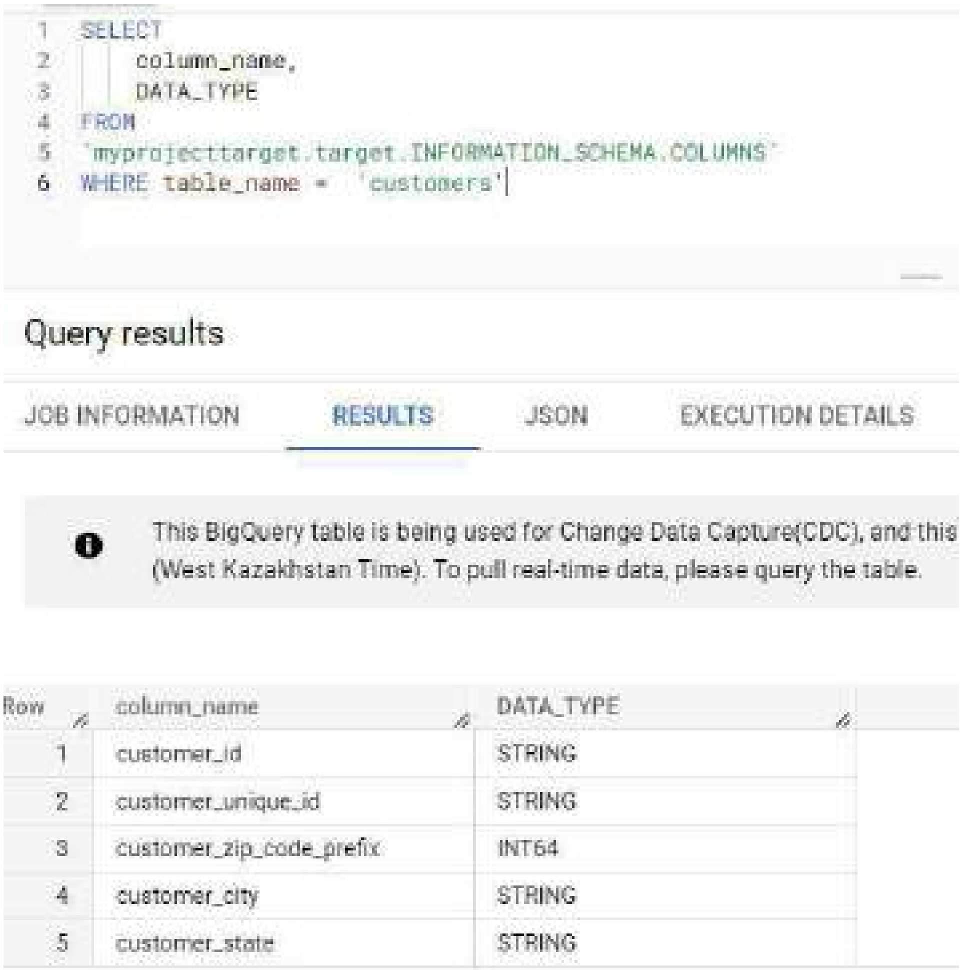
Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset

Data type of columns in a table

'customers'

```
SELECT
    column_name,
    DATA_TYPE
FROM
`myprojecttarget.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name =  'customers'
```



```
SELECT
    column_name,
    DATA_TYPE
FROM
`myprojecttarget.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name =  'customers'
```

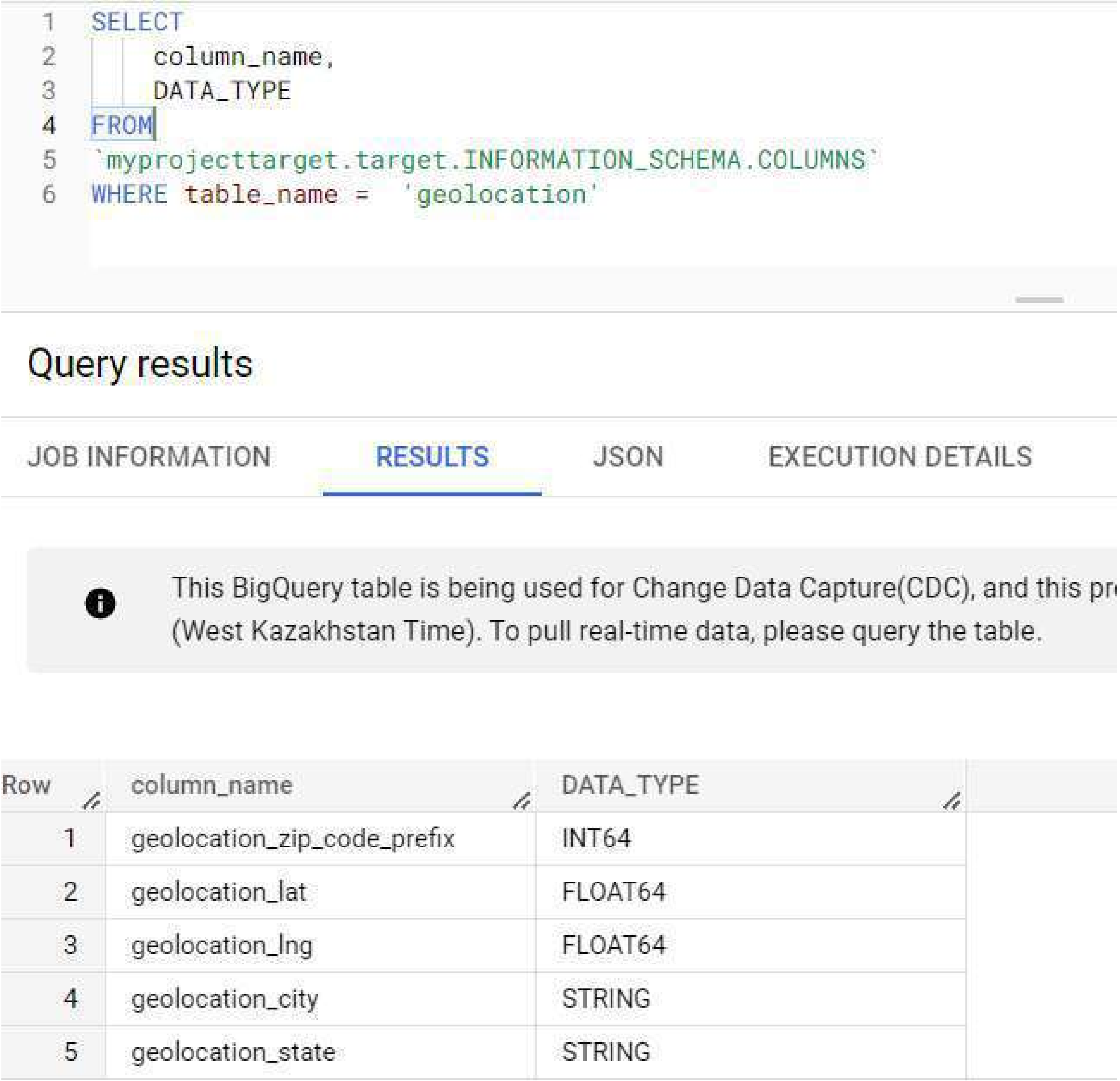
Query results

This BigQuery table is being used for Change Data Capture(CDC), and this (West Kazakhstan Time). To pull real-time data, please query the table.

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

'geolocation'

```
SELECT
    column_name,
    DATA_TYPE
FROM
`myprojecttarget.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name =  'geolocation'
```



```
SELECT
      column_name,
      DATA_TYPE
FROM
`myprojecttarget.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name =  'geolocation'
```

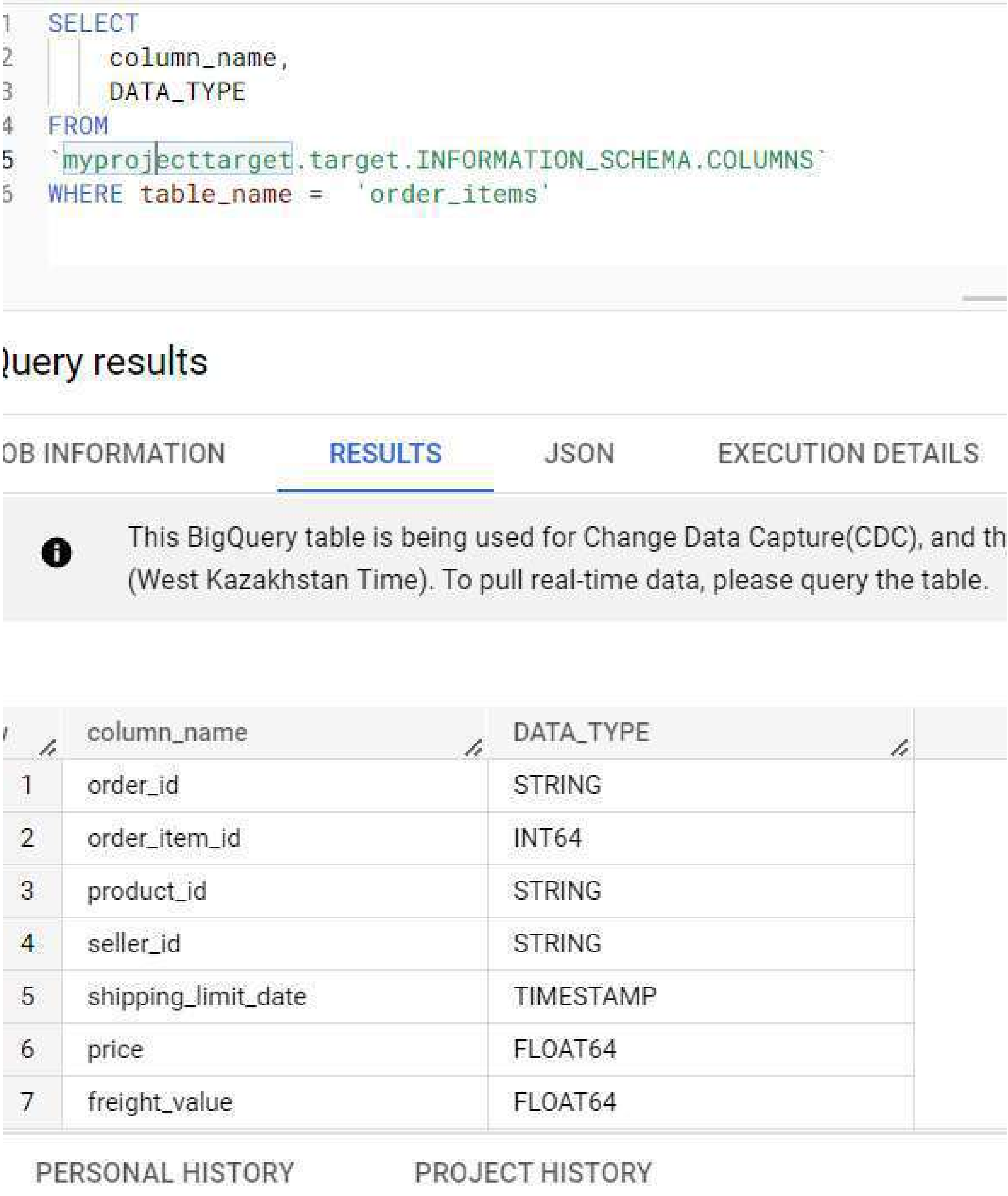
Query results

This BigQuery table is being used for Change Data Capture(CDC), and this pr
(West Kazakhstan Time). To pull real-time data, please query the table.

Row	column_name	DATA_TYPE
1	geolocation_zip_code_prefix	INT64
2	geolocation_lat	FLOAT64
3	geolocation_lng	FLOAT64
4	geolocation_city	STRING
5	geolocation_state	STRING

order_items

```
SELECT
    column_name,
    DATA_TYPE
FROM
`myprojecttarget.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name =  'order_items'
```



```
SELECT
    column_name,
    DATA_TYPE
FROM
`myprojecttarget.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name =  'order_items'
```

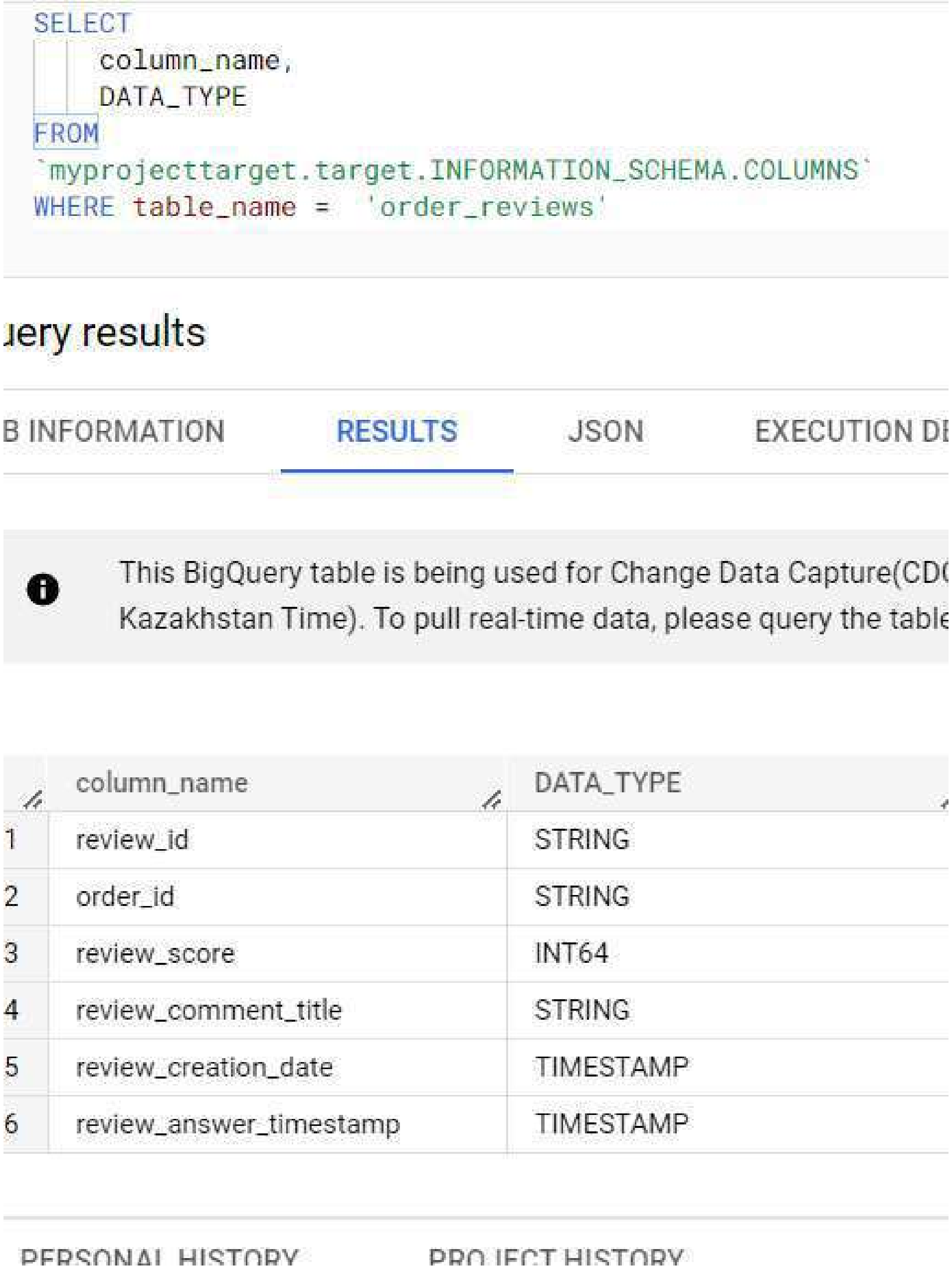
)uery results

This BigQuery table is being used for Change Data Capture(CDC), and th
(West Kazakhstan Time). To pull real-time data, please query the table.

/	column_name	DATA_TYPE
1	order_id	STRING
2	order_item_id	INT64
3	product_id	STRING
4	seller_id	STRING
5	shipping_limit_date	TIMESTAMP
6	price	FLOAT64
7	freight_value	FLOAT64

order_reviews

```
SELECT
   column_name,
   DATA_TYPE
FROM
`myprojecttarget.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name =  'order_reviews'
```



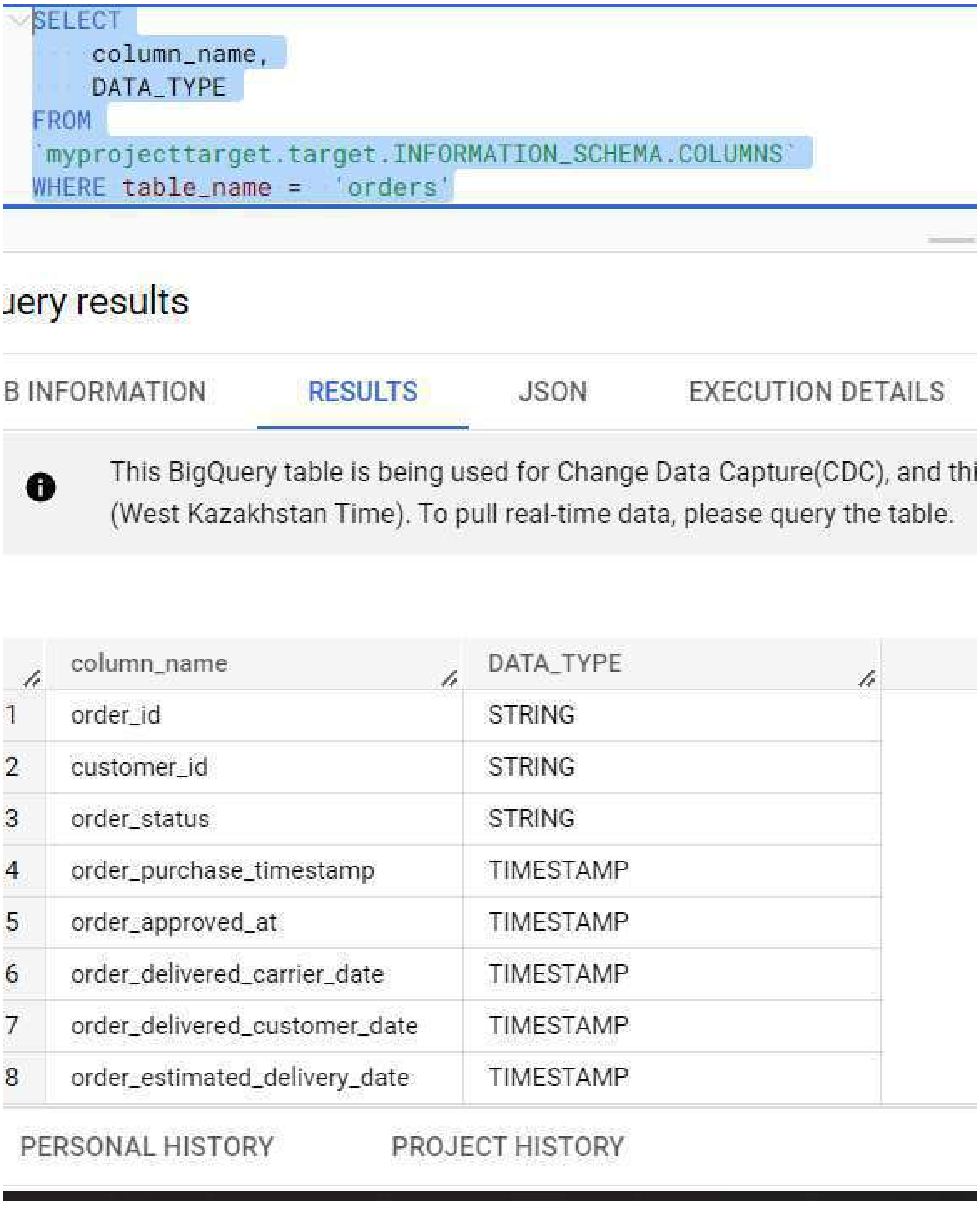
```
SELECT
    column_name,
    DATA_TYPE
FROM
`myprojecttarget.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name =  'order_reviews'
```

This BigQuery table is being used for Change Data Capture(CDC
Kazakhstan Time). To pull real-time data, please query the table

	column_name	DATA_TYPE
1	review_id	STRING
2	order_id	STRING
3	review_score	INT64
4	review_comment_title	STRING
5	review_creation_date	TIMESTAMP
6	review_answer_timestamp	TIMESTAMP

'orders'

```
SELECT
    column_name,
    DATA_TYPE
FROM
`myprojecttarget.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name =  'orders'
```



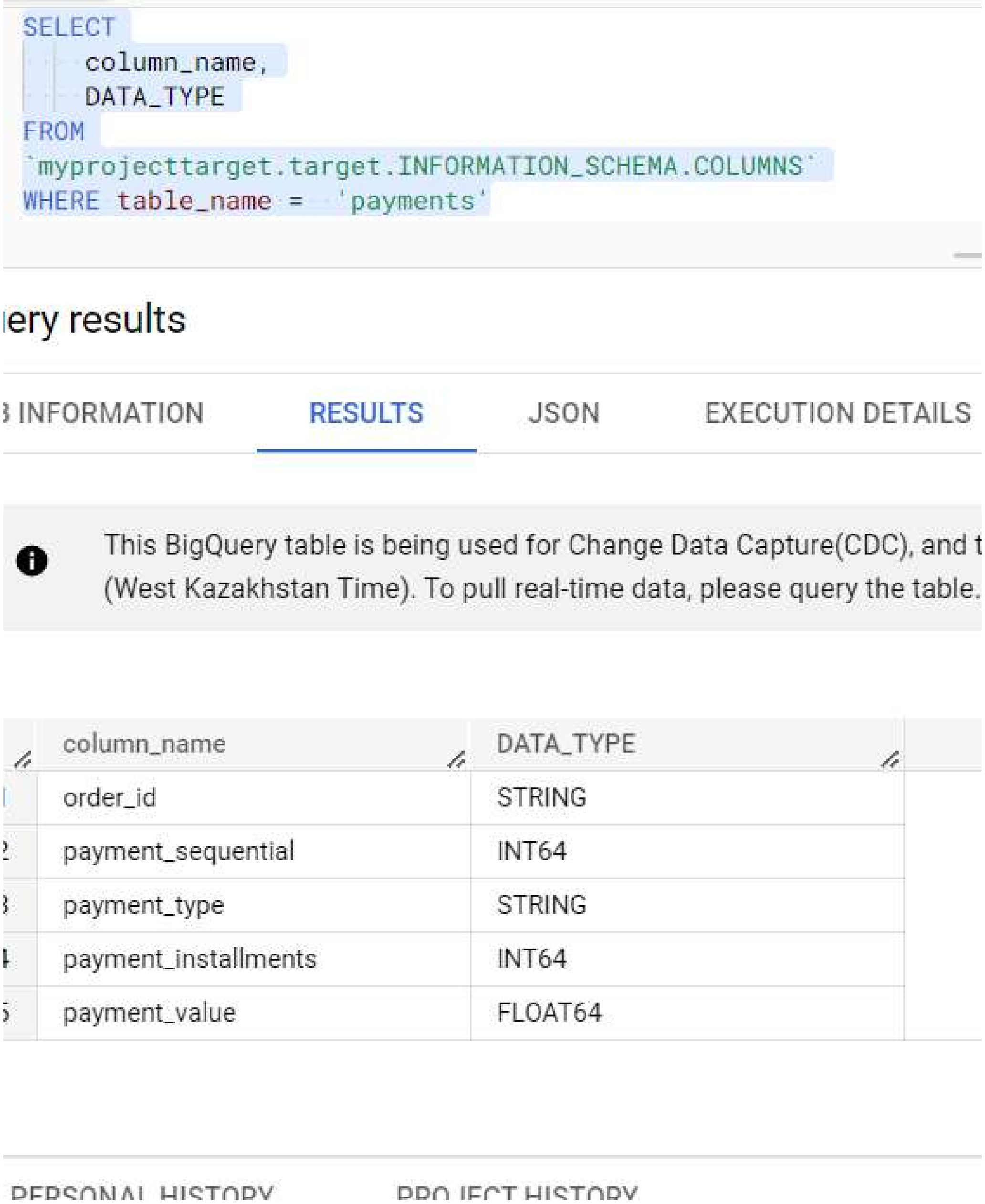
```
SELECT
    column_name,
    DATA_TYPE
FROM
`myprojecttarget.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name =  'orders'
```

This BigQuery table is being used for Change Data Capture(CDC), and thi (West Kazakhstan Time). To pull real-time data, please query the table.

	column_name	DATA_TYPE
1	order_id	STRING
2	customer_id	STRING
3	order_status	STRING
4	order_purchase_timestamp	TIMESTAMP
5	order_approved_at	TIMESTAMP
6	order_delivered_carrier_date	TIMESTAMP
7	order_delivered_customer_date	TIMESTAMP
8	order_estimated_delivery_date	TIMESTAMP

payments

```
SELECT
    column_name,
    DATA_TYPE
FROM
`myprojecttarget.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name =  'payments'
```



```
SELECT
    column_name,
    DATA_TYPE
FROM
`myprojecttarget.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name =  'payments'
```

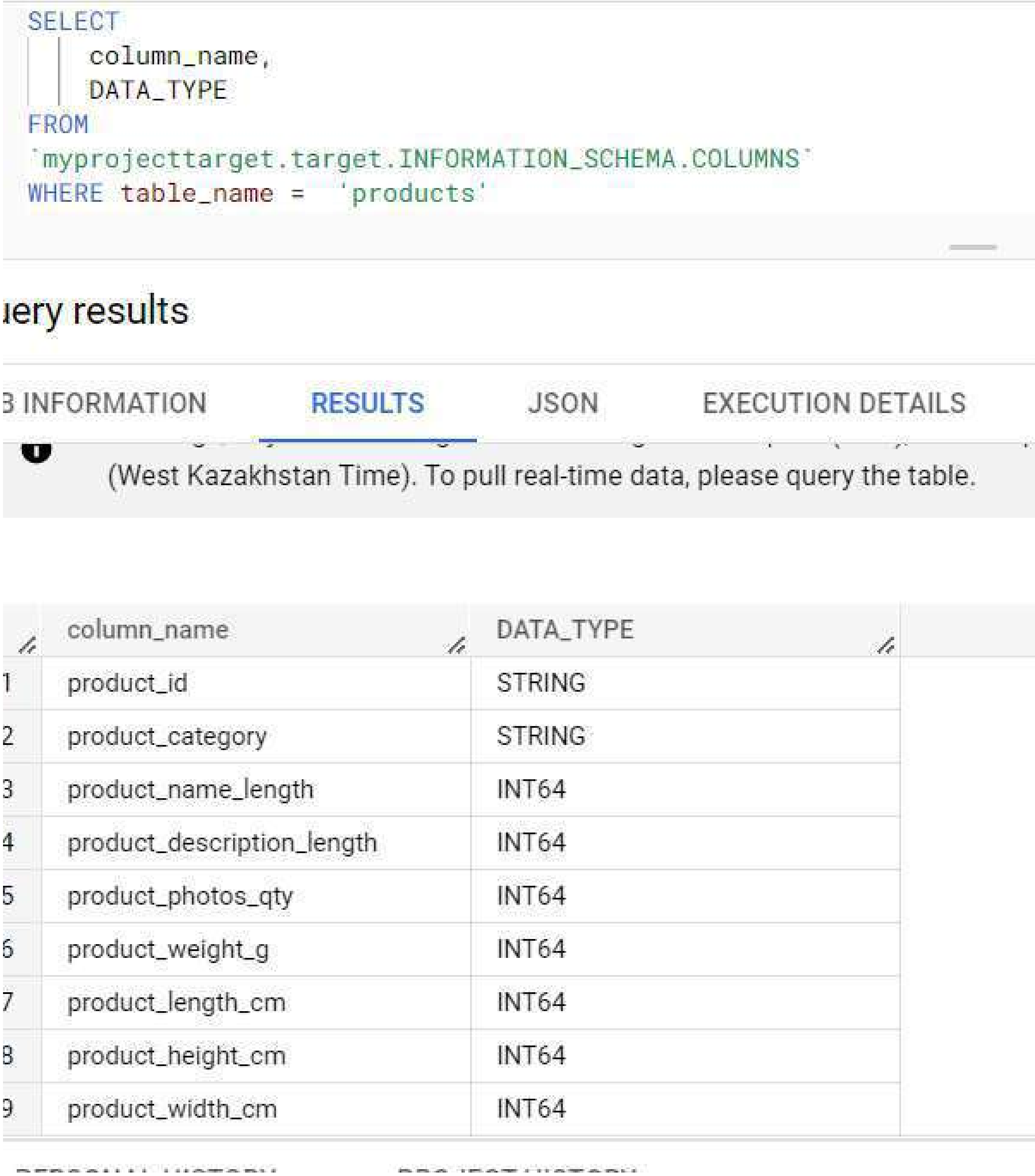
ery results

This BigQuery table is being used for Change Data Capture(CDC), and t (West Kazakhstan Time). To pull real-time data, please query the table.

	column_name	DATA_TYPE
1	order_id	STRING
2	payment_sequential	INT64
3	payment_type	STRING
4	payment_installments	INT64
5	payment_value	FLOAT64

products

```
SELECT
    column_name,
    DATA_TYPE
FROM
`myprojecttarget.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name =  'products'
```



```
SELECT
    column_name,
    DATA_TYPE
FROM
`myprojecttarget.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name =  'products'
```

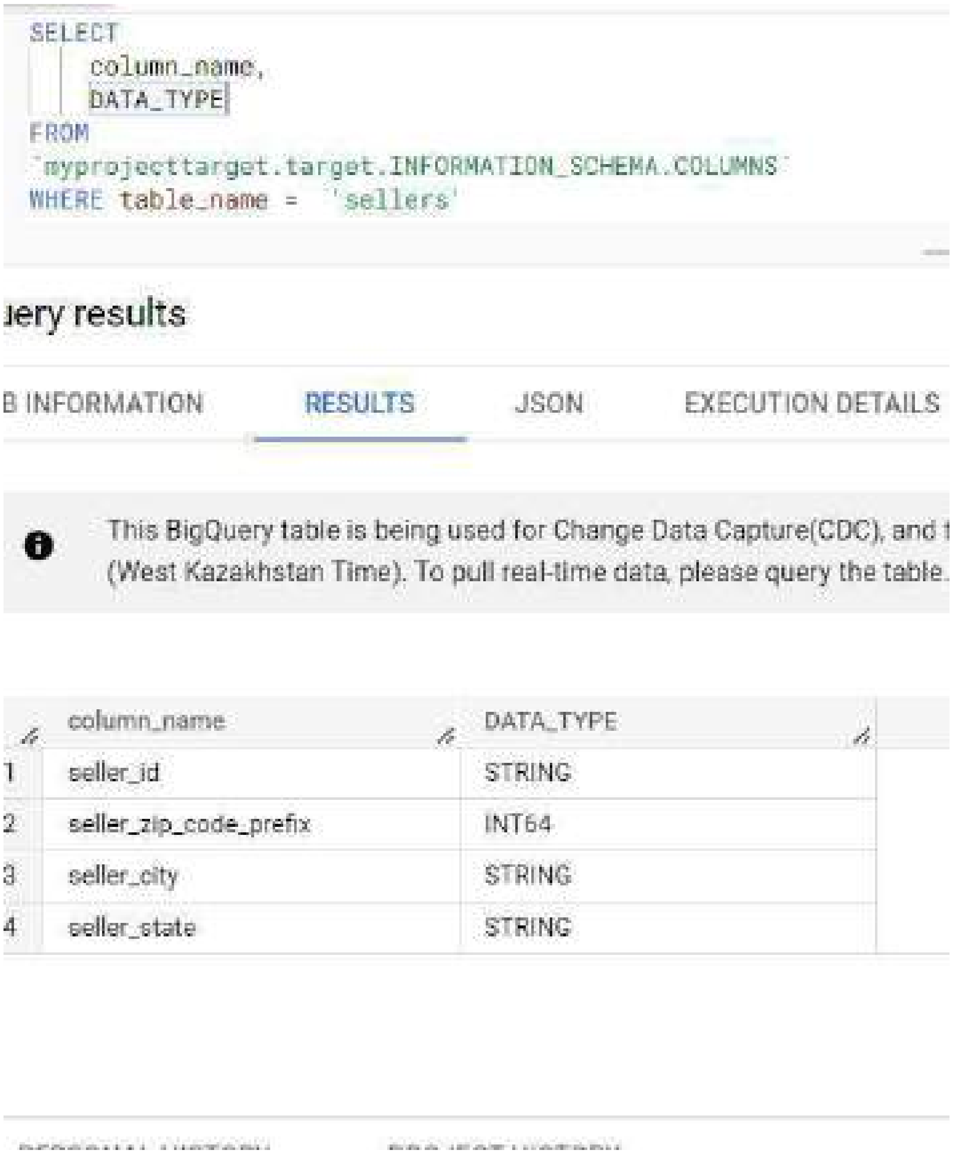
ıery results

(West Kazakhstan Time). To pull real-time data, please query the table.

	column_name	DATA_TYPE
1	product_id	STRING
2	product_category	STRING
3	product_name_length	INT64
4	product_description_length	INT64
5	product_photos_qty	INT64
6	product_weight_g	INT64
7	product_length_cm	INT64
8	product_height_cm	INT64
9	product_width_cm	INT64

sellers

```
SELECT
    column_name,
    DATA_TYPE
FROM
`myprojecttarget.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name =  'sellers'
```



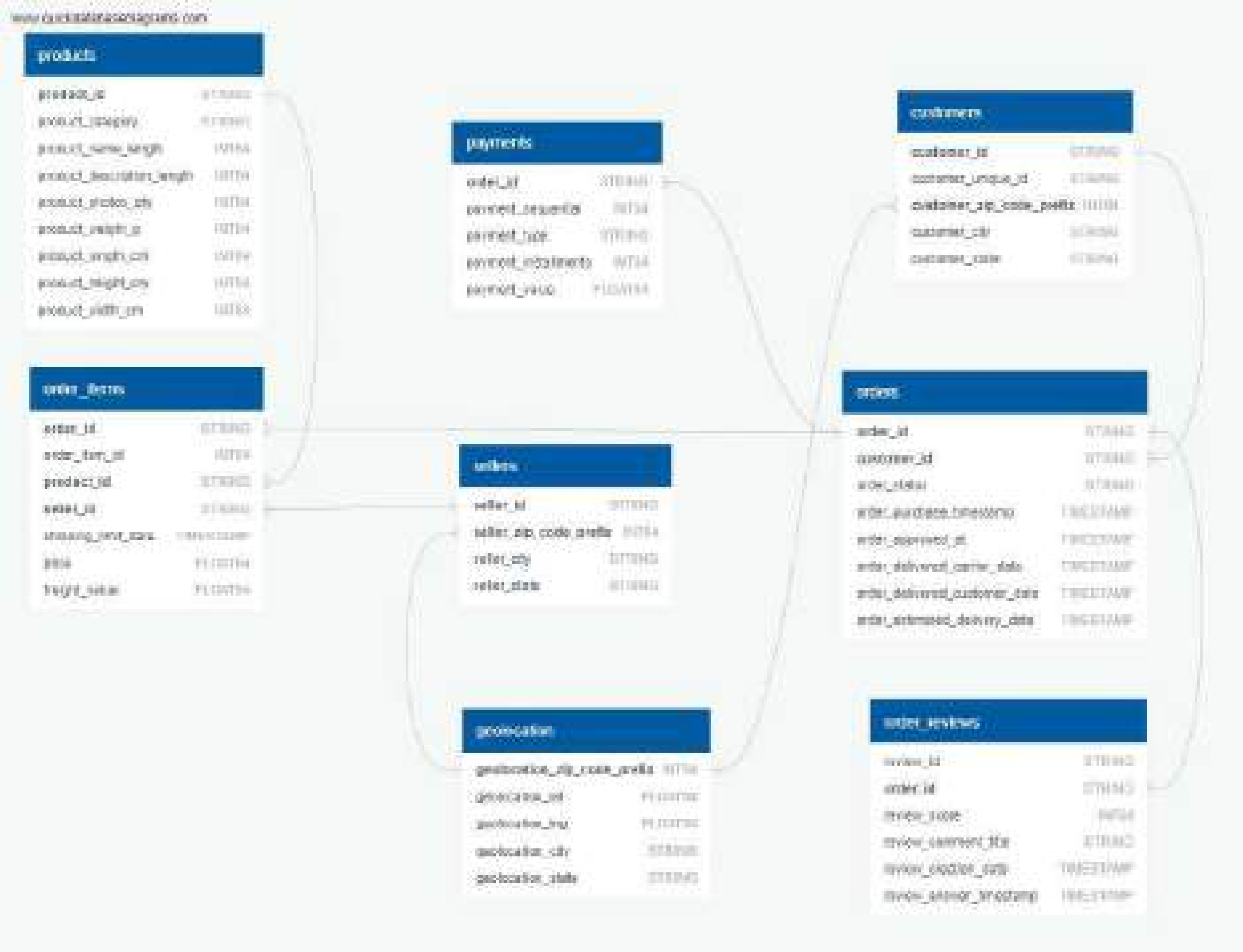
```
SELECT
    column_name,
    DATA_TYPE
FROM
`myprojecttarget.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name =  'sellers'
```

uery results

This BigQuery table is being used for Change Data Capture(CDC), and t
(West Kazakhstan Time). To pull real-time data, please query the table.

	column_name	DATA_TYPE
1	seller_id	STRING
2	seller_zip_code_prefix	INT64
3	seller_city	STRING
4	seller_state	STRING

ERD:

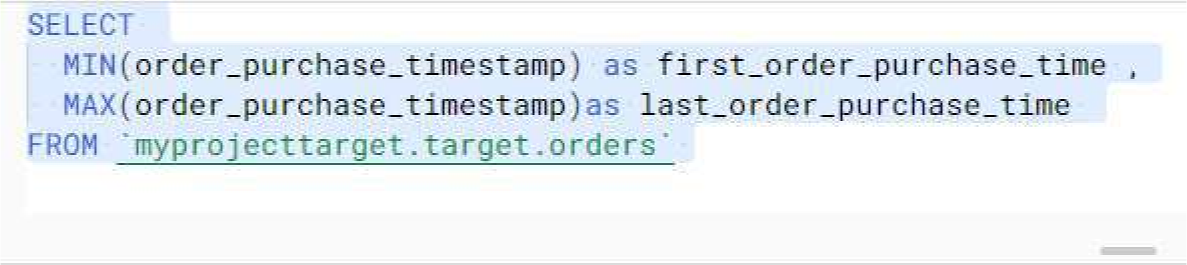


Actionable Insights:

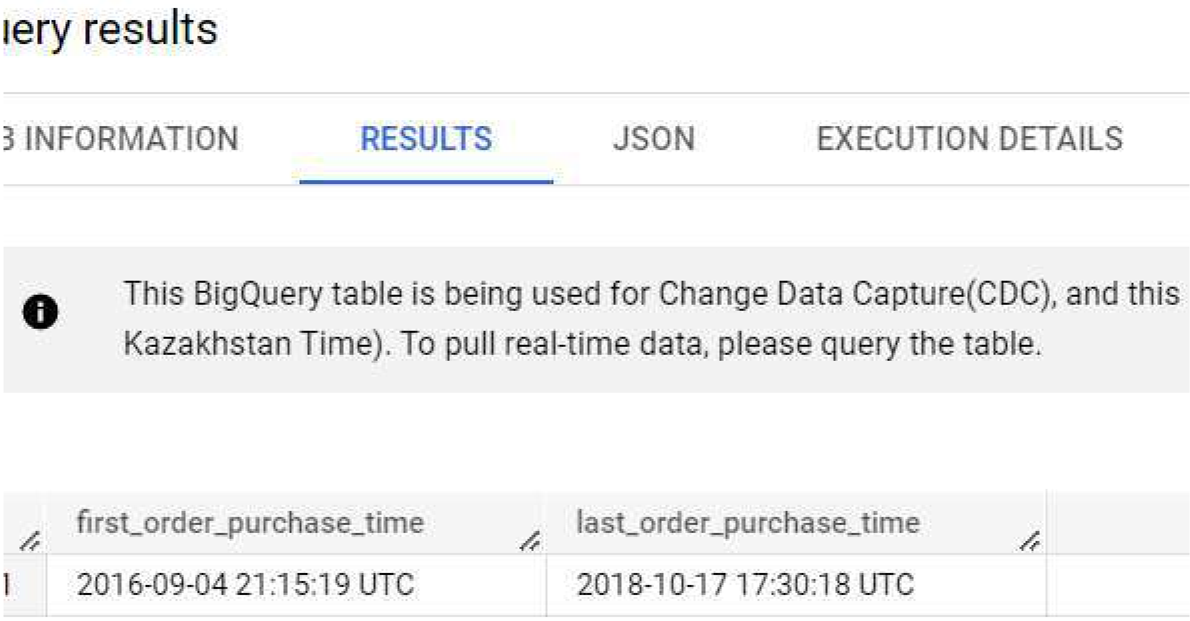
Table	**Interacting with tables**
products	order_items
order_items	Orders, products, sellers
payments	Orders
sellers	order_items, geolocation
geolocation	Sellers, customers
customers	Orders, geolocation
Orders	Customers, order_reviews, payments, order_items
order_reviews	Orders

Time period for which the data is given

```
SELECT
  MIN(order_purchase_timestamp) as first_order_purchase_time ,
  MAX(order_purchase_timestamp)as last_order_purchase_time
FROM `myprojecttarget.target.orders`
```



```
SELECT
  MIN(order_purchase_timestamp) as first_order_purchase_time ,
  MAX(order_purchase_timestamp)as last_order_purchase_time
FROM `myprojecttarget.target.orders`
```



This BigQuery table is being used for Change Data Capture(CDC), and this Kazakhstan Time). To pull real-time data, please query the table.

	first_order_purchase_time	last_order_purchase_time
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Conclusion / Actionable Insights: as a result, Time period for which the data is given is concluded as below

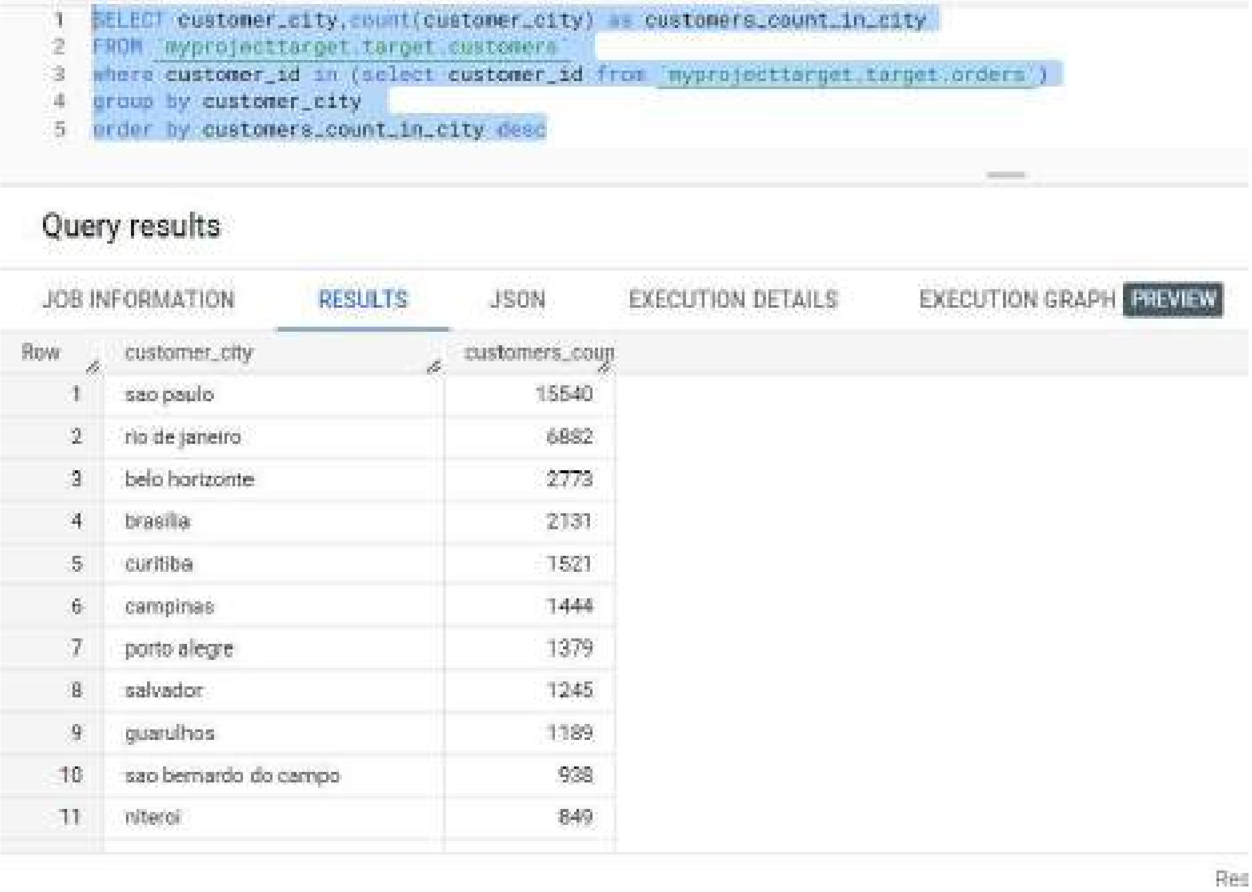
First order was on : 2016-09-04 21:15:19.000000 UTC

Last order was on : 2018-10-17 17:30:18.000000 UTC

Cities and States of customers ordered during the given period

Count of number of customers in Cities during the given period

```
SELECT customer_city,count(customer_city) as customers_count_in_city
FROM `myprojecttarget.target.customers`
where customer_id in (select customer_id from `myprojecttarget.target.orders`)
group by customer_city
order by customers_count_in_city desc
```



```
SELECT customer_city,count(customer_city) as customers_count_in_city
FROM `myprojecttarget.target.customers`
where customer_id in (select customer_id from `myprojecttarget.target.orders`)
group by customer_city
order by customers_count_in_city desc
```

Query results

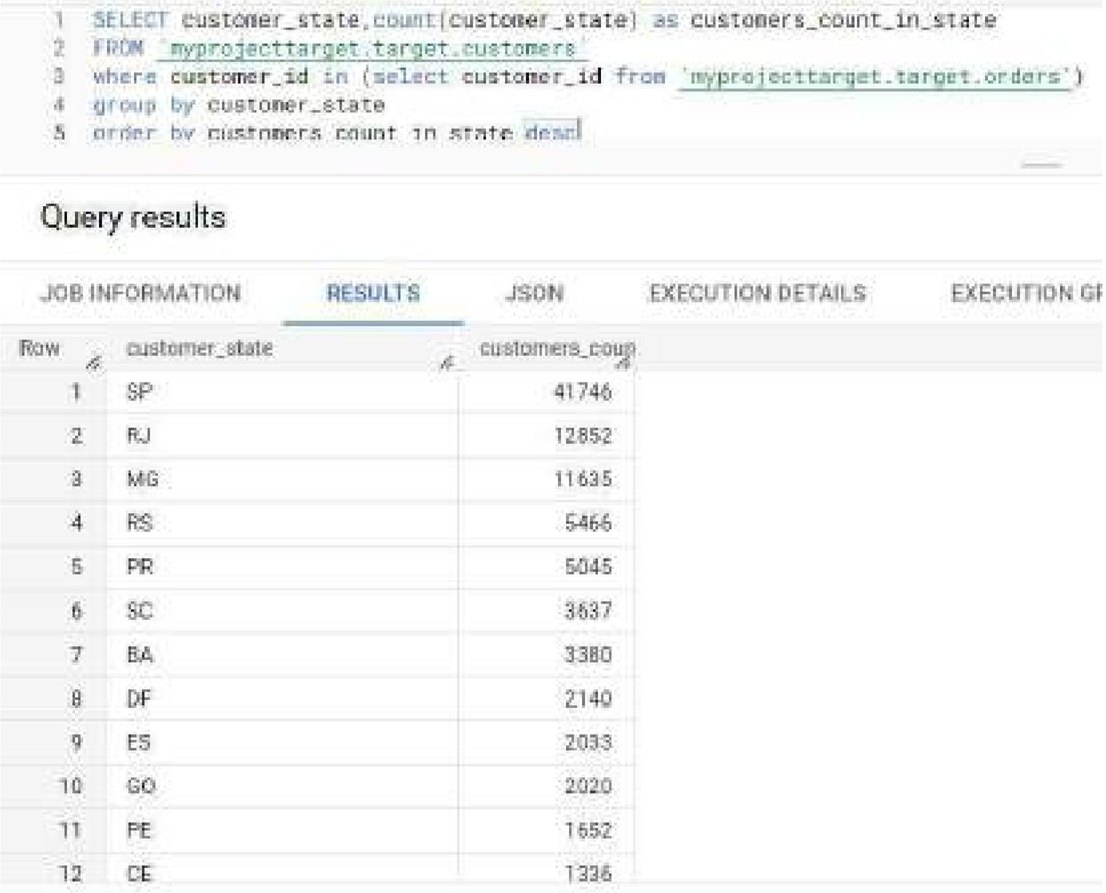
Row	customer_city	customers_coun
1	sao paulo	15540
2	rio de janeiro	6882
3	belo horizonte	2773
4	brasilia	2131
5	curitiba	1521
6	campinas	1444
7	porto alegre	1379
8	salvador	1245
9	guarulhos	1189
10	sao bernardo do campo	938
11	niteroi	849

Actionable Insights:

sao paulo is having highest number of customers and avai and others are having similar lowest number of customers

Count of number of customers in States during the given period

```
SELECT customer_state,count(customer_state) as customers_count_in_state
FROM `myprojecttarget.target.customers`
where customer_id in (select customer_id from `myprojecttarget.target.orders`)
group by customer_state
order by customers_count_in_state desc
```



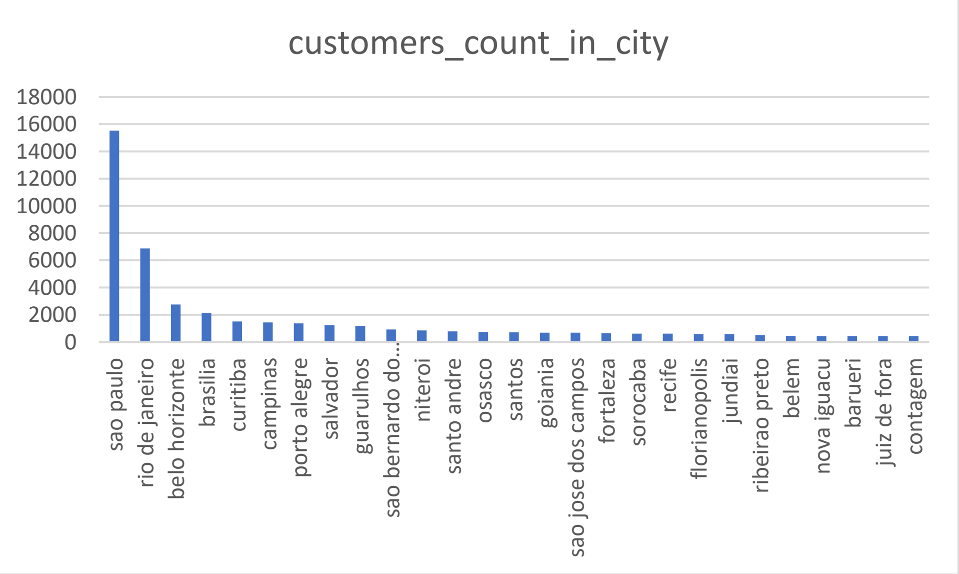
```
SELECT customer_state,count(customer_state) as customers_count_in_state
FROM `myprojecttarget.target.customers`
where customer_id in (select customer_id from `myprojecttarget.target.orders`)
group by customer_state
order by customers_count_in_state desc
```

Query results

Row	customer_state	customers_coun
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020
11	PE	1652
12	CE	1336

Actionable Insights:

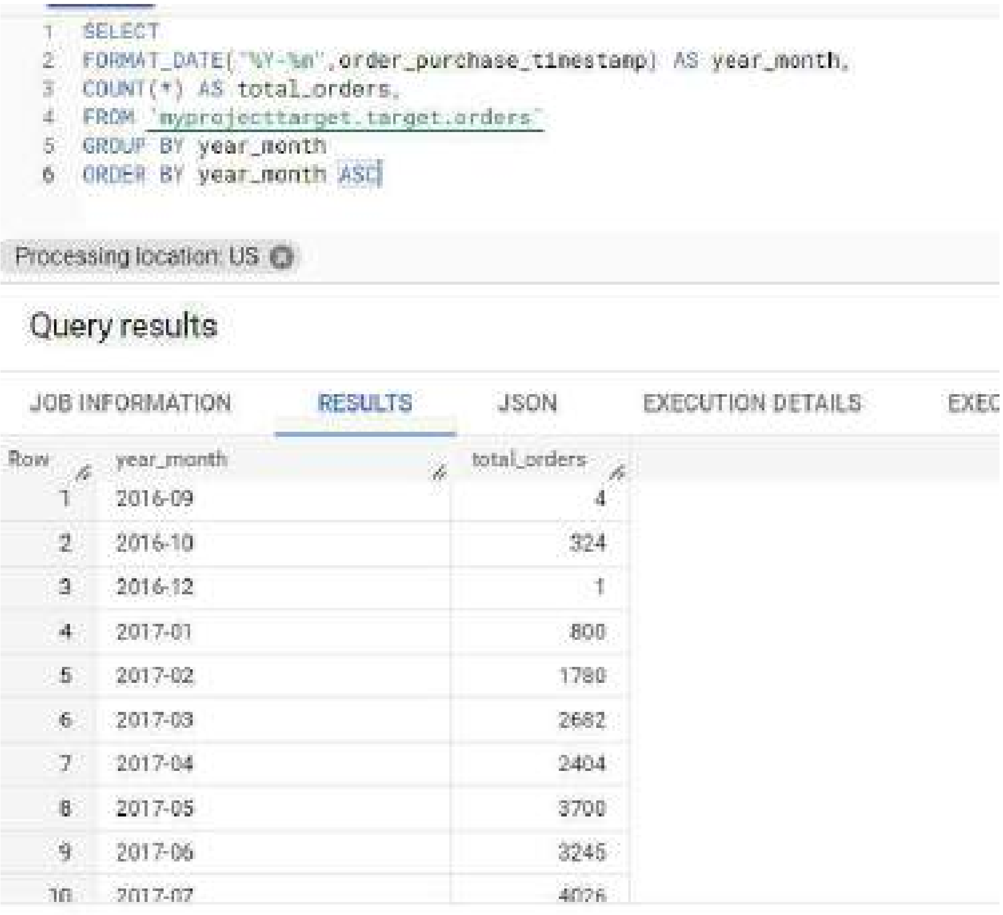
SP is having highest of customers and RR is the lowest among the data present



In-depth Exploration:

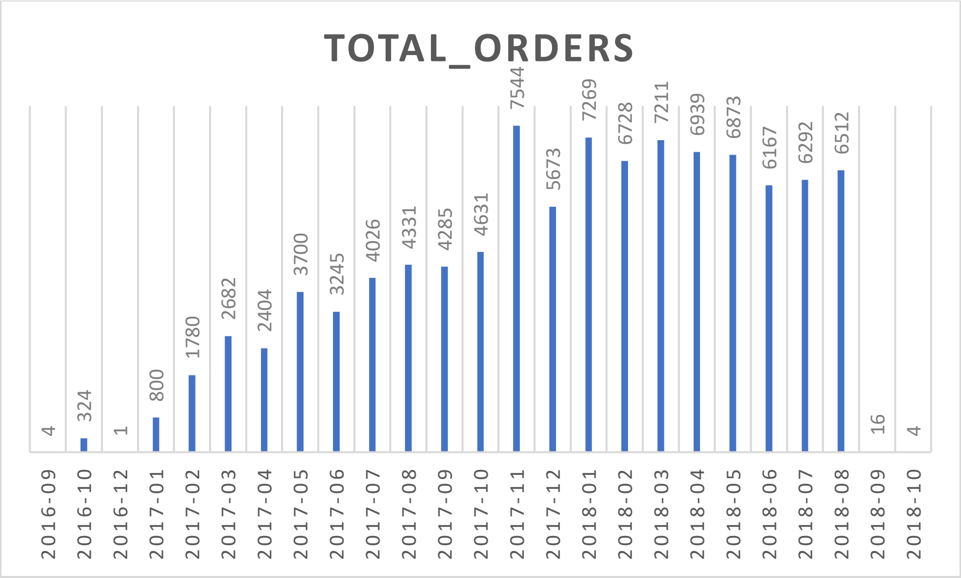
Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

```
SELECT
FORMAT_DATE(“%Y-%m”,order_purchase_timestamp) AS year_month,
COUNT(*) AS total_orders,
FROM `myprojecttarget.target.orders`
GROUP BY year_month
ORDER BY year_month ASC
```



```
SELECT
FORMAT_DATE("%Y-%m",order_purchase_timestamp) AS year_month,
COUNT(*) AS total_orders,
FROM `myprojecttarget.target.orders`
GROUP BY year_month
ORDER BY year_month ASC
```

Row	year_month	total_orders
1	2016-09	4
2	2016-10	324
3	2016-12	1
4	2017-01	800
5	2017-02	1780
6	2017-03	2682
7	2017-04	2404
8	2017-05	3700
9	2017-06	3245
10	2017-07	4026



Is there a growing trend on e-commerce in Brazil?

Growing trend is observed till 2018 august, post that its huge down fall.

How can we describe a complete scenario?

For period 2016-09 to 2016-12 :

October month have highest orders for 2016

For period 2017-01 to 2017-12 :

November have highest orders for 2017.

July to October maintains minimum 4026 order count.

For period 2018-01 to 2018-10 :

January and March have highest orders for 2018

2018 maintains order count minimum as 6167 from January till August.

Huge down fall in September and October

Can we see some seasonality with peaks at specific months?

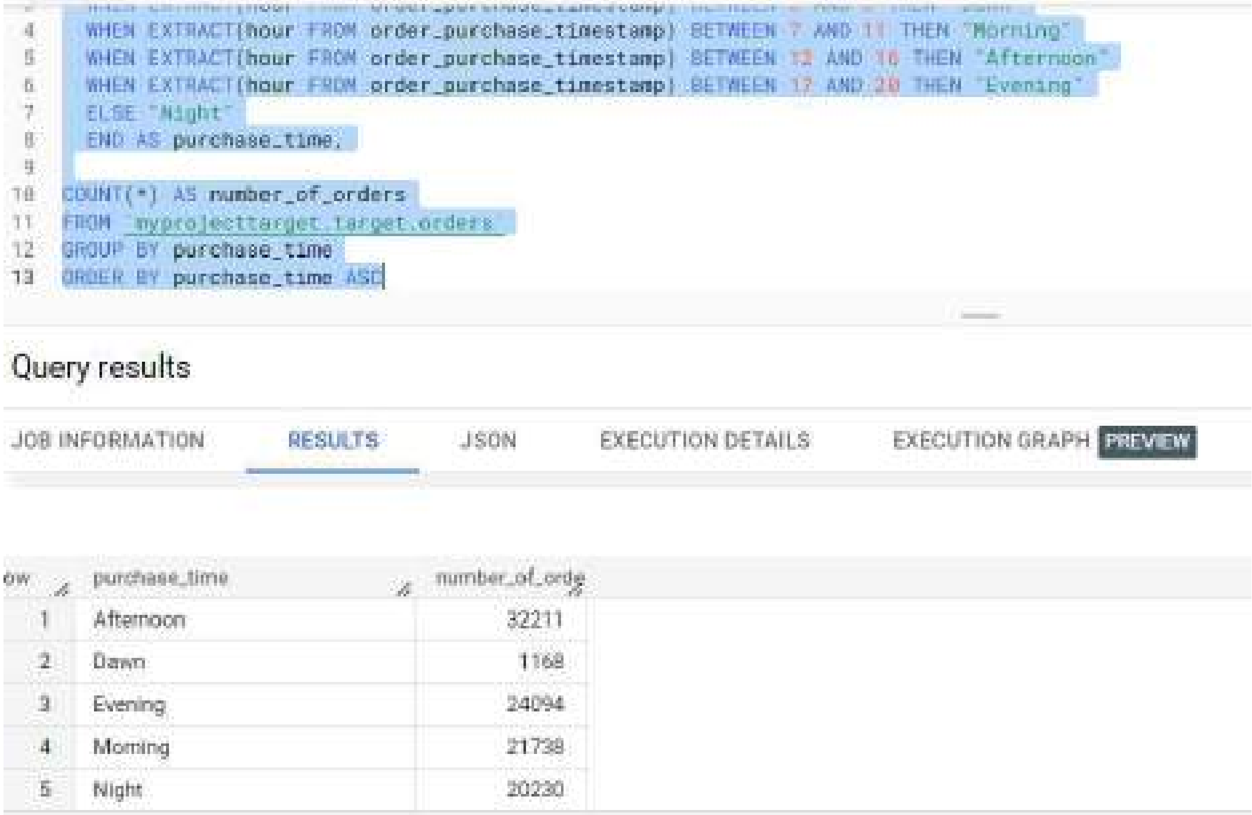
2017 Nov, 2018 January can observe peaks of count of orders

Actionable Insights:
Growing trend is observed till 2018 august, post that its huge down fall.
2017 November have highest orders for the provided range

Huge down fall in 2018 September and October

What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

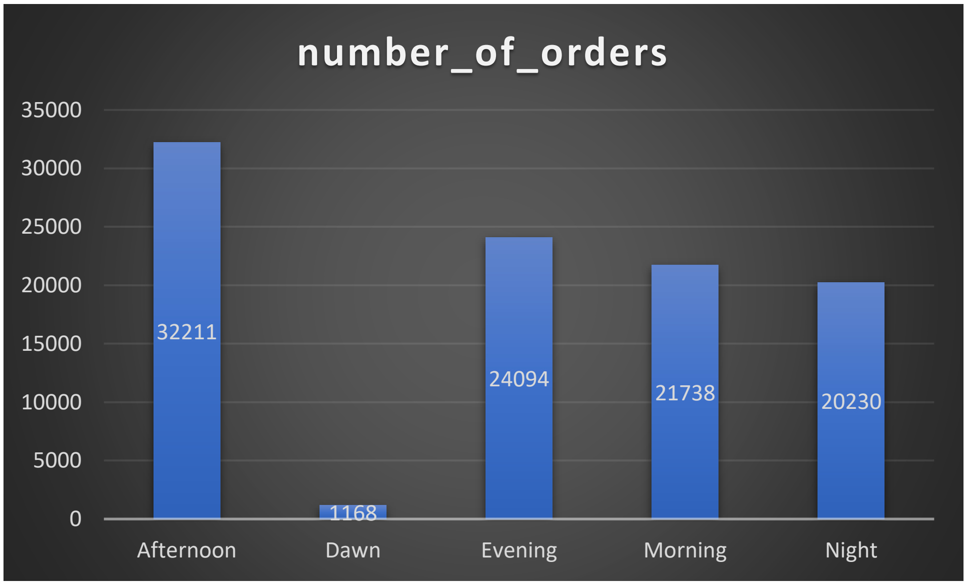
```
SELECT

  CASE
  WHEN EXTRACT(hour FROM order_purchase_timestamp) BETWEEN 3 AND 6 THEN “Dawn”
  WHEN EXTRACT(hour FROM order_purchase_timestamp) BETWEEN 7 AND 11 THEN “Morning”
  WHEN EXTRACT(hour FROM order_purchase_timestamp) BETWEEN 12 AND 16 THEN “Afternoon”
  WHEN EXTRACT(hour FROM order_purchase_timestamp) BETWEEN 17 AND 20 THEN “Evening”
  ELSE “Night”
  END AS purchase_time,

COUNT(*) AS number_of_orders
FROM `myprojecttarget.target.orders`
GROUP BY purchase_time
ORDER BY purchase_time ASC
```



```
  WHEN EXTRACT(hour FROM order_purchase_timestamp) BETWEEN 7 AND 11 THEN "Morning"
  WHEN EXTRACT(hour FROM order_purchase_timestamp) BETWEEN 12 AND 16 THEN "Afternoon"
  WHEN EXTRACT(hour FROM order_purchase_timestamp) BETWEEN 17 AND 20 THEN "Evening"
  ELSE "Night"
  END AS purchase_time;

COUNT(*) AS number_of_orders
FROM `myprojecttarget.target.orders`
GROUP BY purchase_time
ORDER BY purchase_time ASC
```

Query results

ow	purchase_time	number_of_orde
1	Afternoon	32211
2	Dawn	1168
3	Evening	24094
4	Morning	21738
5	Night	20230



Actionable Insights:

Maximum number of orders seen during Afternoon

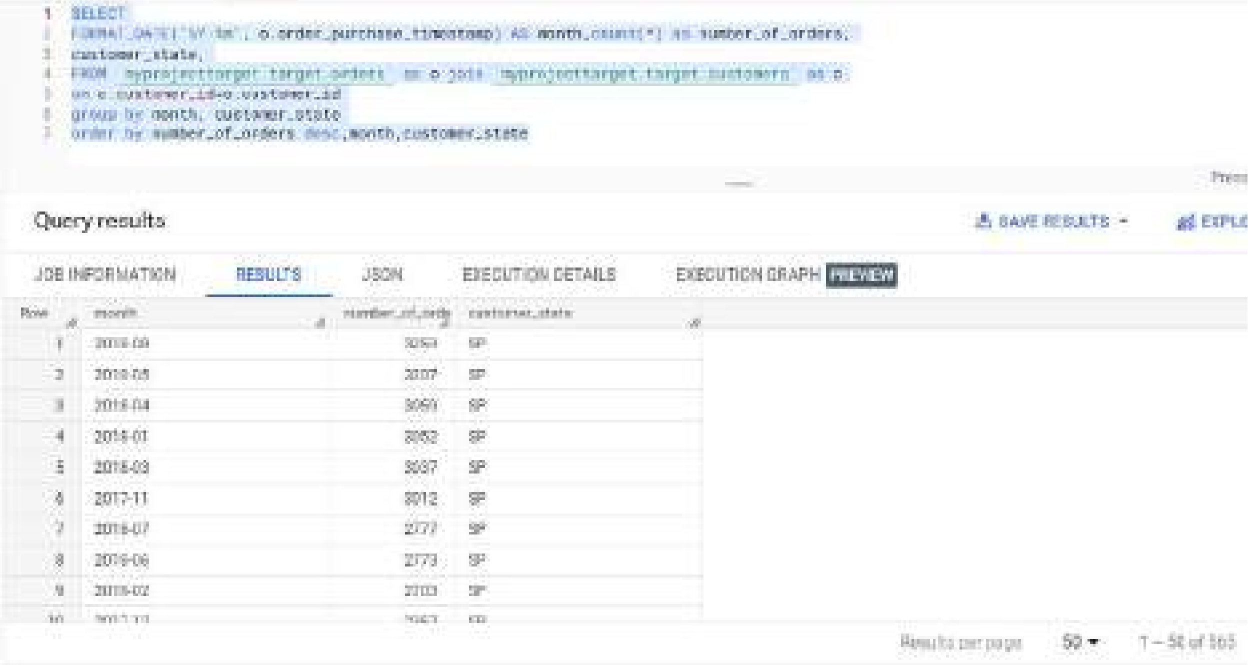
Order count is maintained minimum of 20230 during Evening / Morning / Night.

Lowest number of orders seen during Dwan

Evolution of E-commerce orders in the Brazil region:

Get month on month orders by states

```
SELECT
FORMAT_DATE("%Y-%m", o.order_purchase_timestamp) AS month,count(*) as number_of_orders,
customer_state,
FROM `myprojecttarget.target.orders` as o join `myprojecttarget.target.customers` as c
on c.customer_id=o.customer_id
group by month, customer_state
order by number_of_orders desc,month,customer_state
```



```
SELECT
FORMAT_DATE("%Y-%m", o.order_purchase_timestamp) AS month,count(*) as number_of_orders,
customer_state,
FROM `myprojecttarget.target.orders` as o join `myprojecttarget.target.customers` as c
on c.customer_id=o.customer_id
group by month, customer_state
order by number_of_orders desc,month,customer_state
```

Row	month	number_of_orde	customer_state
1	2018-08	3253	SP
2	2018-05	3207	SP
3	2018-04	3059	SP
4	2018-01	3052	SP
5	2018-03	3037	SP
6	2017-11	3012	SP
7	2018-07	2777	SP
8	2018-06	2773	SP
9	2018-02	2703	SP
10	2017-12	2357	SP

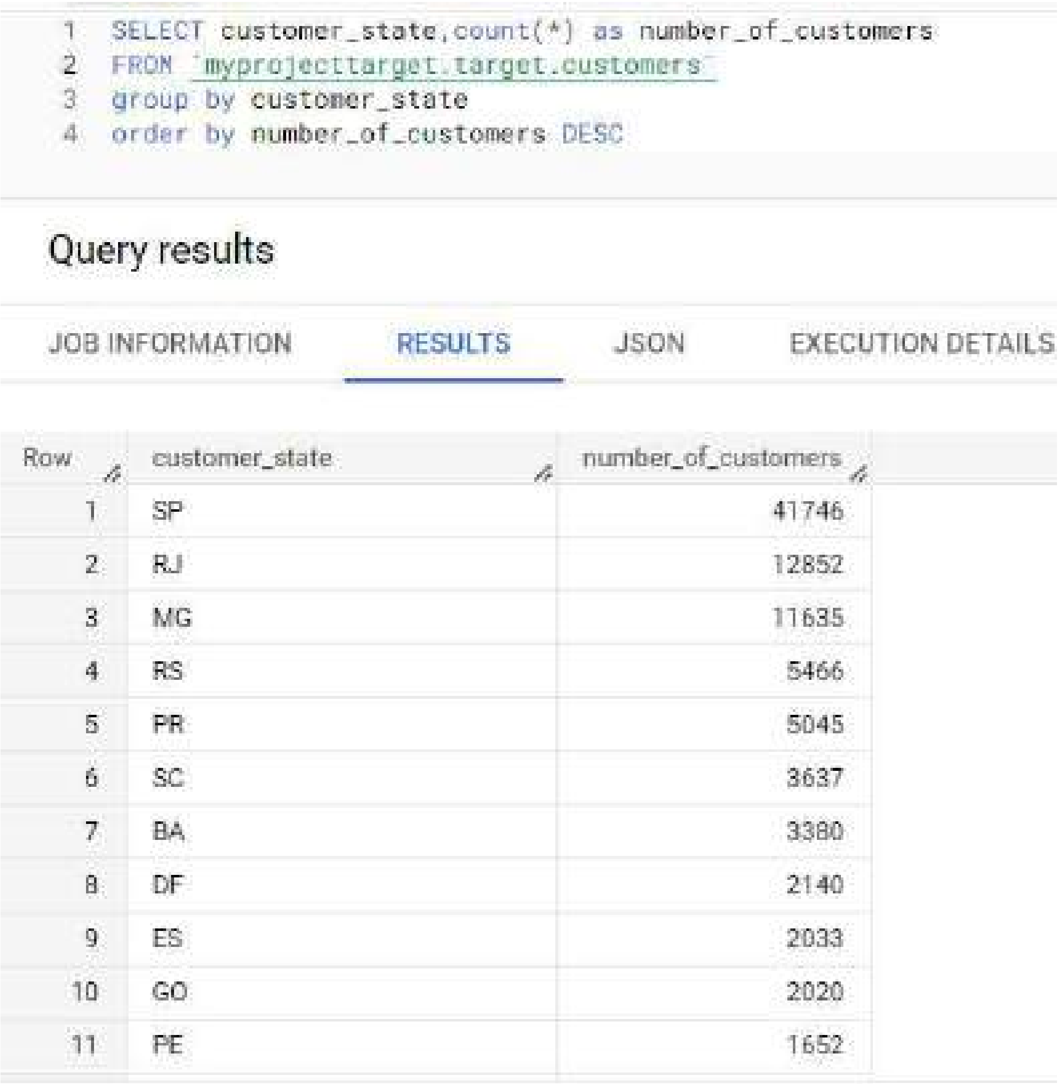
Actionable Insights:

Highest number of orders seen in 2018 August for state: SP and lowest number of orders 1 for

16 states

Distribution of customers across the states in Brazil

```
SELECT customer_state,count(*) as number_of_customers
FROM `myprojecttarget.target.customers`
group by customer_state
order by number_of_customers DESC
```



```
SELECT customer_state,count(*) as number_of_customers
FROM `myprojecttarget.target.customers`
group by customer_state
order by number_of_customers DESC
```

Query results

Row	customer_state	number_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
9	ES	2033
10	GO	2020
11	PE	1652

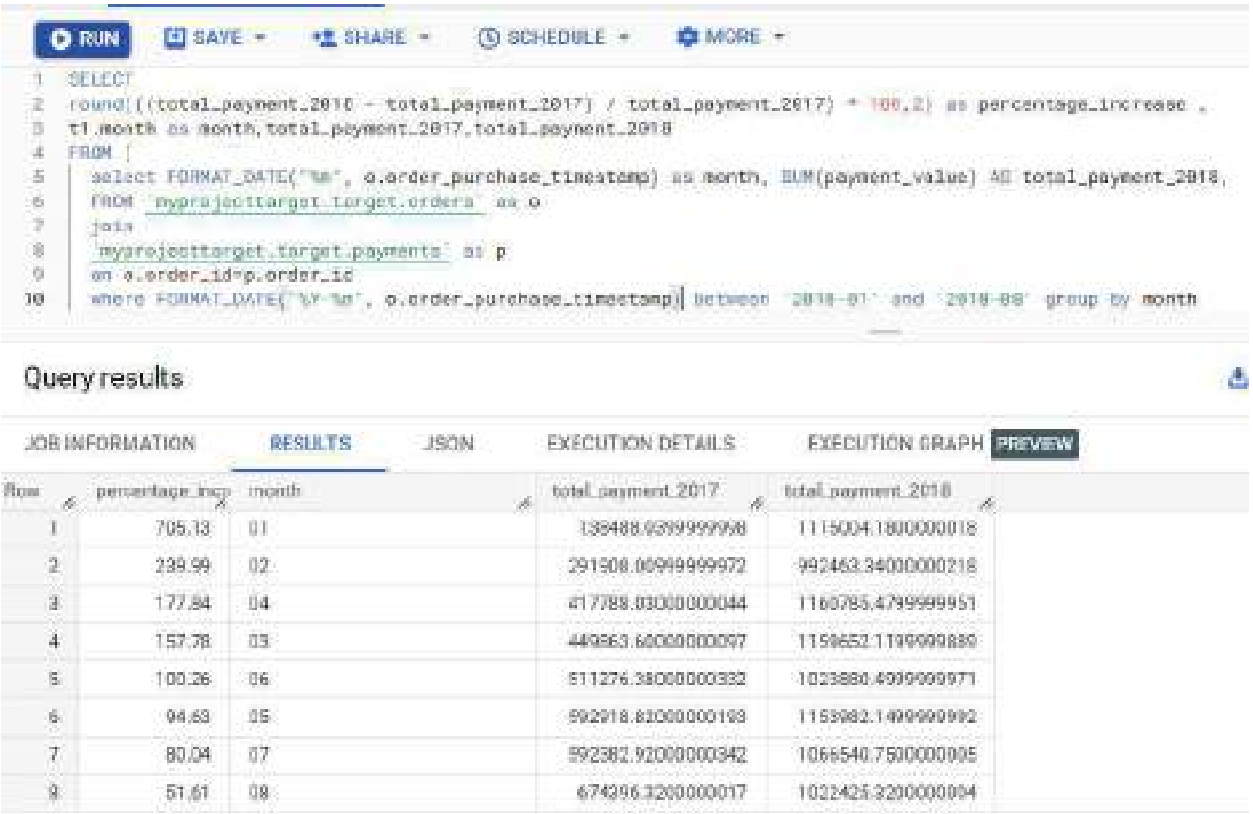
Actionable Insights:
Highest number of customers were seen in state: SP and lowest number of customers were seen in state: RR

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

Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) – You can use “payment_value” column in payments table

Month vice data

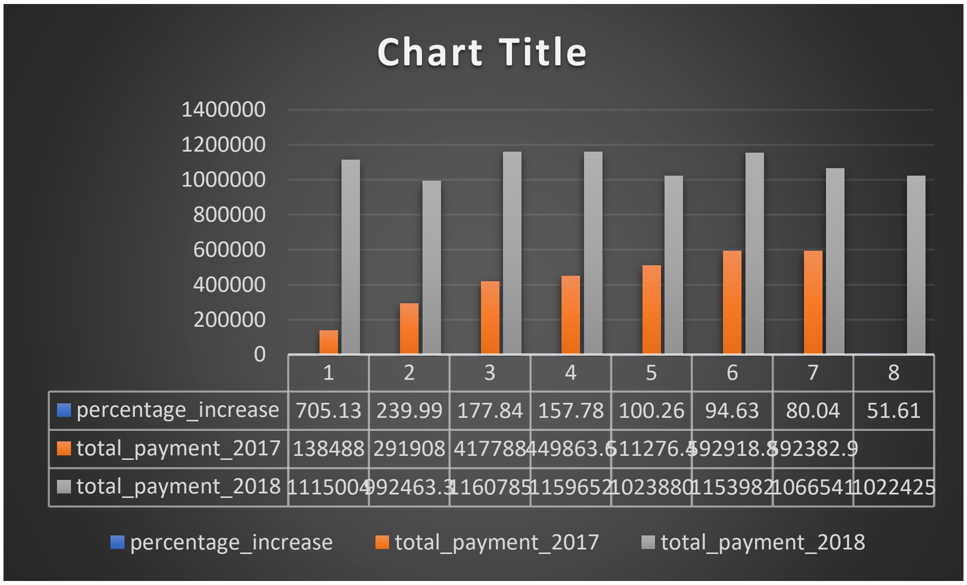
```
SELECT
round(((total_payment_2018 – total_payment_2017) / total_payment_2017) * 100,2) as percentage_increase ,
t1.month as month,total_payment_2017,total_payment_2018
FROM (
  select FORMAT_DATE(“%m”, o.order_purchase_timestamp) as month, SUM(payment_value) AS total_payment_20
18,
  FROM `myprojecttarget.target.orders` as o
  join
  `myprojecttarget.target.payments` as p
  on o.order_id=p.order_id
  where FORMAT_DATE(“%Y-%m”, o.order_purchase_timestamp) between ‘2018-01’ and ‘2018-08’ group by month
) as t1

join

(
SELECT FORMAT_DATE(“%m”, o.order_purchase_timestamp) as month,SUM(payment_value) AS total_payment_2
017,
FROM `myprojecttarget.target.orders` as o
join
`myprojecttarget.target.payments` as p
on o.order_id=p.order_id
where FORMAT_DATE(“%Y-%m”, o.order_purchase_timestamp) between ‘2017-01’ and ‘2017-08’group by month
) as t2

on t1.month=t2.month
order by percentage_increase DESC
```



```
SELECT
round(((total_payment_2018 - total_payment_2017) / total_payment_2017) * 100,2) as percentage_increase ,
t1.month as month, total_payment_2017,total_payment_2018
FROM (
  select FORMAT_DATE("%m", o.order_purchase_timestamp) as month, SUM(payment_value) AS total_payment_2018,
  FROM `myprojecttarget.target.orders` as o
  join
  `myprojecttarget.target.payments` as p
  on o.order_id=p.order_id
  where FORMAT_DATE("%Y-%m", o.order_purchase_timestamp) between '2018-01' and '2018-08' group by month
```

Query results

Row	percentage_incr	month	total_payment_2017	total_payment_2018
1	705.13	01	138488.0399999998	1115004.1800000018
2	239.99	02	291908.0099999972	992463.3400000218
3	177.84	04	417788.0300000044	1160785.4799999951
4	157.78	03	449863.6000000097	1159652.1199999889
5	100.26	06	511276.3800000332	1023880.4999999971
6	94.63	05	592918.8200000193	1153982.1499999992
7	80.04	07	592382.9200000342	1066540.7500000095
8	51.61	08	674396.3200000017	1022425.3200000004



	1	2	3	4	5	6	7	8
■ percentage_increase	705.13	239.99	177.84	157.78	100.26	94.63	80.04	51.61
■ total_payment_2017	138488	291908	417788	449863.6	511276.4	592918.8	592382.9	
■ total_payment_2018	1115004	992463.3	1160785	1159652	1023880	1153982	1066541	1022425

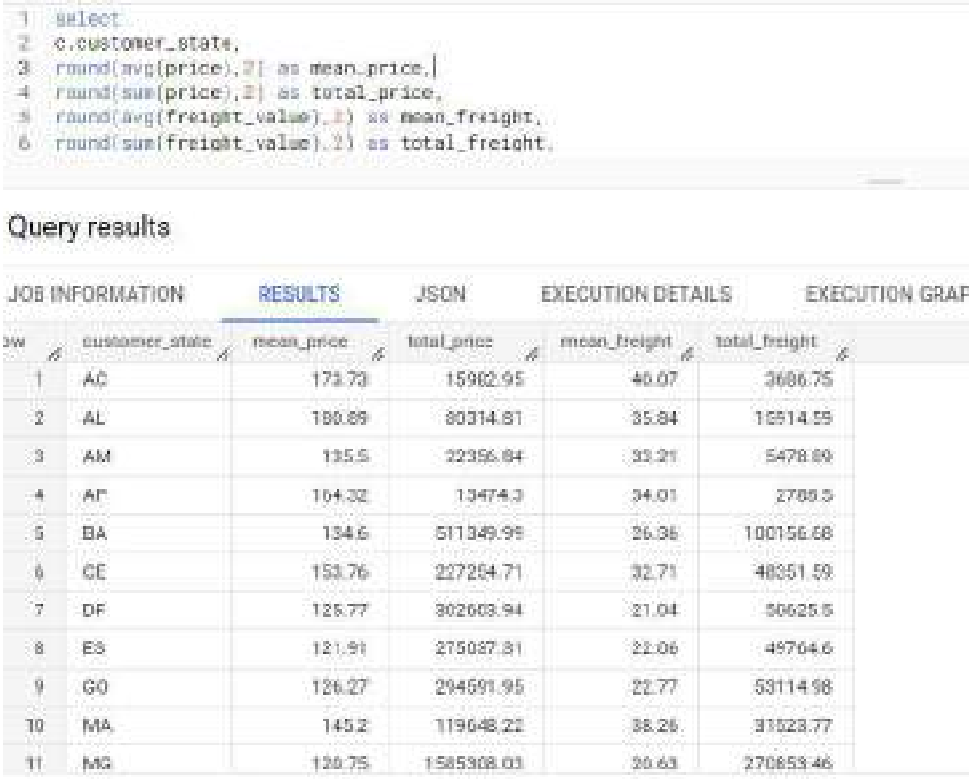
Actionable Insights:

January month have highest percentage increase and august have lowest percentage increase.

For 2017: growing trend seen for cost of orders
For 2018: cost of orders were an average of 1 lakh for the provided period

Mean & Sum of price and freight value by customer state

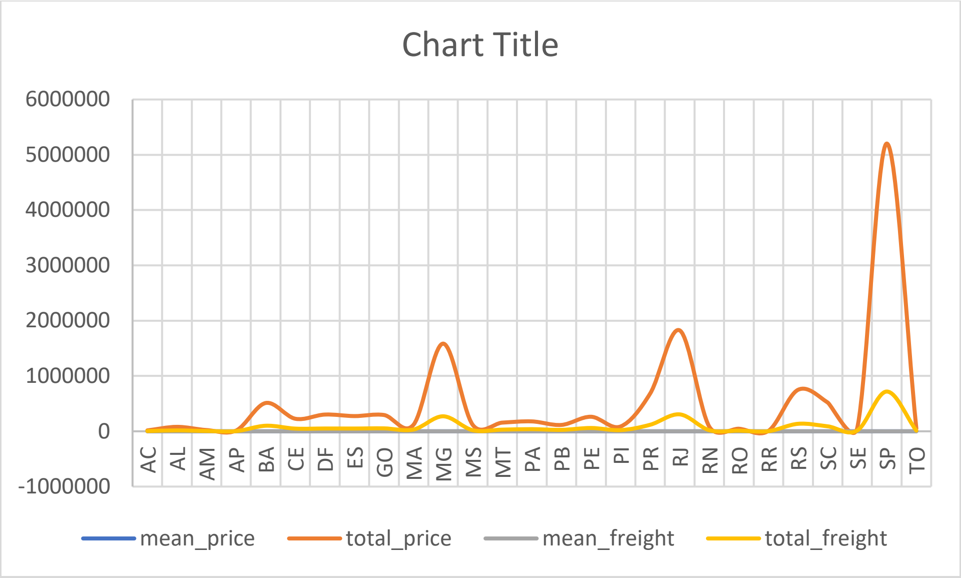
```
select
c.customer_state,
round(avg(price),2) as mean_price,
round(sum(price),2) as total_price,
round(avg(freight_value),2) as mean_freight,
round(sum(freight_value),2) as total_freight,

from `myprojecttarget.target.order_items` as oi join `myprojecttarget.target.orders` as o
on oi.order_id=o.order_id
join
`myprojecttarget.target.customers` as c on c.customer_id=o.customer_id
group by  c.customer_state
order by  c.customer_state
```



```
select
c.customer_state,
round(avg(price),2) as mean_price,
round(sum(price),2) as total_price,
round(avg(freight_value),2) as mean_freight,
round(sum(freight_value),2) as total_freight,
```

Query results

ow	customer_state	mean_price	total_price	mean_freight	total_freight
1	AC	173.73	15982.95	40.07	3686.75
2	AL	180.89	80314.81	35.84	15914.59
3	AM	135.5	22356.84	33.21	5478.89
4	AP	164.32	13474.3	34.01	2788.5
5	BA	134.6	511349.99	26.36	100156.68
6	CE	153.76	227254.71	32.71	48351.59
7	DF	125.77	302603.94	21.04	50625.5
8	ES	121.91	275037.31	22.06	49764.6
9	GO	126.27	294591.95	22.77	53114.98
10	MA	145.2	119648.22	38.26	31523.77
11	MG	120.75	1585308.03	20.63	270853.46



Actionable Insights:

- Highest mean price is seen for state:PB and lowest mean price seen for SP
- Highest total price is seen for state: SP and lowest total price is seen for RR
- Highest mean freight is seen for state:RR and the lowest mean freight seen in SP
- Highest total freight is seen for state: SP and lowest total freight seen in RR

Analysis on sales, freight and delivery time

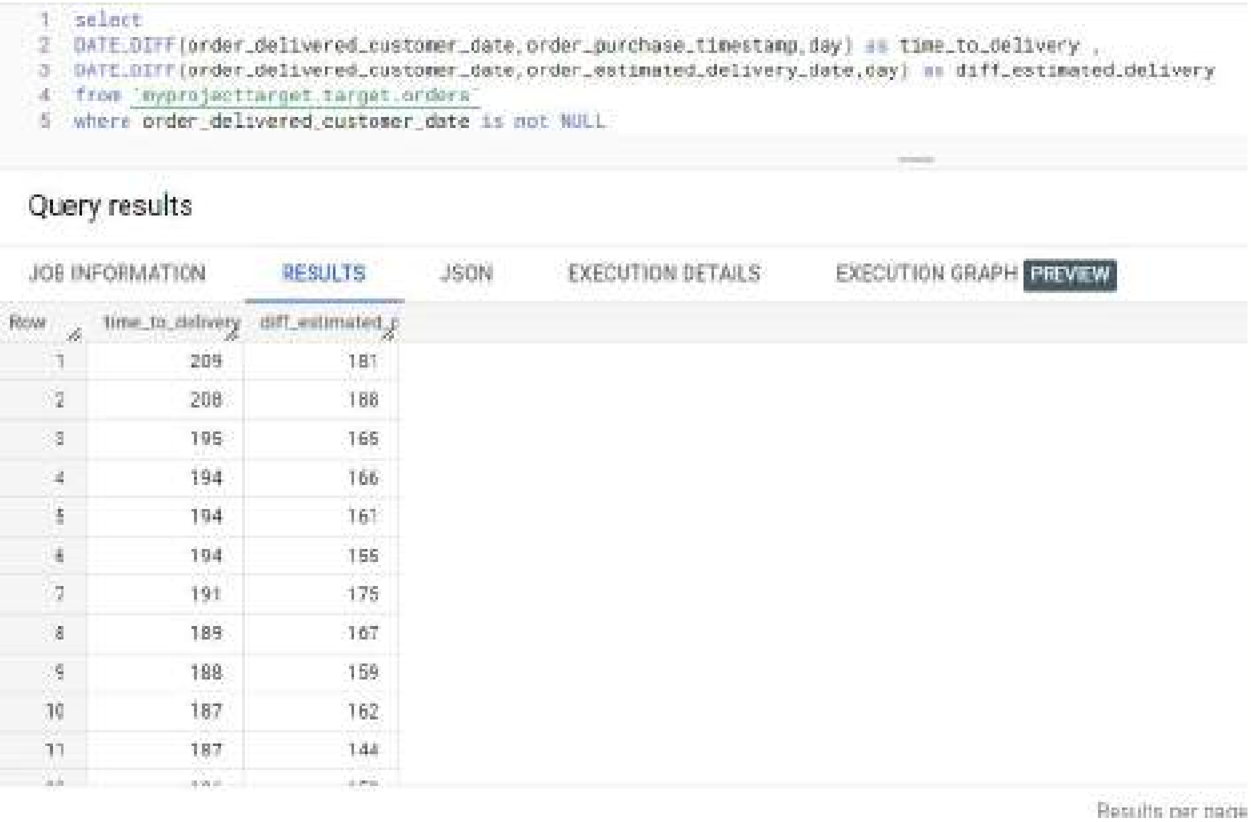
Calculate days between purchasing, delivering and estimated delivery

Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

time_to_delivery = order_purchase_timestamp-order_delivered_customer_date

diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

```
select

DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery ,
DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date,day) as diff_estimated_delivery
from `myprojecttarget.target.orders`
where order_delivered_customer_date is not NULL
order by time_to_delivery desc, diff_estimated_delivery desc
```



```
select
DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery ,
DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date,day) as diff_estimated_delivery
from `myprojecttarget.target.orders`
where order_delivered_customer_date is not NULL
```

Query results

Row	time_to_delivery	diff_estimated_c
1	209	181
2	208	188
3	195	165
4	194	166
5	194	161
6	194	155
7	191	175
8	189	167
9	188	159
10	187	162
11	187	144

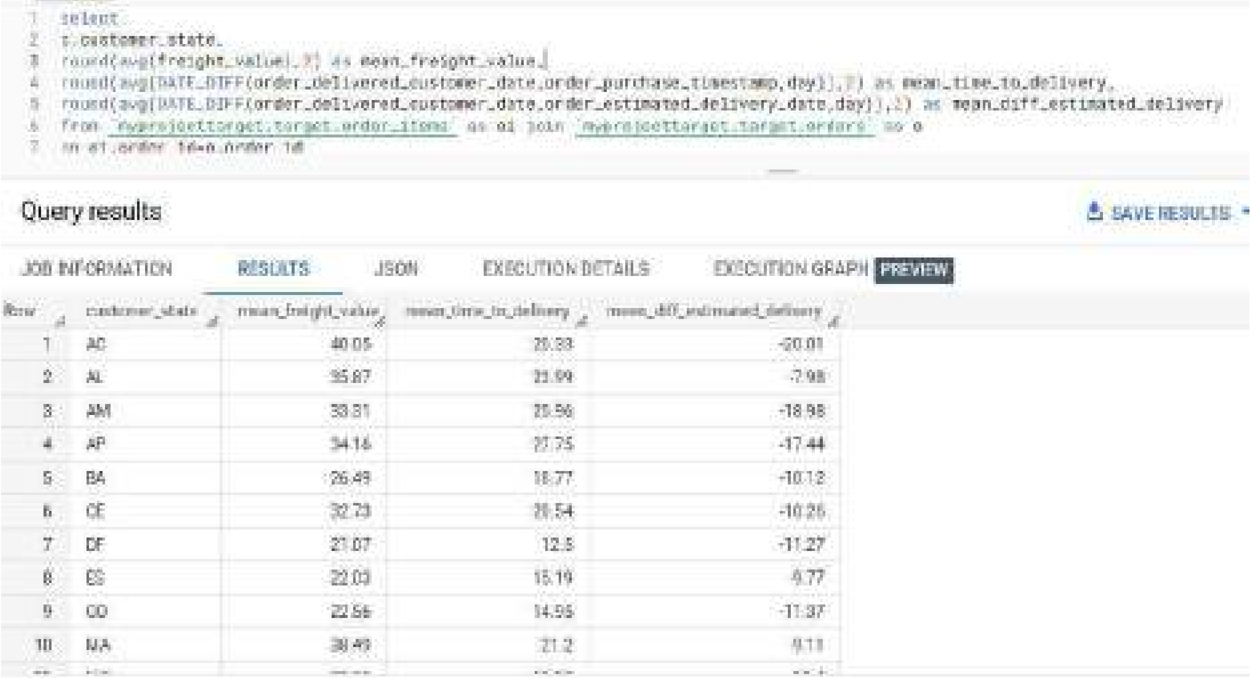
Actionable Insights:

- the highest number of days took time to delivery is 209 and the lowest number of days is 0
- the highest number of days for diff_estimated _delivery is 188 and the lowest is -146

Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
select
c.customer_state,
round(avg(freight_value),2) as mean_freight_value,
round(avg(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,day)),2) as mean_time_to_delivery,
round(avg(DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date,day)),2) as mean_diff_estimated_delivery
from `myprojecttarget.target.order_items` as oi join `myprojecttarget.target.orders` as o
on oi.order_id=o.order_id
join
`myprojecttarget.target.customers` as c on c.customer_id=o.customer_id
```

```
where order_delivered_customer_date is not NULL
group by  c.customer_state
order by  c.customer_state
```



```
select
c.customer_state,
round(avg(freight_value),2) as mean_freight_value,
round(avg(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,day)),2) as mean_time_to_delivery,
round(avg(DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date,day)),2) as mean_diff_estimated_delivery
from `myprojecttarget.target.order_items` as oi join `myprojecttarget.target.orders` as o
on oi.order_id=o.order_id
```

Query results

Row	customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	AC	40.05	20.33	-20.01
2	AL	35.87	23.99	-7.98
3	AM	33.31	25.96	-18.98
4	AP	34.16	27.75	-17.44
5	BA	26.49	18.77	-10.12
6	CE	32.73	20.54	-10.26
7	DF	21.07	12.5	-11.27
8	ES	22.03	15.19	-9.77
9	GO	22.56	14.95	-11.37
10	MA	38.49	21.2	-9.11

Actionable Insights:

- highest mean_freight_value is 43.09 for states RR and PB and lowest value for state: SP
- highest mean_time_to_delivery is 27.83 for RR and the lowest value for 8.26 for state: SP
- highest mean_diff_estimated_delivery is -7.98 for state : AL and the lowest is -20.01 for AC

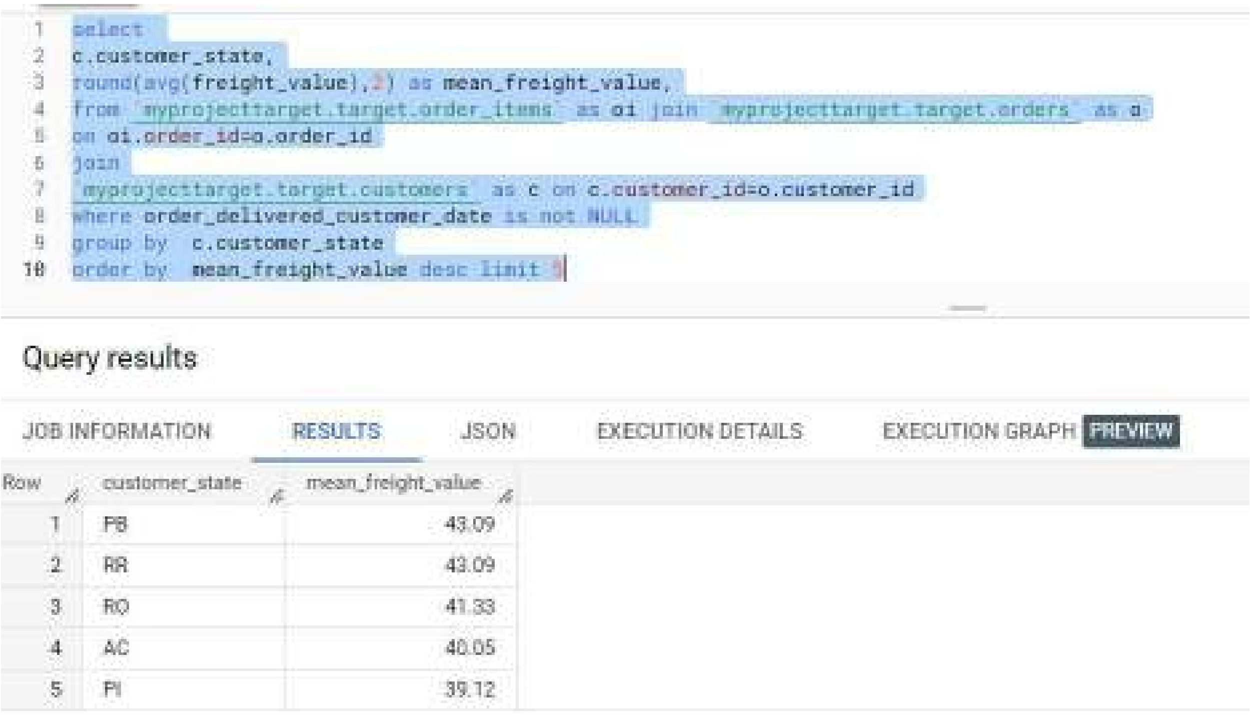
Sort the data to get the following:

Top 5 states with highest/lowest average freight value – sort in desc/asc limit 5

highest average freight value – sort in desc limit 5

```
select

c.customer_state,
round(avg(freight_value),2) as mean_freight_value,
from `myprojecttarget.target.order_items` as oi join `myprojecttarget.target.orders` as o
on oi.order_id=o.order_id
join
```

```
`myprojecttarget.target.customers` as c on c.customer_id=o.customer_id
where order_delivered_customer_date is not NULL
group by  c.customer_state
order by  mean_freight_value desc limit 5
```



```
select
c.customer_state,
round(avg(freight_value),2) as mean_freight_value,
from `myprojecttarget.target.order_items` as oi join `myprojecttarget.target.orders` as o
on oi.order_id=o.order_id
join
`myprojecttarget.target.customers` as c on c.customer_id=o.customer_id
where order_delivered_customer_date is not NULL
group by  c.customer_state
order by  mean_freight_value desc limit 5
```

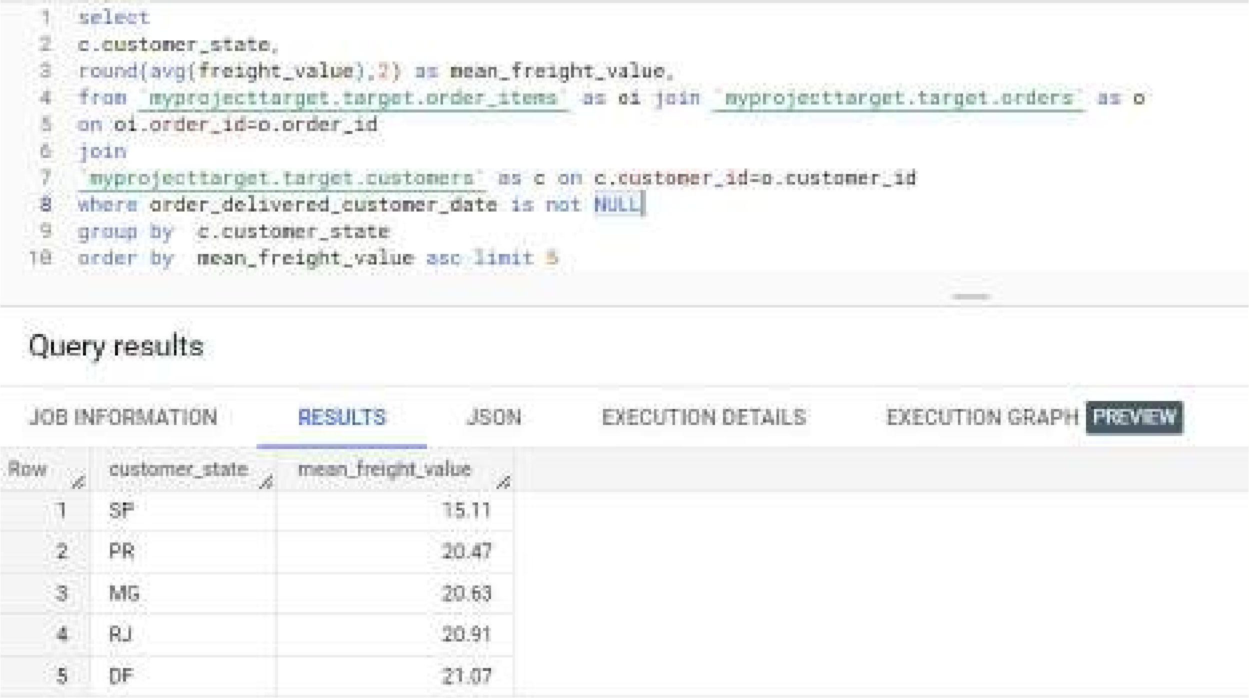
Row	customer_state	mean_freight_value
1	PB	43.09
2	RR	43.09
3	RO	41.33
4	AC	40.05
5	PI	39.12

Actionable Insights:

Top 5 states for highest mean_freight_value are PB,RR,RO,AC,PI those values are between 39.12 and 43.09

Top 5 states with lowest average freight value – sort in asc limit 5

```
select

c.customer_state,
round(avg(freight_value),2) as mean_freight_value,
from `myprojecttarget.target.order_items` as oi join `myprojecttarget.target.orders` as o
on oi.order_id=o.order_id
join
`myprojecttarget.target.customers` as c on c.customer_id=o.customer_id
where order_delivered_customer_date is not NULL
group by  c.customer_state
order by  mean_freight_value asc limit 5
```



```
select
c.customer_state,
round(avg(freight_value),2) as mean_freight_value,
from `myprojecttarget.target.order_items` as oi join `myprojecttarget.target.orders` as o
on oi.order_id=o.order_id
join
`myprojecttarget.target.customers` as c on c.customer_id=o.customer_id
where order_delivered_customer_date is not NULL
group by  c.customer_state
order by  mean_freight_value asc limit 5
```

Query results

Row	customer_state	mean_freight_value
1	SP	15.11
2	PR	20.47
3	MG	20.63
4	RJ	20.91
5	DF	21.07

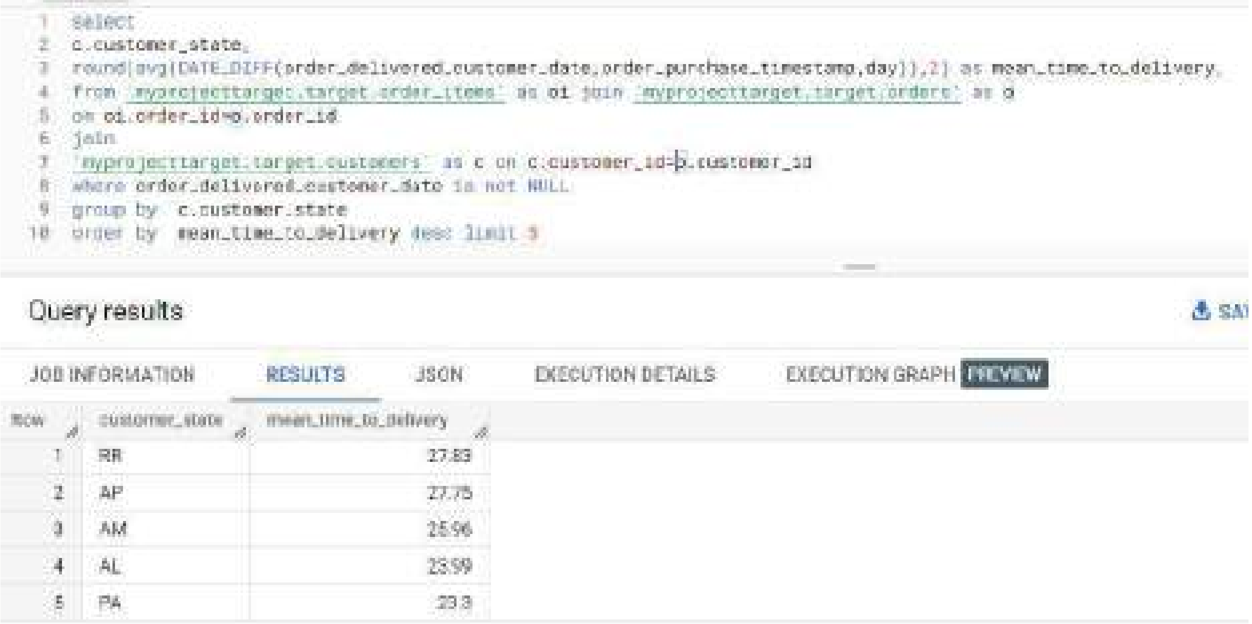
Actionable Insights:

Top 5 states for lowest mean_freight_value are SP,PR,MG,RJ,DF those values are between 15.11 and 21.07

Top 5 states with highest/lowest average time to delivery

Top 5 states with highest average time to delivery

```
select
c.customer_state,
round(avg(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,day)),2) as mean_time_to_delivery,
from `myprojecttarget.target.order_items` as oi join `myprojecttarget.target.orders` as o
on oi.order_id=o.order_id
join
`myprojecttarget.target.customers` as c on c.customer_id=o.customer_id
where order_delivered_customer_date is not NULL
group by  c.customer_state
order by  mean_time_to_delivery desc limit 5
```



```
select
c.customer_state,
round(avg(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,day)),2) as mean_time_to_delivery,
from `myprojecttarget.target.order_items` as oi join `myprojecttarget.target.orders` as o
on oi.order_id=o.order_id
join
`myprojecttarget.target.customers` as c on c.customer_id=o.customer_id
where order_delivered_customer_date is not NULL
group by  c.customer_state
order by  mean_time_to_delivery desc limit 5
```

Query results

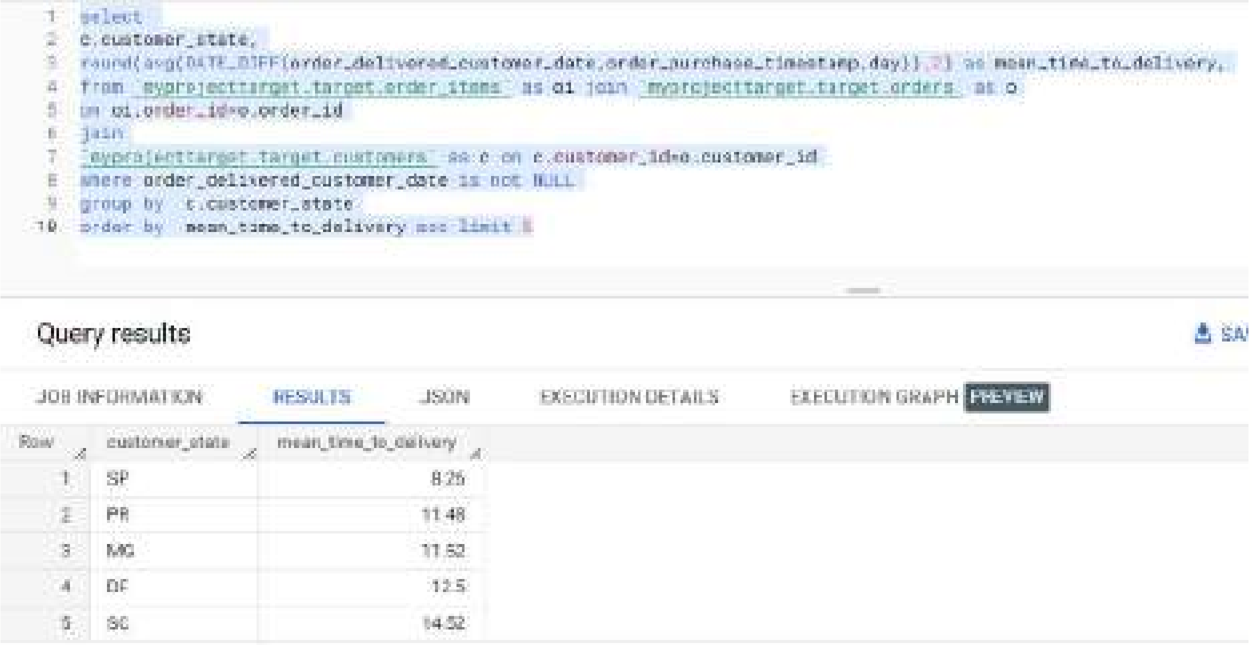
Row	customer_state	mean_time_to_delivery
1	RR	27.83
2	AP	27.75
3	AM	25.96
4	AL	23.99
5	PA	23.3

Actionable Insights:

Top 5 states for highest mean_time_to_delivery are RR,AP,AM,AL,PA those values are between 23.3 and 27.83

Top 5 states with lowest average time to delivery

```
select
c.customer_state,
round(avg(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,day)),2) as mean_time_to_deliv
ery,
from `myprojecttarget.target.order_items` as oi join `myprojecttarget.target.orders` as o
on oi.order_id=o.order_id
join
`myprojecttarget.target.customers` as c on c.customer_id=o.customer_id
where order_delivered_customer_date is not NULL
group by  c.customer_state
order by  mean_time_to_delivery asc limit 5
```



```
select
c.customer_state,
round(avg(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,day)),2) as mean_time_to_delivery,
from `myprojecttarget.target.order_items` as oi join `myprojecttarget.target.orders` as o
on oi.order_id=o.order_id
join
`myprojecttarget.target.customers` as c on c.customer_id=o.customer_id
where order_delivered_customer_date is not NULL
group by  c.customer_state
order by  mean_time_to_delivery asc limit 5
```

Query results

Row	customer_state	mean_time_to_delivery
1	SP	8.25
2	PR	11.48
3	MG	11.52
4	DF	12.5
5	SC	14.52

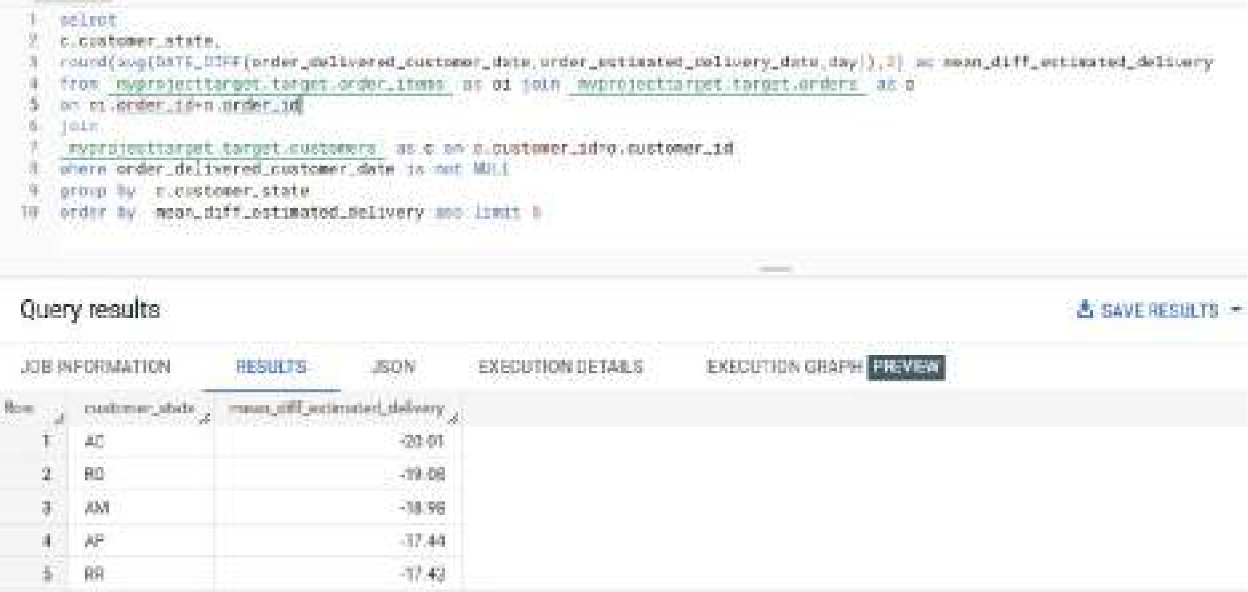
Actionable Insights:

Top 5 states for lowest mean_time_to_delivery are SP,PR,MG,DF,SC those values are between 23.3 and 27.83

Top 5 states where delivery is really fast/ not so fast compared to estimated date

Top 5 states where delivery is really fast compared to estimated date

```
select

c.customer_state,
round(avg(DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date,day)),2) as mean_diff_estimated_delivery
from `myprojecttarget.target.order_items` as oi join `myprojecttarget.target.orders` as o
on oi.order_id=o.order_id
join
`myprojecttarget.target.customers` as c on c.customer_id=o.customer_id
where order_delivered_customer_date is not NULL
group by  c.customer_state
order by  mean_diff_estimated_delivery asc limit 5
```



```
select
c.customer_state,
round(avg(DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date,day)),2) as mean_diff_estimated_delivery
from `myprojecttarget.target.order_items` as oi join `myprojecttarget.target.orders` as o
on oi.order_id=o.order_id
join
`myprojecttarget.target.customers` as c on c.customer_id=o.customer_id
where order_delivered_customer_date is not NULL
group by  c.customer_state
order by  mean_diff_estimated_delivery asc limit 5
```

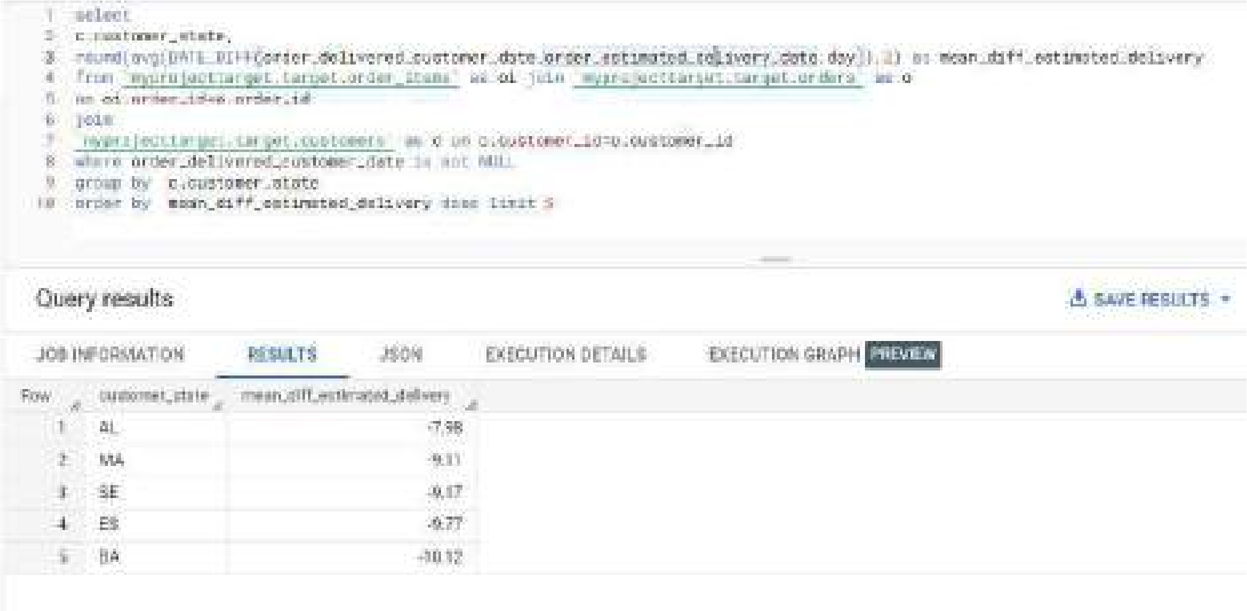
Row	customer_state	mean_diff_estimated_delivery
1	AC	-20.01
2	RO	-19.08
3	AM	-18.98
4	AP	-17.44
5	RR	-17.43

Actionable Insights:

Top 5 states where delivery is really fast compared to estimated date are AC,RO,AM,AP,RR values between -20.01 and -17.43

Top 5 states where delivery is not so fast compared to estimated date

```
select
c.customer_state,
round(avg(DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date,day)),2) as mean_diff_estimated_delivery
from `myprojecttarget.target.order_items` as oi join `myprojecttarget.target.orders` as o
on oi.order_id=o.order_id
join
`myprojecttarget.target.customers` as c on c.customer_id=o.customer_id
where order_delivered_customer_date is not NULL
group by  c.customer_state
order by  mean_diff_estimated_delivery desc limit 5
```



```
select
c.customer_state,
round(avg(DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date,day)),2) as mean_diff_estimated_delivery
from `myprojecttarget.target.order_items` as oi join `myprojecttarget.target.orders` as o
on oi.order_id=o.order_id
join
`myprojecttarget.target.customers` as c on c.customer_id=o.customer_id
where order_delivered_customer_date is not NULL
group by  c.customer_state
order by  mean_diff_estimated_delivery desc limit 5
```

Query results

Row	customer_state	mean_diff_estimated_delivery
1	AL	-7.98
2	MA	-9.11
3	SE	-9.17
4	ES	-9.77
5	BA	-10.12

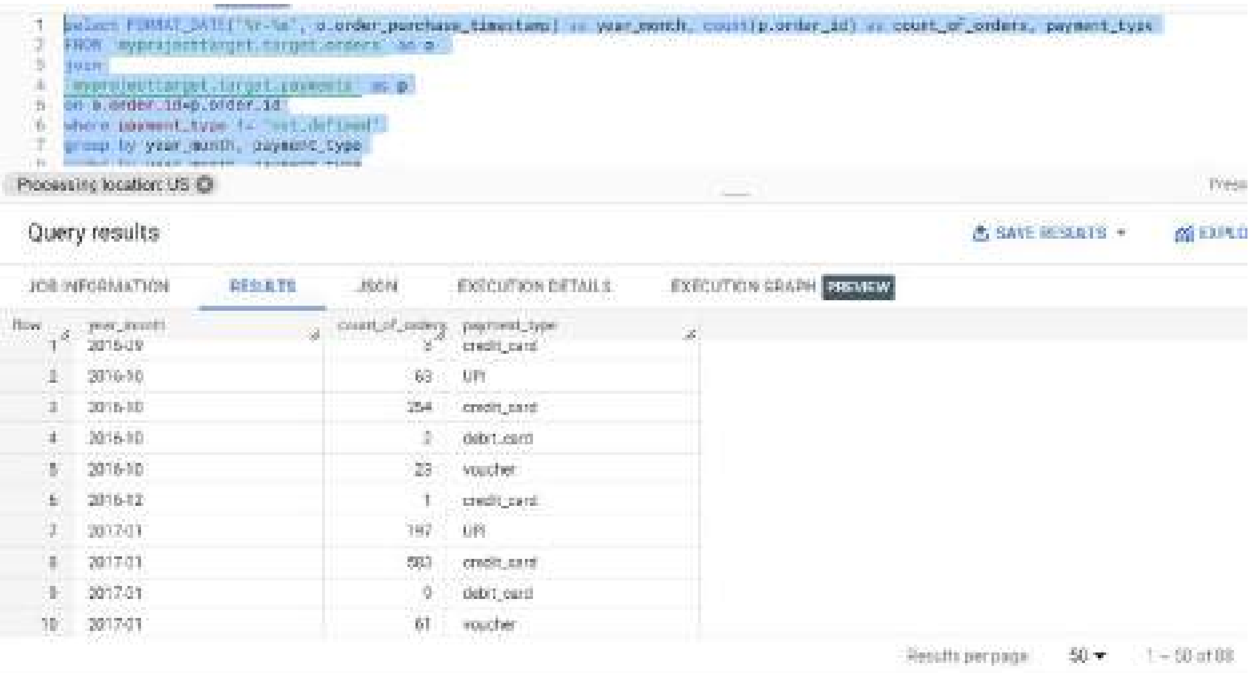
Actionable Insights:

Top 5 states where delivery is not so fast compared to estimated date are AL,MA,SE,ES,BA values between -10.12 and -7.98

Payment type analysis:

Month over Month count of orders for different payment types

```
select FORMAT_DATE("%Y-
%m", o.order_purchase_timestamp) as year_month, count(p.order_id) as count_of_orders, payment_type
FROM `myprojecttarget.target.orders` as o
join
`myprojecttarget.target.payments` as p
on o.order_id=p.order_id
where payment_type != "not_defined"
group by year_month, payment_type
order by year_month, payment_type
```



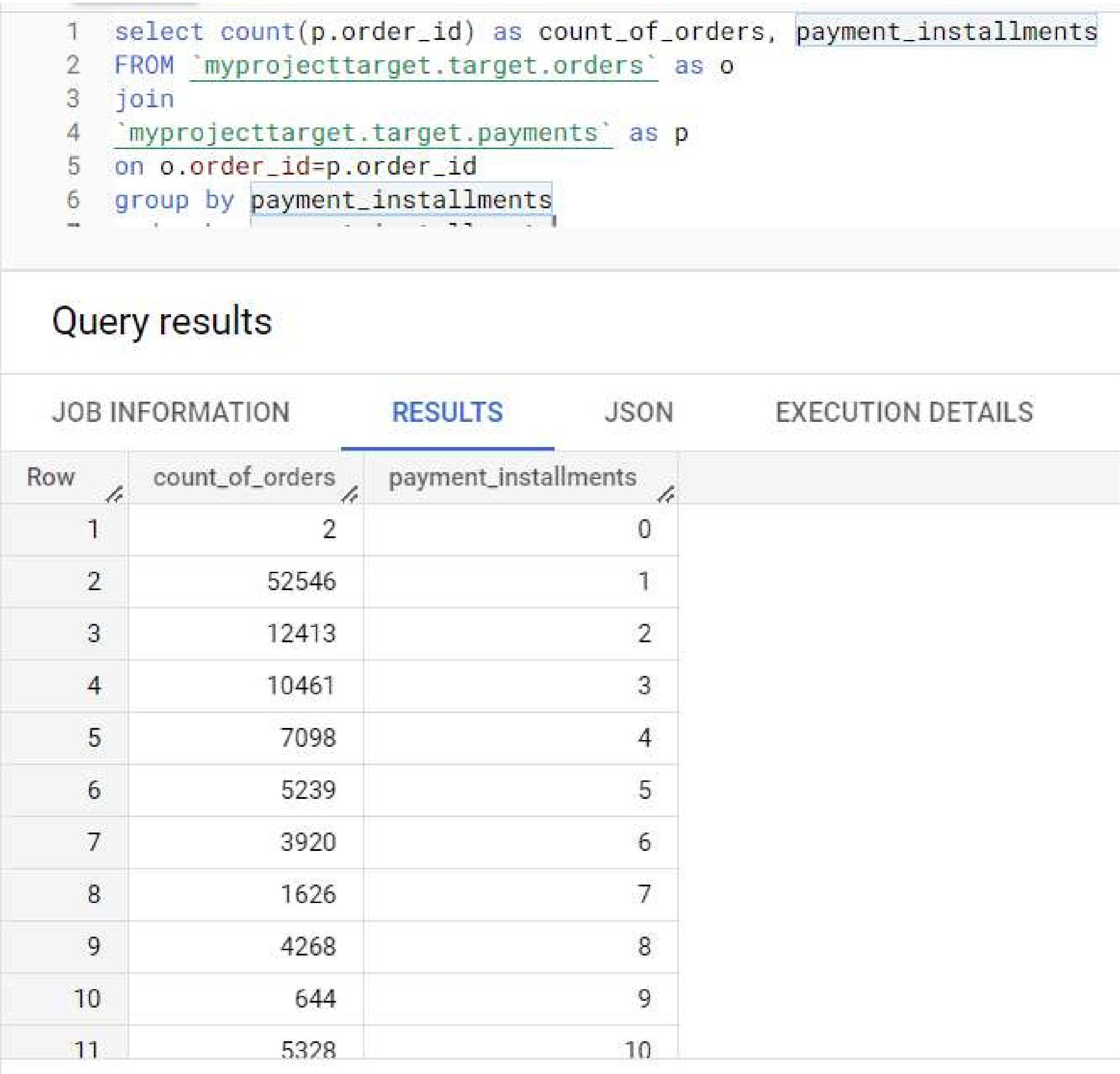
```
select FORMAT_DATE('%Y-%m', o.order_purchase_timestamp) as year_month, count(p.order_id) as count_of_orders, payment_type
FROM `myprojecttarget.target.orders` as o
join
`myprojecttarget.target.payments` as p
on o.order_id=p.order_id
where payment_type != 'not_defined'
group by year_month, payment_type
```

Row	year_month	count_of_orders	payment_type
1	2016-09	3	credit_card
2	2016-10	63	UPI
3	2016-10	254	credit_card
4	2016-10	2	debit_card
5	2016-10	23	voucher
6	2016-12	1	credit_card
7	2017-01	197	UPI
8	2017-01	583	credit_card
9	2017-01	9	debit_card
10	2017-01	61	voucher

Actionable Insights:
highest number of order seen for credit card payment type on 2017-11 and lowest number of orders seen for credit card on 2016-12

Count of orders based on the no. of payment installments

```
select count(p.order_id) as count_of_orders, payment_installments
FROM `myprojecttarget.target.orders` as o
join
`myprojecttarget.target.payments` as p
on o.order_id=p.order_id
group by payment_installments
order by payment_installments
```



```sql
select count(p.order_id) as count_of_orders, payment_installments
FROM `myprojecttarget.target.orders` as o
join
`myprojecttarget.target.payments` as p
on o.order_id=p.order_id
group by payment_installments
```

Row	count_of_orders	payment_installments
1	2	0
2	52546	1
3	12413	2
4	10461	3
5	7098	4
6	5239	5
7	3920	6
8	1626	7
9	4268	8
10	644	9
11	5328	10

Actionable Insights:

Highest number of orders are seen for payment_installments 1 and the lowest is for payment_installments 23

Recommendations

- 1144 cities have only one customer and 6 states having customer count less than 300, can focus on improving customer base by increasing engagement with these customers by providing faster shipping by arranging warehouses near them.
- 2018 september and october months have huge fall in orders need to increase the orders by providing discounts.
- orders during the dawn can be increased by providing special discounts and for evening , morning and night we can increase orders by providing hourly sale for these time periods

- top 10 states as per highest number of customers have relatively lower number of order per customer (below 100) , should concentrate on these customers to increase orders
- SP have the highest total price but mean price is very low so we can look for increasing mean price for this state as the lowest mean freight seen in SP by providing incentive or free gifts or delivery amount discount on certain value.
- we can reduce the diff_estimated delivery days to average of it so that customer likes to visit soon by having warehouses near to the customers.
- SP is the state where lowest mean_freight_value and lowest mean_time_to_delivery so we can should increase orders here as it has low order count
- these are the top 5 states where delivery is not so fast compared to estimated date are AL,MA,SE,ES,BA so we should look to speed te delivery in these states
- for count of orders are less for credit card payment type can increase orders using this type by providing attractive offers.
- count of orders are less for zero installments and larger installments value like 23 we can provide attractive offers if customers choose these payments