

A. Create the different metrics like Sales, customer acquisitions, total no. of orders for each Year across the different states they serve.

Does all the metrics show similar trends or is there any disparity amongst each of them?

--A) Sales

```
WITH CTE1 AS(  
SELECT YEAR(0.order_purchase_timestamp)Years,C.customer_state,ROUND(SUM(I.price),2)Sales  
FROM customers C  
JOIN order1 O  
ON C.customer_id = O.customer_id  
  
JOIN order_items I  
ON O.order_id = I.order_id  
  
WHERE O.order_status NOT IN('unavailable','canceled')  
  
GROUP BY C.customer_state,YEAR(0.order_purchase_timestamp)),  
--  
CTE2 AS(  
SELECT *,ROW_NUMBER() OVER(PARTITION BY Years,customer_state ORDER BY  
customer_state)Ranks  
FROM CTE1)  
--  
SELECT Years,customer_state,sales FROM CTE2;
```

The screenshot displays the Microsoft SQL Server Management Studio interface. The query editor shows the SQL code for CTE1 and CTE2. The Results pane shows the output of the query, which is a table with three columns: Years, customer_state, and sales. The data is sorted by Years, then customer_state, and then sales. The status bar at the bottom indicates that the query was executed successfully and returned 75 rows.

Years	customer_state	sales
2016	AL	82.49
2016	AK	894.05
2016	CA	1689.38
2016	DE	1043.77
2016	ES	917.79
2016	GO	984.39
2016	MA	704.28
2016	MG	4652.22
2016	MT	327.79
2016	PA	1087.6
2016	PB	49.9
2016	PE	1369.1
2016	PI	210
2016	PR	2015.51
2016	RJ	9187.18
2016	RN	728.69
2016	RR	112.59

--B) Customer Acquisitions

```
WITH CTE3 AS(  

```

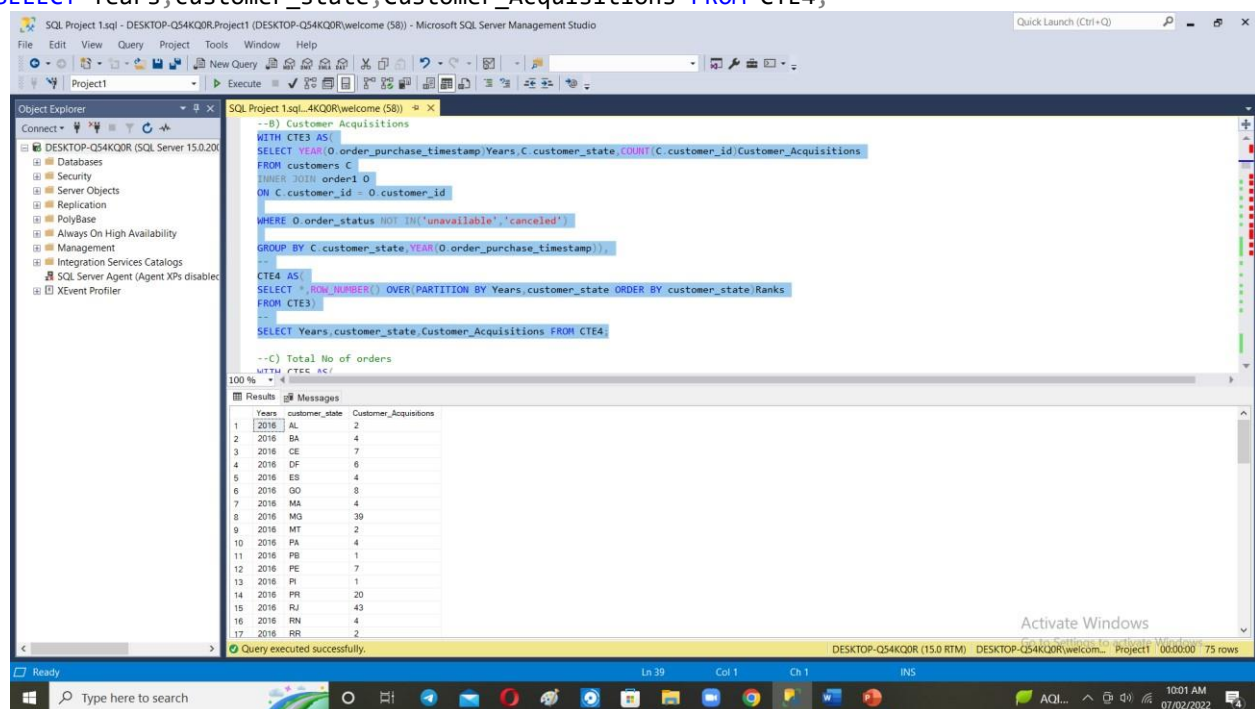
```

SELECT
YEAR(0.order_purchase_timestamp)Years,C.customer_state,COUNT(C.customer_id)Customer_Acqui
sitions FROM customers C
INNER JOIN order1 O
ON C.customer_id = O.customer_id

WHERE 0.order_status NOT IN('unavailable','canceled')

GROUP BY C.customer_state,YEAR(0.order_purchase_timestamp)),
--
CTE4 AS(
SELECT *,ROW_NUMBER() OVER(PARTITION BY Years,customer_state ORDER BY
customer_state)Ranks
FROM CTE3)
--
SELECT Years,customer_state,Customer_Acquisitions FROM CTE4;

```



```

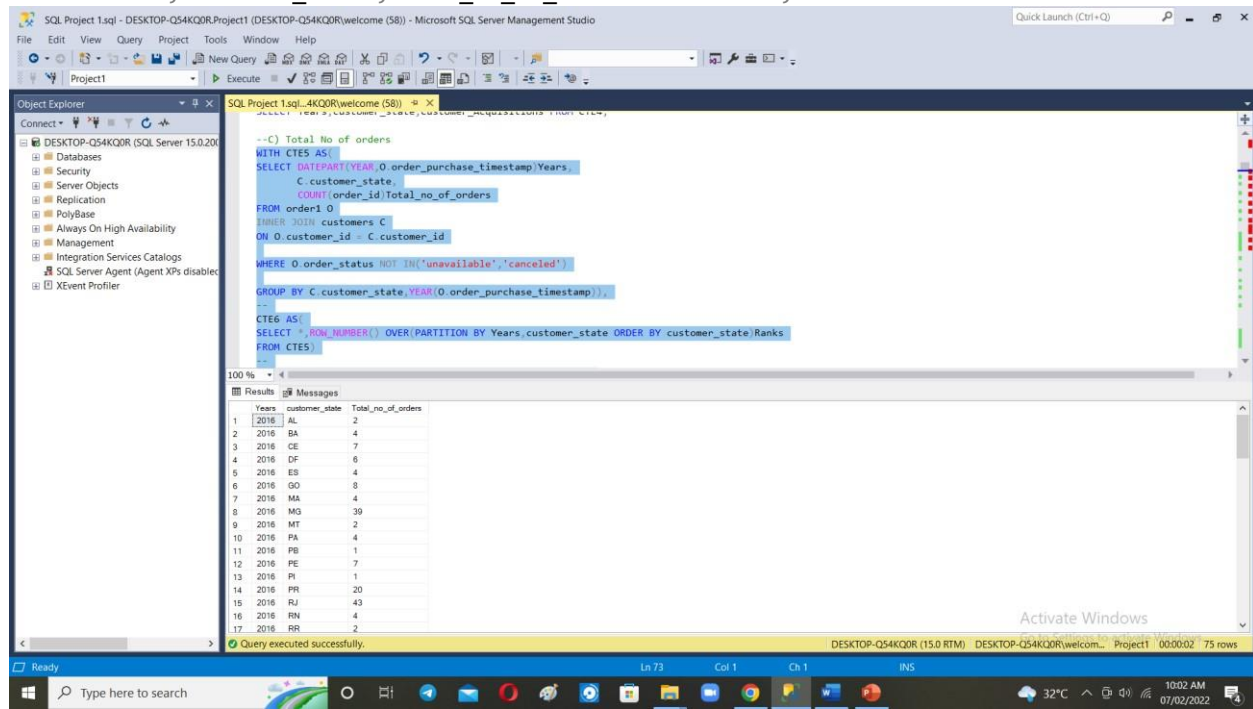
--C) Total No of orders
WITH CTE5 AS(
SELECT DATEPART(YEAR,0.order_purchase_timestamp)Years,
       C.customer_state,
       COUNT(order_id)Total_no_of_orders
FROM order1 O
INNER JOIN customers C
ON O.customer_id = C.customer_id

WHERE 0.order_status NOT IN('unavailable','canceled')
GROUP BY C.customer_state,YEAR(0.order_purchase_timestamp)),
--
CTE6 AS(

```

```
SELECT *,ROW_NUMBER() OVER(PARTITION BY Years,customer_state ORDER BY
customer_state)Ranks
FROM CTE5)
```

```
--
SELECT Years,customer_state,Total_no_of_orders FROM CTE6;
```



C. For the States identified above, do the Root Cause analysis for their performance across a variety of metrics.

You can utilize the following metrics and explore a few yourself as well by analyzing the data.

i)Category level Sales and orders placed,

A) INCREASING

```
SELECT
YEAR(O.order_purchase_timestamp)Years,C.customer_state,P.product_category_name,COUNT(P.pr
oduct_category_name)Order_placed
FROM order1 O
INNER JOIN customers C
ON O.customer_id = C.customer_id

INNER JOIN order_items I
ON O.order_id = I.order_id

INNER JOIN products P
ON I.product_id = P.product_id

WHERE C.customer_state IN('AP','RR') And O.order_status NOT IN('unavailable','canceled')
```

```
GROUP BY YEAR(O.order_purchase_timestamp),C.customer_state,P.product_category_name
```

ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

SQL Project 1.sql - DESKTOP-Q54KQOR (SQL Server 15.0.2000.1) - Microsoft SQL Server Enterprise Manager

Object Explorer: DESKTOP-Q54KQOR (SQL Server 15.0.2000.1) > Databases > Project1

Query: A) INCREASING

```
SELECT YEAR(0.order_purchase_timestamp)Years,C.customer_state,P.product_category_name,COUNT(P.product_category_name)Order_palced
FROM order1 O
INNER JOIN customers C
ON O.customer_id = C.customer_id
INNER JOIN order_items I
ON O.order_id = I.order_id
INNER JOIN products P
ON I.product_id = P.product_id
WHERE C.customer_state IN('AP','RR') And O.order_status NOT IN('unavailable','canceled')
GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,P.product_category_name
```

Results: 60 rows

Years	customer_state	product_category_name	Order_palced
2016	RR	beleza_saude	1
2016	RR	moveis_decoracao	2
2017	AP	agro_industria_e_comercio	1
2017	AP	automotivo	1
2017	AP	beleza_saude	7
2017	AP	brinquedos	2
2017	AP	camas_mesa_banho	6
2017	AP	cool_stuff	1
2017	AP	eletronicos	3
2017	AP	ferramentas_jardim	1
2017	AP	informatica_acessorios	2
2017	AP	instrumentos_musicais	1
2017	AP	moveis_decoracao	2
2017	AP	pets	1
2017	AP	pet_shop	2
2017	AP	religiao_presentes	1
2017	AP	telefonia	1

Query executed successfully.

B) DECREASING

```
SELECT
YEAR(0.order_purchase_timestamp)Years,C.customer_state,P.product_category_name,COUNT(P.pr
oduct_category_name)Order_palced
FROM order1 O
INNER JOIN customers C
ON O.customer_id = C.customer_id

INNER JOIN order_items I
ON O.order_id = I.order_id

INNER JOIN products P
ON I.product_id = P.product_id

WHERE C.customer_state IN('AC','SE') And O.order_status NOT IN('unavailable','canceled')
GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,P.product_category_name
```

ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

SQL Project 1.sql - DESKTOP-Q54KQQR.Project1 (DESKTOP-Q54KQQR\welcome (58)) - Microsoft SQL Server Management Studio

Object Explorer: DESKTOP-Q54KQQR (SQL Server 15.0.2000.1)

Query: ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

Results:

Years	customer_state	product_category_name	Order_paiced
2016	SE	console_games	1
2016	SE	informatica_acessorios	1
2016	SE	moveis_decoracao	1
2017	AC	NULL	0
2017	AC	artigos_de_natal	1
2017	AC	automotivo	4
2017	AC	bebidas	3
2017	AC	beleza_saude	4
2017	AC	brinquedos	2
2017	AC	cama_mesa_banho	4
2017	AC	console_games	1
2017	AC	cost_stuff	1
2017	AC	eletronicos	3
2017	AC	eletronicos	1
2017	AC	esportes_lazer	7
2017	AC	fashion_calçados	1
2017	AC	informatica_acessorios	7

Query executed successfully.

2)post-order reviews

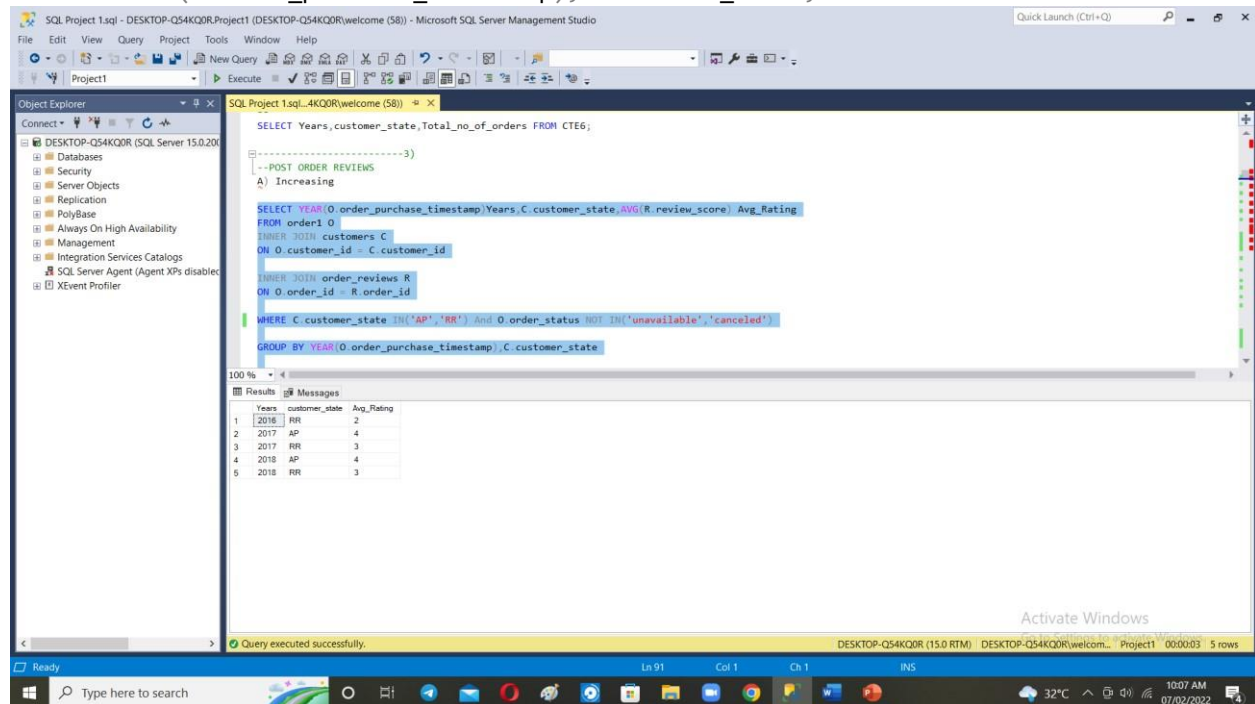
A) Increasing

```
SELECT YEAR(0.order_purchase_timestamp)Years,C.customer_state,AVG(R.review_score)
Avg_Rating
FROM order1 O
INNER JOIN customers C
ON O.customer_id = C.customer_id

INNER JOIN order_reviews R
ON O.order_id = R.order_id

WHERE C.customer_state IN('AP','RR') And O.order_status NOT IN('unavailable','canceled')
GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state
```

ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;



SQL Project 1.sql - DESKTOP-Q54KQOR.Project1 (DESKTOP-Q54KQOR\welcome (58)) - Microsoft SQL Server Management Studio

Object Explorer: DESKTOP-Q54KQOR (SQL Server 15.0.2000.1)

Query: SQL Project 1.sql...4KQOR\welcome (58)

```

SELECT Years,customer_state,Total_no_of_orders FROM CTE6;

--POST ORDER REVIEWS
A) Increasing

SELECT YEAR(0.order_purchase_timestamp)Years,C.customer_state,AVG(R.review_score) Avg_Rating
FROM order1 O
INNER JOIN customers C
ON O.customer_id = C.customer_id
INNER JOIN order_reviews R
ON O.order_id = R.order_id
WHERE C.customer_state IN('AP','RR') And O.order_status NOT IN('unavailable','canceled')
GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state
  
```

Results: 5 rows

Years	customer_state	Avg_Rating
2016	RR	2
2017	AP	4
2017	RR	3
2018	AP	4
2018	RR	3

Query executed successfully.

B) DECREASING

```

SELECT YEAR(0.order_purchase_timestamp)Years,C.customer_state,AVG(R.review_score)
Avg_Rating
FROM order1 O
INNER JOIN customers C
ON O.customer_id = C.customer_id

INNER JOIN order_reviews R
ON O.order_id = R.order_id

WHERE C.customer_state IN('AC','SE') And O.order_status NOT IN('unavailable','canceled')
GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state
  
```

ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

SQL Project 1.sql - DESKTOP-Q54KQOR.Project1 (DESKTOP-Q54KQOR\welcome (58)) - Microsoft SQL Server Management Studio

Object Explorer: DESKTOP-Q54KQOR (SQL Server 15.0.2000.1)

Query: ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

B) DECREASING

```
SELECT YEAR(0.order_purchase_timestamp)Years,C.customer_state,AVG(R.review_score) Avg_Rating
FROM order1 O
INNER JOIN customers C
ON O.customer_id = C.customer_id
INNER JOIN order_reviews R
ON O.order_id = R.order_id
WHERE C.customer_state IN('AC','SE') And O.order_status NOT IN('unavailable','canceled')
GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state
ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;
```

Results:

	Years	customer_state	Avg_Rating
1	2016	SE	4
2	2017	AC	3
3	2017	SE	3
4	2018	AC	4
5	2018	SE	3

Query executed successfully.

3)Seller performance in terms of deliveries,

A) INCREASING

```
SELECT YEAR(0.order_purchase_timestamp)Years , S.seller_id , C.customer_state ,
DATEDIFF(DAY,0.order_delivered_carrier_date,0.order_delivered_customer_date)
Del_days
FROM sellers S
INNER JOIN order_items I
ON S.seller_id = I.seller_id

INNER JOIN order1 O
ON I.order_id = O.order_id

INNER JOIN customers C
ON O.customer_id = C.customer_id

WHERE C.customer_state IN('AP','RR') And O.order_status NOT IN('unavailable','canceled')
```


ORDER BY

`YEAR(0.order_purchase_timestamp) , S.seller_id , C.customer_state ;`

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The query editor displays the following SQL query:

```
--seller performance
A) INCREASING
SELECT YEAR(0.order_purchase_timestamp)Years , S.seller_id , C.customer_state
      DATEDIFF(DAY,0.order_delivered_carrier_date,0.order_delivered_customer_date) Del_days
FROM sellers S
INNER JOIN order_items I
ON S.seller_id = I.seller_id
INNER JOIN order1 O
ON I.order_id = O.order_id
INNER JOIN customers C
ON O.customer_id = C.customer_id
WHERE C.customer_state IN('AP','RR') And 0.order_status NOT IN('unavailable','canceled')
```

The query results are displayed in a table with the following columns: Years, seller_id, customer_state, and Del_days. The results show 17 rows of data, with the first row having a NULL value for Del_days.

Years	seller_id	customer_state	Del_days
2016	1554a68530182680a5f8b042c3ab563	RR	NULL
2016	1554a68530182680a5f8b042c3ab563	RR	NULL
2016	84aba7c0ba51484c30ab7bdc34bcd81	RR	3
2017	1354d5165364534064725ea204e85e	RR	6
2017	1792ce2ab72c3da9d27d9e7a898104e1	RR	169
2017	1da3aeb70d7998f1a6d90a037807623	AP	22
2017	2138cb85b11a4ee1e37afbf1cd8da1f	AP	4
2017	2138cb85b11a4ee1e37afbf1cd8da1f	AP	27
2017	31ae0774e178ab00f070cf8bde009f	AP	13
2017	37be5a7c7911698b098c8ba4119e043	AP	14
2017	4080e5d1529077ba41e38b092ad3	AP	8
2017	4a3ca9315b744ce98e9374361493884	AP	13
2017	4a3ca9315b744ce98e9374361493884	AP	13
2017	4a3ca9315b744ce98e9374361493884	AP	13
2017	4a17c85a51695d023a2ae78b842288	RR	28
2017	4a022999e969a388248c378af1c9395	AP	43
2017	53245955a1a6c20430214f1853a8905	AP	15

B) DECREASING

`SELECT YEAR(0.order_purchase_timestamp)Years , S.seller_id , C.customer_state ,
 DATEDIFF(DAY,0.order_delivered_carrier_date,0.order_delivered_customer_date)`

`Del_days`

`FROM sellers S`

`INNER JOIN order_items I`

`ON S.seller_id = I.seller_id`

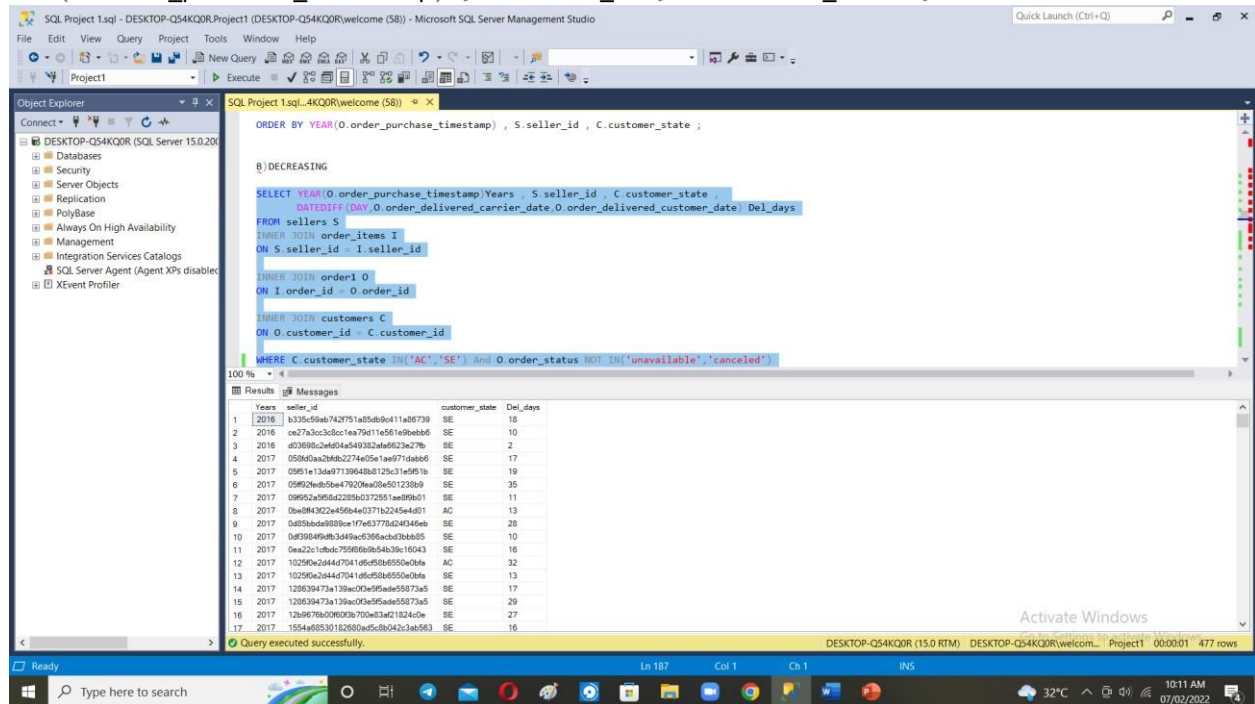
`INNER JOIN order1 O`

`ON I.order_id = O.order_id`

`INNER JOIN customers C`

`ON O.customer_id = C.customer_id`

```
ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;
WHERE C.customer_state IN('AC','SE') And 0.order_status NOT IN('unavailable','canceled')
YEAR(0.order_purchase_timestamp) , S.seller_id , C.customer_state ;
```



4)product-level sales & orders placed,

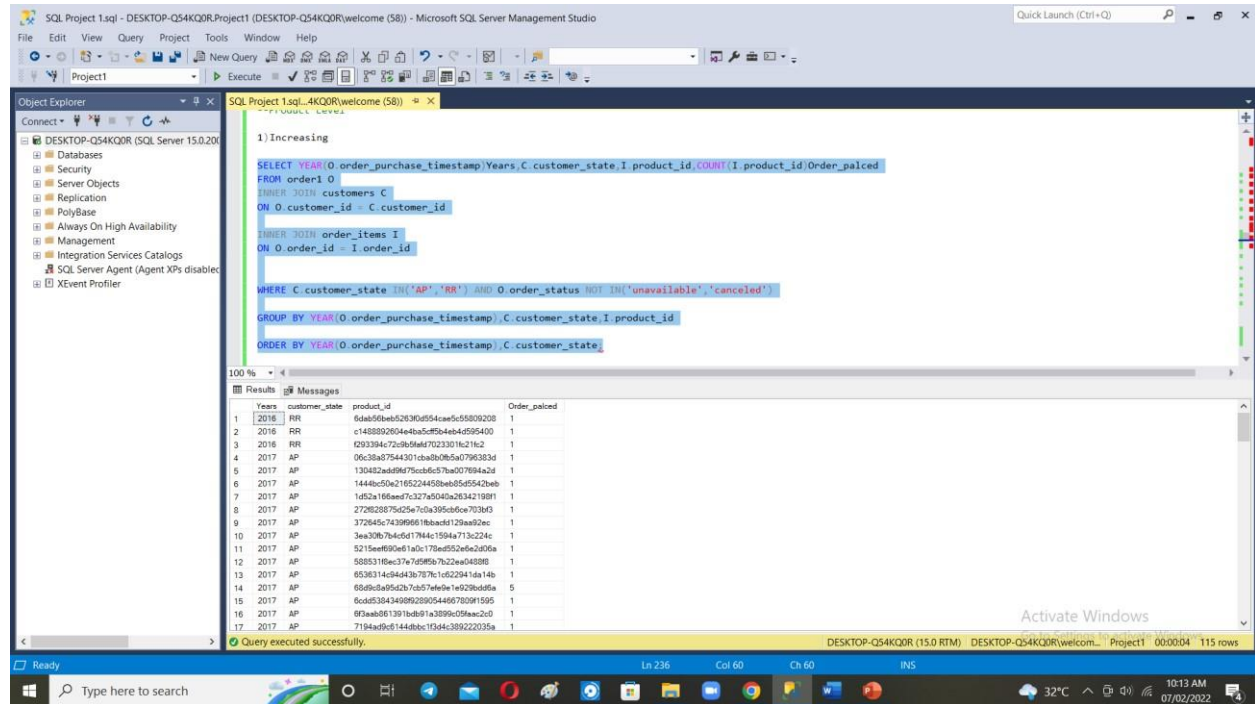
1)Increasing

```
SELECT
YEAR(0.order_purchase_timestamp)Years,C.customer_state,I.product_id,COUNT(I.product_id)Or
der_placed FROM order1 0
INNER JOIN customers C
ON 0.customer_id = C.customer_id

INNER JOIN order_items I
ON 0.order_id = I.order_id
```

```
WHERE C.customer_state IN('AP','RR') AND 0.order_status NOT IN('unavailable','canceled')
GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,I.product_id
```

ORDER BY



1) Increasing

```
SELECT YEAR(O.order_purchase_timestamp)Years,C.customer_state,I.product_id,COUNT(I.product_id)Order_palced
FROM order1 O
INNER JOIN customers C
ON O.customer_id = C.customer_id
INNER JOIN order_items I
ON O.order_id = I.order_id
WHERE C.customer_state IN('AP','RR') AND O.order_status NOT IN('unavailable','canceled')
GROUP BY YEAR(O.order_purchase_timestamp),C.customer_state,I.product_id
ORDER BY YEAR(O.order_purchase_timestamp),C.customer_state
```

Years	customer_state	product_id	Order_palced
2016	RR	6da958ba652830d554aef5c5809202	1
2016	RR	c1488892604e4ba5c5f5b4e4d4595400	1
2016	RR	0293394c72c9b5af7023301c21e2	1
2017	AP	06c38a87544301c8a80b5a0796383d	1
2017	AP	130482ad946750c6c657ba007694a2d	1
2017	AP	1444b0502165224450ba055f5542ba0	1
2017	AP	1d52a165aed7c327a050a26342189f1	1
2017	AP	272828785d25a7c0a395c0b6c7038d3	1
2017	AP	372645c743996618baad8120ea02ec	1
2017	AP	3ea30b764cd61784c1594a713c224c	1
2017	AP	5215ea090a61a1c173aed53af5c3a05a	1
2017	AP	585318ac37a7d9f6a7622a040886	1
2017	AP	6536314c04443b787c1c522941da14b	1
2017	AP	68d9c0a95d2b7cb57efefde1e029b0d8a	5
2017	AP	6cd6384349892890544667809f1595	1
2017	AP	6f2aeb861391bd691a3899c09aac2c0	1
2017	AP	7194ad9c9144dbbc1554c389222035a	1

Query executed successfully.

2) Decreasing

```
SELECT
YEAR(O.order_purchase_timestamp)Years,C.customer_state,I.product_id,COUNT(I.product_id)Or
der_palced FROM order1 O
INNER JOIN customers C
ON O.customer_id = C.customer_id

INNER JOIN order_items I
ON O.order_id = I.order_id

WHERE C.customer_state IN('AC','SE') AND O.order_status NOT IN('unavailable','canceled')
GROUP BY YEAR(O.order_purchase_timestamp),C.customer_state,I.product_id
```

ORDER BY YEAR(O.order_purchase_timestamp),C.customer_state;

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'DESKTOP-Q54KQOR (SQL Server 15.0.2000)' server selected. The right pane shows a query window with the following SQL code:

```

2) Decreasing
SELECT YEAR(O.order_purchase_timestamp)Years,C.customer_state,I.product_id,COUNT(I.product_id)Order_pai
FROM order1 O
INNER JOIN customers C
ON O.customer_id = C.customer_id
INNER JOIN order_items I
ON O.order_id = I.order_id
WHERE C.customer_state IN('AC','SE') AND O.order_status NOT IN('unavailable','canceled')
GROUP BY YEAR(O.order_purchase_timestamp),C.customer_state,I.product_id
ORDER BY YEAR(O.order_purchase_timestamp),C.customer_state

```

Below the query window, the 'Results' pane shows the output of the query. The results are grouped by year and customer state, with columns: Years, customer_state, product_id, and Order_pai.

Years	customer_state	product_id	Order_pai
2016	SE	6a2b3252a7b6a56c39571d4d4748	1
2016	SE	056a03b4d54a5d1d371db7d5534136	1
2016	SE	d32c445d9d3d61c8f015ed14ee44d	1
2017	AC	09dbbe2c420cad4d50aea0439632c	1
2017	AC	0a44da3eaa5a5b7894a9a4ec8e2660	1
2017	AC	1a2c2b19f5c3d93d4d3d3d3d3d3d3d	1
2017	AC	3a8b8c3084a3e388585a1b43e185d81	3
2017	AC	0a41699cb5740d605d3d3d3d3d3d3d	1
2017	AC	108e01163b7588c7924edcb5ad81b63	1
2017	AC	22c546d22ea8cae84f9917de8924c6	1
2017	AC	4a679ed3448d6e1c2273d3d3d3d3d3d	1
2017	AC	1a5f5c524a5b52b5b519b5a5a5a5a5	1
2017	AC	29427de78ba8ee683d8db51ec569b4	1
2017	AC	2f28eac328b9a107418d5900a59ab	1
2017	AC	6de8b8a41c6882049d3e4c2016d1bbb	1
2017	AC	3de8b3ae011b40803a508b23392e15d	1
2017	AC	40e8b425d1a26c2b9cb77363523e95e	1

The status bar at the bottom indicates 'Query executed successfully.' and '399 rows'.

5)% of orders delivered earlier than the expected date,

A) INCREASING/DECREASING

```

SELECT
YEAR(O.order_purchase_timestamp)Years,C.customer_state,COUNT(C.customer_state)Order_Deliv
ered
FROM customers C
INNER JOIN order1 O
ON O.customer_id = C.customer_id

WHERE O.order_status IN('delivered') AND O.order_delivered_customer_date <
O.order_estimated_delivery_date

GROUP BY YEAR(O.order_purchase_timestamp),C.customer_state

```

ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```

ORDER BY YEAR(0.order_purchase_timestamp), S.seller_id, C.customer_state;

---% OF ORDER DELIVERED EARLIER

A) INCREASING/DECREASING

SELECT YEAR(0.order_purchase_timestamp)Years,C.customer_state,COUNT(C.customer_state)Order_Delivered
FROM customers C
INNER JOIN order1 O
ON O.customer_id = C.customer_id
WHERE O.order_status IN('delivered') AND O.order_delivered_customer_date < O.order_estimated_delivery_date
GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state
ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state

```

The Results pane shows the following data:

Years	customer_state	Order_Delivered
2016	AL	1
2016	BA	3
2016	CE	6
2016	DF	6
2016	ES	3
2016	GO	7
2016	MA	3
2016	MG	35
2016	MT	1
2016	PA	4
2016	PE	1
2016	PI	6
2016	PR	1
2016	RR	20
2016	RJ	40
2016	RN	4
2016	RS	1

The status bar at the bottom indicates "Query executed successfully." and "75 rows".

6)% of orders delivered later than the expected date, etc.e

A) INCREASING/DECREASING

```

SELECT
YEAR(0.order_purchase_timestamp)Years,C.customer_state,COUNT(C.customer_state)Order_Delivered
FROM customers C
INNER JOIN order1 O
ON O.customer_id = C.customer_id

WHERE O.order_status IN('delivered') AND O.order_delivered_customer_date >
O.order_estimated_delivery_date

GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state

```

ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

SQL Project 1.sql - DESKTOP-Q54KQ0R (SQL Server 15.0.2000.1) - Microsoft SQL Server Enterprise Manager

Object Explorer: DESKTOP-Q54KQ0R (SQL Server 15.0.2000.1) > Databases > Order1 > Tables > order1

Query: **SELECT YEAR(0.order_purchase_timestamp)Years,C.customer_state,COUNT(C.customer_state)Order_Delivered FROM customers C INNER JOIN order1 O ON O.customer_id = C.customer_id WHERE O.order_status IN('delivered') AND O.order_delivered_customer_date > O.order_estimated_delivery_date GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state**

Results: 54 rows

Years	customer_state	Order_Delivered
2018	AP	3
2017	AC	2
2017	AL	41
2017	AM	1
2017	AP	2
2017	BA	153
2017	CE	55
2017	DF	38
2017	EB	70
2017	GO	48
2017	MA	56
2017	MG	143
2017	MS	9
2017	MT	25
2017	PA	31
2017	PB	30
2017	PE	68

D. Do the above analysis for the top 2 cities which are causing the trend for each of the states identified in point (b)

1)Category level Sales and orders placed,

A)INCREASING

```
WITH CTE1 AS( SELECT
YEAR(0.order_purchase_timestamp)Years,C.customer_state,C.customer_city,P.product_category
_name,COUNT(P.product_category_name)Order_placed
FROM order1 O
INNER JOIN customers C
ON O.customer_id = C.customer_id
```

```
INNER JOIN order_items I
ON O.order_id = I.order_id
```

```
INNER JOIN products P
ON I.product_id = P.product_id
```

```
WHERE C.customer_state IN('AP','RR') And O.order_status NOT IN('unavailable','canceled')
```

```
GROUP BY
YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city,P.product_category_name
```

```
),
```

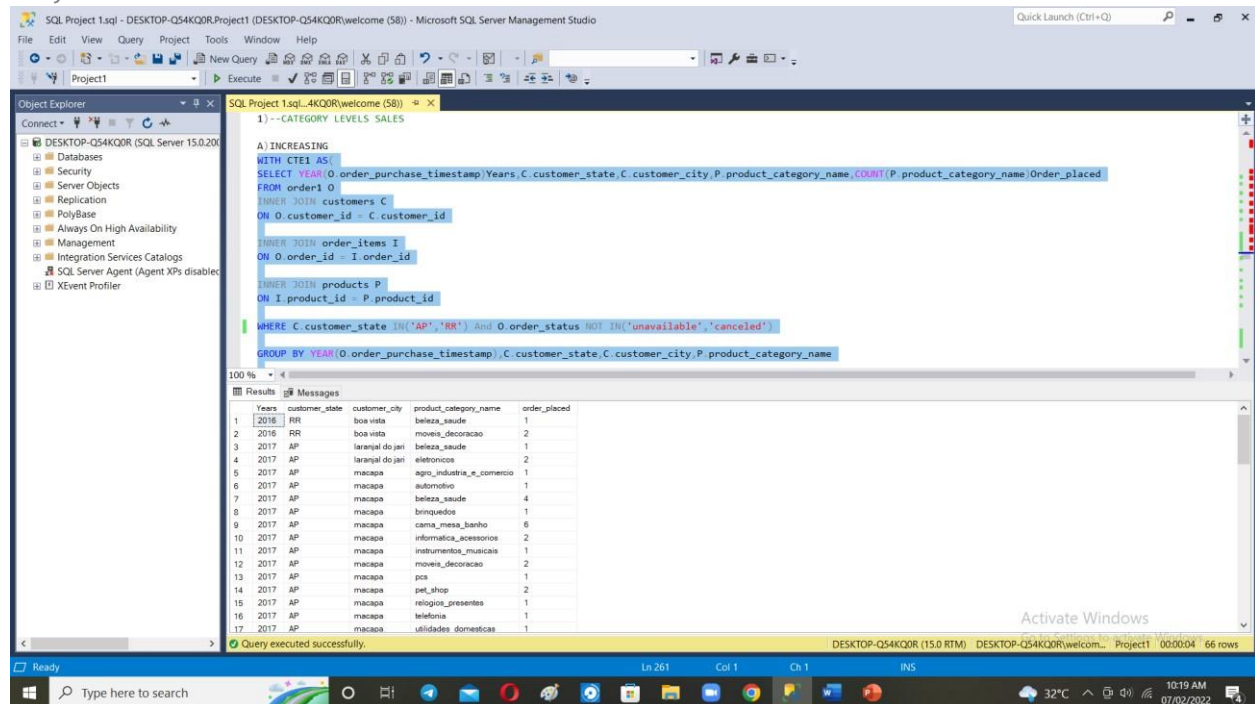
```
--
```

```
CTE2 AS(
SELECT *, DENSE_RANK() OVER(PARTITION BY Years,customer_state ORDER BY Order_placed
DESC)Ranks
```

```
ORDER BY YEAR(O.order_purchase_timestamp),C.customer_state;  
FROM CTE1)
```


--

```
SELECT Years, customer_state, customer_city, product_category_name, order_placed FROM  
CTE2;
```



B) DECREASING

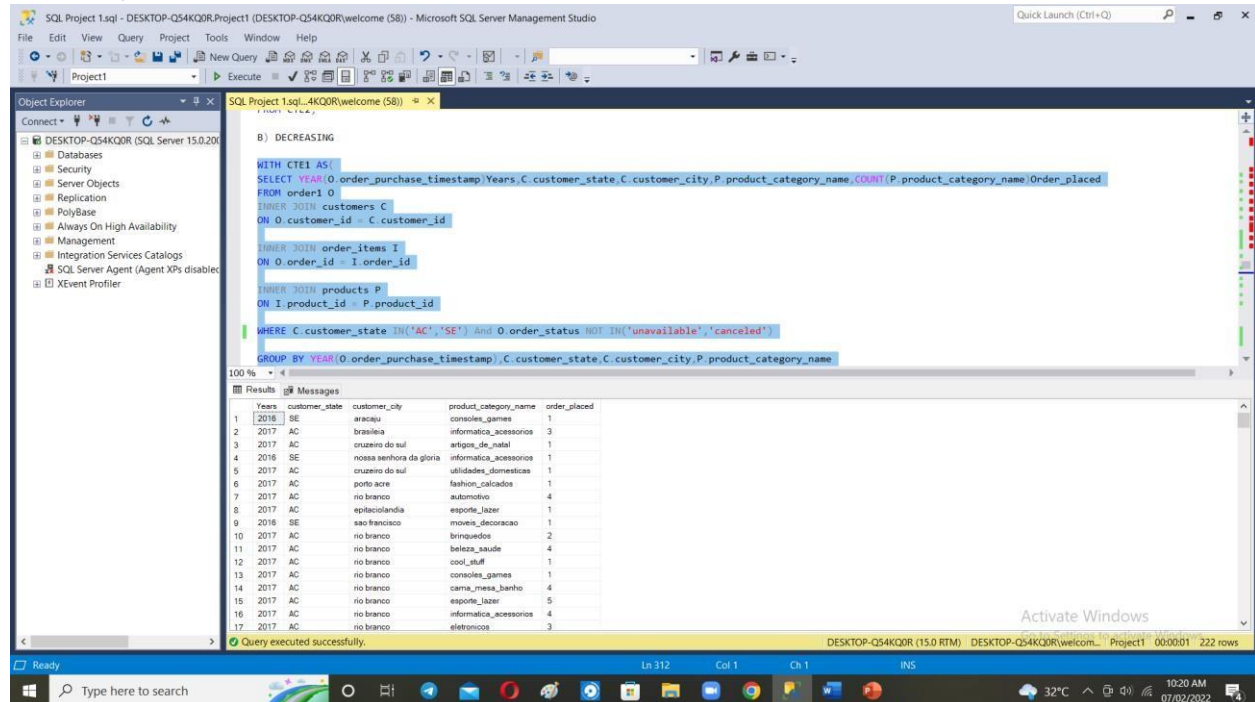
```
WITH CTE1 AS( SELECT  
YEAR(0.order_purchase_timestamp)Years,C.customer_state,C.customer_city,P.product_category  
_name,COUNT(P.product_category_name)Order_placed  
FROM order1 O  
INNER JOIN customers C  
ON O.customer_id = C.customer_id  
  
INNER JOIN order_items I  
ON O.order_id = I.order_id  
  
INNER JOIN products P  
ON I.product_id = P.product_id  
  
WHERE C.customer_state IN('AC','SE') And O.order_status NOT IN('unavailable','canceled')  
  
GROUP BY  
YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city,P.product_category_name  
)
```

--

```
CTE2 AS(  
SELECT *, DENSE_RANK() OVER(PARTITION BY Years, customer_state ORDER BY Order_placed  
DESC)Ranks  
FROM CTE1)
```

--


```
SELECT Years, customer_state, customer_city, product_category_name, order_placed
FROM CTE2;
```



2) post-order reviews,

A) Increasing

```
SELECT
YEAR(0.order_purchase_timestamp)Years,C.customer_state,C.customer_city,AVG(R.review_score
) Avg_Rating
FROM order1 0
INNER JOIN customers C
ON 0.customer_id = C.customer_id

INNER JOIN order_reviews R
ON 0.order_id = R.order_id

WHERE C.customer_state IN('AP','RR') And 0.order_status NOT IN('unavailable','canceled')
GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city
YEAR(0.order_purchase_timestamp),C.customer_state;
```

ORDER BY

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays a SQL query that calculates the average rating for orders, grouped by year, customer state, and customer city. The query is as follows:

```
FROM CTE2;  
2) --POST ORDER REVIEWS  
A) Increasing  
  
SELECT YEAR(O.order_purchase_timestamp) Years, C.customer_state, C.customer_city, AVG(R.review_score) Avg_Rating  
FROM order1 O  
INNER JOIN customers C  
ON O.customer_id = C.customer_id  
  
INNER JOIN order_reviews R  
ON O.order_id = R.order_id  
  
WHERE C.customer_state IN('AP','RR') And O.order_status NOT IN('unavailable','canceled')  
GROUP BY YEAR(O.order_purchase_timestamp), C.customer_state, C.customer_city  
ORDER BY YEAR(O.order_purchase_timestamp), C.customer_state
```

The Results pane shows the following data:

	Years	customer_state	customer_city	Avg_Rating
1	2016	RR	boa vista	2
2	2017	AP	laranjal do jari	5
3	2017	AP	macapa	4
4	2017	AP	oiaoque	5
5	2017	AP	porto grande	5
6	2017	AP	santana	4
7	2017	RR	boa vista	3
8	2018	AP	macapa	4
9	2018	AP	santana	3
10	2018	AP	vitoria do jari	4
11	2018	RR	boa vista	3
12	2018	RR	bonfim	4

The status bar at the bottom indicates "Query executed successfully." and "12 rows".

B) DECREASING

```
SELECT  
YEAR(O.order_purchase_timestamp) Years, C.customer_state, C.customer_city, AVG(R.review_score)  
) Avg_Rating  
FROM order1 O  
INNER JOIN customers C  
ON O.customer_id = C.customer_id  
  
INNER JOIN order_reviews R  
ON O.order_id = R.order_id  
  
WHERE C.customer_state IN('AC', 'SE') And O.order_status NOT IN('unavailable', 'canceled')  
GROUP BY YEAR(O.order_purchase_timestamp), C.customer_state, C.customer_city
```

ORDER BY

YEAR(0.order_purchase_timestamp),C.customer_state;

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
SELECT YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city,AVG(R.review_score) Avg_Rating
FROM order1 O
INNER JOIN customers C
ON O.customer_id = C.customer_id
INNER JOIN order_reviews R
ON O.order_id = R.order_id
WHERE C.customer_state IN('AC','SE') And O.order_status NOT IN('unavailable','canceled')
GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city
ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;
```

The Results pane shows the following data:

Years	customer_state	customer_city	Avg_Rating
2016	SE	aracaju	4
2016	SE	nossa senhora da gloria	5
2016	SE	sao francisco	5
2017	AC	brasilia	5
2017	AC	xapuri	5
2017	AC	porto aore	5
2017	AC	senedor guimard	5
2017	AC	cruzeiro do sul	4
2017	AC	epitaciolandia	1
2017	AC	manoel urbano	5
2017	AC	rio branco	3
2017	SE	aracaju	3
2017	SE	barra dos coqueiros	1
2017	SE	capela	5
2017	SE	aquidaba	5
2017	SE	cedro de sao joao	3
2017	SE	nossa senhora da gloria	3

The status bar at the bottom indicates "Query executed successfully." and "73 rows".

3)Seller performance in terms of deliveries,

A) INCREASING

```
SELECT YEAR(0.order_purchase_timestamp)Years , S.seller_id , C.customer_state
,C.customer_city,
DATEDIFF(DAY,0.order_delivered_carrier_date,0.order_delivered_customer_date)
```

Del_days

FROM sellers S

INNER JOIN order_items I

ON S.seller_id = I.seller_id

INNER JOIN order1 O

ON I.order_id = O.order_id

INNER JOIN customers C

ORDER BY

ON O.customer_id = C.customer_id

WHERE C.customer_state IN('AP', 'RR') And O.order_status NOT IN('unavailable', 'canceled')
YEAR(O.order_purchase_timestamp), C.customer_state , C.customer_city;

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
3) seller performance  
A) INCREASING  
  
SELECT YEAR(O.order_purchase_timestamp) Years , S.seller_id , C.customer_state , C.customer_city,  
DATEDIFF(DAY, O.order_delivered_carrier_date, O.order_delivered_customer_date) Del_days  
FROM sellers S  
INNER JOIN order_items I  
ON S.seller_id = I.seller_id  
INNER JOIN order1 O  
ON I.order_id = O.order_id  
INNER JOIN customers C  
ON O.customer_id = C.customer_id  
WHERE C.customer_state IN('AP', 'RR') And O.order_status NOT IN('unavailable', 'canceled')
```

The Results pane shows the following data:

Years	seller_id	customer_state	customer_city	Del_days
2016	1554a68530182680ad5db042c3ab563	RR	boa vista	NULL
2016	1554a68530182680ad5db042c3ab563	RR	boa vista	NULL
2016	MaBa7d08ca51484c30ab7bdc34bcd1	RR	boa vista	3
2017	d27e33c5d2965138a9967bdc445b6d5	AP	laranjal do jari	16
2017	d859f1ab788714a40e7954c3be6d745	AP	laranjal do jari	20
2017	d859f1ab788714a40e7954c3be6d745	AP	laranjal do jari	20
2017	d87d6d04e1b6c2d614352b383efc2d36	AP	macapa	14
2017	d87d6d04e1b6c2d614352b383efc2d36	AP	macapa	14
2017	910eac23d4a81d7d7cacefa30ad1ca	AP	macapa	41
2017	9d44643a8dbd6434e5962109a92273	AP	macapa	25
2017	a4186a04da11724393259414a6d95e	AP	macapa	21
2017	da8622b14ab17ae28314ac5b9dab84a	AP	macapa	21
2017	da8622b14ab17ae28314ac5b9dab84a	AP	macapa	21
2017	cca3071e3e9bb7612640c09ba2301306	AP	macapa	182
2017	cca3071e3e9bb7612640c09ba2301306	AP	macapa	182
2017	cca4bba9f2afae2b7066a4130f14e3	AP	macapa	13
2017	14d33ae704798b11e6d9b0e8787c23	AP	macapa	22

The status bar at the bottom indicates "Query executed successfully." and "133 rows".

B) DECREASING

```
SELECT YEAR(O.order_purchase_timestamp) Years , S.seller_id , C.customer_state ,  
C.customer_city,  
DATEDIFF(DAY, O.order_delivered_carrier_date, O.order_delivered_customer_date)  
Del_days  
FROM sellers S  
INNER JOIN order_items I  
ON S.seller_id = I.seller_id  
  
INNER JOIN order1 O
```

ORDER BY

ON I.order_id = O.order_id

INNER JOIN customers C

ON O.customer_id = C.customer_id

WHERE C.customer_state IN('AC','SE') And O.order_status NOT IN('unavailable','canceled')
YEAR(0.order_purchase_timestamp),C.customer_state , C.customer_city;

The screenshot shows the SQL Server Management Studio interface. The query editor contains the following SQL code:

```
B) DECREASING
SELECT YEAR(0.order_purchase_timestamp)Years , S.seller_id , C.customer_state , C.customer_city,
DATEDIFF(DAY,0.order_delivered_carrier_date,0.order_delivered_customer_date) Del_days
FROM sellers S
INNER JOIN order_items I
ON S.seller_id = I.seller_id
INNER JOIN order1 O
ON I.order_id = O.order_id
INNER JOIN customers C
ON O.customer_id = C.customer_id
WHERE C.customer_state IN('AC','SE') And O.order_status NOT IN('unavailable','canceled')
ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state , C.customer_city;
```

The Results pane shows the following data:

Years	seller_id	customer_state	customer_city	Del_days
2016	ea27a3bc3d8c1ea79d411e581e9beb6d	SE	aracaju	10
2016	d03696c2e6d4a545382af6623e27b	SE	nossa senhora da gloria	2
2016	b335c59ab742f751e05db8c411a86739	SE	sao francisco	18
2017	897060da8ba21855304d50a935913	AC	brasileia	26
2017	897060da8ba21855304d50a935913	AC	brasileia	26
2017	897060da8ba21855304d50a935913	AC	brasileia	26
2017	483919c944b6fc8f12086c8a899aach	AC	cruzeiro do sul	31
2017	0ba084302a456b4e0371b2245e4d01	AC	cruzeiro do sul	13
2017	274978279e0e0d033addec1474776d7	AC	epitaciolandia	12
2017	788d8864b5bc1e436ca8e151f8ec77d	AC	manoel urbano	11
2017	714254068d91a22237e67f811e4eb2	AC	porto aze	24
2017	1a6a303571468144bae4a93bc1e0441	AC	rio branco	8
2017	2138c3b85b11a4ec1e37afbd1c8eda1f	AC	rio branco	21
2017	a9ee440659948b7849d83a62734150b	AC	rio branco	10
2017	5dbeca129747e928ef7a9974c4f8ca	AC	rio branco	23
2017	36a4898d44895394e4e8d7572683988	AC	rio branco	21
2017	37ba5a7c751198b6c60cb4119e043	AC	rio branco	13

The status bar at the bottom indicates "Query executed successfully." and "477 rows".

4)product-level sales & orders placed,
1)Increasing

SELECT

YEAR(0.order_purchase_timestamp)Years,C.customer_state,C.customer_city,I.product_id,COUNT
(I.product_id)Order_palced

FROM order1 O

INNER JOIN customers C

```
ORDER BY
ON O.customer_id = C.customer_id

INNER JOIN order_items I
ON O.order_id = I.order_id

WHERE C.customer_state IN('AP', 'RR') AND O.order_status NOT IN('unavailable', 'canceled')
GROUP BY YEAR(O.order_purchase_timestamp), C.customer_state, C.customer_city, I.product_id
```

ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The 'Query' window displays a SQL query that has been executed successfully. The 'Results' window shows the output of the query, which is a table with 5 columns: Years, customer_state, customer_city, product_id, and Order_palced. The table contains 17 rows of data, showing orders from 2016 and 2017 for various customer states and cities. The status bar at the bottom indicates that the query was executed successfully and returned 116 rows.

```

6)Product Level
1)Increasing

SELECT YEAR(0.order_purchase_timestamp)Years,C.customer_state,C.customer_city,I.product_id,COUNT(I.product_id)Order_palced
FROM order1 O
INNER JOIN customers C
ON O.customer_id = C.customer_id

INNER JOIN order_items I
ON O.order_id = I.order_id

WHERE C.customer_state IN('AP','RR') AND O.order_status NOT IN('unavailable','canceled')

GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city,I.product_id

```

Years	customer_state	customer_city	product_id	Order_palced
2016	RR	boa vista	6da556ae52c36d5f54ae5c55020208	1
2016	RR	boa vista	c1480892604e4ba5d95b4eb4d595400	1
2016	RR	boa vista	d93394c72c9b9fdd702330162182	1
2017	AP	laranja do jari	05aab861391bb91a3899c059aac2c0	1
2017	AP	laranja do jari	9750263dae39930ce039eeb21c12029	2
2017	AP	macapa	130402a6d9d75cc0b5c57ba007094a2a	1
2017	AP	macapa	1d52a186aed7c327a004a2a26342180f	1
2017	AP	macapa	372645c74399661fbbaed129aa02ec	1
2017	AP	macapa	3ea30b7b4c6d17844c1594a713c224c	1
2017	AP	macapa	5215ee90ae1a0c178ae552ae2a006a	1
2017	AP	macapa	58853186a27e7d998b7622a04d88f	1
2017	AP	macapa	6ed4f38d3416952095a4667809f1595	1
2017	AP	macapa	7194af9cd144d8bc1f584c389222035a	1
2017	AP	macapa	72b88a0c0910a3d1db068070c1515029	1
2017	AP	macapa	750a49a336ad13c0d84a76130949302	3
2017	AP	macapa	780b6890ae544337ae0837649b0606ad	1
2017	AP	macapa	a62a25a09e05e9af71a90d0ea1aa3f1	1

2) Decreasing

```

SELECT
YEAR(0.order_purchase_timestamp)Years,C.customer_state,C.customer_city,I.product_id,COUNT
(I.product_id)Order_palced
FROM order1 O
INNER JOIN customers C
ON O.customer_id = C.customer_id

INNER JOIN order_items I
ON O.order_id = I.order_id

WHERE C.customer_state IN('AC','SE') AND O.order_status NOT IN('unavailable','canceled')
GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city,I.product_id

```


ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

SQL Project 1.sql - DESKTOP-Q54KQOR (SQL Server 15.0.2000.1) - Microsoft SQL Server Enterprise Manager

Object Explorer: DESKTOP-Q54KQOR (SQL Server 15.0.2000.1) > Databases > Project1 > Tables > customers

Query: 2) Decreasing

```
SELECT YEAR(0.order_purchase_timestamp)Years,C.customer_state,C.customer_city,I.product_id,COUNT(I.product_id)Order_paired
FROM order1 0
INNER JOIN customers C
ON 0.customer_id = C.customer_id
INNER JOIN order_items I
ON 0.order_id = I.order_id
WHERE C.customer_state IN('AC','SE') AND 0.order_status NOT IN('unavailable','canceled')
GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city,I.product_id
ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;
```

Results: 415 rows

Years	customer_state	customer_city	product_id	Order_paired
2016	SE	aracaju	8a232323a2baf65b4395714d04748	1
2016	SE	nostra senhora da gloria	05fa03b4d54a5d14371db7d5534136	1
2016	SE	sao francisco	d32c445d9d38f1c8f515ed14ee44cd	1
2017	AC	porto acre	fe6614ba400caca2d8d310002d5ecd5	1
2017	AC	brasilia	3a8b6cf3084a3e303550a1b43e118d81	3
2017	AC	crusneiro do sul	a9188544b2c2e4ae0229b8aceca825	1
2017	AC	crusneiro do sul	40e8b425a1a25a2d5b77363523a05e	1
2017	AC	rio branco	0a4dea8eaaaa5b7894a9a4ec5e960	1
2017	AC	epitaciolandia	bba1af8c7467ef18e2b2878335ae55	1
2017	AC	rio branco	09d8ba2c4028cad4f90a0a04396532c	1
2017	AC	rio branco	44679e03948d9e1a2277b889927ac7	1
2017	AC	rio branco	109a01163b7958b7924e0d5b5a0d1a3	1
2017	AC	manuel urbano	1a2d3b19f5c3d9344dbcc3cd963a9f	1
2017	AC	rio branco	0a41699cb57940d05d3d4d4d461673b	1
2017	AC	rio branco	63085b4366de027bcb63abb59e4103a	1
2017	AC	rio branco	7a08c924ba052beffa18b55a9a4d5e	1
2017	AC	rio branco	294275e78ba9ee98349bdc51ee599b4	1

5)% of orders delivered earlier than the expected date,
A) INCREASING

```
SELECT
YEAR(0.order_purchase_timestamp)Years,C.customer_state,C.customer_city,COUNT(C.customer_s
tate)Order_Delivered FROM customers C
INNER JOIN order1 0
ON 0.customer_id = C.customer_id

WHERE C.customer_state IN('AP','RR') AND 0.order_status IN('delivered') AND
0.order_delivered_customer_date < 0.order_estimated_delivery_date

GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city
```


ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state , C.customer_city;

4)% OF ORDER DELIVERED EARLIER

A) INCREASING

SELECT YEAR(0.order_purchase_timestamp)Years,C.customer_state,C.customer_city,COUNT(C.customer_state)Order_Delivered
FROM customers C
INNER JOIN order1 O
ON O.customer_id = C.customer_id
WHERE C.customer_state IN('AP','RR') AND O.order_status IN('delivered') AND O.order_delivered_customer_date < O.order_estimated_delivery_date
GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city
ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state
```

The Results pane shows the following data:

Years	customer_state	customer_city	Order_Delivered
2016	RR	boa vista	1
2017	AP	laranjal do jari	2
2017	AP	macapa	17
2017	AP	porto grande	1
2017	AP	santana	5
2017	RR	boa vista	15
2018	AP	macapa	35
2018	AP	santana	3
2018	AP	vitoria do jari	1
2018	RR	boa vista	19
2018	RR	bonfim	1

The status bar at the bottom indicates "Query executed successfully." and "DESKTOP-Q54KQOR (15.0 RTM) DESKTOP-Q54KQOR(welcom... Project1 00:00:03 11 rows".

B) DECREASING

SELECT

YEAR(0.order_purchase_timestamp)Years,C.customer_state,C.customer_city,COUNT(C.customer_state)Order_Delivered FROM customers C
INNER JOIN order1 O
ON O.customer_id = C.customer_id

WHERE C.customer_state IN('AC','SE') AND O.order_status IN('delivered') AND
O.order_delivered_customer_date < O.order_estimated_delivery_date

GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city

ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

SQL Project 1sql_4KQOR(welcome (58)) - Microsoft SQL Server Management Studio

WHERE C.customer_state IN('AP','RR') AND O.order_status IN('delivered') AND O.order_delivered_customer_date < O.order_estimated_delivery_date

GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city

ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

B) DECREASING

SELECT YEAR(0.order_purchase_timestamp)Years,C.customer_state,C.customer_city,COUNT(C.customer_state)Order_Delivered

FROM customers C

INNER JOIN order1 O

ON O.customer_id = C.customer_id

WHERE C.customer_state IN('AC','SE') AND O.order_status IN('delivered') AND O.order_delivered_customer_date < O.order_estimated_delivery_date

GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city

ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state

Years	customer_state	customer_city	Order_Delivered	
1	2016	SE	aracaju	1
2	2016	SE	nossa senhora da gloria	1
3	2016	SE	sao francisco	1
4	2017	AC	brasileia	1
5	2017	AC	cruzeiro do sul	2
6	2017	AC	epitaciolandia	1
7	2017	AC	manuel urbano	1
8	2017	AC	porto acre	1
9	2017	AC	rio branco	42
10	2017	AC	senador guimard	2
11	2017	AC	valemi	1
12	2017	SE	aquidaba	1
13	2017	SE	aracaju	104
14	2017	SE	campo do brito	1
15	2017	SE	capela	2
16	2017	SE	cedro de sao joao	1
17	2017	SE	estancia	6

Query executed successfully.

6)% of orders delivered later than the expected date, etc.e

A) INCREASING

SELECT

YEAR(0.order_purchase_timestamp)Years,C.customer_state,C.customer_city,**COUNT**(C.customer_state)Order_Delivered **FROM** customers C

INNER JOIN order1 O

ON O.customer_id = C.customer_id

WHERE C.customer_state IN('AP','RR') AND O.order_status IN('delivered') AND O.order_delivered_customer_date > O.order_estimated_delivery_date

GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city

ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays a query with the following structure:

```
ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

5) % OF ORDER DELIVERED LATER

A) INCREASING

SELECT YEAR(0.order_purchase_timestamp)Years,C.customer_state,C.customer_city,COUNT(C.customer_state)Order_Delivered
FROM customers C
INNER JOIN order1 O
ON O.customer_id = C.customer_id
WHERE C.customer_state IN('AP','RR') AND O.order_status IN('delivered') AND O.order_delivered_customer_date > O.order_estimated_delivery_date
GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city
ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

B) DECREASING
```

The Results pane shows the following data:

	Years	customer_state	customer_city	Order_Delivered
1	2017	AP	matapa	1
2	2017	AP	santana	1
3	2017	RR	boa vista	3
4	2018	RR	boa vista	2

The status bar at the bottom indicates "Query executed successfully." and "DESKTOP-Q54KQOR (15.0 RTM) | DESKTOP-Q54KQOR\welcom... | Project1 | 00:00:03 | 4 rows".

B)DECREASING

SELECT

YEAR(0.order_purchase_timestamp)Years,C.customer_state,C.customer_city,COUNT(C.customer_state)Order_Delivered FROM customers C
INNER JOIN order1 O
ON O.customer_id = C.customer_id

WHERE C.customer_state IN('AC','SE') AND O.order_status IN('delivered') AND
O.order_delivered_customer_date > O.order_estimated_delivery_date

GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city

ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

The screenshot displays the Microsoft SQL Server Management Studio (SSMS) interface. The main window shows a SQL query being executed. The query filters for orders with status 'delivered' and groups them by year, customer state, and customer city, ordered by year. The results pane shows 21 rows of data.

Query:

```
WHERE C.customer_state IN('AP','RR') AND O.order_status IN('delivered') AND O.order_delivered_customer_date > O.order_estimated_delivery_date

GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city

ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;

B) DECREASING

SELECT YEAR(0.order_purchase_timestamp)Years,C.customer_state,C.customer_city,COUNT(C.customer_state)Order_Delivered
FROM customers C
INNER JOIN orders O ON O.customer_id = C.customer_id
WHERE C.customer_state IN('AC','SE') AND O.order_status IN('delivered') AND O.order_delivered_customer_date > O.order_estimated_delivery_date
GROUP BY YEAR(0.order_purchase_timestamp),C.customer_state,C.customer_city
ORDER BY YEAR(0.order_purchase_timestamp),C.customer_state;
```

Results:

Years	customer_state	customer_city	Order_Delivered
2017	AC	rio branco	2
2017	SE	aracaju	13
2017	SE	barra dos coqueiros	1
2017	SE	cedro de sao joao	1
2017	SE	estancia	1
2017	SE	indaiaba	1
2017	SE	laporanga d'ajuda	1
2017	SE	lagarto	2
2017	SE	macambira	1
2017	SE	nossa senhora do socorro	4
2017	SE	pinhao	1
2017	SE	umbaua	1
2018	AC	rio branco	1
2018	SE	aracaju	16
2018	SE	barra dos coqueiros	1
2018	SE	capela	1
2018	SE	carmopolis	1

Query executed successfully.