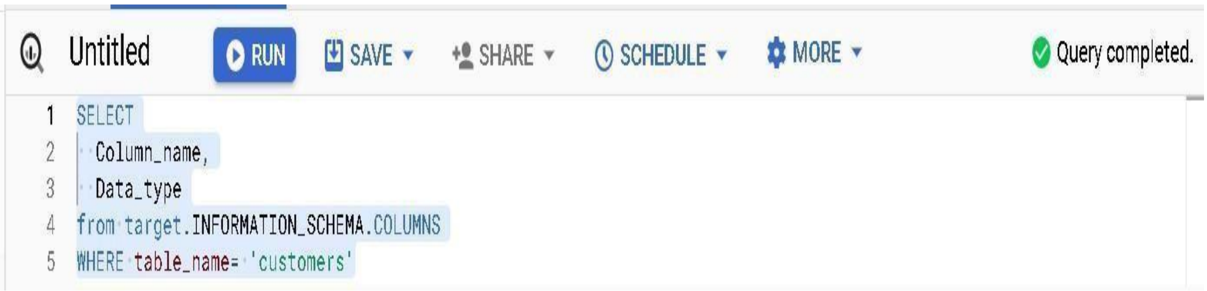
Business Case: E-Commerce

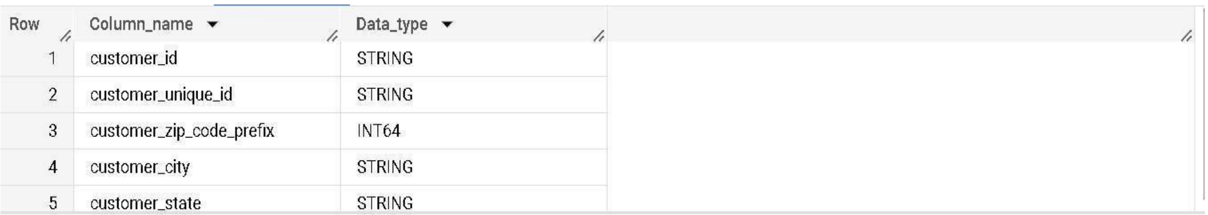
- Data type of all columns in the "customers" table.



Query completed.

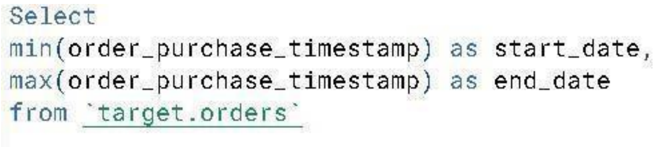
```
SELECT
  Column_name,
  Data_type
from target.INFORMATION_SCHEMA.COLUMNS
WHERE table_name= 'customers'
```

Output:



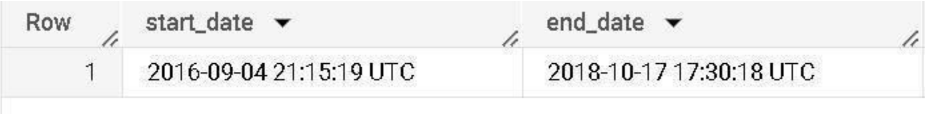
Row	Column_name	Data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

- Get the time range between which the orders were placed.



```
Select
min(order_purchase_timestamp) as start_date,
max(order_purchase_timestamp) as end_date
from `target.orders`
```

Output:

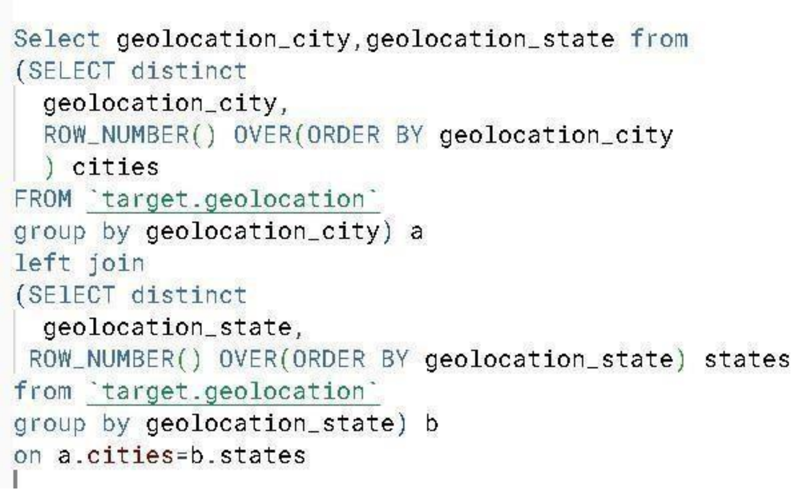


Row	start_date	end_date
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Insight:

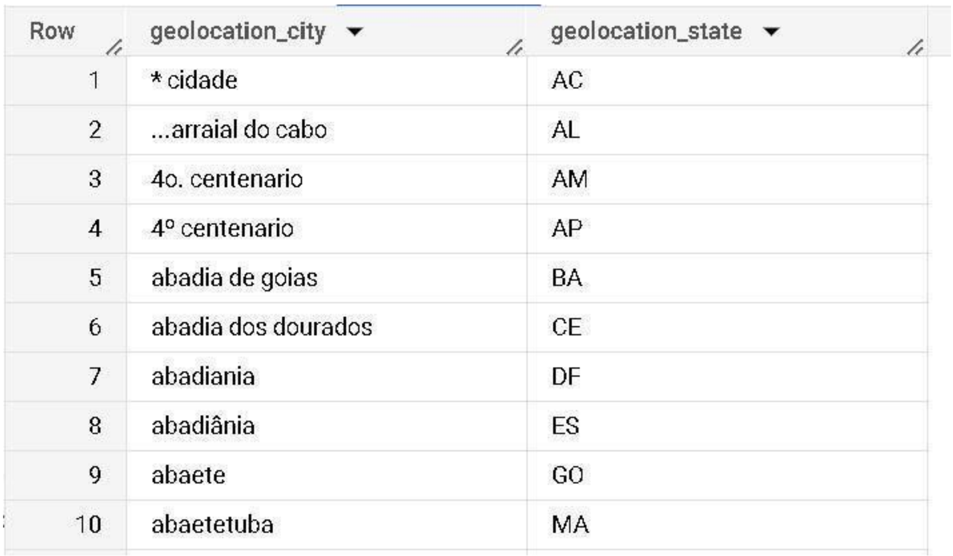
The first order date is 2016-09-04 and the last date in which order placed is 2018-10-17 in the given dataset.

- Count the number of Cities and States in the dataset.



```
Select geolocation_city,geolocation_state from
(SELECT distinct
  geolocation_city,
  ROW_NUMBER() OVER(ORDER BY geolocation_city
  ) cities
FROM `target.geolocation`
group by geolocation_city) a
left join
(SElECT distinct
  geolocation_state,
 ROW_NUMBER() OVER(ORDER BY geolocation_state) states
from `target.geolocation`
group by geolocation_state) b
on a.cities=b.states
```

Output:

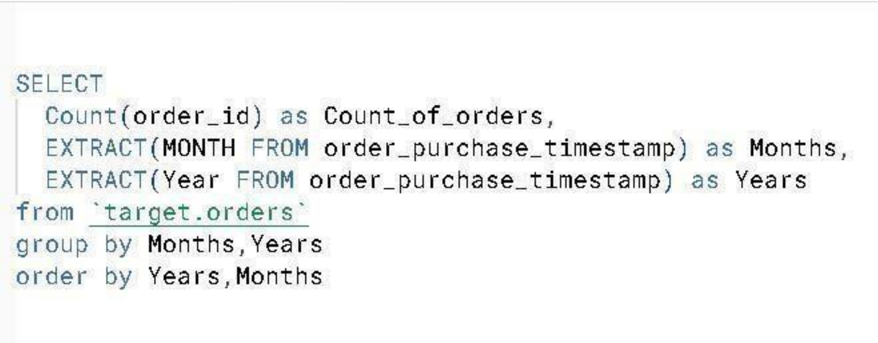


Row	geolocation_city	geolocation_state
1	* cidade	AC
2	...arraial do cabo	AL
3	4o. centenario	AM
4	4º centenario	AP
5	abadia de goias	BA
6	abadia dos dourados	CE
7	abadiania	DF
8	abadiânia	ES
9	abaete	GO
10	abaetetuba	MA

Insight:

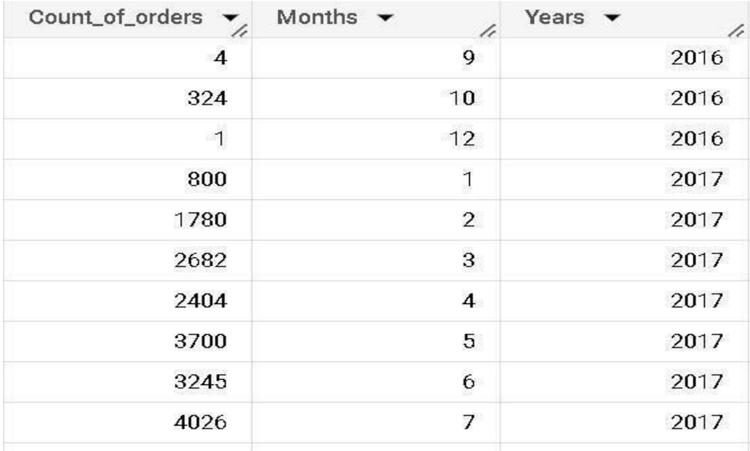
There are total of 27 states and 50 cities in our dataset.

- Is there a growing trend in the no. of orders placed over the past years?



```
SELECT
  Count(order_id) as Count_of_orders,
  EXTRACT(MONTH FROM order_purchase_timestamp) as Months,
  EXTRACT(Year FROM order_purchase_timestamp) as Years
from `target.orders`
group by Months,Years
order by Years,Months
```

Output:

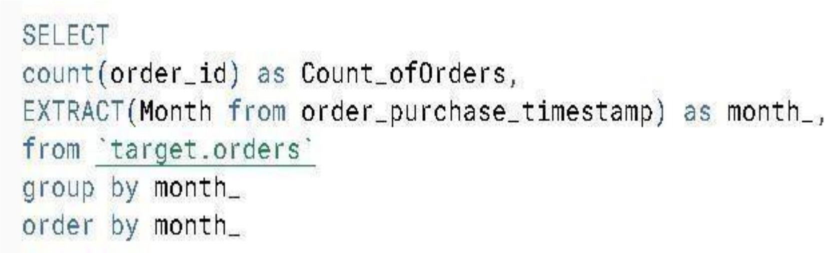


Count_of_orders	Months	Years
4	9	2016
324	10	2016
1	12	2016
800	1	2017
1780	2	2017
2682	3	2017
2404	4	2017
3700	5	2017
3245	6	2017
4026	7	2017

Insight:

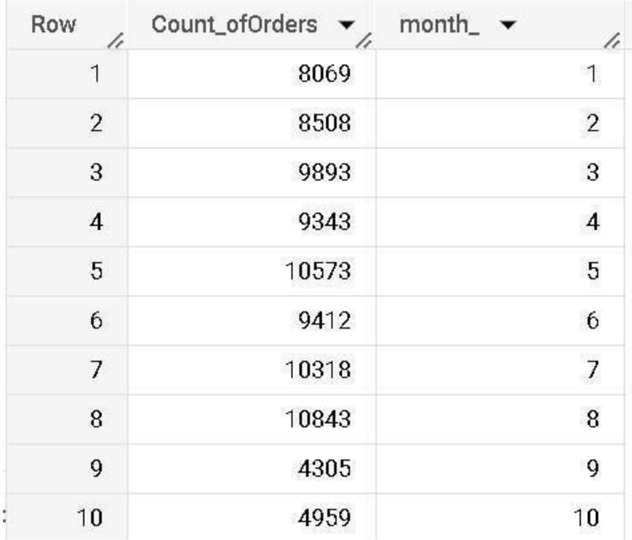
There is increase in the no. of orders month by month through the years in 2016,2017 and 2018 except the 9th and 10th month of 2018 where orders were really low.

- Can we see some kind of monthly seasonality in terms of the no. of orders being placed?



```
SELECT
count(order_id) as Count_ofOrders,
EXTRACT(Month from order_purchase_timestamp) as month_,
from `target.orders`
group by month_
order by month_
```

Output:



Row	Count_ofOrders	month_
1	8069	1
2	8508	2
3	9893	3
4	9343	4
5	10573	5
6	9412	6
7	10318	7
8	10843	8
9	4305	9
10	4959	10

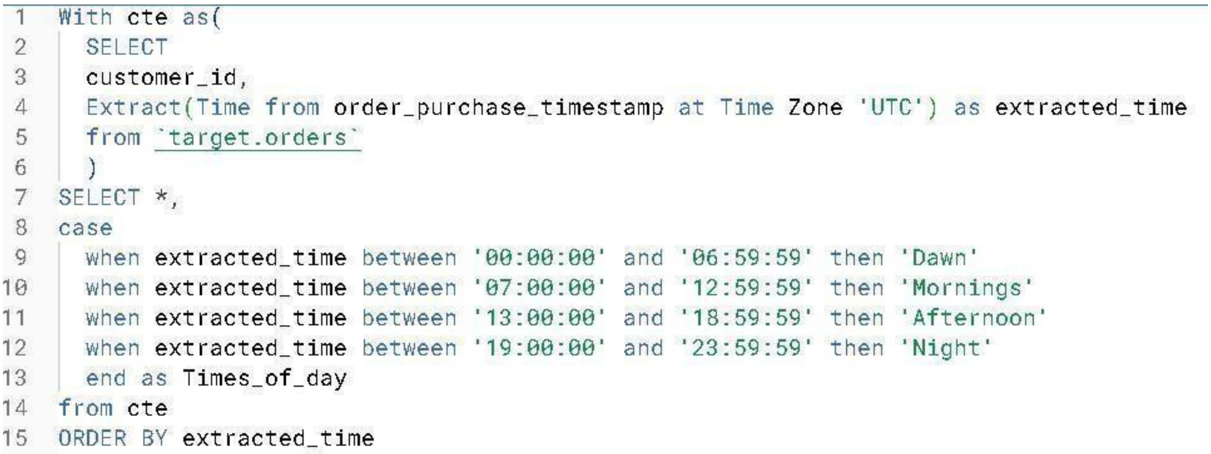
Insight:

In the month of August, the no. of orders are highest in the year it is followed by May and July.

The orders are lowest in the month of September and October.

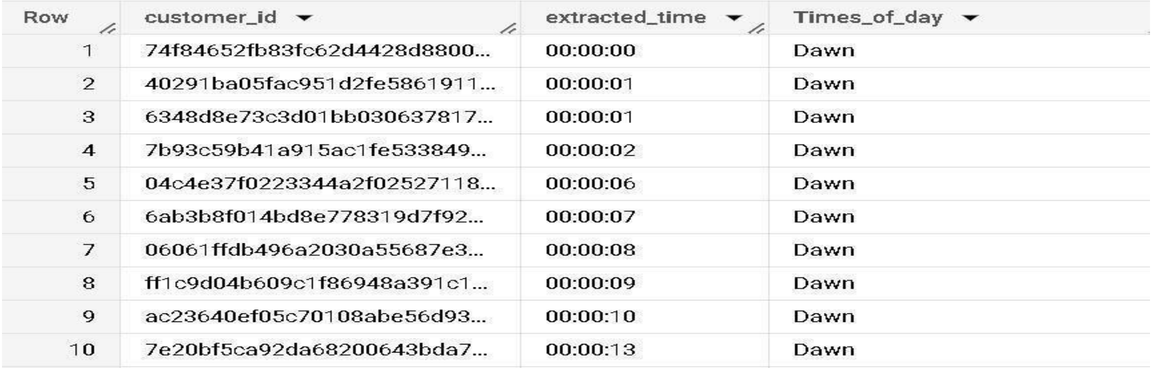
- During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

0-6 hrs: Dawn ,7-12 hrs: Mornings, 13-18 hrs: Afternoon, 19-23 hrs: Night



```sql
With cte as(
  SELECT
  customer_id,
  Extract(Time from order_purchase_timestamp at Time Zone 'UTC') as extracted_time
  from `target.orders`
  )
SELECT *,
case
  when extracted_time between '00:00:00' and '06:59:59' then 'Dawn'
  when extracted_time between '07:00:00' and '12:59:59' then 'Mornings'
  when extracted_time between '13:00:00' and '18:59:59' then 'Afternoon'
  when extracted_time between '19:00:00' and '23:59:59' then 'Night'
  end as Times_of_day
from cte
ORDER BY extracted_time
```

Output:



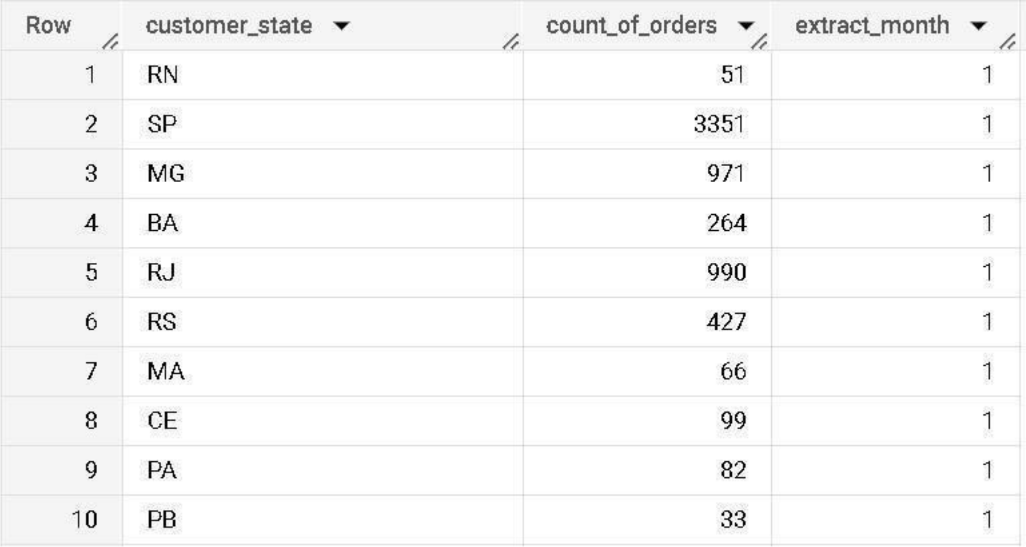
Row	customer_id	extracted_time	Times_of_day
1	74f84652fb83fc62d4428d8800...	00:00:00	Dawn
2	40291ba05fac951d2fe5861911...	00:00:01	Dawn
3	6348d8e73c3d01bb030637817...	00:00:01	Dawn
4	7b93c59b41a915ac1fe533849...	00:00:02	Dawn
5	04c4e37f0223344a2f02527118...	00:00:06	Dawn
6	6ab3b8f014bd8e778319d7f92...	00:00:07	Dawn
7	06061ffdb496a2030a55687e3...	00:00:08	Dawn
8	ff1c9d04b609c1f86948a391c1...	00:00:09	Dawn
9	ac23640ef05c70108abe56d93...	00:00:10	Dawn
10	7e20bf5ca92da68200643bda7...	00:00:13	Dawn

Insight:

38135 Brazilian customers placed their orders in afternoon followed by night 28831, morning 27733 and dawn 5242.

- Get the month-on-month no. of orders placed in each state.

Output:

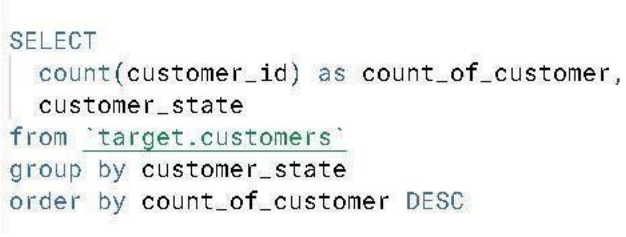


Row	customer_state	count_of_orders	extract_month
1	RN	51	1
2	SP	3351	1
3	MG	971	1
4	BA	264	1
5	RJ	990	1
6	RS	427	1
7	MA	66	1
8	CE	99	1
9	PA	82	1
10	PB	33	1

Insight:

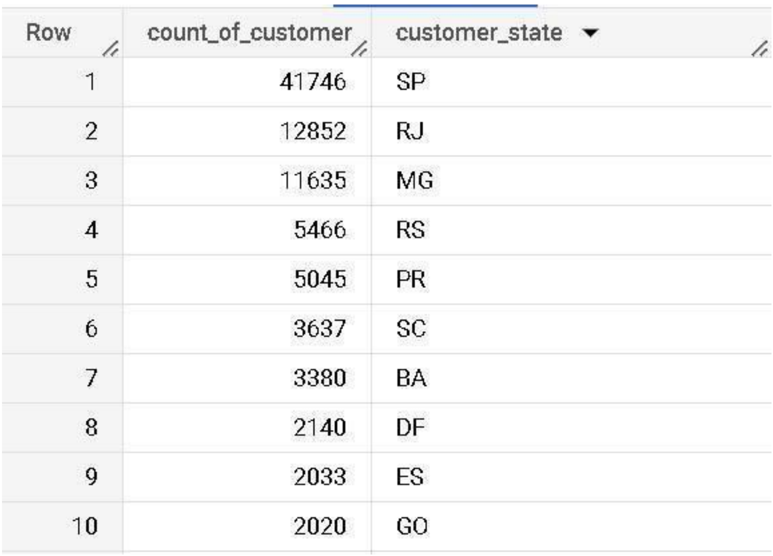
Sales are fluctuating in every month in some states no. of orders are in single digits in certainmonths. Highest sales were recorded in the month of August which were 4982.

- How are the customers distributed across all the states?



```
SELECT
  count(customer_id) as count_of_customer,
  customer_state
from `target.customers`
group by customer_state
order by count_of_customer DESC
```

Output:

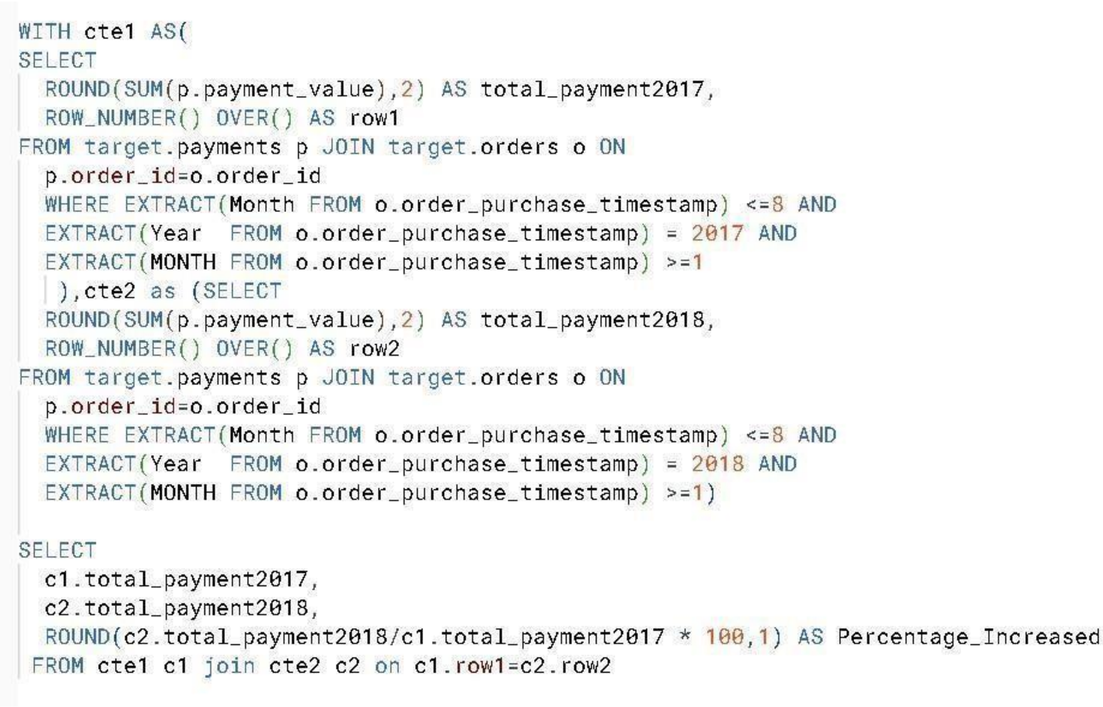


Row	count_of_customer	customer_state
1	41746	SP
2	12852	RJ
3	11635	MG
4	5466	RS
5	5045	PR
6	3637	SC
7	3380	BA
8	2140	DF
9	2033	ES
10	2020	GO

Insight:

The state with highest count of unique customers is SP with 41746 and the state which has lowest count of unique customer is RR with 46 customers.

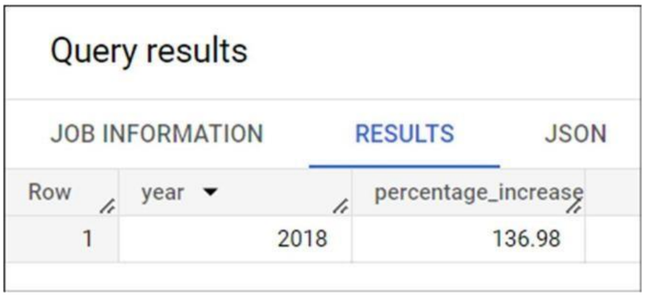
- Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).



```
WITH cte1 AS(
SELECT
  ROUND(SUM(p.payment_value),2) AS total_payment2017,
  ROW_NUMBER() OVER() AS row1
FROM target.payments p JOIN target.orders o ON
  p.order_id=o.order_id
  WHERE EXTRACT(Month FROM o.order_purchase_timestamp) <=8 AND
  EXTRACT(Year  FROM o.order_purchase_timestamp) = 2017 AND
  EXTRACT(MONTH FROM o.order_purchase_timestamp) >=1
   ),cte2 as (SELECT
  ROUND(SUM(p.payment_value),2) AS total_payment2018,
  ROW_NUMBER() OVER() AS row2
FROM target.payments p JOIN target.orders o ON
  p.order_id=o.order_id
  WHERE EXTRACT(Month FROM o.order_purchase_timestamp) <=8 AND
  EXTRACT(Year  FROM o.order_purchase_timestamp) = 2018 AND
  EXTRACT(MONTH FROM o.order_purchase_timestamp) >=1)

SELECT
  c1.total_payment2017,
  c2.total_payment2018,
  ROUND(c2.total_payment2018/c1.total_payment2017 * 100,1) AS Percentage_Increased
 FROM cte1 c1 join cte2 c2 on c1.row1=c2.row2
```

Output:

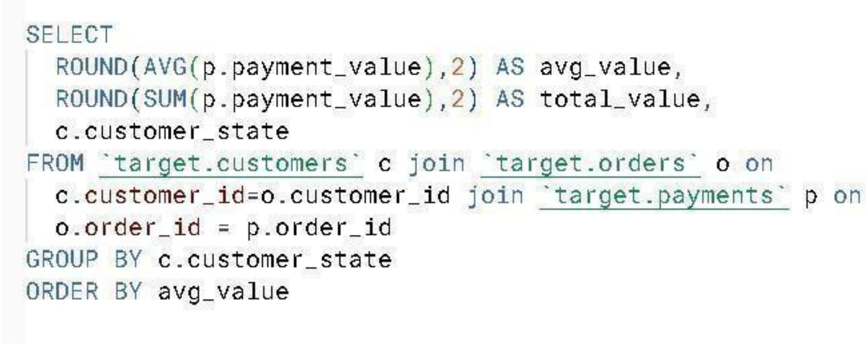


Row	year	percentage_increase
1	2018	136.98

Insight:

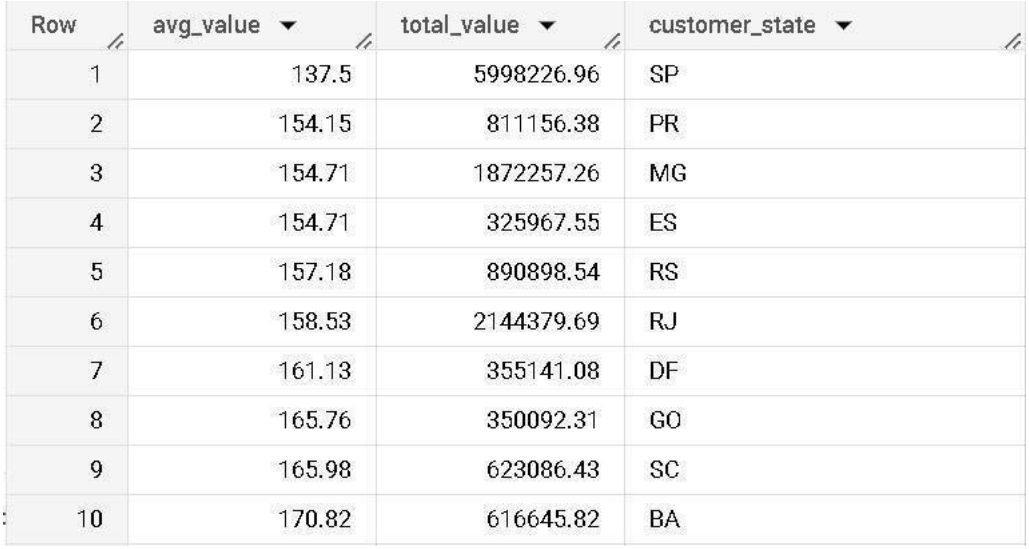
There has been a significant growth of 237 percent in the order amount from 2017 to 2018 between months January and August.

- Calculate the Total & Average value of order price for each state.



```
SELECT
  ROUND(AVG(p.payment_value),2) AS avg_value,
  ROUND(SUM(p.payment_value),2) AS total_value,
  c.customer_state
FROM `target.customers` c join `target.orders` o on
  c.customer_id=o.customer_id join `target.payments` p on
  o.order_id = p.order_id
GROUP BY c.customer_state
ORDER BY avg_value
```

Output:

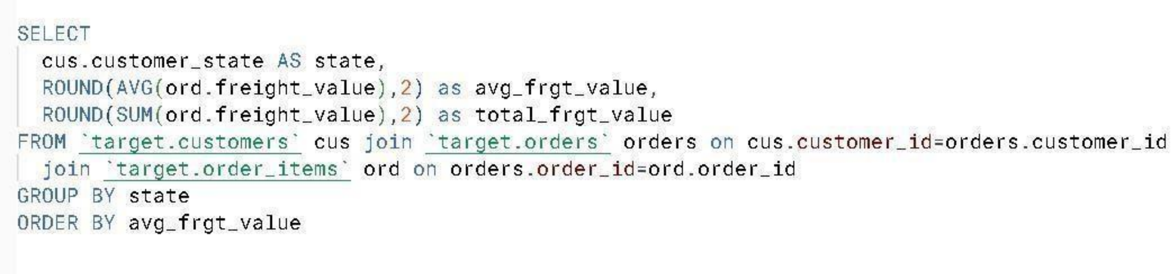


Row	avg_value	total_value	customer_state
1	137.5	5998226.96	SP
2	154.15	811156.38	PR
3	154.71	1872257.26	MG
4	154.71	325967.55	ES
5	157.18	890898.54	RS
6	158.53	2144379.69	RJ
7	161.13	355141.08	DF
8	165.76	350092.31	GO
9	165.98	623086.43	SC
10	170.82	616645.82	BA

Insight:

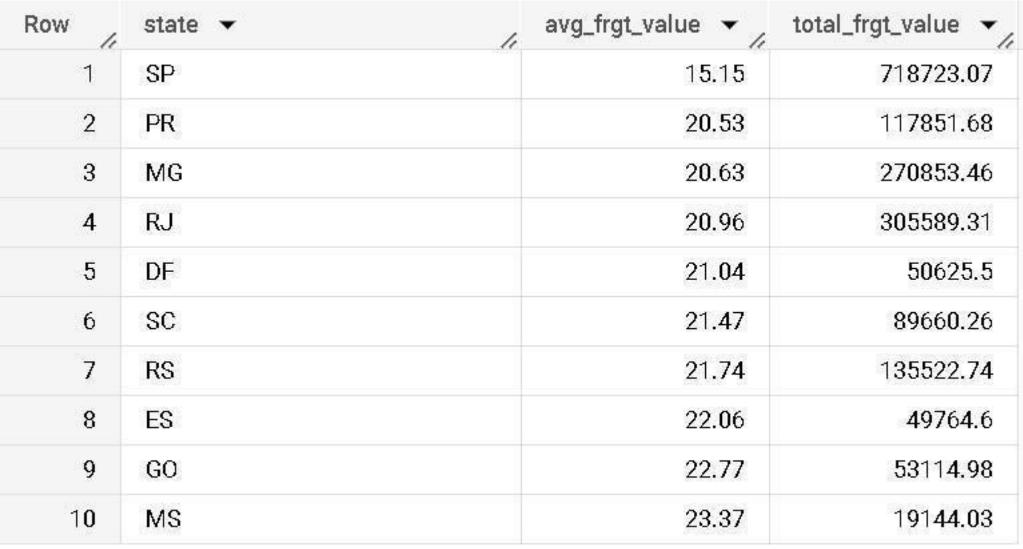
There are 27 states in total of which 10 states here have average value between 200 and 250, 16 states having average value between 150 and 200 one state having 137.5 average value. While the state which has the lowest average value per order also has the highest total order value among all states.

- Calculate the Total & Average value of order freight for each state.



```
SELECT
  cus.customer_state AS state,
  ROUND(AVG(ord.freight_value),2) as avg_frgt_value,
  ROUND(SUM(ord.freight_value),2) as total_frgt_value
FROM `target.customers` cus join `target.orders` orders on cus.customer_id=orders.customer_id
  join `target.order_items` ord on orders.order_id=ord.order_id
GROUP BY state
ORDER BY avg_frgt_value
```

Output:



Row	state	avg_frgt_value	total_frgt_value
1	SP	15.15	718723.07
2	PR	20.53	117851.68
3	MG	20.63	270853.46
4	RJ	20.96	305589.31
5	DF	21.04	50625.5
6	SC	21.47	89660.26
7	RS	21.74	135522.74
8	ES	22.06	49764.6
9	GO	22.77	53114.98
10	MS	23.37	19144.03

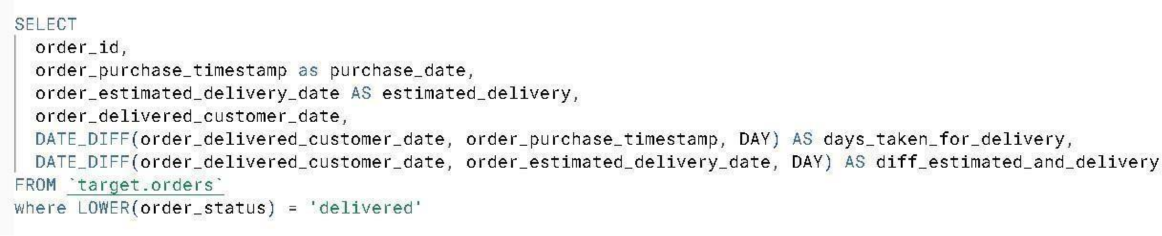
Insight:

The lowest average freight value $15.15 and highest total freight value $7187723.07 is for SP state and the highest average freight value $42.98 and lowest total freight value $2235.19 is for RR state.

- Find the no. of days taken to deliver each order from the order's purchase date as delivery time.

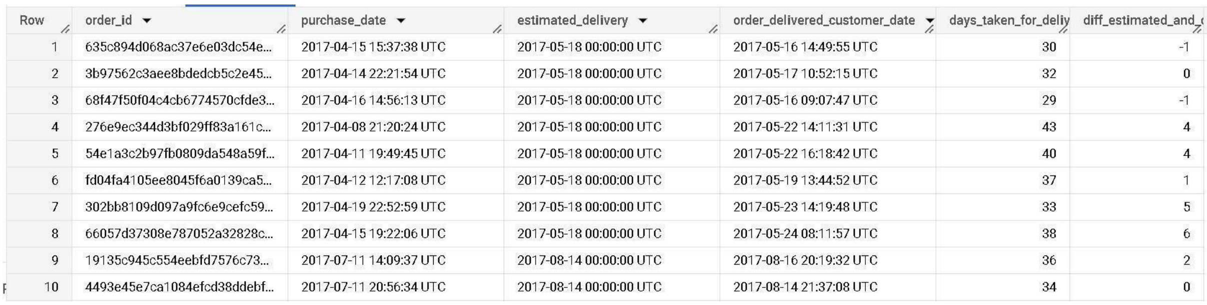
Also, calculate the difference (in days) between the estimated & actual delivery date

of an order. Do this in a single query.



```
SELECT
  order_id,
  order_purchase_timestamp as purchase_date,
  order_estimated_delivery_date AS estimated_delivery,
  order_delivered_customer_date,
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS days_taken_for_delivery,
  DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date, DAY) AS diff_estimated_and_delivery
FROM `target.orders`
where LOWER(order_status) = 'delivered'
```

Output:

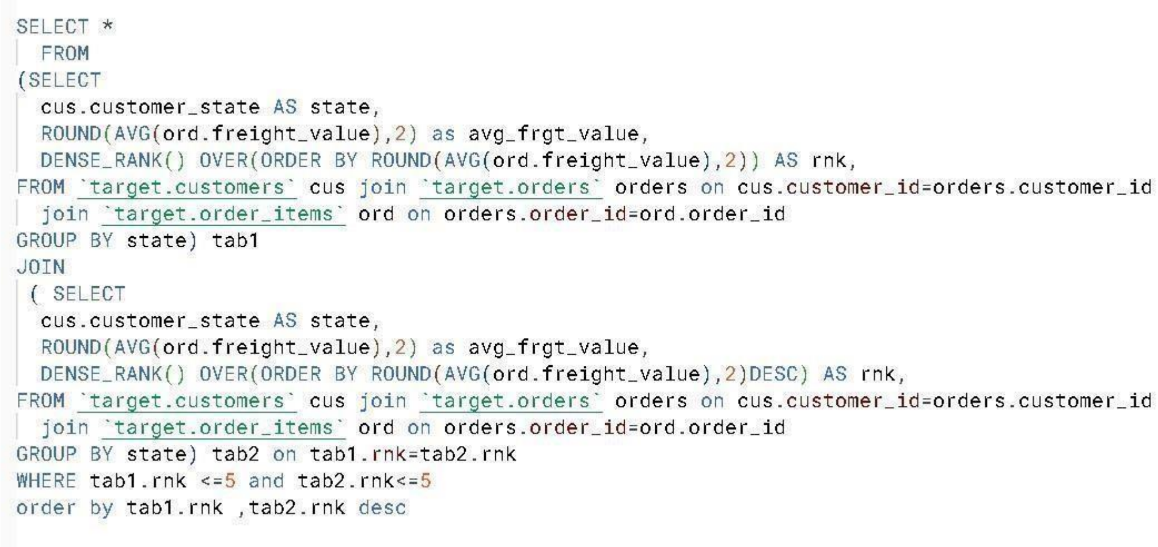


Row	order_id	purchase_date	estimated_delivery	order_delivered_customer_date	days_taken_for_deliy	diff_estimated_and_
1	635c894d068ac37e6e03dc54e...	2017-04-15 15:37:38 UTC	2017-05-18 00:00:00 UTC	2017-05-16 14:49:55 UTC	30	-1
2	3b97562c3aee8bdedcb5c2e45...	2017-04-14 22:21:54 UTC	2017-05-18 00:00:00 UTC	2017-05-17 10:52:15 UTC	32	0
3	68f47f50f04c4cb6774570cfde3...	2017-04-16 14:56:13 UTC	2017-05-18 00:00:00 UTC	2017-05-16 09:07:47 UTC	29	-1
4	276e9ec344d3bf029ff83a161c...	2017-04-08 21:20:24 UTC	2017-05-18 00:00:00 UTC	2017-05-22 14:11:31 UTC	43	4
5	54e1a3c2b97fb0809da548a59f...	2017-04-11 19:49:45 UTC	2017-05-18 00:00:00 UTC	2017-05-22 16:18:42 UTC	40	4
6	fd04fa4105ee8045f6a0139ca5...	2017-04-12 12:17:08 UTC	2017-05-18 00:00:00 UTC	2017-05-19 13:44:52 UTC	37	1
7	302bb8109d097a9fc6e9cefc59...	2017-04-19 22:52:59 UTC	2017-05-18 00:00:00 UTC	2017-05-23 14:19:48 UTC	33	5
8	66057d37308e787052a32828c...	2017-04-15 19:22:06 UTC	2017-05-18 00:00:00 UTC	2017-05-24 08:11:57 UTC	38	6
9	19135c945c554eebfd7576c73...	2017-07-11 14:09:37 UTC	2017-08-14 00:00:00 UTC	2017-08-16 20:19:32 UTC	36	2
10	4493e45e7ca1084efcd38ddebf...	2017-07-11 20:56:34 UTC	2017-08-14 00:00:00 UTC	2017-08-14 21:37:08 UTC	34	0

Insight:

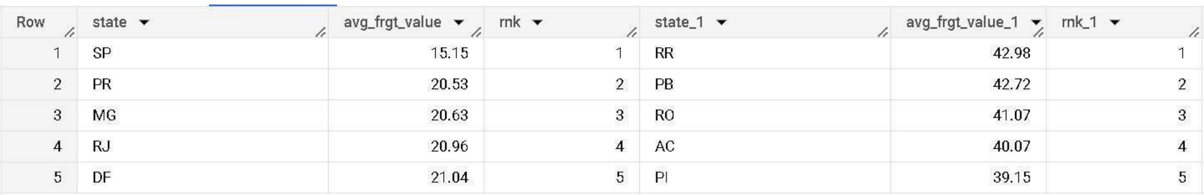
The minus sign in diff_estimated_and_delivery shows that there are orders which were delivered before the estimated delivery date. The average time taken for delivery is 12.09 days.

- Find out the top 5 states with the highest & lowest average freight value.



```
SELECT *
  FROM
(SELECT
  cus.customer_state AS state,
  ROUND(AVG(ord.freight_value),2) as avg_frgt_value,
  DENSE_RANK() OVER(ORDER BY ROUND(AVG(ord.freight_value),2)) AS rnk,
FROM `target.customers` cus join `target.orders` orders on cus.customer_id=orders.customer_id
  join `target.order_items` ord on orders.order_id=ord.order_id
GROUP BY state) tab1
JOIN
 ( SELECT
  cus.customer_state AS state,
  ROUND(AVG(ord.freight_value),2) as avg_frgt_value,
  DENSE_RANK() OVER(ORDER BY ROUND(AVG(ord.freight_value),2)DESC) AS rnk,
FROM `target.customers` cus join `target.orders` orders on cus.customer_id=orders.customer_id
  join `target.order_items` ord on orders.order_id=ord.order_id
GROUP BY state) tab2 on tab1.rnk=tab2.rnk
WHERE tab1.rnk <=5 and tab2.rnk<=5
order by tab1.rnk ,tab2.rnk desc
```

Output:

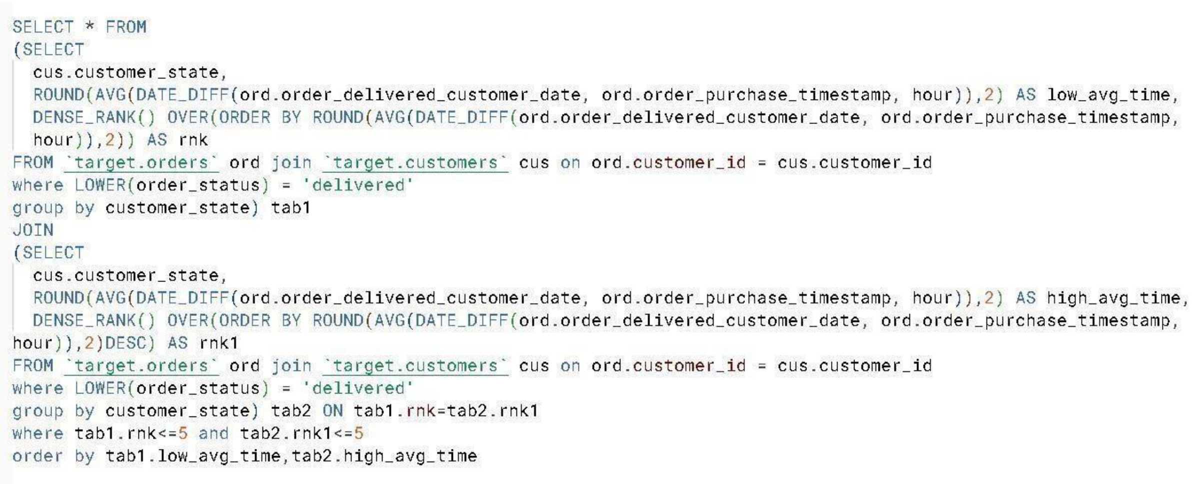


Row	state	avg_frgt_value	rnk	state_1	avg_frgt_value_1	rnk_1
1	SP	15.15	1	RR	42.98	1
2	PR	20.53	2	PB	42.72	2
3	MG	20.63	3	RO	41.07	3
4	RJ	20.96	4	AC	40.07	4
5	DF	21.04	5	PI	39.15	5

Insight:

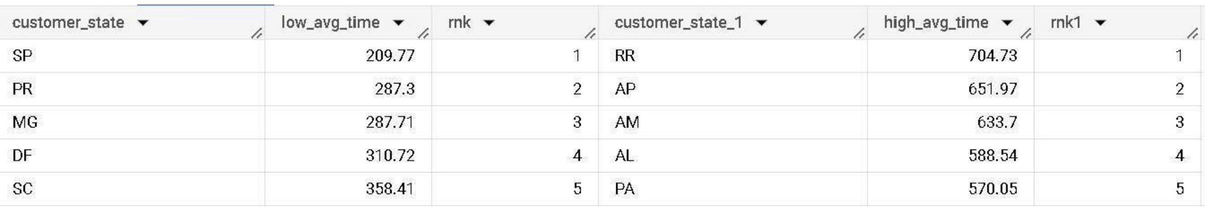
Top 5 states with the highest and lowest average freight value lowest being SP state with $15.15 and highest being RR state with $42.98.

- Find out the top 5 states with the highest & lowest average delivery time.



```
SELECT * FROM
(SELECT
  cus.customer_state,
  ROUND(AVG(DATE_DIFF(ord.order_delivered_customer_date, ord.order_purchase_timestamp, hour)),2) AS low_avg_time,
  DENSE_RANK() OVER(ORDER BY ROUND(AVG(DATE_DIFF(ord.order_delivered_customer_date, ord.order_purchase_timestamp,
  hour)),2)) AS rnk
FROM `target.orders` ord join `target.customers` cus on ord.customer_id = cus.customer_id
where LOWER(order_status) = 'delivered'
group by customer_state) tab1
JOIN
(SELECT
  cus.customer_state,
  ROUND(AVG(DATE_DIFF(ord.order_delivered_customer_date, ord.order_purchase_timestamp, hour)),2) AS high_avg_time,
  DENSE_RANK() OVER(ORDER BY ROUND(AVG(DATE_DIFF(ord.order_delivered_customer_date, ord.order_purchase_timestamp,
hour)),2)DESC) AS rnk1
FROM `target.orders` ord join `target.customers` cus on ord.customer_id = cus.customer_id
where LOWER(order_status) = 'delivered'
group by customer_state) tab2 ON tab1.rnk=tab2.rnk1
where tab1.rnk<=5 and tab2.rnk1<=5
order by tab1.low_avg_time,tab2.high_avg_time
```

Output:

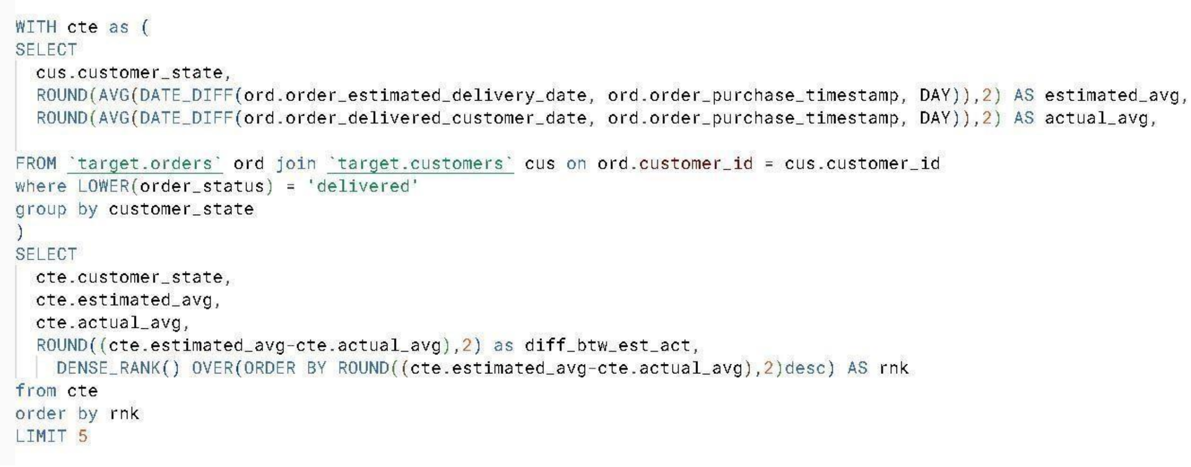


customer_state	low_avg_time	rnk	customer_state_1	high_avg_time	rnk1
SP	209.77	1	RR	704.73	1
PR	287.3	2	AP	651.97	2
MG	287.71	3	AM	633.7	3
DF	310.72	4	AL	588.54	4
SC	358.41	5	PA	570.05	5

Insight:

The time in avg_time columns is in hours, the top 5 states with the highest and lowest time taken to deliver to customers are above. Lowest average being SP state with 209.77 hrs and highest average being RR state with 704.73 hrs.

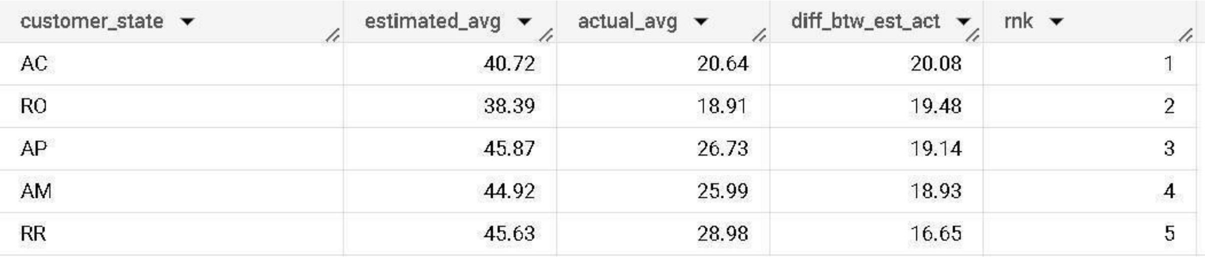
- Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.



```
WITH cte as (
SELECT
  cus.customer_state,
  ROUND(AVG(DATE_DIFF(ord.order_estimated_delivery_date, ord.order_purchase_timestamp, DAY)),2) AS estimated_avg,
  ROUND(AVG(DATE_DIFF(ord.order_delivered_customer_date, ord.order_purchase_timestamp, DAY)),2) AS actual_avg,

FROM `target.orders` ord join `target.customers` cus on ord.customer_id = cus.customer_id
where LOWER(order_status) = 'delivered'
group by customer_state
)
SELECT
  cte.customer_state,
  cte.estimated_avg,
  cte.actual_avg,
  ROUND((cte.estimated_avg-cte.actual_avg),2) as diff_btw_est_act,
    DENSE_RANK() OVER(ORDER BY ROUND((cte.estimated_avg-cte.actual_avg),2)desc) AS rnk
from cte
order by rnk
LIMIT 5
```

Output:

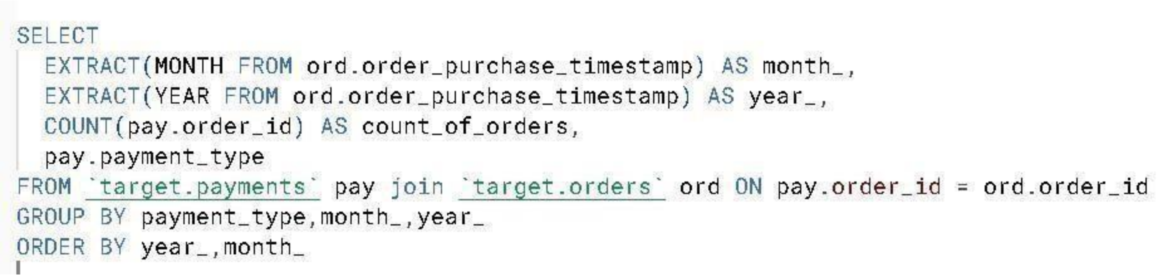


customer_state	estimated_avg	actual_avg	diff_btw_est_act	rnk
AC	40.72	20.64	20.08	1
RO	38.39	18.91	19.48	2
AP	45.87	26.73	19.14	3
AM	44.92	25.99	18.93	4
RR	45.63	28.98	16.65	5

Insight:

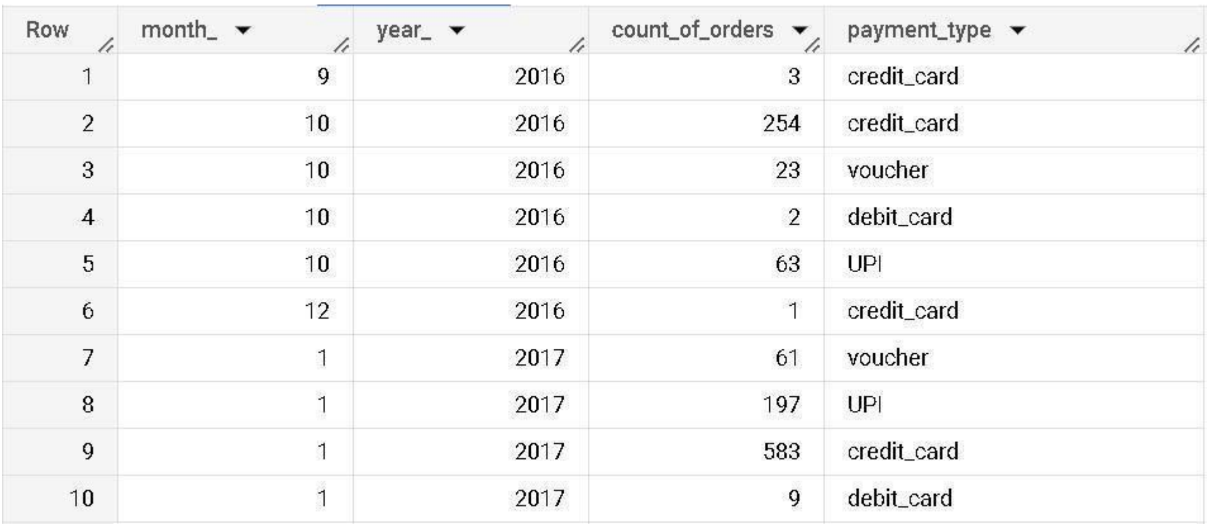
Top 5 states where the order delivery is really fast as compared to the estimated date of delivery are AC, RO, AP, AM, and RR with highest average between estimated and actual delivery date with the AC state with 20.08 days.

- Find the month-on-month no. of orders placed using different payment types.



```
SELECT
  EXTRACT(MONTH FROM ord.order_purchase_timestamp) AS month_,
  EXTRACT(YEAR FROM ord.order_purchase_timestamp) AS year_,
  COUNT(pay.order_id) AS count_of_orders,
  pay.payment_type
FROM `target.payments` pay join `target.orders` ord ON pay.order_id = ord.order_id
GROUP BY payment_type,month_,year_
ORDER BY year_,month_
```

Output:

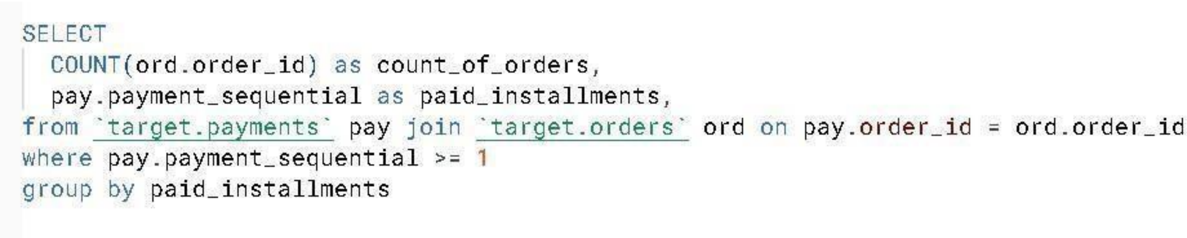


Row	month_	year_	count_of_orders	payment_type
1	9	2016	3	credit_card
2	10	2016	254	credit_card
3	10	2016	23	voucher
4	10	2016	2	debit_card
5	10	2016	63	UPI
6	12	2016	1	credit_card
7	1	2017	61	voucher
8	1	2017	197	UPI
9	1	2017	583	credit_card
10	1	2017	9	debit_card

Insight:

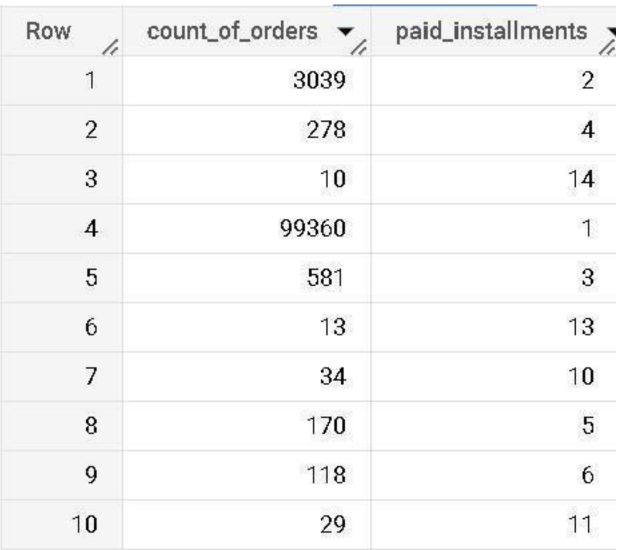
Credit-Card has been the most used payment mode by customers followed by UPI, Voucher and Debit-Card

- Find the no. of orders placed on the basis of the payment installments that have been paid.



```
SELECT
  COUNT(ord.order_id) as count_of_orders,
  pay.payment_sequential as paid_installments,
from `target.payments` pay join `target.orders` ord on pay.order_id = ord.order_id
where pay.payment_sequential >= 1
group by paid_installments
```

Output:



Row	count_of_orders	paid_installments
1	3039	2
2	278	4
3	10	14
4	99360	1
5	581	3
6	13	13
7	34	10
8	170	5
9	118	6
10	29	11

Insight:

103886 orders are placed on the basis of payment installments.

Recommendations

- During peak seasons, the monthly orders growth rate can reach as high Jan 2017 percentage. To meet such high demands, it is crucial to adjust inventory levels, accordingly, ensuring that all items are adequately stocked. Failing to do so may result in Target losing sales from potential customers, who may turn to competitors to fulfil their needs. Consequently, Target would not only miss out on these sales but also lose potential profits.

- Implementing discount pricing strategies before the peak seasons can be an effective approach for Target to attract new customers from different regions of Brazil, retain existing customers, boost sales, and generate interest in new products. By offering competitive pricing, Target can entice potential customers to choose their products over competitors, leading to increased market share and higher profits. Moreover, attracting new customers and retaining existing ones through discounted pricing can build long-term customer loyalty and create a positive brand image. Overall, a wellexecuted discount pricing strategy can contribute significantly to Target's profitability and overall success.

- To cater to the high volume of orders made during the night and afternoon, Target should prioritize ensuring the smooth operation of their website during these peak times. By focusing on website performance and stability, they can provide a seamless online shopping experience for their customers.

- Additionally, implementing website personalization techniques can greatly enhance the online shopping experience for each customer. By analysing past preferences and purchase history, Target can customize the website to display relevant product recommendations, tailored promotions, and personalized content. This level of personalization can help customers find items of interest more efficiently, leading to increased satisfaction and potentially higher sales.
- By investing in website optimization and personalization, Target can not only improve customer satisfaction and loyalty but also gain a competitive edge in the online retail market.
- To minimize freight costs and reduce delivery times, it is crucial for Target to focus on building a robust seller network. By expanding and strengthening their network of

sellers across various regions, Target can enhance the overall customer buying experience.

- A strong seller network allows Target to have a wider range of products available for customers, reducing the need for long-distance shipments. With local sellers in different regions, Target can utilize their inventory to fulfil orders more efficiently, resulting in shorter delivery times and potentially lower freight costs.
- By working closely with sellers, Target can establish partnerships that prioritize fast and reliable shipping methods. This collaboration enables the implementation of streamlined logistics processes, optimizing the delivery of products to customers.
- Moreover, a strong seller network enables Target to have a deeper understanding of regional markets and customer preferences. This knowledge can be utilized to tailor product offerings and marketing strategies to specific regions, further enhancing the customer buying experience.
- Overall, by focusing on building and nurturing a strong seller network, Target can improve its operational efficiency, reduce freight costs, and provide faster delivery times, thereby enhancing the overall customer buying experience across all regions.

○

○