Analysis for AC and RO

**Category level orders placed for AC and RO**

Select e.customer\_state,c.column2 as category, count(b.order\_id) as orders from products a

Join items b

On a.product\_id = b.product\_id

join product\_category\_name\_translation c

On a.product\_category\_name = c.column1

join orders d

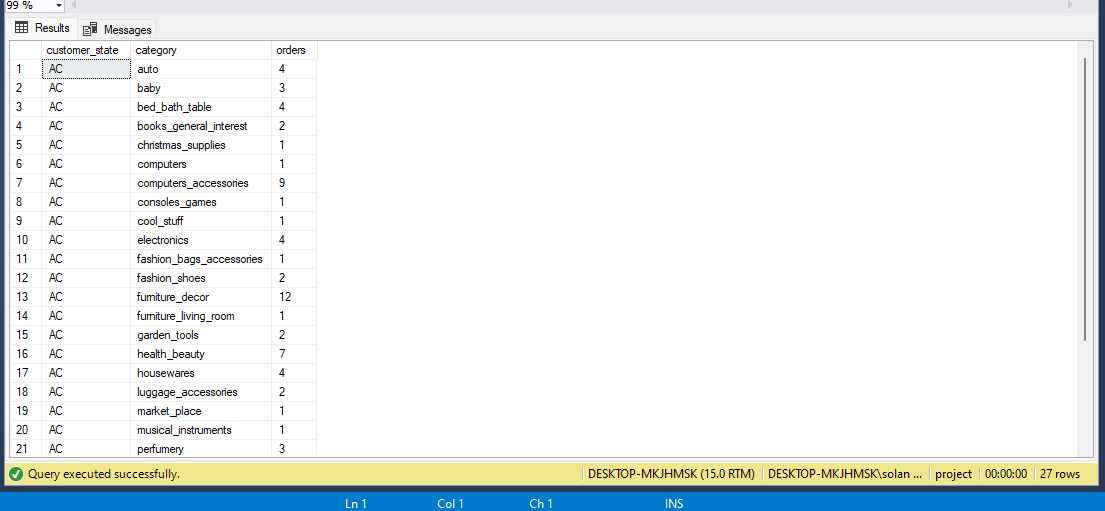
On b.order\_id = d.order\_id

Join customers e

On d.customer\_id = e.customer\_id

group by c.column2, e.customer\_state

Having e.customer\_state = 'AC';



Select e.customer\_state,c.column2 as category, count(b.order\_id) as orders from products a

Join items b

On a.product\_id = b.product\_id

join product\_category\_name\_translation c

On a.product\_category\_name = c.column1

join orders d

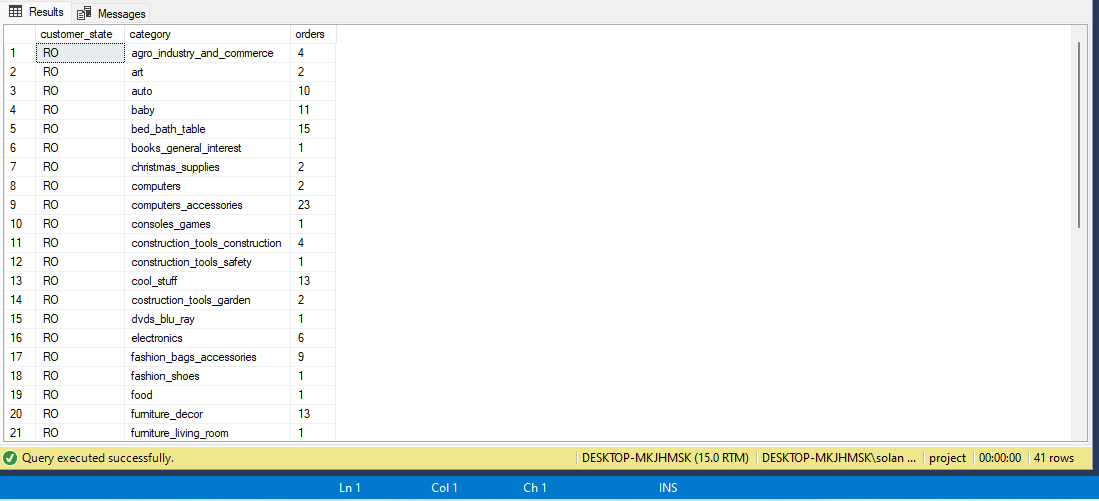
On b.order\_id = d.order\_id

Join customers e

On d.customer\_id = e.customer\_id

group by c.column2, e.customer\_state

Having e.customer\_state = 'RO';



**Analysis of orders delivered before and after the expected date for AC and RO**

select count(\*) as delay from orders a

Join items b

On a.order\_id = b.order\_id

Join customers d

On a.customer\_id = d.customer\_id

Join reviews e

On a.order\_id = e.order\_id

where customer\_state like 'AC' and datediff(day,a.order\_estimated\_delivery\_date,a.order\_delivered\_customer\_date) > 0;

select count(\*) as before\_time from orders a

Join items b

On a.order\_id = b.order\_id

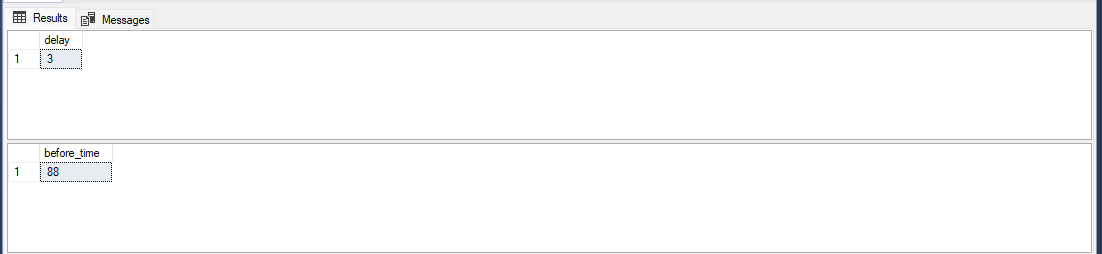
Join customers d

On a.customer\_id = d.customer\_id

Join reviews e

On a.order\_id = e.order\_id

where customer\_state like 'AC' and datediff(day,a.order\_estimated\_delivery\_date,a.order\_delivered\_customer\_date) <= 0;



select count(\*) as delay from orders a

Join items b

On a.order\_id = b.order\_id

Join customers d

On a.customer\_id = d.customer\_id

Join reviews e

On a.order\_id = e.order\_id

where customer\_state like 'RO' and datediff(day,a.order\_estimated\_delivery\_date,a.order\_delivered\_customer\_date) > 0;

select count(\*) as before\_time from orders a

Join items b

On a.order\_id = b.order\_id

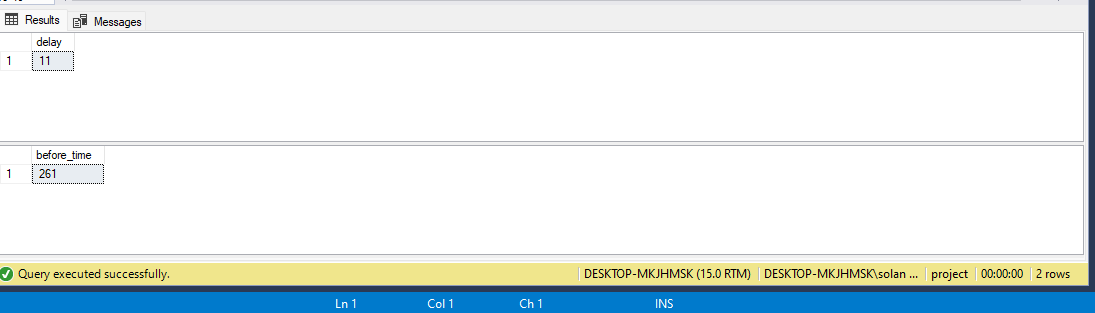
Join customers d

On a.customer\_id = d.customer\_id

Join reviews e

On a.order\_id = e.order\_id

where customer\_state like 'RO' and datediff(day,a.order\_estimated\_delivery\_date,a.order\_delivered\_customer\_date) <= 0;



**Delay in shipping by the sellers for AC and RO**

select count(\*) as delay

from orders a

Join items b

On a.order\_id = b.order\_id

Join sellers c

On b.seller\_id = c.seller\_id

Join customers d

On a.customer\_id = d.customer\_id

where customer\_state like 'AC' and datediff(day,b.shipping\_limit\_date,a.order\_delivered\_carrier\_date) > 0;

select count(\*) as before\_time

from orders a

Join items b

On a.order\_id = b.order\_id

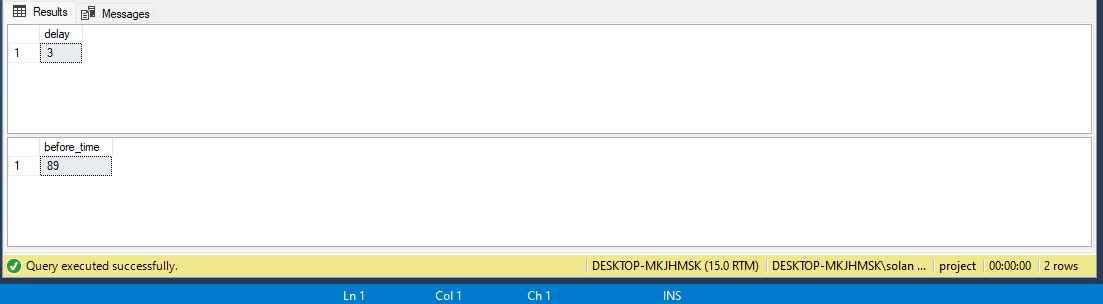
Join sellers c

On b.seller\_id = c.seller\_id

Join customers d

On a.customer\_id = d.customer\_id

where customer\_state like 'AC' and datediff(day,b.shipping\_limit\_date,a.order\_delivered\_carrier\_date) <= 0;



select count(\*) as delay

from orders a

Join items b

On a.order\_id = b.order\_id

Join sellers c

On b.seller\_id = c.seller\_id

Join customers d

On a.customer\_id = d.customer\_id

where customer\_state like 'RO' and datediff(day,b.shipping\_limit\_date,a.order\_delivered\_carrier\_date) > 0;

select count(\*) as before\_time

from orders a

Join items b

On a.order\_id = b.order\_id

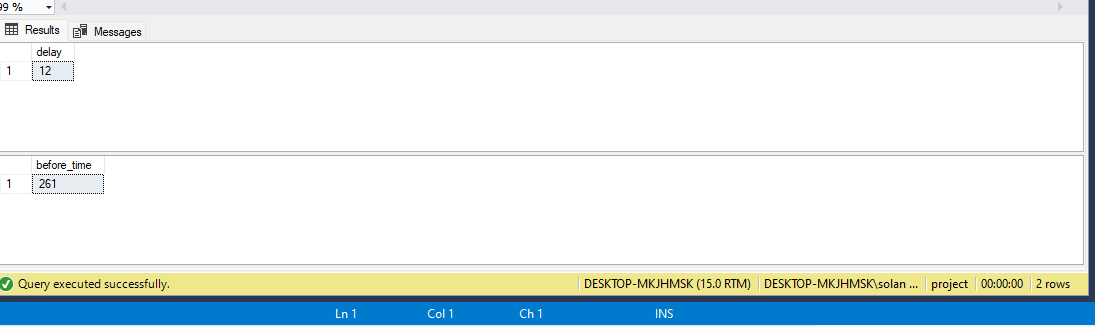
Join sellers c

On b.seller\_id = c.seller\_id

Join customers d

On a.customer\_id = d.customer\_id

where customer\_state like 'RO' and datediff(day,b.shipping\_limit\_date,a.order\_delivered\_carrier\_date) <= 0;



Cities with orders placed per year for AC and RO

Select e.customer\_state,e.customer\_city,datepart(year,d.order\_delivered\_customer\_date) as Year, count(b.order\_id) as orders from products a

Join items b

On a.product\_id = b.product\_id

join product\_category\_name\_translation c

On a.product\_category\_name = c.column1

join orders d

On b.order\_id = d.order\_id

Join customers e

On d.customer\_id = e.customer\_id

group by e.customer\_state, e.customer\_city,datepart(year,d.order\_delivered\_customer\_date)

Having e.customer\_state = 'AC'

Order by count(b.order\_id) desc;

Select e.customer\_state,e.customer\_city,datepart(year,d.order\_delivered\_customer\_date) as Year, count(b.order\_id) as orders from products a

Join items b

On a.product\_id = b.product\_id

join product\_category\_name\_translation c

On a.product\_category\_name = c.column1

join orders d

On b.order\_id = d.order\_id

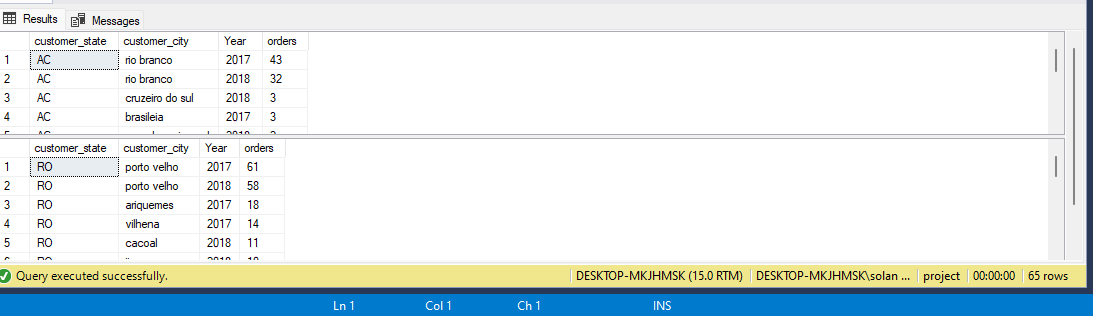
Join customers e

On d.customer\_id = e.customer\_id

group by e.customer\_state, e.customer\_city,datepart(year,d.order\_delivered\_customer\_date)

Having e.customer\_state = 'RO'

Order by count(b.order\_id) desc;



Cities with max delayed orders for AC and RO

select d.customer\_state,d.customer\_city,datediff(day,a.order\_estimated\_delivery\_date,a.order\_delivered\_customer\_date) as delay,

sum(count(\*)) over(partition by d.customer\_city) as total\_delayed\_orders

from orders a

Join items b

On a.order\_id = b.order\_id

Join customers d

On a.customer\_id = d.customer\_id

Join reviews e

On a.order\_id = e.order\_id

group by d.customer\_state,d.customer\_city,datediff(day,a.order\_estimated\_delivery\_date,a.order\_delivered\_customer\_date)

having customer\_state like 'AC' and datediff(day,a.order\_estimated\_delivery\_date,a.order\_delivered\_customer\_date) > 0

order by sum(count(\*)) over(partition by d.customer\_city) desc;

select d.customer\_state,d.customer\_city,datediff(day,a.order\_estimated\_delivery\_date,a.order\_delivered\_customer\_date) as delay,

sum(count(\*)) over(partition by d.customer\_city) as total\_delayed\_orders

from orders a

Join items b

On a.order\_id = b.order\_id

Join customers d

On a.customer\_id = d.customer\_id

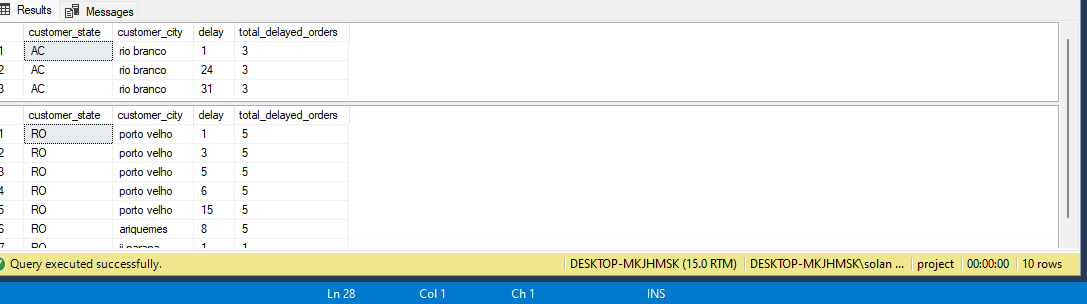
Join reviews e

On a.order\_id = e.order\_id

group by d.customer\_state,d.customer\_city,datediff(day,a.order\_estimated\_delivery\_date,a.order\_delivered\_customer\_date)

having customer\_state like 'RO' and datediff(day,a.order\_estimated\_delivery\_date,a.order\_delivered\_customer\_date) > 0

order by sum(count(\*)) over(partition by d.customer\_city) desc;



**We see that No Branco is the city that sets the trend for AC in terms of maximum number of orders and also delayed orders**

**Porto Velho sets the trend for AC in terms of max number of orders and delayed orders**