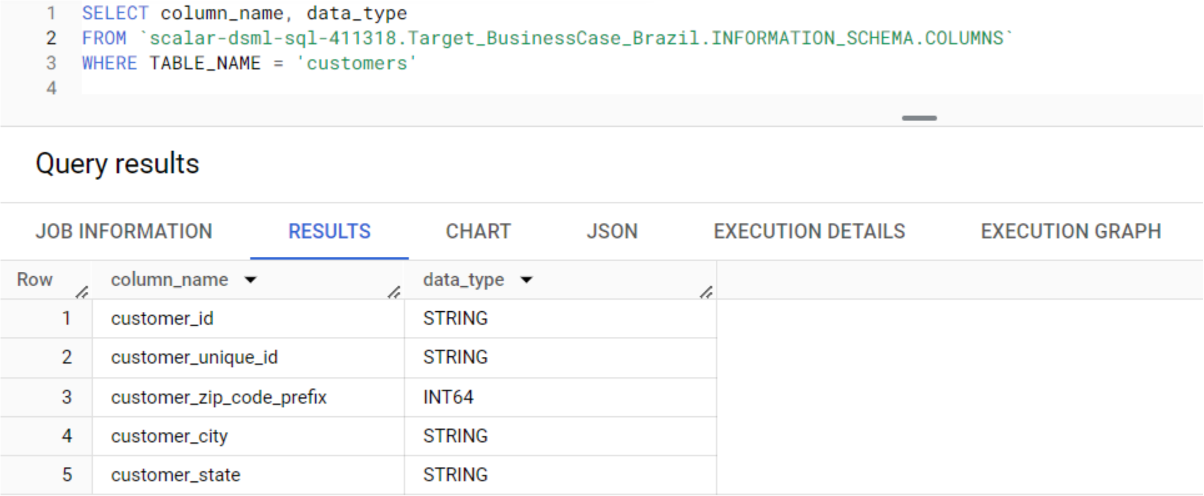
Operations of Target in Brazil

1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**

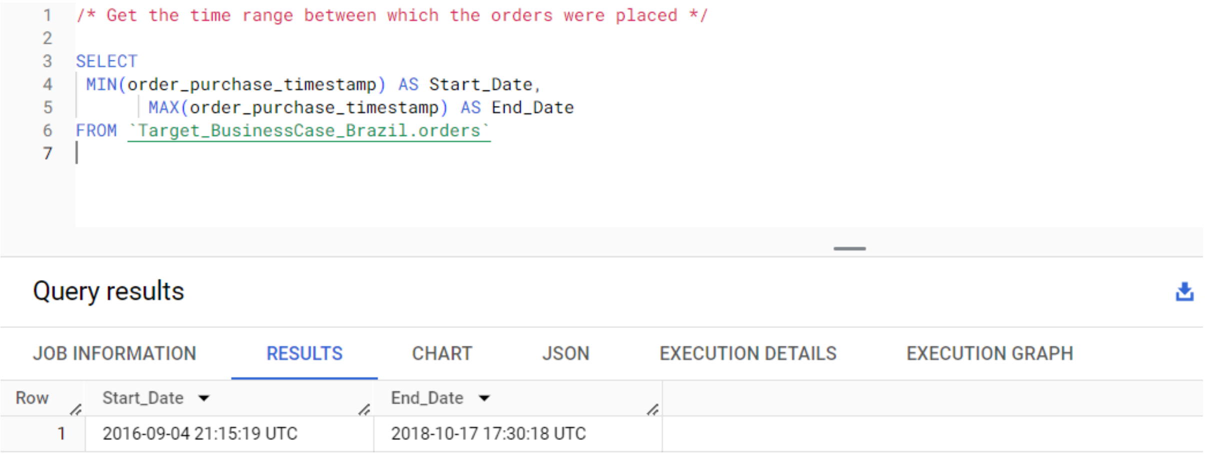
1. Data type of all columns in the "customers" table:



```sql
SELECT column_name, data_type
FROM `scalar-dsml-sql-411318.Target_BusinessCase_Brazil.INFORMATION_SCHEMA.COLUMNS`
WHERE TABLE_NAME = 'customers'

```

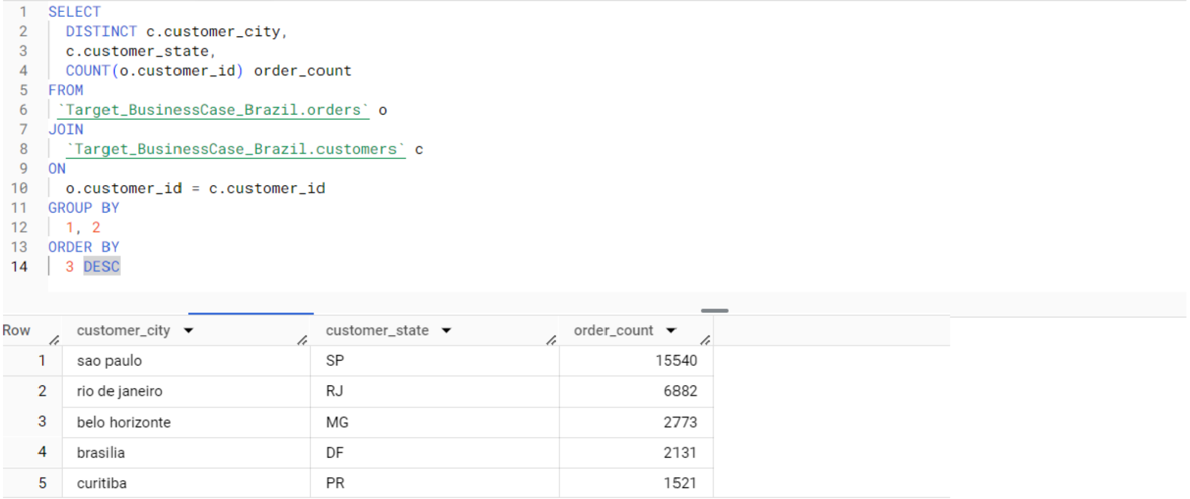
Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

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


```
/* 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_BusinessCase_Brazil.orders`
```

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

3. Count the Cities & States of customers who ordered during the given period


```
SELECT
  DISTINCT c.customer_city,
  c.customer_state,
  COUNT(o.customer_id) order_count
FROM
 `Target_BusinessCase_Brazil.orders` o
JOIN
  `Target_BusinessCase_Brazil.customers` c
ON
  o.customer_id = c.customer_id
GROUP BY
  1, 2
ORDER BY
  3 DESC
```

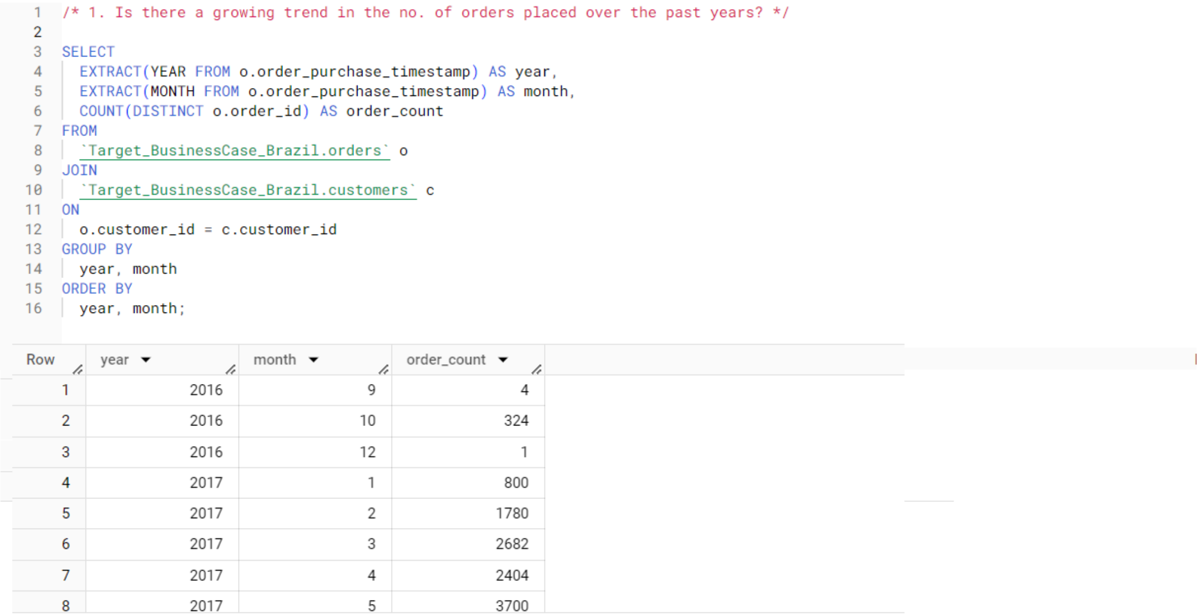
Row	customer_city	customer_state	order_count
1	sao paulo	SP	15540
2	rio de janeiro	RJ	6882
3	belo horizonte	MG	2773
4	brasilia	DF	2131
5	curitiba	PR	1521

Insights:

Here, we can see that sao paulo city from SP state alone has more orders than the following 5 cities combined. This is because sao paulo might be the most populous and the richest state in Brazil

2. In-depth Exploration:

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



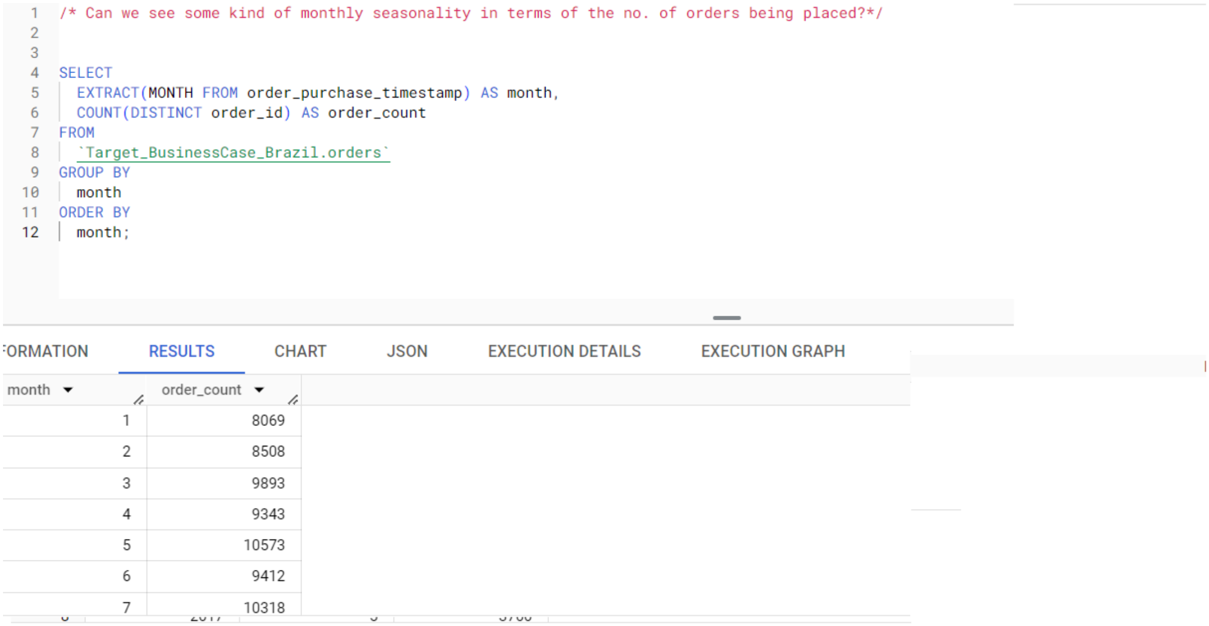
```sql
/* 1. Is there a growing trend in the no. of orders placed over the past years? */

SELECT
  EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
  COUNT(DISTINCT o.order_id) AS order_count
FROM
  `Target_BusinessCase_Brazil.orders` o
JOIN
  `Target_BusinessCase_Brazil.customers` c
ON
  o.customer_id = c.customer_id
GROUP BY
  year, month
ORDER BY
  year, month;
```

Row	year	month	order_count
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700

Insights:

Based on the analysis of order count, it can be observed that there is a growing trend. The count of purchases has shown an overall upward trend, with some fluctuations.

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



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


SELECT
  EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
  COUNT(DISTINCT order_id) AS order_count
FROM
  `Target_BusinessCase_Brazil.orders`
GROUP BY
  month
ORDER BY
  month;
```

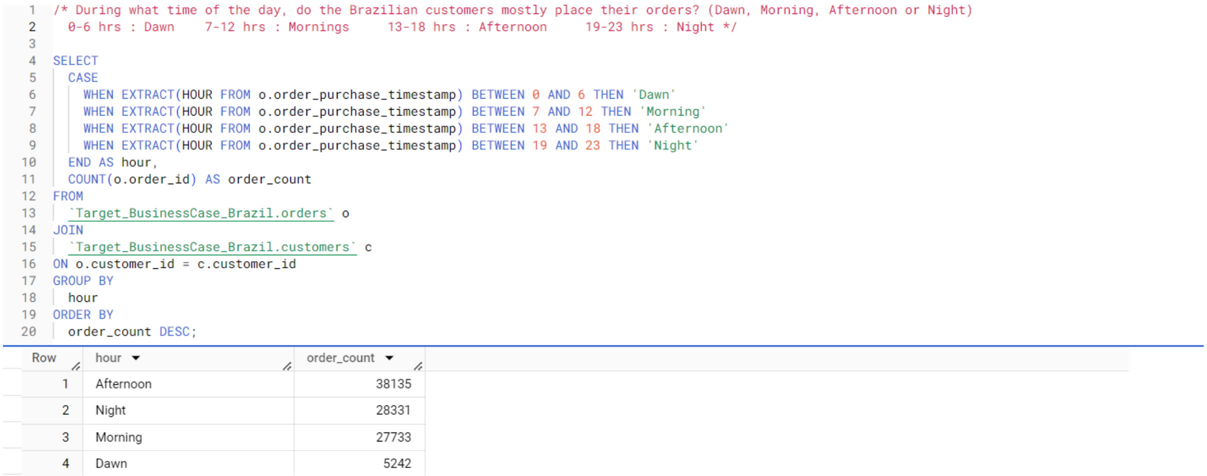
month	order_count
1	8069
2	8508
3	9893
4	9343
5	10573
6	9412
7	10318

Insights:

The count of orders generally increases from March to August with fluctuations in between. Notably, there is an increase in orders during February and March. Additionally, the month of August shows a peak in order count.

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

   i. 0-6 hrs : Dawn
   ii. 7-12 hrs : Mornings
   iii. 13-18 hrs : Afternoon
   iv. 19-23 hrs : Night



```sql
/* 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 */

SELECT
  CASE
    WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 0 AND 6 THEN 'Dawn'
    WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 7 AND 12 THEN 'Morning'
    WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 13 AND 18 THEN 'Afternoon'
    WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 19 AND 23 THEN 'Night'
  END AS hour,
  COUNT(o.order_id) AS order_count
FROM
  `Target_BusinessCase_Brazil.orders` o
JOIN
  `Target_BusinessCase_Brazil.customers` c
ON o.customer_id = c.customer_id
GROUP BY
  hour
ORDER BY
  order_count DESC;
```

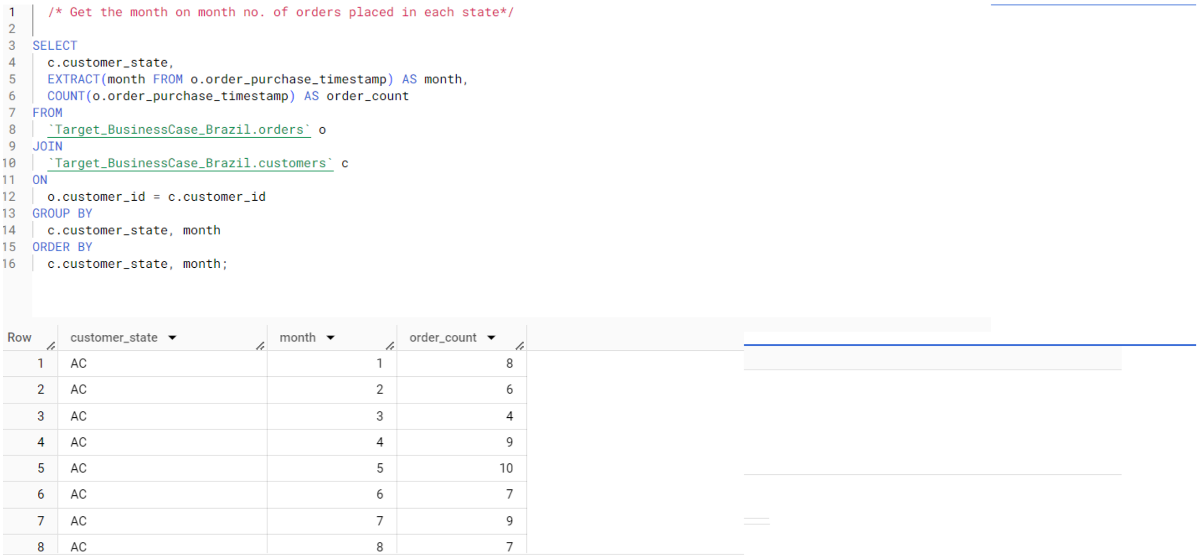
Row	hour	order_count
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

Insights:

Based on the analysis, we found that Brazilian customers tend to place most orders during the afternoon, specifically in the morning and night. This indicates that customers prefer to place orders when they have leisure time or after completing their daily activities.

3. Evolution of E-commerce orders in the Brazil region:

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



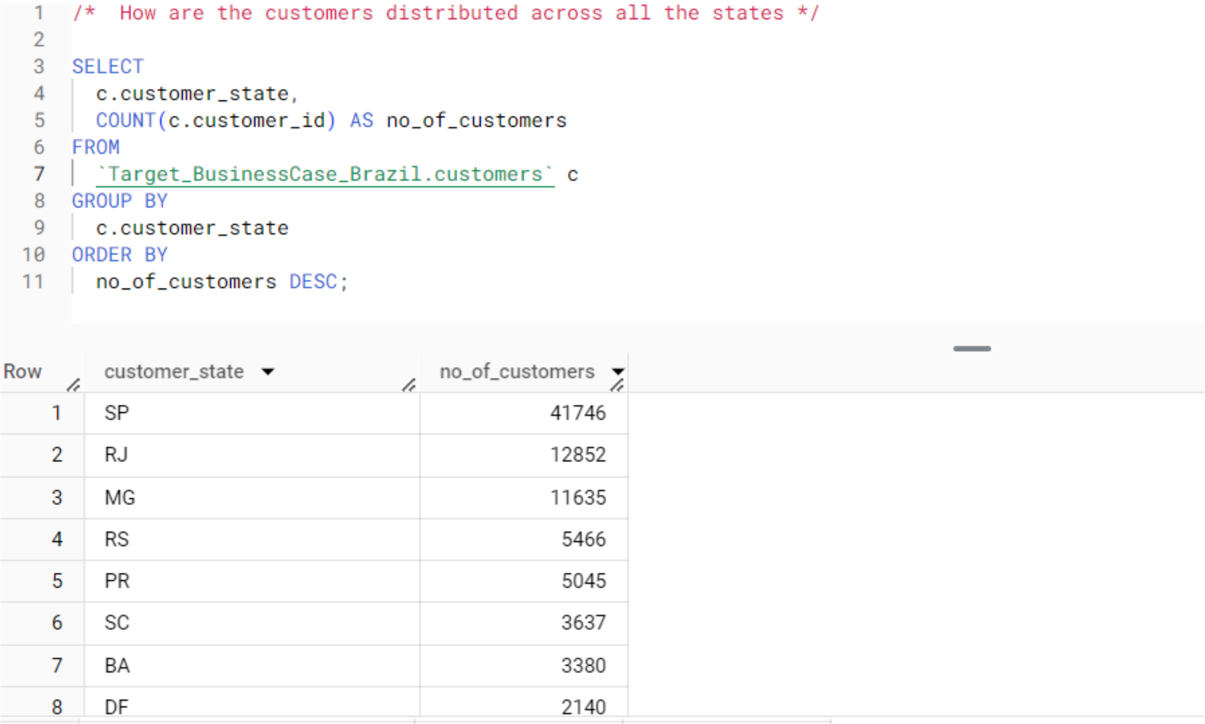
```
/* Get the month on month no. of orders placed in each state*/

SELECT
  c.customer_state,
  EXTRACT(month FROM o.order_purchase_timestamp) AS month,
  COUNT(o.order_purchase_timestamp) AS order_count
FROM
  `Target_BusinessCase_Brazil.orders` o
JOIN
  `Target_BusinessCase_Brazil.customers` c
ON
  o.customer_id = c.customer_id
GROUP BY
  c.customer_state, month
ORDER BY
  c.customer_state, month;
```

Row	customer_state	month	order_count
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7

Insights:

Based on the analysis, it is observed that state BA have highest orders in the month of Jan and Feb.
But there is huge margin of gap on orders places for remaining states.

b. How are the customers distributed across all the states?



```
/*  How are the customers distributed across all the states */

SELECT
  c.customer_state,
  COUNT(c.customer_id) AS no_of_customers
FROM
  `Target_BusinessCase_Brazil.customers` c
GROUP BY
  c.customer_state
ORDER BY
  no_of_customers DESC;
```

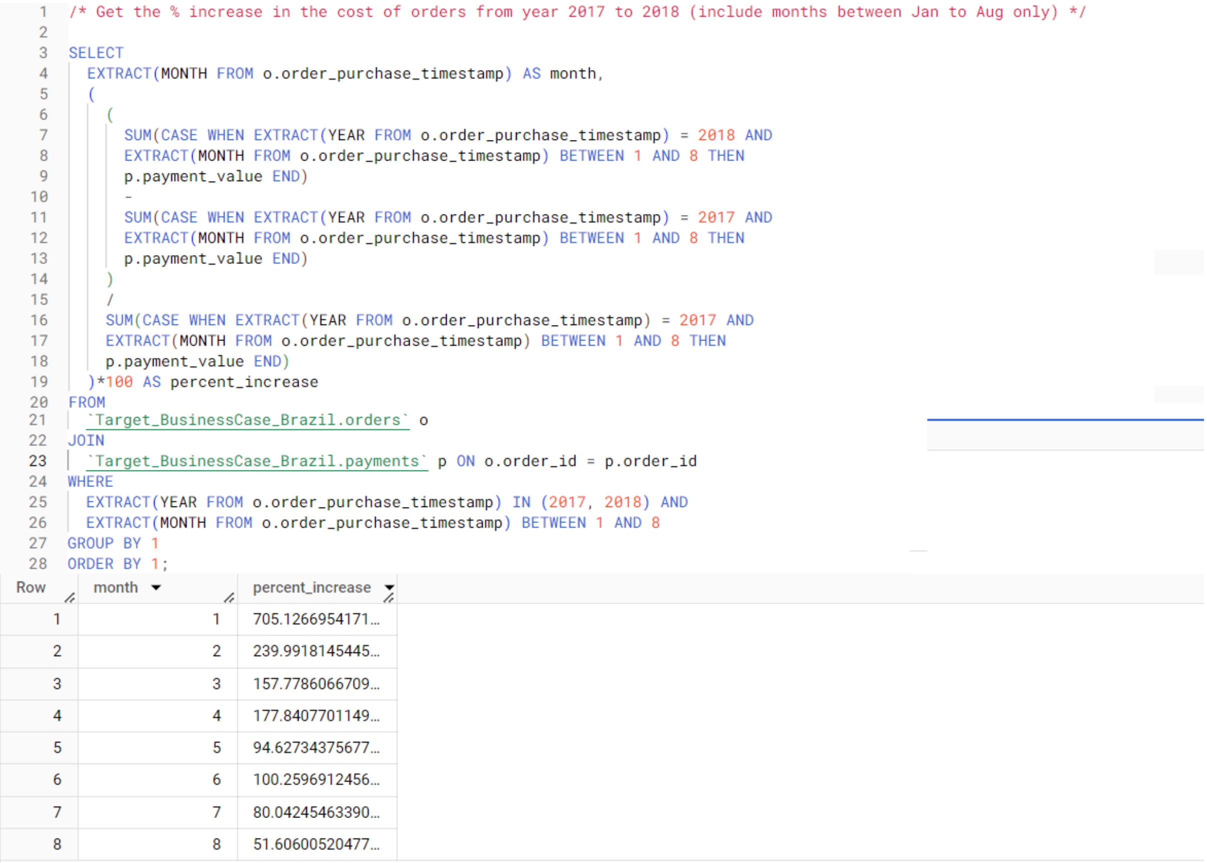
Row	customer_state	no_of_customers
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140

Insights:

The data reveals that the state of São Paulo (SP) has the highest number of customers, which can be attributed to its status as the most populous state in Brazil.

4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

a. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).
You can use the "payment_value" column in the payments table to get the cost of orders.



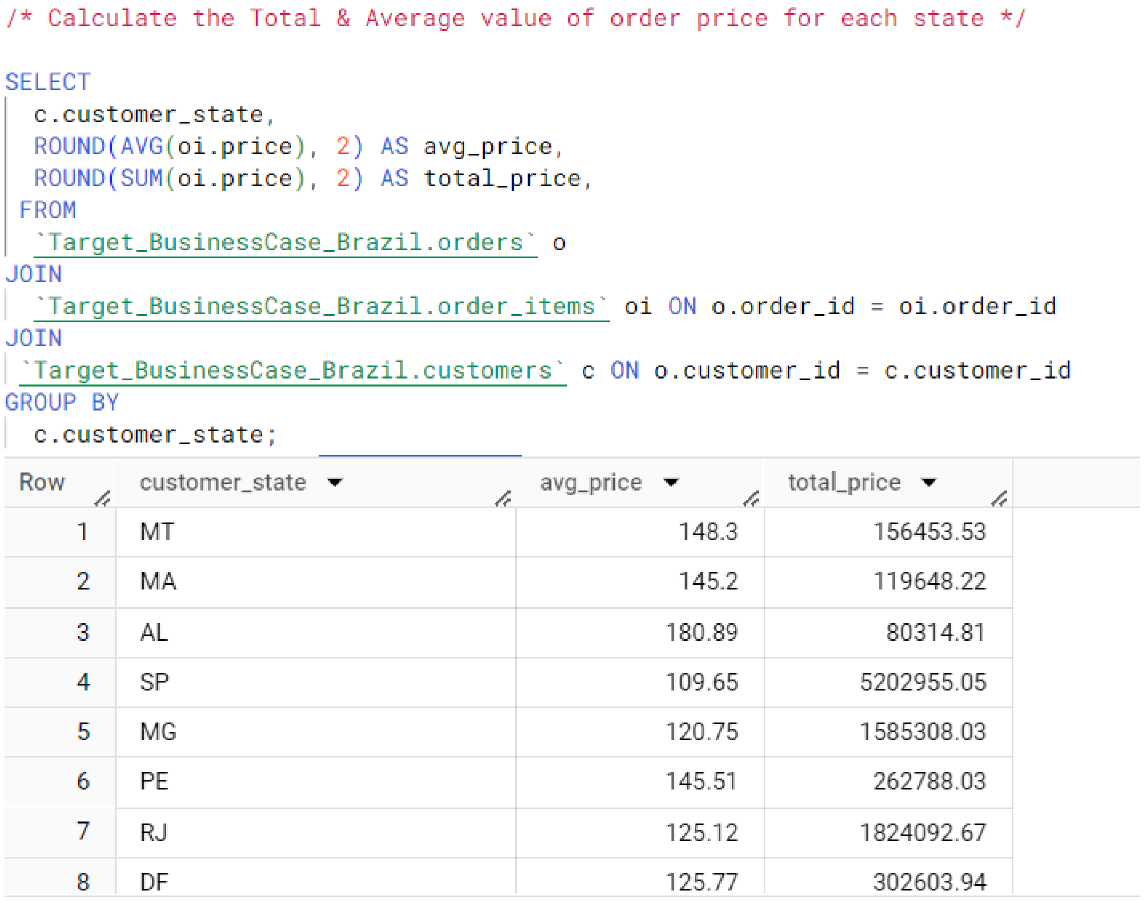
```
/* Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only) */

SELECT
  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
  (
    (
      SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2018 AND
      EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8 THEN
      p.payment_value END)
      -
      SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017 AND
      EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8 THEN
      p.payment_value END)
    )
    /
    SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017 AND
    EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8 THEN
    p.payment_value END)
  )*100 AS percent_increase
FROM
  `Target_BusinessCase_Brazil.orders` o
JOIN
  `Target_BusinessCase_Brazil.payments` p ON o.order_id = p.order_id
WHERE
  EXTRACT(YEAR FROM o.order_purchase_timestamp) IN (2017, 2018) AND
  EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
GROUP BY 1
ORDER BY 1;
```

Row	month	percent_increase
1	1	705.1266954171…
2	2	239.9918145445…
3	3	157.7786066709…
4	4	177.8407701149…
5	5	94.62734375677…
6	6	100.2596912456…
7	7	80.04245463390…
8	8	51.60600520477…

Insights:

The overall percentage increase in the cost of orders from 2017 to 2018, including only the months from January to August, is 138.53%. Upon examining the month-wise increase, January shows the highest percentage increase, followed by February and April.

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



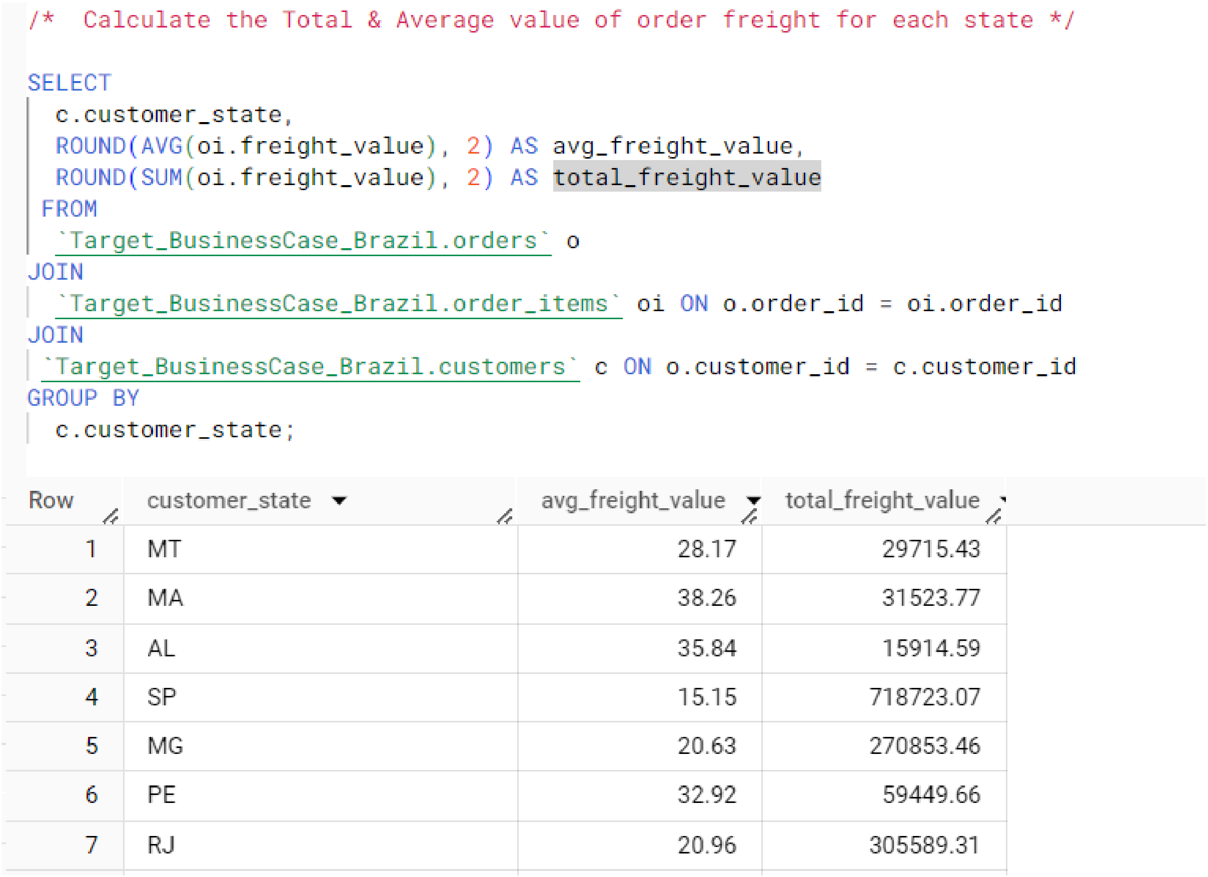
```
/* Calculate the Total & Average value of order price for each state */

SELECT
  c.customer_state,
  ROUND(AVG(oi.price), 2) AS avg_price,
  ROUND(SUM(oi.price), 2) AS total_price,
 FROM
  `Target_BusinessCase_Brazil.orders` o
JOIN
  `Target_BusinessCase_Brazil.order_items` oi ON o.order_id = oi.order_id
JOIN
 `Target_BusinessCase_Brazil.customers` c ON o.customer_id = c.customer_id
GROUP BY
  c.customer_state;
```

Row	customer_state	avg_price	total_price
1	MT	148.3	156453.53
2	MA	145.2	119648.22
3	AL	180.89	80314.81
4	SP	109.65	5202955.05
5	MG	120.75	1585308.03
6	PE	145.51	262788.03
7	RJ	125.12	1824092.67
8	DF	125.77	302603.94

Insights:

The analysis reveals interesting findings. While São Paulo (SP) has the highest total price and surprisingly it has the lowest average price value among all states.

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



```
/*  Calculate the Total & Average value of order freight for each state */

SELECT
  c.customer_state,
  ROUND(AVG(oi.freight_value), 2) AS avg_freight_value,
  ROUND(SUM(oi.freight_value), 2) AS total_freight_value
 FROM
  `Target_BusinessCase_Brazil.orders` o
JOIN
  `Target_BusinessCase_Brazil.order_items` oi ON o.order_id = oi.order_id
JOIN
 `Target_BusinessCase_Brazil.customers` c ON o.customer_id = c.customer_id
GROUP BY
  c.customer_state;
```

Row	customer_state	avg_freight_value	total_freight_value
1	MT	28.17	29715.43
2	MA	38.26	31523.77
3	AL	35.84	15914.59
4	SP	15.15	718723.07
5	MG	20.63	270853.46
6	PE	32.92	59449.66
7	RJ	20.96	305589.31

Insights:

The analysis reveals interesting findings. While São Paulo (SP) has the highest total freight value and surprisingly has the lowest average freight value among all states.

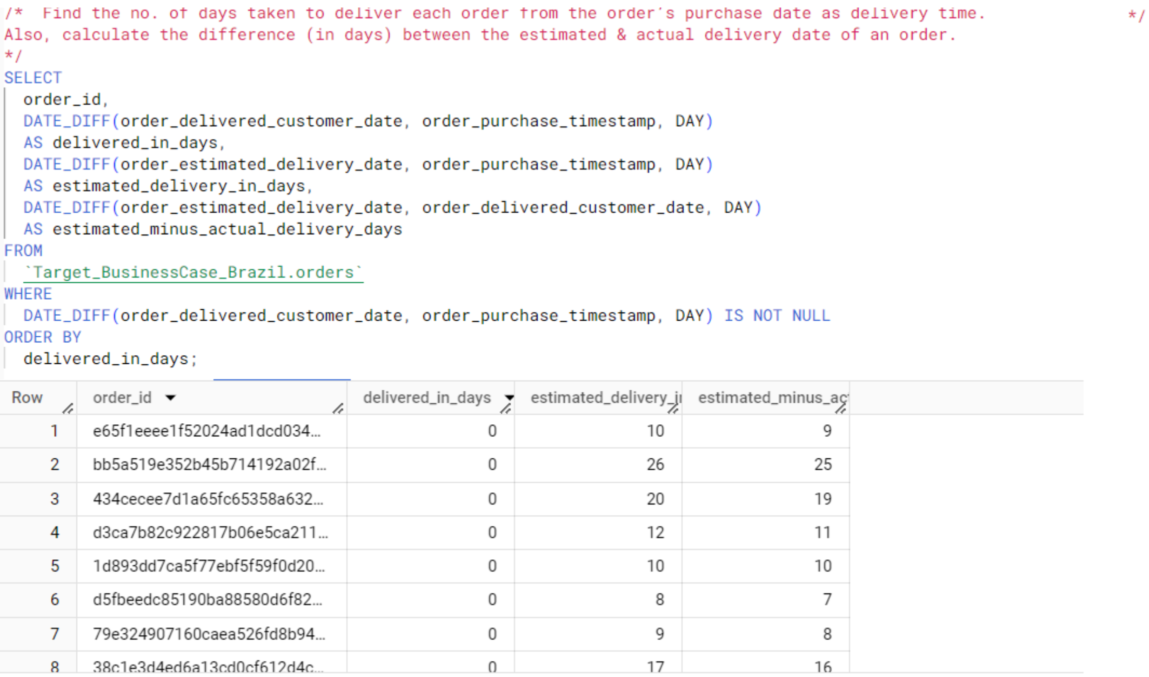
5. Analysis based on sales, freight and delivery time.

a. 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.

You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

i. **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp

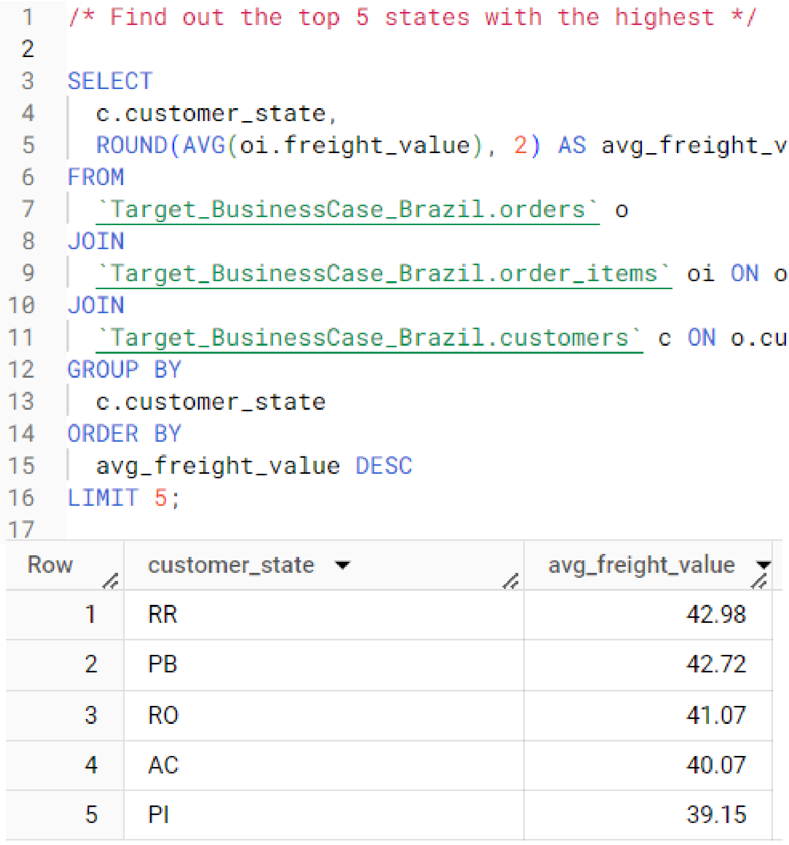
ii. **diff_estimated_delivery** = order_delivered_customer_date - order_estimated_delivery_date



```
/*  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.
*/
SELECT
  order_id,
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)
  AS delivered_in_days,
  DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp, DAY)
  AS estimated_delivery_in_days,
  DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY)
  AS estimated_minus_actual_delivery_days
FROM
  `Target_BusinessCase_Brazil.orders`
WHERE
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) IS NOT NULL
ORDER BY
  delivered_in_days;
```

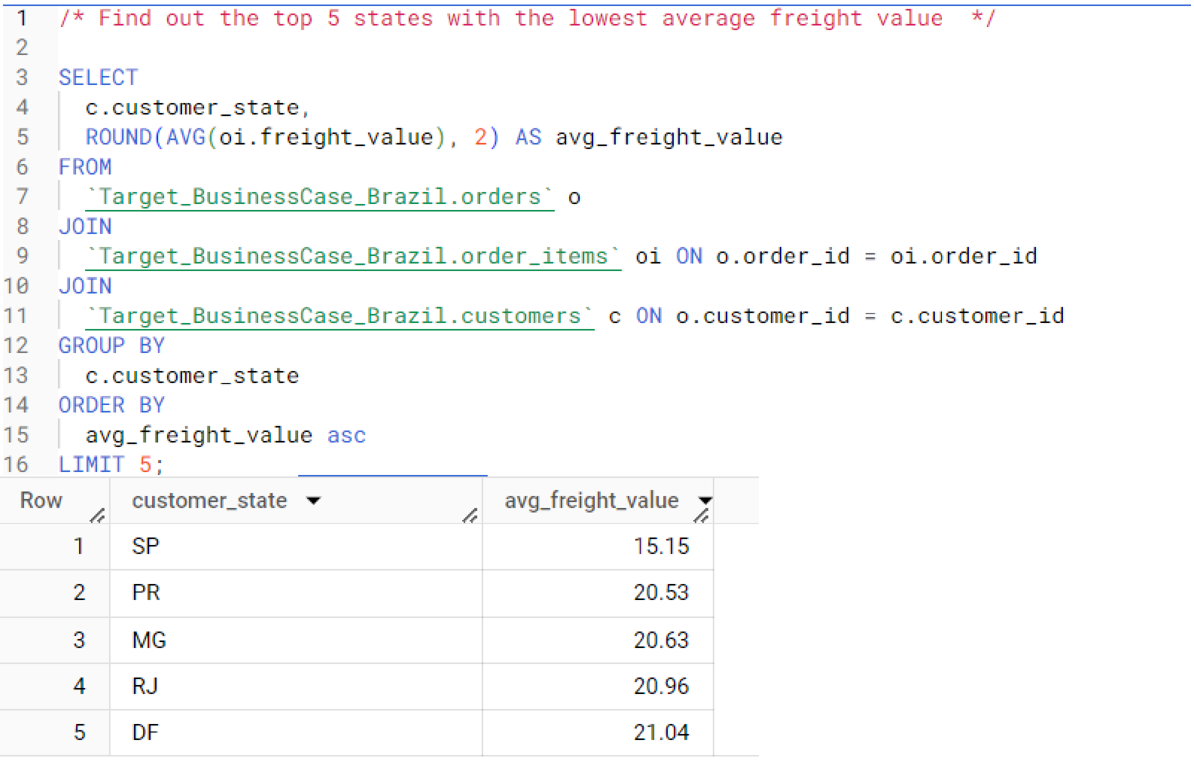
Row	order_id	delivered_in_days	estimated_delivery_i	estimated_minus_ac
1	e65f1eeee1f52024ad1dcd034...	0	10	9
2	bb5a519e352b45b714192a02f...	0	26	25
3	434cecee7d1a65fc65358a632...	0	20	19
4	d3ca7b82c922817b06e5ca211...	0	12	11
5	1d893dd7ca5f77ebf5f59f0d20...	0	10	10
6	d5fbeedc85190ba88580d6f82...	0	8	7
7	79e324907160caea526fd8b94...	0	9	8
8	38c1e3d4ed6a13cd0cf612d4c...	0	17	16

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



```sql
/* Find out the top 5 states with the highest */

SELECT
  c.customer_state,
  ROUND(AVG(oi.freight_value), 2) AS avg_freight_value
FROM
  `Target_BusinessCase_Brazil.orders` o
JOIN
  `Target_BusinessCase_Brazil.order_items` oi ON o.order_id = oi.order_id
JOIN
  `Target_BusinessCase_Brazil.customers` c ON o.customer_id = c.customer_id
GROUP BY
  c.customer_state
ORDER BY
  avg_freight_value DESC
LIMIT 5;

```

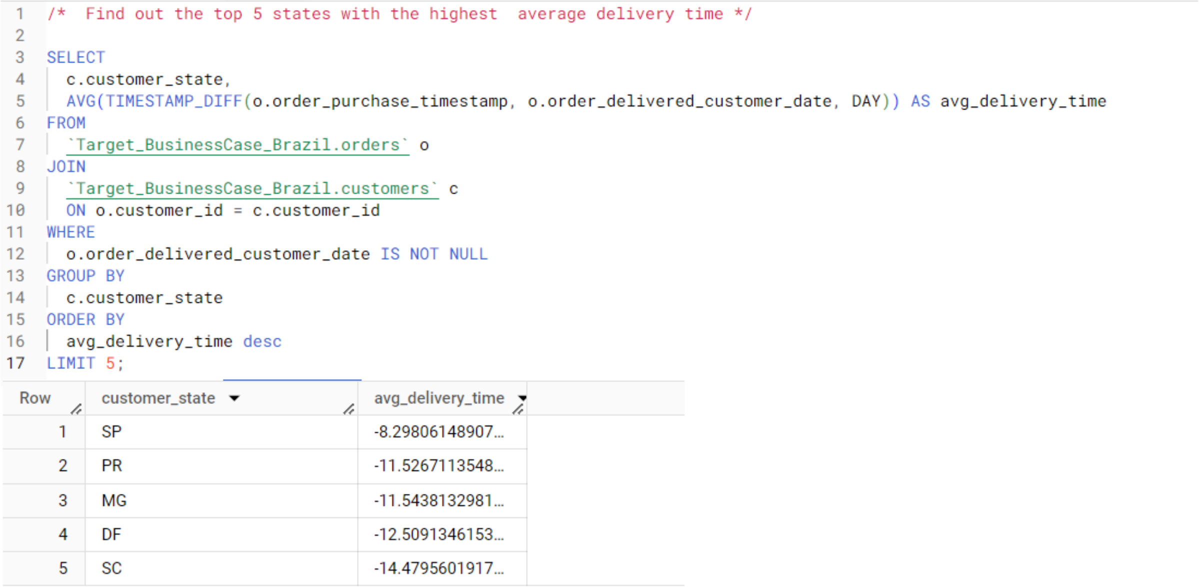
Row	customer_state	avg_freight_value
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15



```sql
/* Find out the top 5 states with the lowest average freight value  */

SELECT
  c.customer_state,
  ROUND(AVG(oi.freight_value), 2) AS avg_freight_value
FROM
  `Target_BusinessCase_Brazil.orders` o
JOIN
  `Target_BusinessCase_Brazil.order_items` oi ON o.order_id = oi.order_id
JOIN
  `Target_BusinessCase_Brazil.customers` c ON o.customer_id = c.customer_id
GROUP BY
  c.customer_state
ORDER BY
  avg_freight_value asc
LIMIT 5;
```

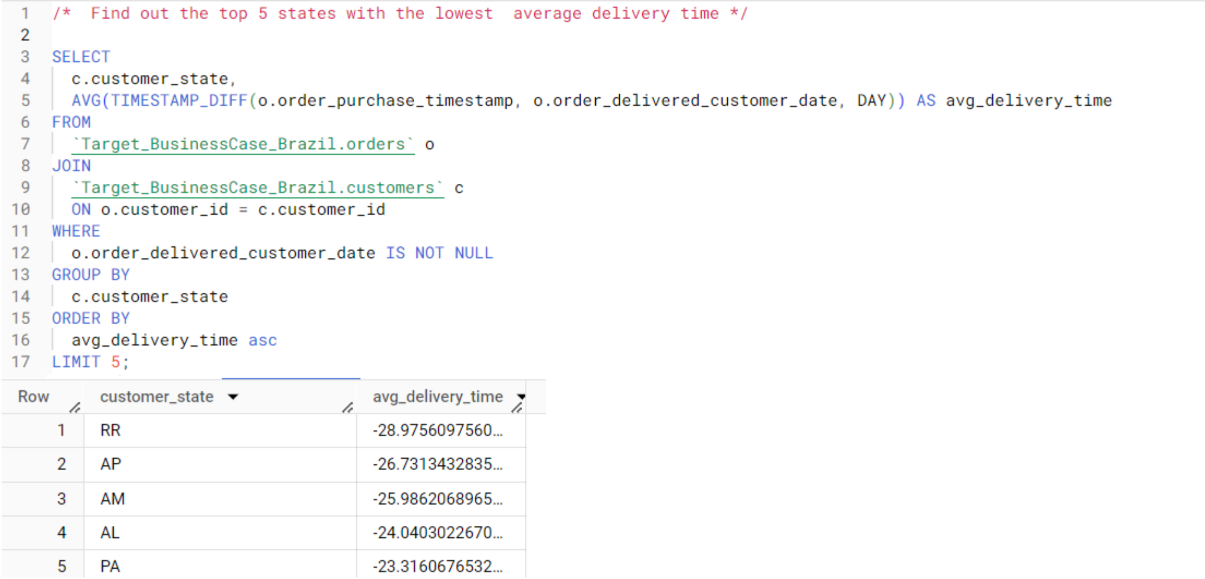
Row	customer_state	avg_freight_value
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04

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



```
/*  Find out the top 5 states with the highest  average delivery time */

SELECT
  c.customer_state,
  AVG(TIMESTAMP_DIFF(o.order_purchase_timestamp, o.order_delivered_customer_date, DAY)) AS avg_delivery_time
FROM
  `Target_BusinessCase_Brazil.orders` o
JOIN
  `Target_BusinessCase_Brazil.customers` c
  ON o.customer_id = c.customer_id
WHERE
  o.order_delivered_customer_date IS NOT NULL
GROUP BY
  c.customer_state
ORDER BY
  avg_delivery_time desc
LIMIT 5;
```

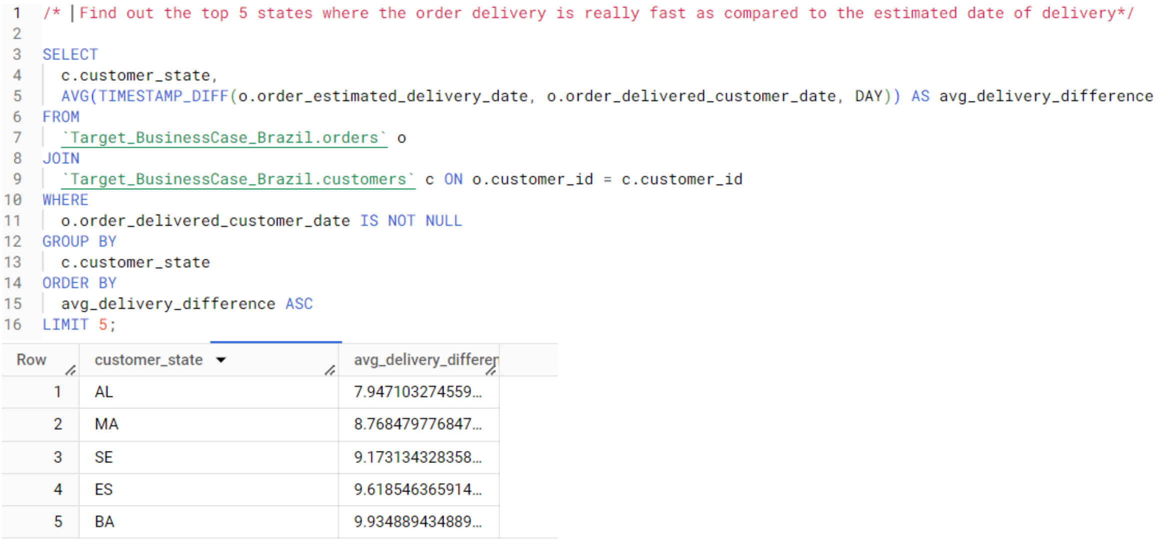
Row	customer_state	avg_delivery_time
1	SP	-8.29806148907...
2	PR	-11.5267113548...
3	MG	-11.5438132981...
4	DF	-12.5091346153...
5	SC	-14.4795601917...



```
/*  Find out the top 5 states with the lowest  average delivery time */

SELECT
  c.customer_state,
  AVG(TIMESTAMP_DIFF(o.order_purchase_timestamp, o.order_delivered_customer_date, DAY)) AS avg_delivery_time
FROM
  `Target_BusinessCase_Brazil.orders` o
JOIN
  `Target_BusinessCase_Brazil.customers` c
  ON o.customer_id = c.customer_id
WHERE
  o.order_delivered_customer_date IS NOT NULL
GROUP BY
  c.customer_state
ORDER BY
  avg_delivery_time asc
LIMIT 5;
```

Row	customer_state	avg_delivery_time
1	RR	-28.9756097560...
2	AP	-26.7313432835...
3	AM	-25.9862068965...
4	AL	-24.0403022670...
5	PA	-23.3160676532...

d. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

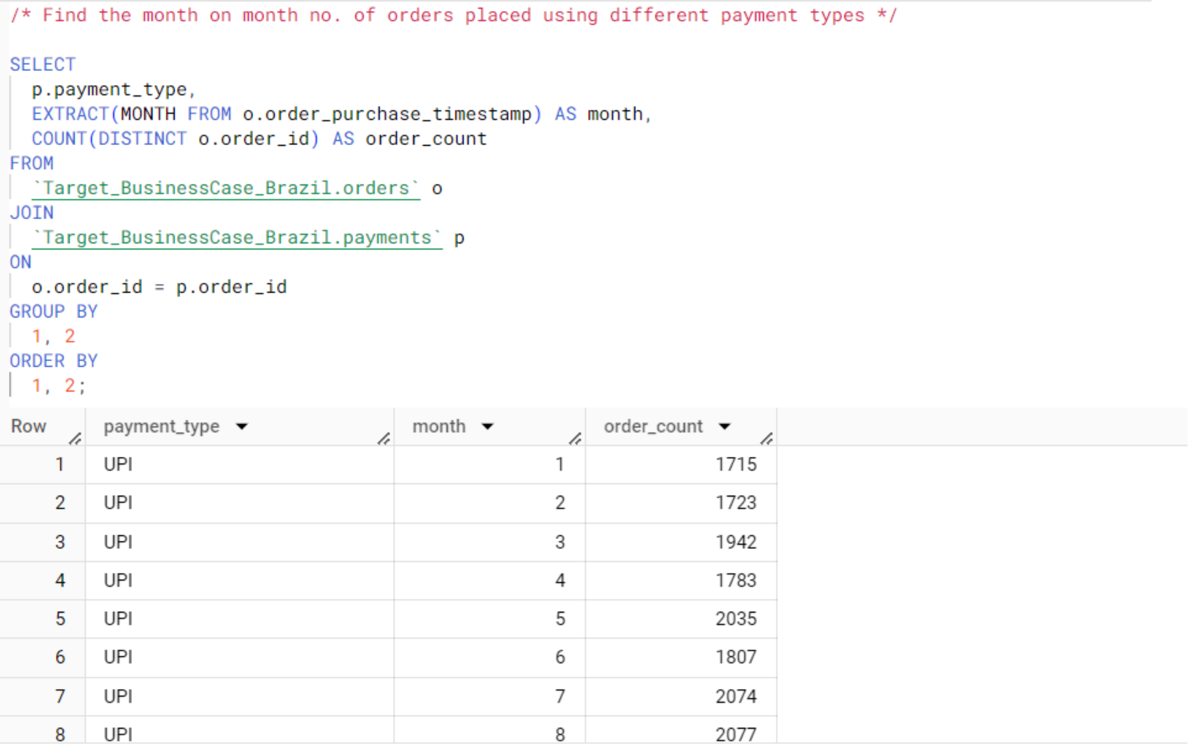


```sql
/* Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery*/

SELECT
  c.customer_state,
  AVG(TIMESTAMP_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY)) AS avg_delivery_difference
FROM
  `Target_BusinessCase_Brazil.orders` o
JOIN
  `Target_BusinessCase_Brazil.customers` c ON o.customer_id = c.customer_id
WHERE
  o.order_delivered_customer_date IS NOT NULL
GROUP BY
  c.customer_state
ORDER BY
  avg_delivery_difference ASC
LIMIT 5;
```

Row	customer_state	avg_delivery_differen
1	AL	7.947103274559...
2	MA	8.768479776847...
3	SE	9.173134328358...
4	ES	9.618546365914...
5	BA	9.934889434889...

6. Analysis based on the payments:

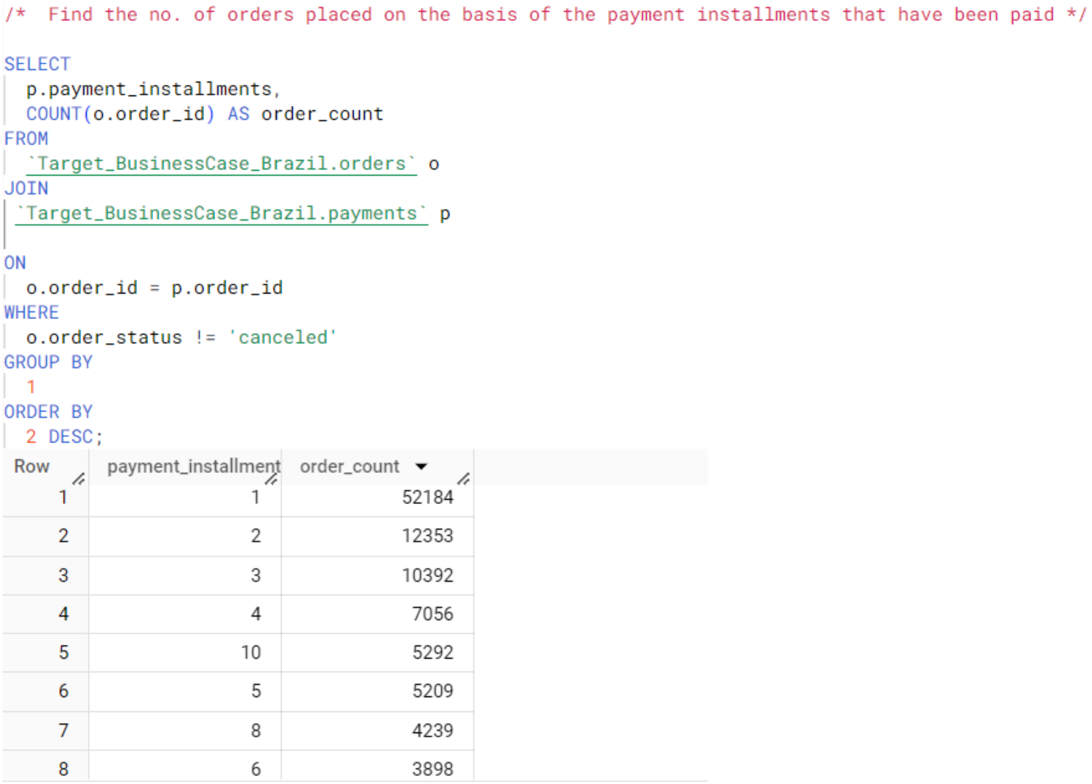
a. Find the month on month no. of orders placed using different payment types.



```
/* Find the month on month no. of orders placed using different payment types */

SELECT
  p.payment_type,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
  COUNT(DISTINCT o.order_id) AS order_count
FROM
  `Target_BusinessCase_Brazil.orders` o
JOIN
  `Target_BusinessCase_Brazil.payments` p
ON
  o.order_id = p.order_id
GROUP BY
  1, 2
ORDER BY
  1, 2;
```

Row	payment_type	month	order_count
1	UPI	1	1715
2	UPI	2	1723
3	UPI	3	1942
4	UPI	4	1783
5	UPI	5	2035
6	UPI	6	1807
7	UPI	7	2074
8	UPI	8	2077

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



```
/*  Find the no. of orders placed on the basis of the payment installments that have been paid */

SELECT
  p.payment_installments,
  COUNT(o.order_id) AS order_count
FROM
  `Target_BusinessCase_Brazil.orders` o
JOIN
 `Target_BusinessCase_Brazil.payments` p

ON
  o.order_id = p.order_id
WHERE
  o.order_status != 'canceled'
GROUP BY
  1
ORDER BY
  2 DESC;
```

Row	payment_installment	order_count
1	1	52184
2	2	12353
3	3	10392
4	4	7056
5	10	5292
6	5	5209
7	8	4239
8	6	3898