BUSINESS CASE STUDY

Target is one of the world's most recognized brands and one of America's leading retailers. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

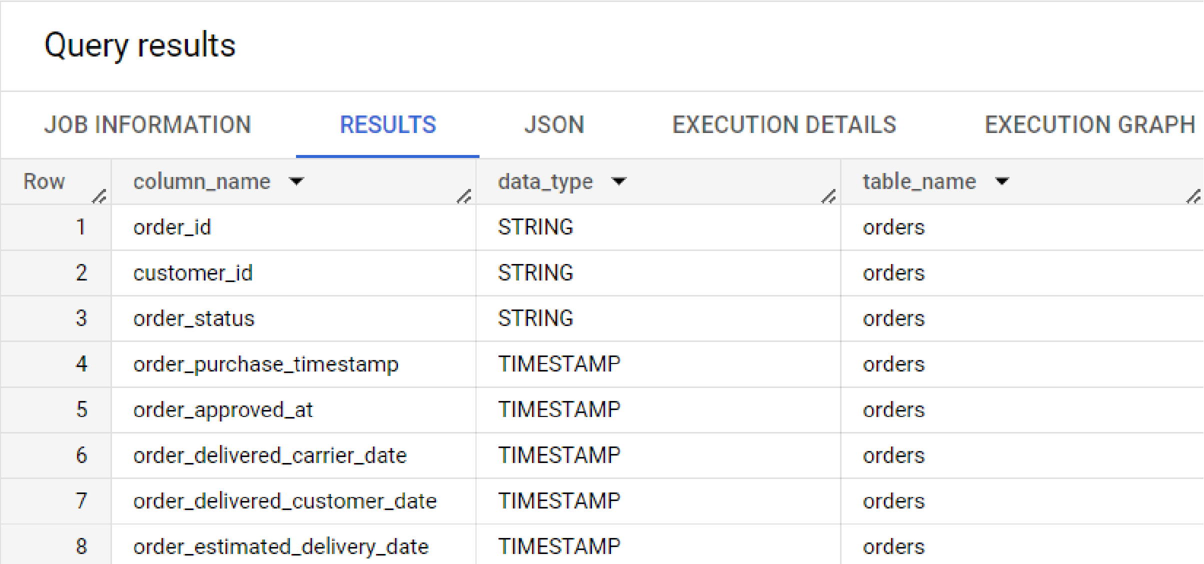
This business case has information of 100k orders from 2016 to 2018 made at Target in Brazil. Its features allows viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers.

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

a) Data type of columns in a table :

(i)orders

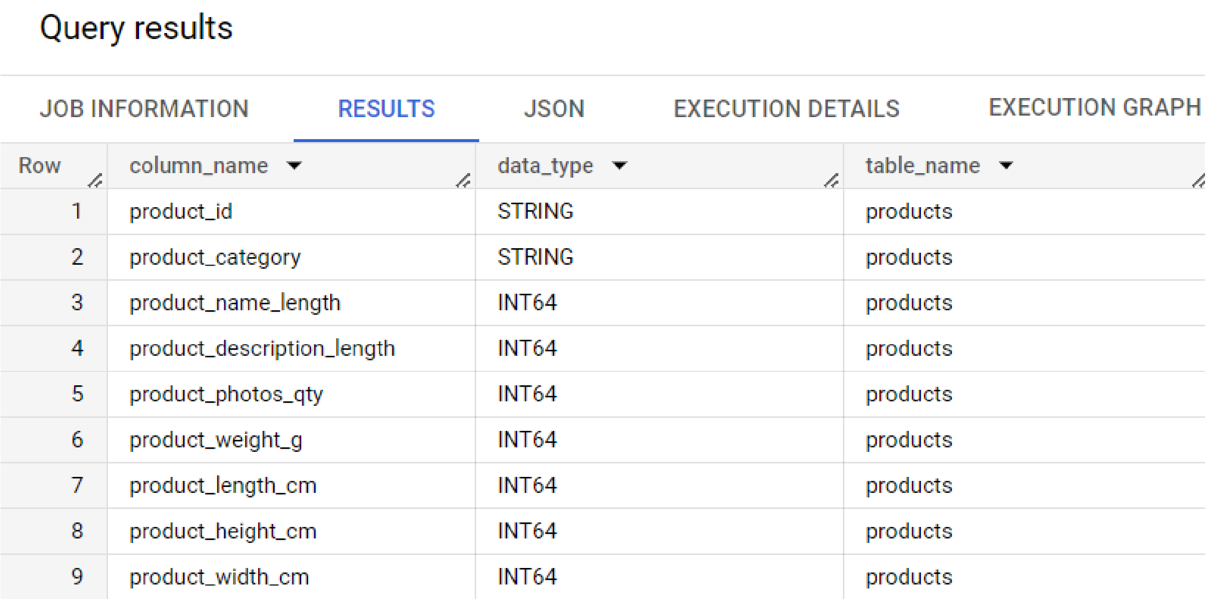
```
select column_name, data_type, table_name

from Ecommerce.INFORMATION_SCHEMA.COLUMNS

where table_name
in('orders')
```



Row	column_name	data_type	table_name
1	order_id	STRING	orders
2	customer_id	STRING	orders
3	order_status	STRING	orders
4	order_purchase_timestamp	TIMESTAMP	orders
5	order_approved_at	TIMESTAMP	orders
6	order_delivered_carrier_date	TIMESTAMP	orders
7	order_delivered_customer_date	TIMESTAMP	orders
8	order_estimated_delivery_date	TIMESTAMP	orders

(ii) products

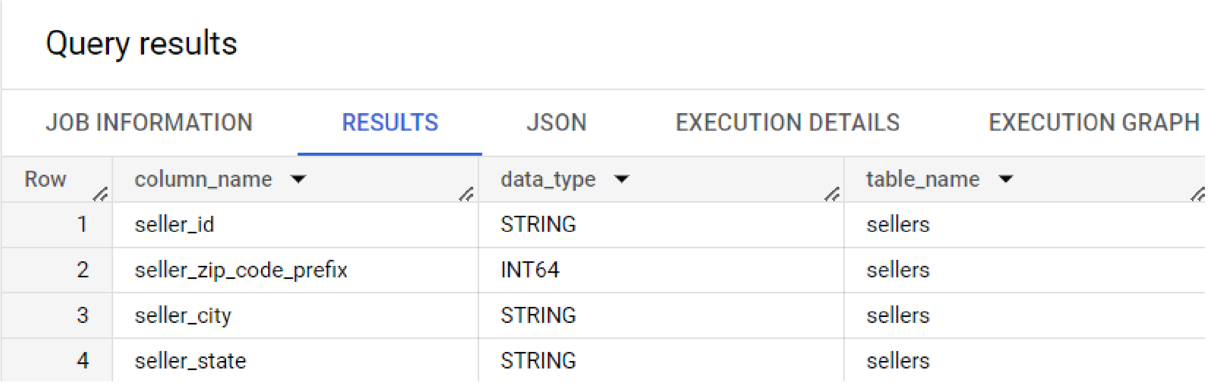
```
select column_name, data_type, table_name

from Ecommerce.INFORMATION_SCHEMA.COLUMNS

where table_name
in('products')
```



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

(iii)sellers

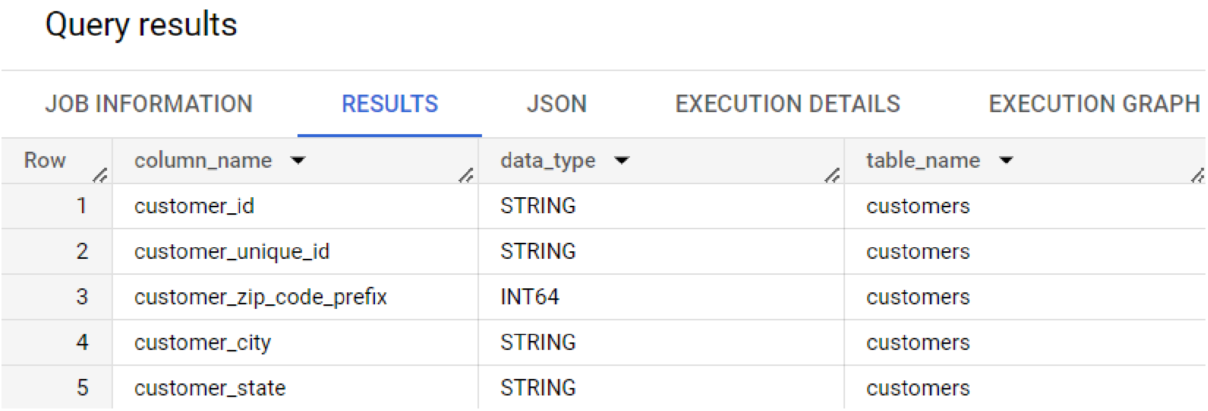
```
select column_name, data_type, table_name
from Ecommerce.INFORMATION_SCHEMA.COLUMNS
where table_name
in('sellers')
```



Row	column_name	data_type	table_name
1	seller_id	STRING	sellers
2	seller_zip_code_prefix	INT64	sellers
3	seller_city	STRING	sellers
4	seller_state	STRING	sellers

(iv)customers

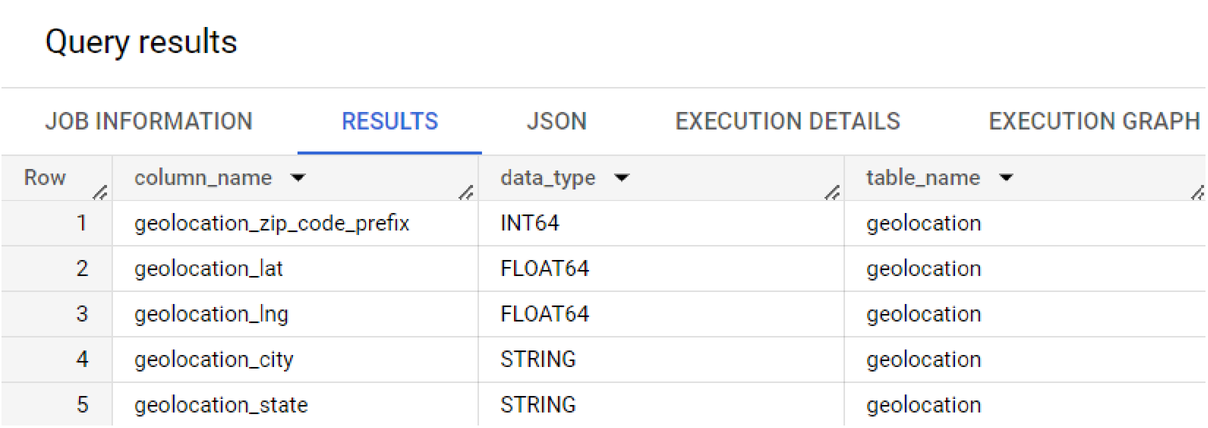
```
select column_name, data_type, table_name

from Ecommerce.INFORMATION_SCHEMA.COLUMNS
where table_name
in('customers')
```



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

(v)geolocation

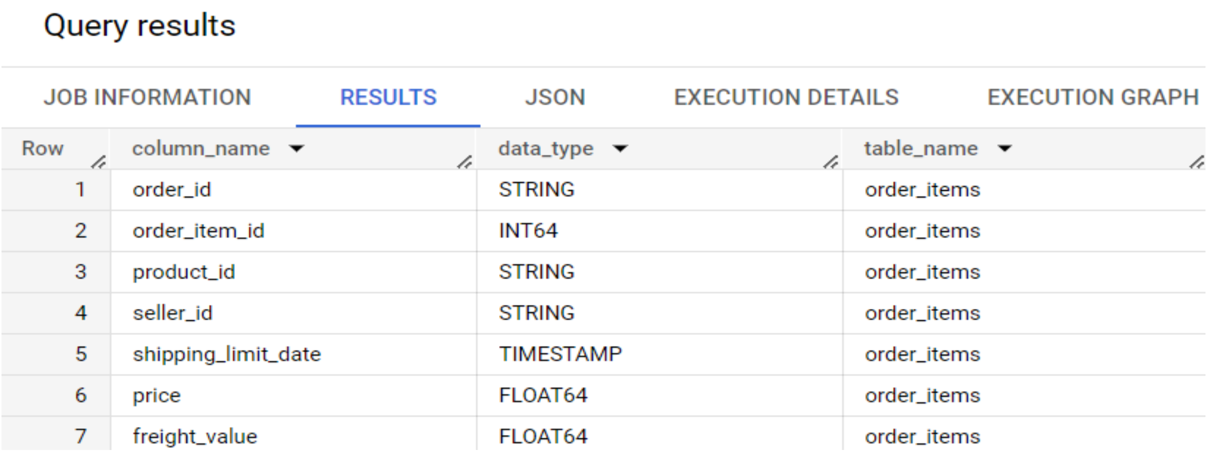
```
select column_name, data_type, table_name
from Ecommerce.INFORMATION_SCHEMA.COLUMNS
where table_name
in('geolocation')
```



Row	column_name	data_type	table_name
1	geolocation_zip_code_prefix	INT64	geolocation
2	geolocation_lat	FLOAT64	geolocation
3	geolocation_lng	FLOAT64	geolocation
4	geolocation_city	STRING	geolocation
5	geolocation_state	STRING	geolocation

(vi) order_items

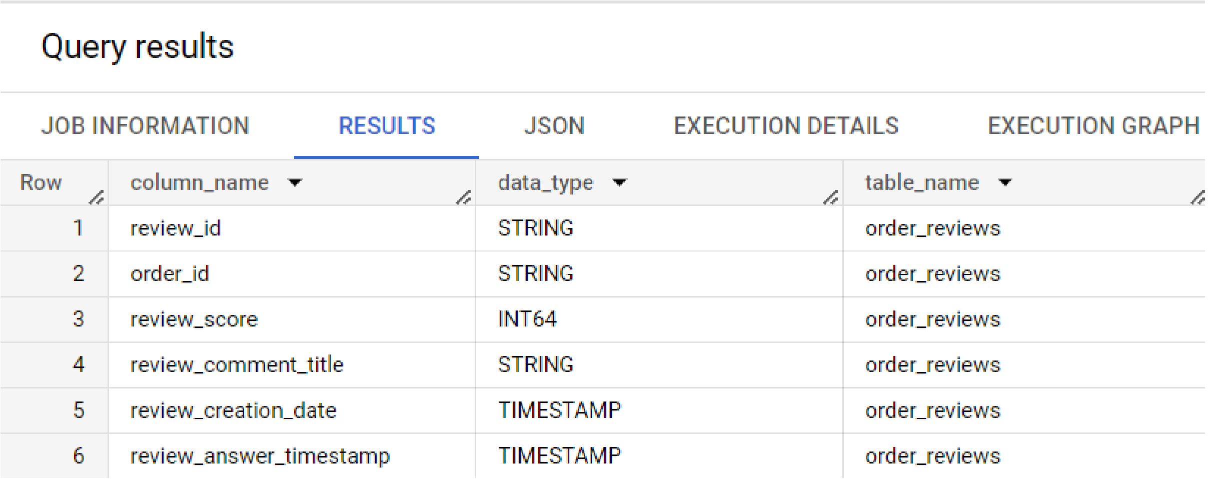
```
select column_name, data_type, table_name
from Ecommerce.INFORMATION_SCHEMA.COLUMNS
where table_name
in('order_items')
```



Row	column_name	data_type	table_name
1	order_id	STRING	order_items
2	order_item_id	INT64	order_items
3	product_id	STRING	order_items
4	seller_id	STRING	order_items
5	shipping_limit_date	TIMESTAMP	order_items
6	price	FLOAT64	order_items
7	freight_value	FLOAT64	order_items

(vii)order_reviews

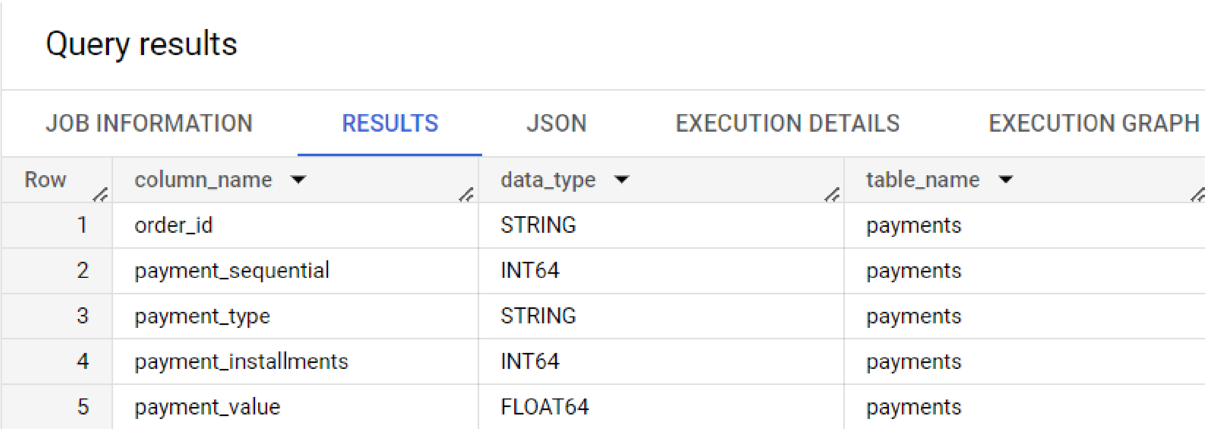
```
select column_name, data_type, table_name
from Ecommerce.INFORMATION_SCHEMA.COLUMNS
where table_name
in('order_reviews')
```



Row	column_name	data_type	table_name
1	review_id	STRING	order_reviews
2	order_id	STRING	order_reviews
3	review_score	INT64	order_reviews
4	review_comment_title	STRING	order_reviews
5	review_creation_date	TIMESTAMP	order_reviews
6	review_answer_timestamp	TIMESTAMP	order_reviews

(viii)payments

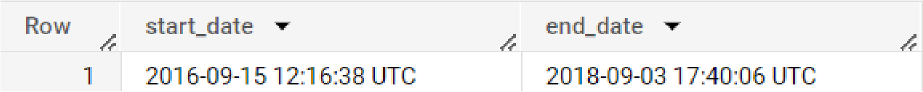
```
select column_name, data_type, table_name
from Ecommerce.INFORMATION_SCHEMA.COLUMNS
where table_name
in('payments')
```



Row	column_name	data_type	table_name
1	order_id	STRING	payments
2	payment_sequential	INT64	payments
3	payment_type	STRING	payments
4	payment_installments	INT64	payments
5	payment_value	FLOAT64	payments

b)Time period for which the data is given:

```
select min(order_approved_at) as start_date,
max(order_approved_at) as end_date
from Ecommerce.orders;
```

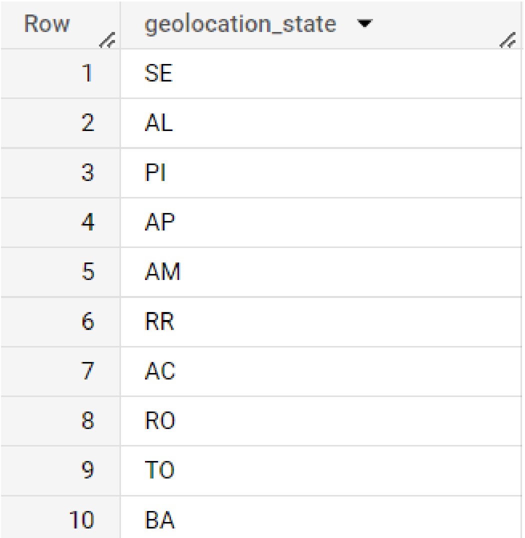


Row	start_date	end_date
1	2016-09-15 12:16:38 UTC	2018-09-03 17:40:06 UTC

c)Cities and States of customers ordered during the given period:

(i)States:

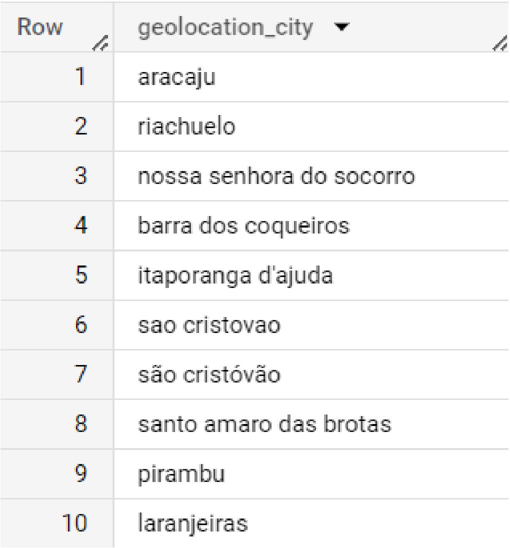
```
select distinct geolocation_state
from Ecommerce.geolocation
limit 10;
```



Row	geolocation_state
1	SE
2	AL
3	PI
4	AP
5	AM
6	RR
7	AC
8	RO
9	TO
10	BA

(ii)city:

```
select distinct geolocation_city
from Ecommerce.geolocation
limit 10;
```



Row	geolocation_city
1	aracaju
2	riachuelo
3	nossa senhora do socorro
4	barra dos coqueiros
5	itaporanga d'ajuda
6	sao cristovao
7	são cristóvão
8	santo amaro das brotas
9	pirambu
10	laranjeiras

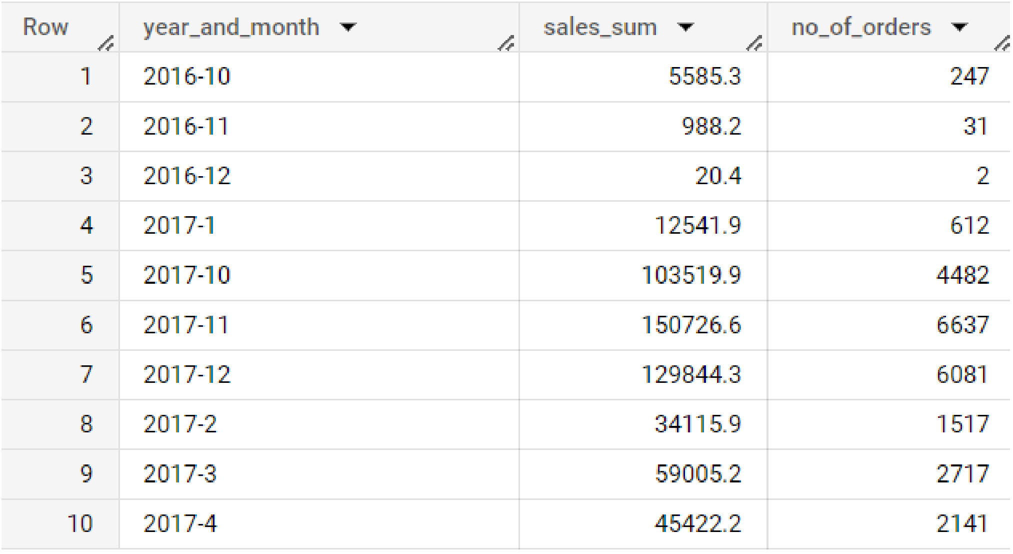
INSIGHTS:

- These are the unique states (27) & cities (8011) present in Brazil from Geo-location data.

2) In-depth Exploration:

a) 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 concat(EXTRACT(Year FROM O.order_delivered_carrier_date), '-',
EXTRACT(Month FROM O.order_delivered_carrier_date)) AS year_and_month,
round(sum(oI.freight_value), 1) as sales_sum, count(distinct oI.order_id)
as no_of_orders
from `Ecommerce.order_items` as oI
join `Ecommerce.orders` as O on O.order_id=oI.order_id
where O.order_delivered_carrier_date is not null
group by year_and_month order by year_and_month
limit 10;
```



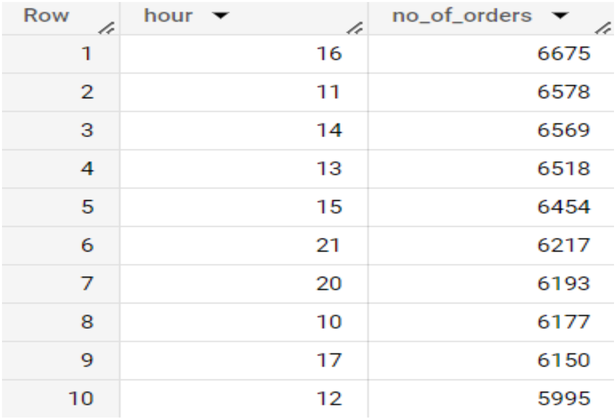
Row	year_and_month	sales_sum	no_of_orders
1	2016-10	5585.3	247
2	2016-11	988.2	31
3	2016-12	20.4	2
4	2017-1	12541.9	612
5	2017-10	103519.9	4482
6	2017-11	150726.6	6637
7	2017-12	129844.3	6081
8	2017-2	34115.9	1517
9	2017-3	59005.2	2717
10	2017-4	45422.2	2141

INSIGHTS:

- There was a growing trend along the time.
- We can see some seasonality with peaks at specific months, but in general we can see clear that customers are more prone to buy things online than before.

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

```
select distinct extract(hour from order_purchase_timestamp) as hour,
count(distinct order_id) as no_of_orders
from `Ecommerce.orders`
group by hour order by no_of_orders desc
limit 10;
```



Row	hour	no_of_orders
1	16	6675
2	11	6578
3	14	6569
4	13	6518
5	15	6454
6	21	6217
7	20	6193
8	10	6177
9	17	6150
10	12	5995

INSIGHTS:

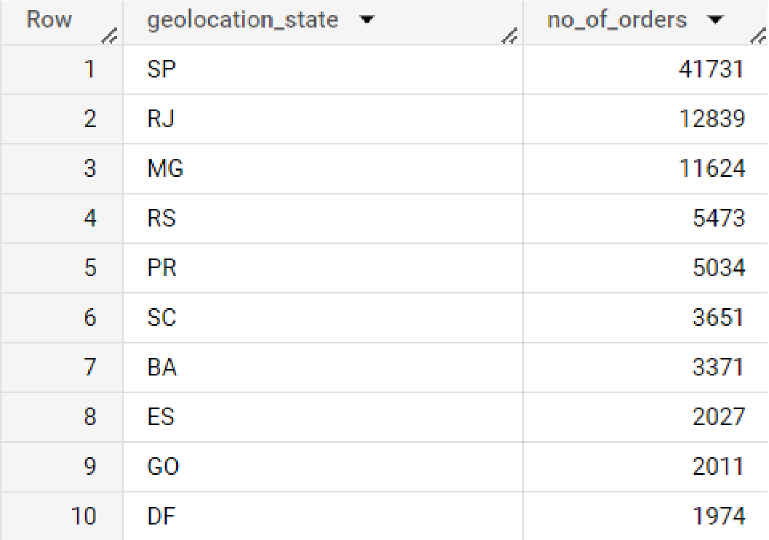
- As we can see here most customers tend to buy in evening, we can consider this is as peak time.

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

a) Get month on month orders by states:

(i) state wise:

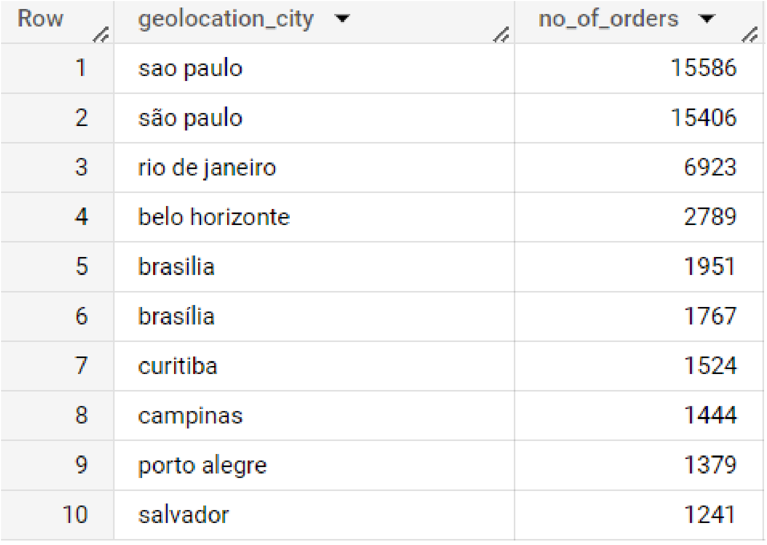
```
select distinct G.geolocation_state, count(distinct O.order_id) as
no_of_orders
from `Ecommerce.orders` O
left join `Ecommerce.order_items` as OI on OI.order_id=O.order_id
join `Ecommerce.customers` as C on C.customer_id=O.customer_id
left join `Ecommerce.geolocation` G on
G.geolocation_zip_code_prefix=C.customer_zip_code_prefix
group by G.geolocation_state order by no_of_orders desc
limit 10;
```



Row	geolocation_state	no_of_orders
1	SP	41731
2	RJ	12839
3	MG	11624
4	RS	5473
5	PR	5034
6	SC	3651
7	BA	3371
8	ES	2027
9	GO	2011
10	DF	1974

(ii) city wise:

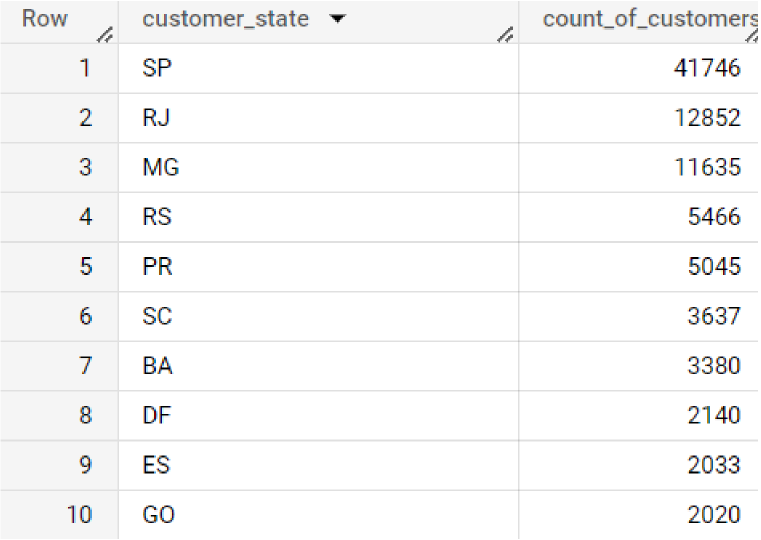
```
select distinct G.geolocation_city, count(distinct O.order_id) as
no_of_orders
from `Ecommerce.orders` O
left join `Ecommerce.order_items` as OI on OI.order_id=O.order_id
join `Ecommerce.customers` as C on C.customer_id=O.customer_id
left join `Ecommerce.geolocation` G on
G.geolocation_zip_code_prefix=C.customer_zip_code_prefix
group by G.geolocation_city order by no_of_orders desc
limit 10;
```



Row	geolocation_city	no_of_orders
1	sao paulo	15586
2	são paulo	15406
3	rio de janeiro	6923
4	belo horizonte	2789
5	brasilia	1951
6	brasília	1767
7	curitiba	1524
8	campinas	1444
9	porto alegre	1379
10	salvador	1241

b)How are customers distributed in Brazil:

```
select customer_state,
count(customer_id)as count_of_customers
from Ecommerce.customers
group by customer_state order by count_of_customers desc
limit 10;
```

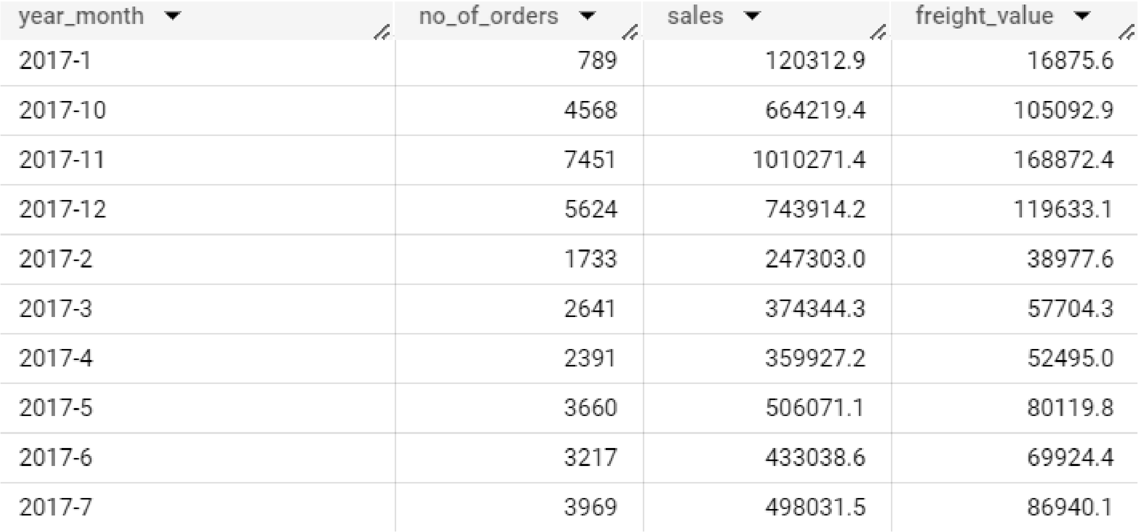


Row	customer_state	count_of_customers
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

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

a) Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only)

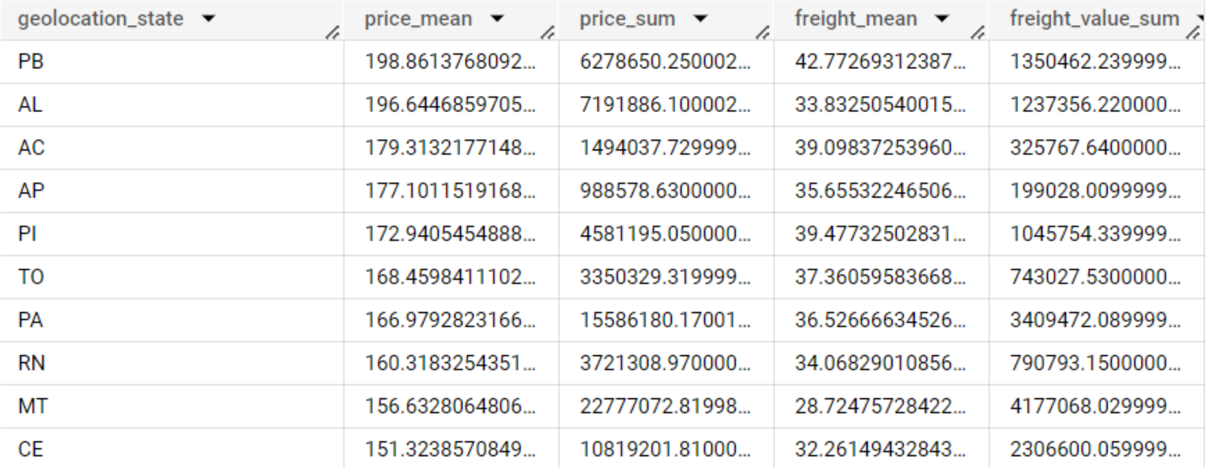
```
select distinct concat(Extract(Year from O.order_purchase_timestamp),'-',
Extract (Month from O.order_purchase_timestamp)) as year_month,
count(distinct O.order_id) as no_of_orders, round(sum(oI.price), 1) as
sales,
round(sum(oI.freight_value), 1) as freight_value from `Ecommerce.orders` O
join `Ecommerce.order_items` oI on oI.order_id=O.order_id
group by year_month having year_month >= '2017-1' and year_month <= '2018-
8'
order by year_month
limit 10;
```



year_month	no_of_orders	sales	freight_value
2017-1	789	120312.9	16875.6
2017-10	4568	664219.4	105092.9
2017-11	7451	1010271.4	168872.4
2017-12	5624	743914.2	119633.1
2017-2	1733	247303.0	38977.6
2017-3	2641	374344.3	57704.3
2017-4	2391	359927.2	52495.0
2017-5	3660	506071.1	80119.8
2017-6	3217	433038.6	69924.4
2017-7	3969	498031.5	86940.1

b) Mean & Sum of price and freight value by customer state:

```
select distinct G.geolocation_state, avg(OI.price) as price_mean,
sum(OI.price) as price_sum, avg(OI.freight_value) as freight_mean,
sum(OI.freight_value) as freight_value_sum from `Ecommerce.orders` O
left join `Ecommerce.order_items` as OI on OI.order_id=O.order_id
join `Ecommerce.customers` as C on C.customer_id=O.customer_id
left join `Ecommerce.geolocation` G on
G.geolocation_zip_code_prefix=C.customer_zip_code_prefix
group by G.geolocation_state order by price_mean desc
limit 10;
```

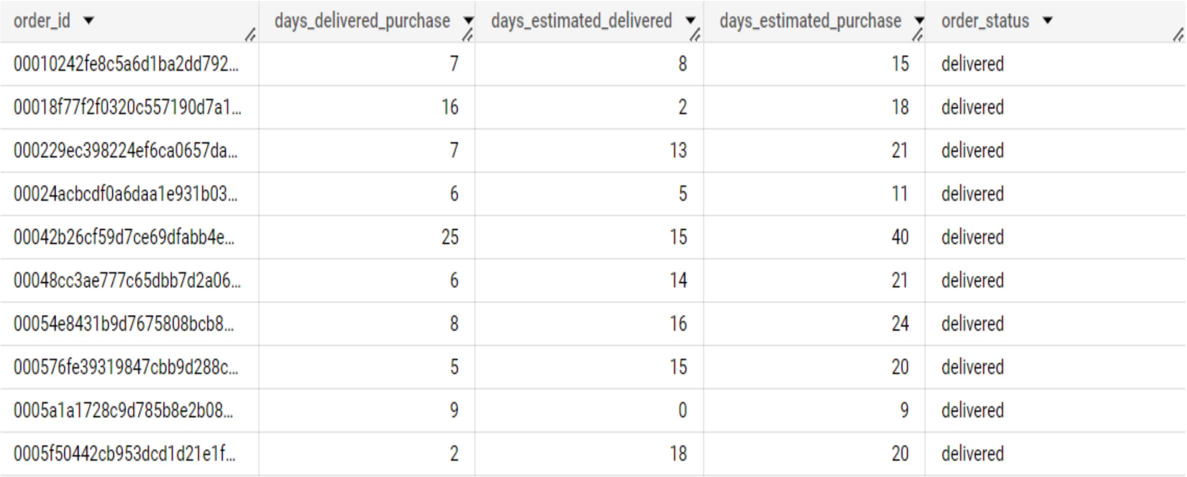


geolocation_state	price_mean	price_sum	freight_mean	freight_value_sum
PB	198.8613768092...	6278650.250002...	42.77269312387...	1350462.239999...
AL	196.6446859705...	7191886.100002...	33.83250540015...	1237356.220000...
AC	179.3132177148...	1494037.729999...	39.09837253960...	325767.6400000...
AP	177.1011519168...	988578.6300000...	35.65532246506...	199028.0099999...
PI	172.9405454888...	4581195.050000...	39.47732502831...	1045754.339999...
TO	168.4598411102...	3350329.319999...	37.36059583668...	743027.5300000...
PA	166.9792823166...	15586180.17001...	36.52666634526...	3409472.089999...
RN	160.3183254351...	3721308.970000...	34.06829010856...	790793.1500000...
MT	156.6328064806...	22777072.81998...	28.72475728422...	4177068.029999...
CE	151.3238570849...	10819201.81000...	32.26149432843...	2306600.059999...

5) Analysis on sales, freight and delivery time:

a) Calculate days between purchasing, delivering and estimated delivery :

```
select  order_id,
DATETIME_DIFF(order_delivered_customer_date,order_purchase_timestamp,day)
as days_delivered_purchase,
DATETIME_DIFF(order_estimated_delivery_date,order_delivered_customer_date,
day) as days_estimated_delivered,
DATETIME_DIFF(order_estimated_delivery_date,order_purchase_timestamp,day)
as days_estimated_purchase,
order_status from `Ecommerce.orders`
order by order_id
limit 10;
```

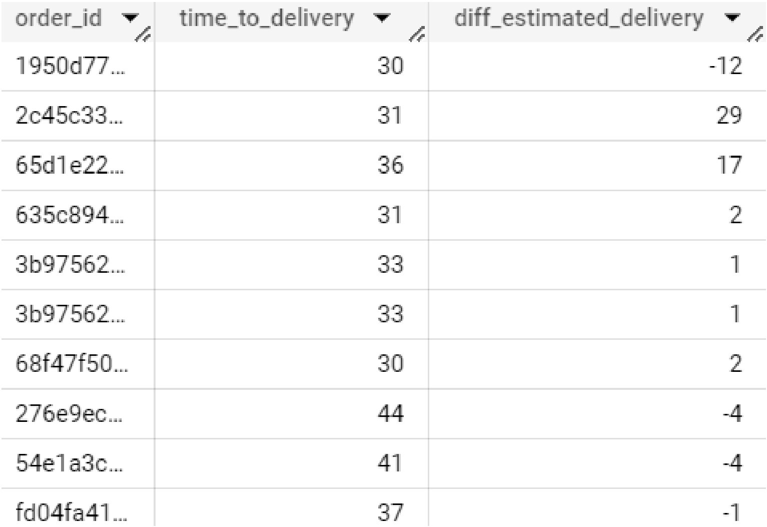


order_id	days_delivered_purchase	days_estimated_delivered	days_estimated_purchase	order_status
00010242fe8c5a6d1ba2dd792...	7	8	15	delivered
00018f77f2f0320c557190d7a1...	16	2	18	delivered
000229ec398224ef6ca0657da...	7	13	21	delivered
00024acbcdf0a6daa1e931b03...	6	5	11	delivered
00042b26cf59d7ce69dfabb4e...	25	15	40	delivered
00048cc3ae777c65dbb7d2a06...	6	14	21	delivered
00054e8431b9d7675808bcb8...	8	16	24	delivered
000576fe39319847cbb9d288c...	5	15	20	delivered
0005a1a1728c9d785b8e2b08...	9	0	9	delivered
0005f50442cb953dcd1d21e1f...	2	18	20	delivered

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

- time_to_delivery = order_delivered_customer_date-order_purchase_timestamp
- diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

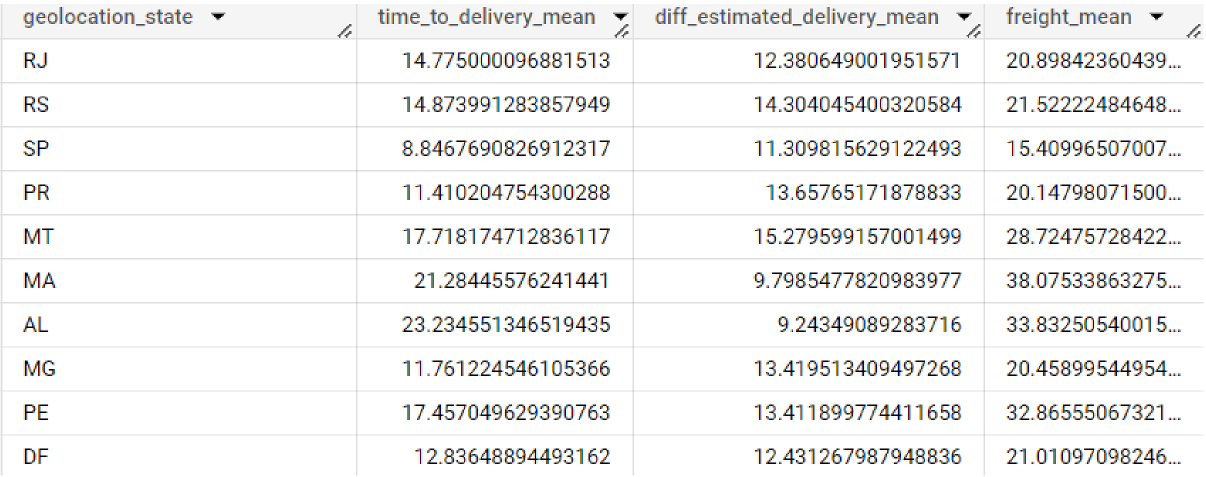
```
select O.order_id,
extract(day from (date(O.order_delivered_customer_date)-
date(O.order_purchase_timestamp))) as time_to_delivery,
extract(day from (date(O.order_estimated_delivery_date)-
date(O.order_delivered_customer_date))) as diff_estimated_delivery
from `Ecommerce.orders` O
left join `Ecommerce.order_items` as OI on OI.order_id=O.order_id
```



order_id	time_to_delivery	diff_estimated_delivery
1950d77...	30	-12
2c45c33...	31	29
65d1e22...	36	17
635c894...	31	2
3b97562...	33	1
3b97562...	33	1
68f47f50...	30	2
276e9ec...	44	-4
54e1a3c...	41	-4
fd04fa41...	37	-1

c)Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery:

```
select distinct G.geolocation_state,
avg(extract(day from (date(O.order_delivered_customer_date)-
date(O.order_purchase_timestamp)))) as time_to_delivery_mean,
avg(extract(day from (date(O.order_estimated_delivery_date)-
date(O.order_delivered_customer_date)))) as diff_estimated_delivery_mean,
avg(OI.freight_value) as freight_mean
from `Ecommerce.orders` O
left join `Ecommerce.order_items` as OI on OI.order_id=O.order_id
join `Ecommerce.customers` as C on C.customer_id=O.customer_id
left join `Ecommerce.geolocation` G
on G.geolocation_zip_code_prefix=C.customer_zip_code_prefix
group by G.geolocation_state having G.geolocation_state is not null
limit 10;
```



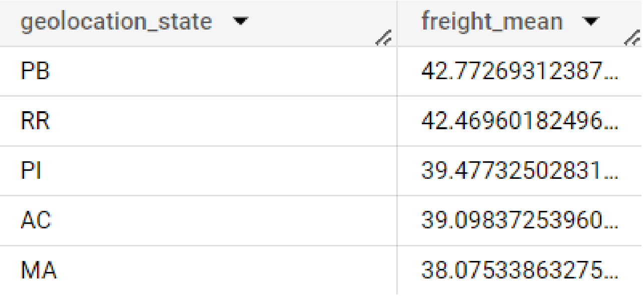
geolocation_state	time_to_delivery_mean	diff_estimated_delivery_mean	freight_mean
RJ	14.775000096881513	12.380649001951571	20.89842360439...
RS	14.873991283857949	14.304045400320584	21.52222484648...
SP	8.8467690826912317	11.309815629122493	15.40996507007...
PR	11.410204754300288	13.65765171878833	20.14798071500...
MT	17.718174712836117	15.279599157001499	28.72475728422...
MA	21.28445576241441	9.7985477820983977	38.07533863275...
AL	23.234551346519435	9.24349089283716	33.83250540015...
MG	11.761224546105366	13.419513409497268	20.45899544954...
PE	17.457049629390763	13.411899774411658	32.86555067321...
DF	12.83648894493162	12.431267987948836	21.01097098246...

d) Sort the data to get the following:

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

(i)Top 5 States with Highest freight value:

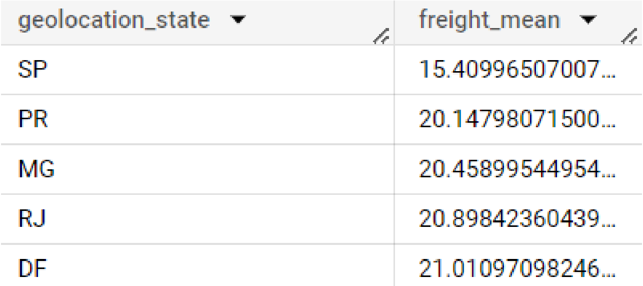
```
select distinct G.geolocation_state, avg(OI.freight_value) as freight_mean
from `Ecommerce.orders` O
left join `Ecommerce.order_items` as OI on OI.order_id=O.order_id
join `Ecommerce.customers` as C on C.customer_id=O.customer_id
left join `Ecommerce.geolocation` G on
G.geolocation_zip_code_prefix=C.customer_zip_code_prefix
group by G.geolocation_state having G.geolocation_state is not null
order by freight_mean desc
limit 5;
```



geolocation_state	freight_mean
PB	42.77269312387...
RR	42.46960182496...
PI	39.47732502831...
AC	39.09837253960...
MA	38.07533863275...

(ii)Top 5 States with Lowest freight value:

```
select distinct G.geolocation_state, avg(OI.freight_value) as freight_mean
from `Ecommerce.orders` O
left join `Ecommerce.order_items` as OI on OI.order_id=O.order_id
join `Ecommerce.customers` as C on C.customer_id=O.customer_id
left join `Ecommerce.geolocation` G on
G.geolocation_zip_code_prefix=C.customer_zip_code_prefix
group by G.geolocation_state having G.geolocation_state is not null
order by freight_mean asc
limit 5;
```

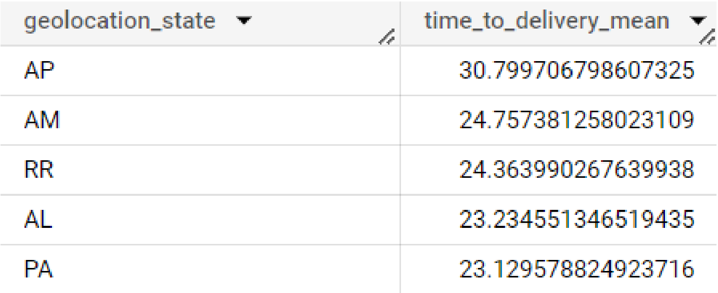


geolocation_state	freight_mean
SP	15.40996507007...
PR	20.14798071500...
MG	20.45899544954...
RJ	20.89842360439...
DF	21.01097098246...

e) Top 5 states with highest/lowest average time to delivery:

Highest state:

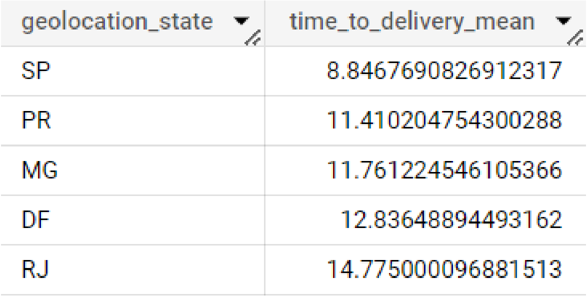
```
select distinct G.geolocation_state, avg(extract(day from
(date(O.order_delivered_customer_date)-
date(O.order_purchase_timestamp)))) as
time_to_delivery_mean from `Ecommerce.orders` O
left join `Ecommerce.order_items` as OI on OI.order_id=O.order_id
join `Ecommerce.customers` as C on C.customer_id=O.customer_id
left join `Ecommerce.geolocation` G on
G.geolocation_zip_code_prefix=C.customer_zip_code_prefix group by
G.geolocation_state
having G.geolocation_state is not null
order by time_to_delivery_mean desc
limit 5;
```



geolocation_state	time_to_delivery_mean
AP	30.799706798607325
AM	24.757381258023109
RR	24.363990267639938
AL	23.234551346519435
PA	23.129578824923716

Lowest state:

```
select distinct G.geolocation_state, avg(extract(day from
(date(O.order_delivered_customer_date)-
date(O.order_purchase_timestamp)))) as time_to_delivery_mean
from `Ecommerce.orders` O
left join `Ecommerce.order_items` as OI on OI.order_id=O.order_id
join `Ecommerce.customers` as C on C.customer_id=O.customer_id
left join `Ecommerce.geolocation` G on
G.geolocation_zip_code_prefix=C.customer_zip_code_prefix
group by G.geolocation_state having G.geolocation_state is not null
order by time_to_delivery_mean asc
limit 5;
```

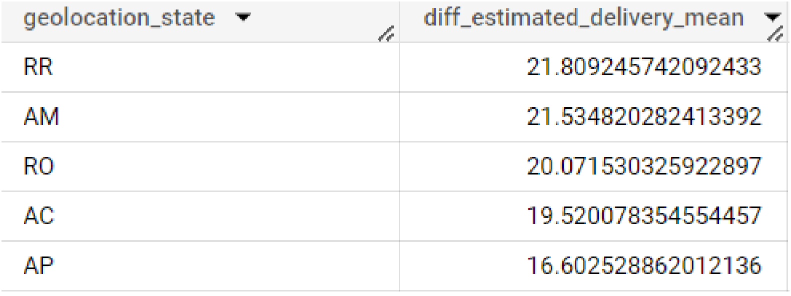


geolocation_state	time_to_delivery_mean
SP	8.8467690826912317
PR	11.410204754300288
MG	11.761224546105366
DF	12.83648894493162
RJ	14.775000096881513

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

(i)Top 5 with Fast Delivery:

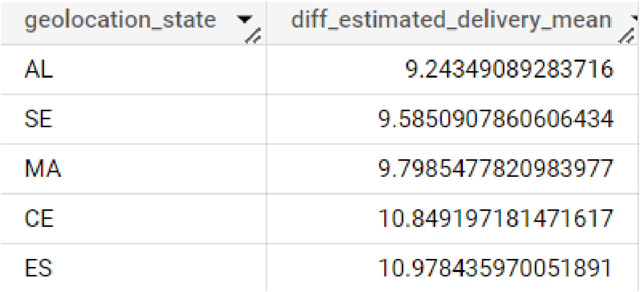
```
select distinct G.geolocation_state, avg(extract(day from
(date(O.order_estimated_delivery_date)-
date(O.order_delivered_customer_date)))) as diff_estimated_delivery_mean
from `Ecommerce.orders` O
left join `Ecommerce.order_items` as OI on OI.order_id=O.order_id
join `Ecommerce.customers` as C on C.customer_id=O.customer_id
left join `Ecommerce.geolocation` G on
G.geolocation_zip_code_prefix=C.customer_zip_code_prefix
group by G.geolocation_state having G.geolocation_state is not null
order by diff_estimated_delivery_mean desc
limit 5;
```



geolocation_state	diff_estimated_delivery_mean
RR	21.809245742092433
AM	21.534820282413392
RO	20.071530325922897
AC	19.520078354554457
AP	16.602528862012136

(ii)Top 5 with Slow Delivery:

```
select distinct G.geolocation_state, avg(extract(day from
(date(O.order_estimated_delivery_date)-
date(O.order_delivered_customer_date)))) as diff_estimated_delivery_mean
from `Ecommerce.orders` O
left join `Ecommerce.order_items` as OI on OI.order_id=O.order_id
join `Ecommerce.customers` as C on C.customer_id=O.customer_id
left join `Ecommerce.geolocation` G on
G.geolocation_zip_code_prefix=C.customer_zip_code_prefix
group by G.geolocation_state having G.geolocation_state is not null
order by diff_estimated_delivery_mean asc
limit 5;
```



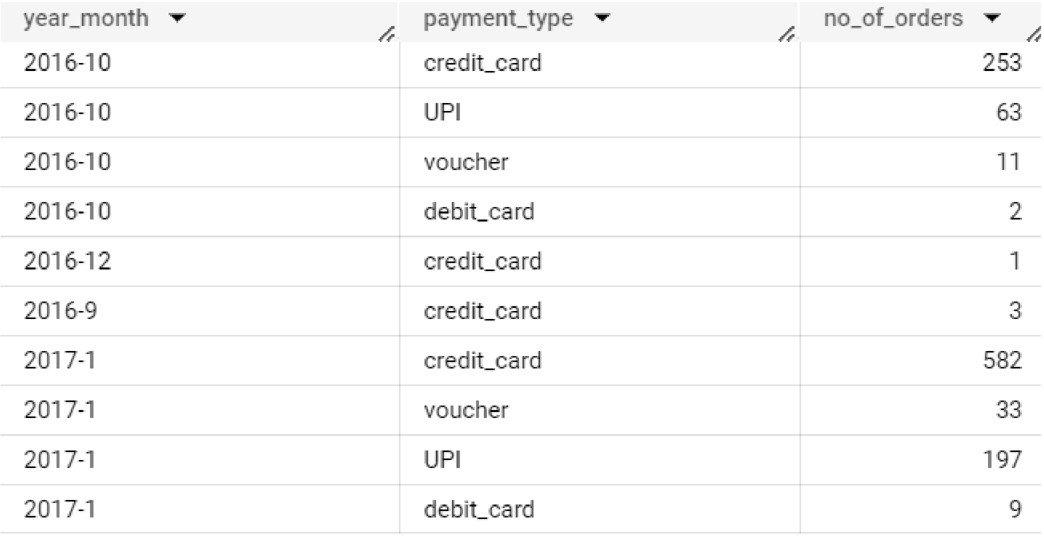
geolocation_state	diff_estimated_delivery_mean
AL	9.24349089283716
SE	9.5850907860606434
MA	9.7985477820983977
CE	10.849197181471617
ES	10.978435970051891

6) Payment type analysis:

a) Month over Month count of orders for different payment types:

```
select distinct concat(extract(Year from O.order_purchase_timestamp),'-',
extract(Month from O.order_purchase_timestamp))as year_month,
P.payment_type,
count(distinct O.order_id) as no_of_orders
```

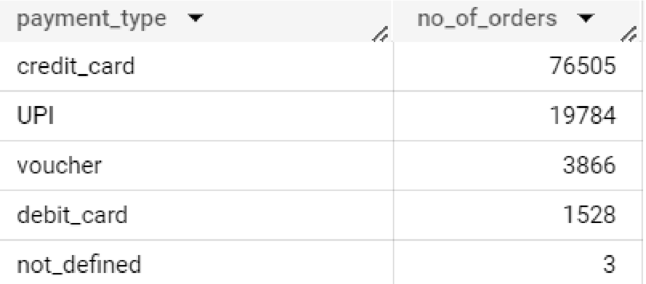
```
from `Ecommerce.orders` O
left join `Ecommerce.order_items` as OI on OI.order_id=O.order_id
join `Ecommerce.payments` P on P.order_id=O.order_id
group by year_month, P.payment_type order by year_month
limit 10;
```



year_month	payment_type	no_of_orders
2016-10	credit_card	253
2016-10	UPI	63
2016-10	voucher	11
2016-10	debit_card	2
2016-12	credit_card	1
2016-9	credit_card	3
2017-1	credit_card	582
2017-1	voucher	33
2017-1	UPI	197
2017-1	debit_card	9

b) Count of orders based on the no. of payment instalments:

```
select distinct P.payment_type, count(distinct O.order_id) as no_of_orders
from `Ecommerce.orders` O
left join `Ecommerce.order_items` as OI on OI.order_id=O.order_id
join `Ecommerce.payments` P on P.order_id=O.order_id
group by P.payment_type order by no_of_orders desc;
```



payment_type	no_of_orders
credit_card	76505
UPI	19784
voucher	3866
debit_card	1528
not_defined	3

ACTIONABLE INSIGHTS :

- From the findings of payments table we get to know that credit card payments are more often in brazil.

- Payments made by debit card is showing a growing trend since 2018-05, which is a good opportunity for investor to improve services for payments like this.

RECOMMENDATIONS:

- **If we look at the average time to carrier to start the delivery is around 2-3 days, this should be optimized as low as possible, and that can result into faster delivery.**

- **From the analysis we observe the average time to complete the delivery is 12days, as there is high competition in e-commerce market, is should be reduced to half.**

- **The delivery is really slow when compared to estimate date at Top States, delivering faster may create and increase new customers and revenue.**

- **It was observed an increasing trend in revenue and orders over time, but there was decrease in order during September and October month, Introducing discount or offer during low going month.**

- **We need to explore options for same-day or next-day delivery services to stay competitive in the market.**

- **In Month on Month orders by states, no of orders are certainly lower in states like GO,DF,SC,BA where it must be given attention to increase the orders by taking required actions.**