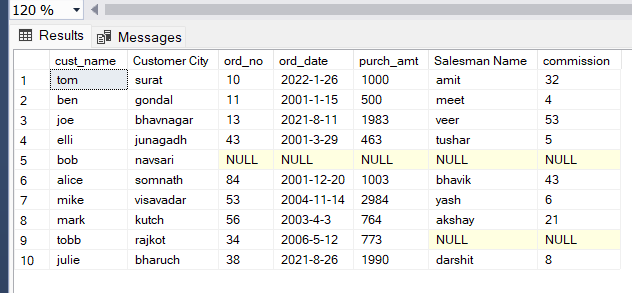
from customer C

full outer join salesman S

on C.salesman\_id=S.salesman\_id

full outer join orders O

on C.customer\_id=O.customer\_id



Output of query12:

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

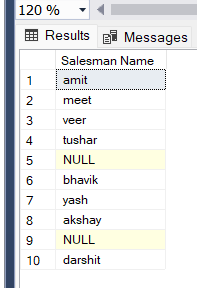
select S.name as "Salesman Name"

from customer C

left outer join salesman S

on S.salesman\_id=C.salesman\_id

order by S.salesman\_id asc



Output of query13:

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

select S.name as "Salesman Name",C.cust\_name,C.city as "Customer City",C.grade,O.ord\_no,O.ord\_date,O.purch\_amt

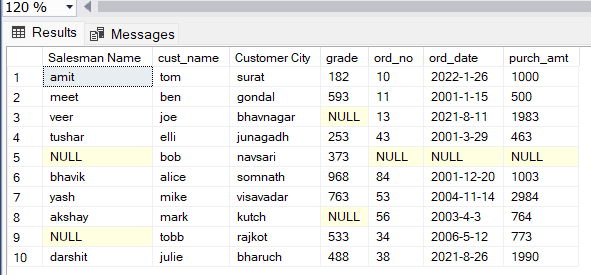
from salesman S

full outer join customer C

full outer join orders O

On O.customer\_id=C.customer\_id

on O.salesman\_id=S.salesman\_id



Output of query14:

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 C.cust\_name,O.ord\_no,O.ord\_date,O.purch\_amt,S.name,C.grade

from customer C

right outer join salesman S

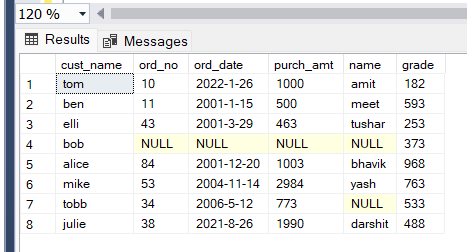
on C.salesman\_id=S.salesman\_id

left outer join orders O

on S.salesman\_id=O.salesman\_id

where O.purch\_amt>2000

or C.grade is not null



Output of query15:

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 C.cust\_name,O.ord\_no,O.ord\_date,O.purch\_amt,S.name,C.grade

from customer C

right outer join salesman S

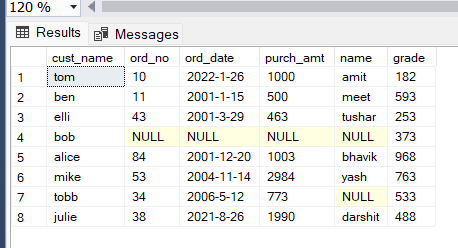
on C.salesman\_id=S.salesman\_id

left outer join orders O

on S.salesman\_id=O.salesman\_id

where O.purch\_amt>2000

or C.grade is not null



Output of query16:

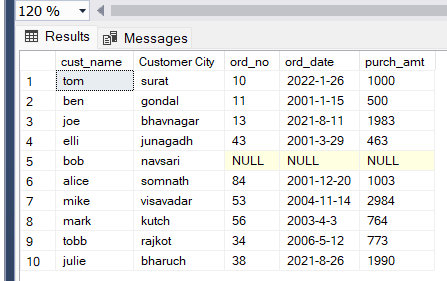
Write a SQL statement to generate a report with the customer name, city, order no. order date, purchase amount for only those customers on the list who must have a grade and placed one or more orders or which order(s) have been placed by the customer who neither is on the list nor has a grade.

select C.cust\_name,C.city as "Customer City",O.ord\_no,O.ord\_date,O.purch\_amt

from customer C

left outer join orders O

on C.customer\_id=O.customer\_id



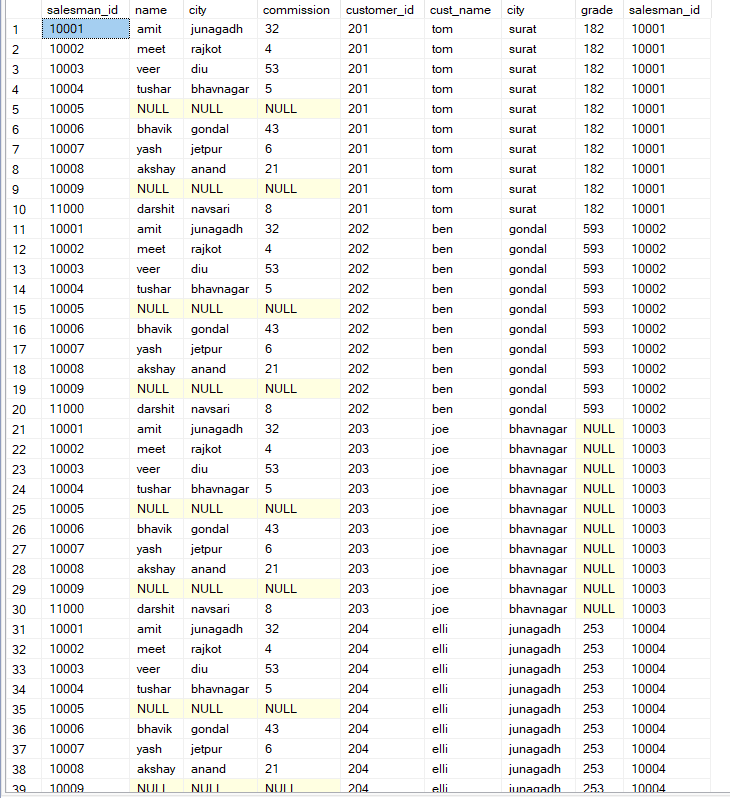
Output of query17:

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



Output of query18:

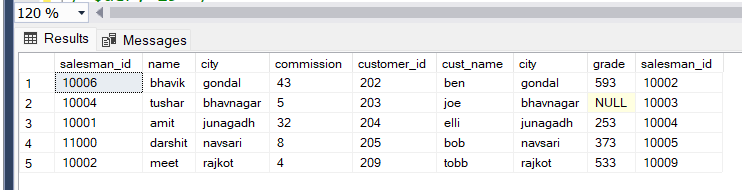
Write a SQL statement to create a Cartesian product between salesperson and customer, i.e. each salesperson will appear for all customers and vice versa for that salesperson who belongs to that city

select \*

from salesman S

cross join customer C

where S.city=C.city



Output of query19:

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

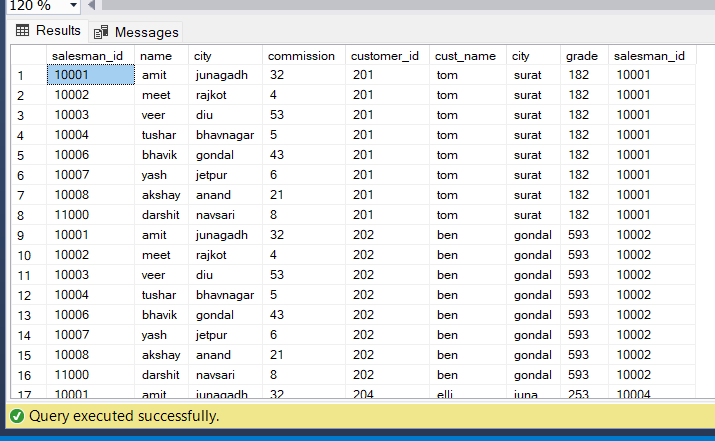
select \*

from salesman S

cross join customer C

where S.city is not null

and C.grade is not null



Output of query20:

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!=C.city

and C.grade is not null

