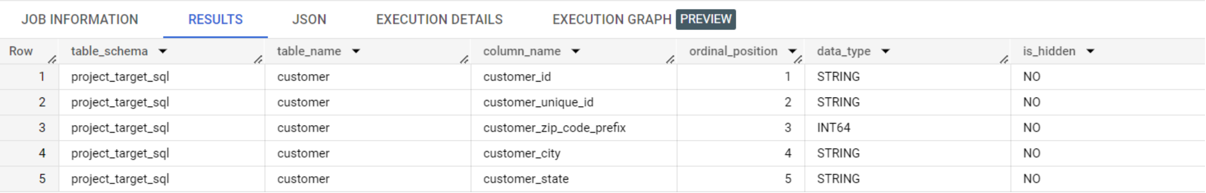
Name: Pritam kudale
Topic: Project - Business Case: Target SQL
Batch: March 2023

1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset**
    1. **Data type of columns in a table**

Query : For customer Table

```
SELECT
  table_schema, table_name, column_name, ordinal_position, data_type, is_hidden
FROM `project-387005.project_target_sql.INFORMATION_SCHEMA.COLUMNS`
where table_name = 'customer';
```

Output

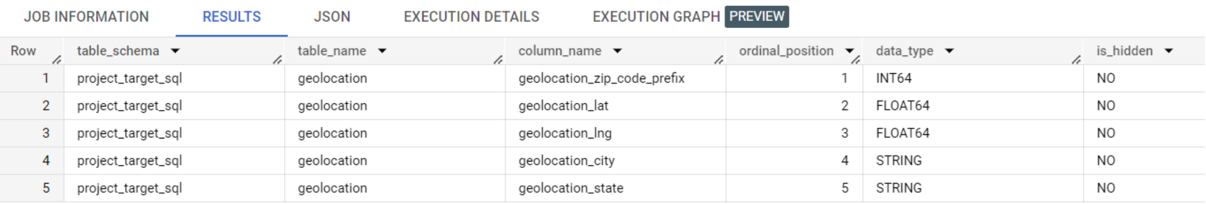


Row	table_schema	table_name	column_name	ordinal_position	data_type	is_hidden
1	project_target_sql	customer	customer_id	1	STRING	NO
2	project_target_sql	customer	customer_unique_id	2	STRING	NO
3	project_target_sql	customer	customer_zip_code_prefix	3	INT64	NO
4	project_target_sql	customer	customer_city	4	STRING	NO
5	project_target_sql	customer	customer_state	5	STRING	NO

Query : For geolocation Table

```
SELECT
  table_schema, table_name, column_name, ordinal_position, data_type, is_hidden
FROM  `project-387005.project_target_sql.INFORMATION_SCHEMA.COLUMNS`
where  table_name = 'geolocation';
```

Output

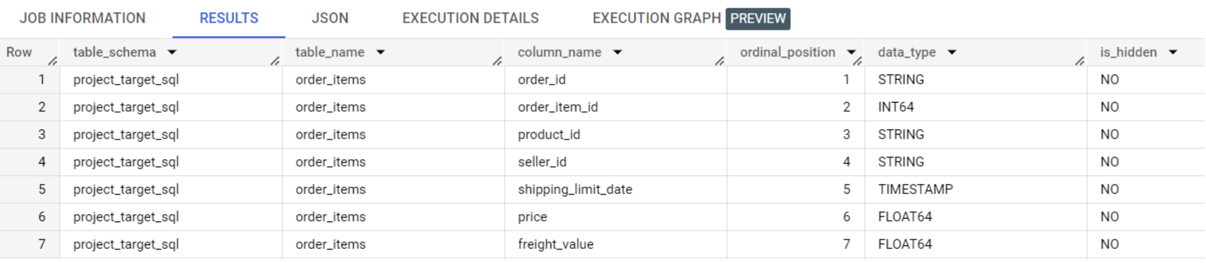


Row	table_schema	table_name	column_name	ordinal_position	data_type	is_hidden
1	project_target_sql	geolocation	geolocation_zip_code_prefix	1	INT64	NO
2	project_target_sql	geolocation	geolocation_lat	2	FLOAT64	NO
3	project_target_sql	geolocation	geolocation_lng	3	FLOAT64	NO
4	project_target_sql	geolocation	geolocation_city	4	STRING	NO
5	project_target_sql	geolocation	geolocation_state	5	STRING	NO

Query : For order_items Table

```
SELECT
  table_schema, table_name, column_name, ordinal_position, data_type, is_hidden
FROM `project-387005.project_target_sql.INFORMATION_SCHEMA.COLUMNS`
where  table_name = 'order_items';
```

Output

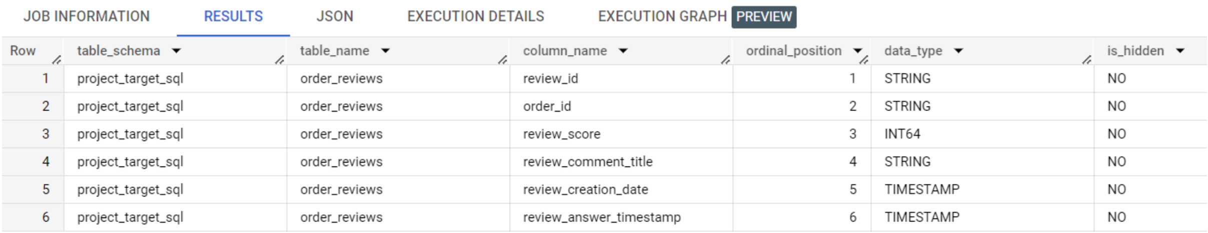


Row	table_schema	table_name	column_name	ordinal_position	data_type	is_hidden
1	project_target_sql	order_items	order_id	1	STRING	NO
2	project_target_sql	order_items	order_item_id	2	INT64	NO
3	project_target_sql	order_items	product_id	3	STRING	NO
4	project_target_sql	order_items	seller_id	4	STRING	NO
5	project_target_sql	order_items	shipping_limit_date	5	TIMESTAMP	NO
6	project_target_sql	order_items	price	6	FLOAT64	NO
7	project_target_sql	order_items	freight_value	7	FLOAT64	NO

Query : For order_reviews Table

```
SELECT
  table_schema, table_name, column_name, ordinal_position, data_type, is_hidden
FROM  `project-387005.project_target_sql.INFORMATION_SCHEMA.COLUMNS`
where  table_name = 'order_reviews';
```

Output

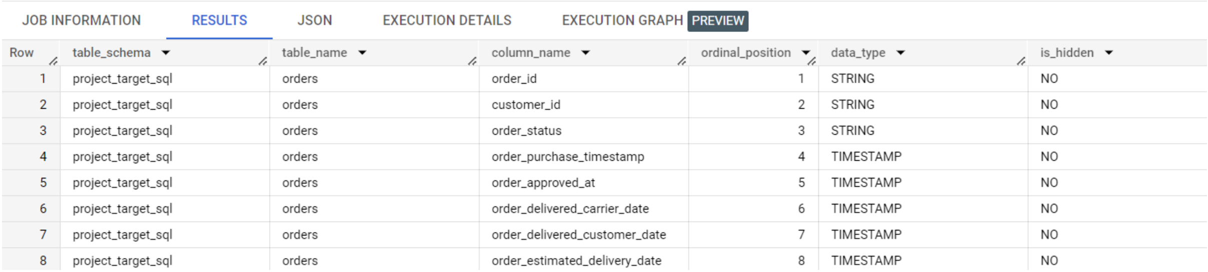


Row	table_schema	table_name	column_name	ordinal_position	data_type	is_hidden
1	project_target_sql	order_reviews	review_id	1	STRING	NO
2	project_target_sql	order_reviews	order_id	2	STRING	NO
3	project_target_sql	order_reviews	review_score	3	INT64	NO
4	project_target_sql	order_reviews	review_comment_title	4	STRING	NO
5	project_target_sql	order_reviews	review_creation_date	5	TIMESTAMP	NO
6	project_target_sql	order_reviews	review_answer_timestamp	6	TIMESTAMP	NO

Query : For orders Table

```
SELECT
  table_schema, table_name, column_name, ordinal_position, data_type, is_hidden
FROM
  `project-387005.project_target_sql.INFORMATION_SCHEMA.COLUMNS`
where
  table_name = 'orders';
```

Output

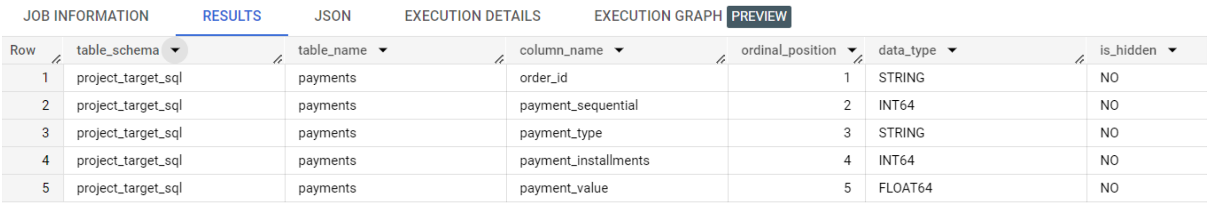


Row	table_schema	table_name	column_name	ordinal_position	data_type	is_hidden
1	project_target_sql	orders	order_id	1	STRING	NO
2	project_target_sql	orders	customer_id	2	STRING	NO
3	project_target_sql	orders	order_status	3	STRING	NO
4	project_target_sql	orders	order_purchase_timestamp	4	TIMESTAMP	NO
5	project_target_sql	orders	order_approved_at	5	TIMESTAMP	NO
6	project_target_sql	orders	order_delivered_carrier_date	6	TIMESTAMP	NO
7	project_target_sql	orders	order_delivered_customer_date	7	TIMESTAMP	NO
8	project_target_sql	orders	order_estimated_delivery_date	8	TIMESTAMP	NO

Query : For payments Table

```
SELECT
  table_schema, table_name, column_name, ordinal_position, data_type, is_hidden
FROM
  `project-387005.project_target_sql.INFORMATION_SCHEMA.COLUMNS`
where
  table_name = 'payments';
```

Output

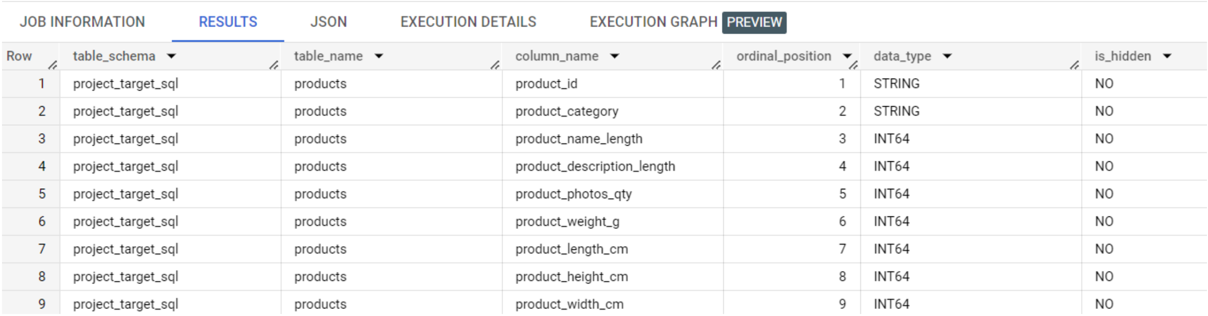


Row	table_schema	table_name	column_name	ordinal_position	data_type	is_hidden
1	project_target_sql	payments	order_id	1	STRING	NO
2	project_target_sql	payments	payment_sequential	2	INT64	NO
3	project_target_sql	payments	payment_type	3	STRING	NO
4	project_target_sql	payments	payment_installments	4	INT64	NO
5	project_target_sql	payments	payment_value	5	FLOAT64	NO

Query : For products Table

```
SELECT
  table_schema, table_name, column_name, ordinal_position, data_type, is_hidden
FROM
  `project-387005.project_target_sql.INFORMATION_SCHEMA.COLUMNS`
where
  table_name = 'products';
```

Output

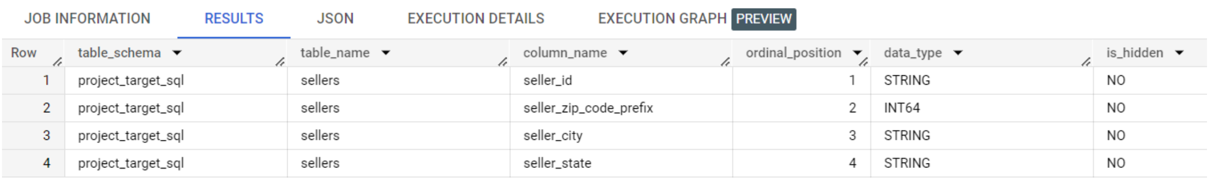


Row	table_schema	table_name	column_name	ordinal_position	data_type	is_hidden
1	project_target_sql	products	product_id	1	STRING	NO
2	project_target_sql	products	product_category	2	STRING	NO
3	project_target_sql	products	product_name_length	3	INT64	NO
4	project_target_sql	products	product_description_length	4	INT64	NO
5	project_target_sql	products	product_photos_qty	5	INT64	NO
6	project_target_sql	products	product_weight_g	6	INT64	NO
7	project_target_sql	products	product_length_cm	7	INT64	NO
8	project_target_sql	products	product_height_cm	8	INT64	NO
9	project_target_sql	products	product_width_cm	9	INT64	NO

Query : For sellers Table

```
SELECT
  table_schema, table_name, column_name, ordinal_position, data_type, is_hidden
FROM
  `project-387005.project_target_sql.INFORMATION_SCHEMA.COLUMNS`
where
  table_name = 'sellers';
```

Output



Row	table_schema	table_name	column_name	ordinal_position	data_type	is_hidden
1	project_target_sql	sellers	seller_id	1	STRING	NO
2	project_target_sql	sellers	seller_zip_code_prefix	2	INT64	NO
3	project_target_sql	sellers	seller_city	3	STRING	NO
4	project_target_sql	sellers	seller_state	4	STRING	NO

Insights:

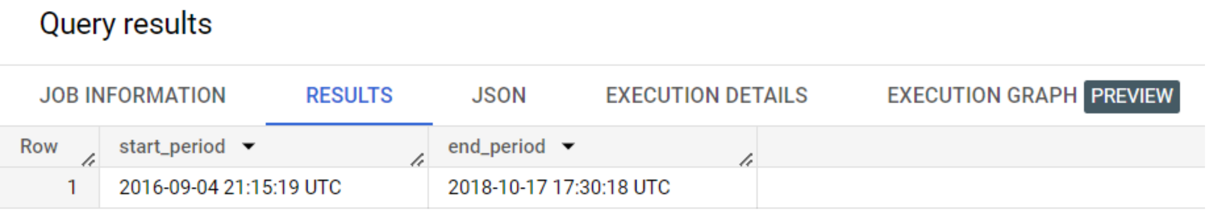
All dates have a timestamp format.

All tables have a unique primary key.

2. Time period for which the data is given

Query :

```
select
  min(order_purchase_timestamp) as start_period,
  max(order_purchase_timestamp) as end_period,
  date_diff(max(order_purchase_timestamp),min(order_purchase_timestamp),day) as
no_of_days
from
  `project_target_sql.orders`;
```



Query results

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

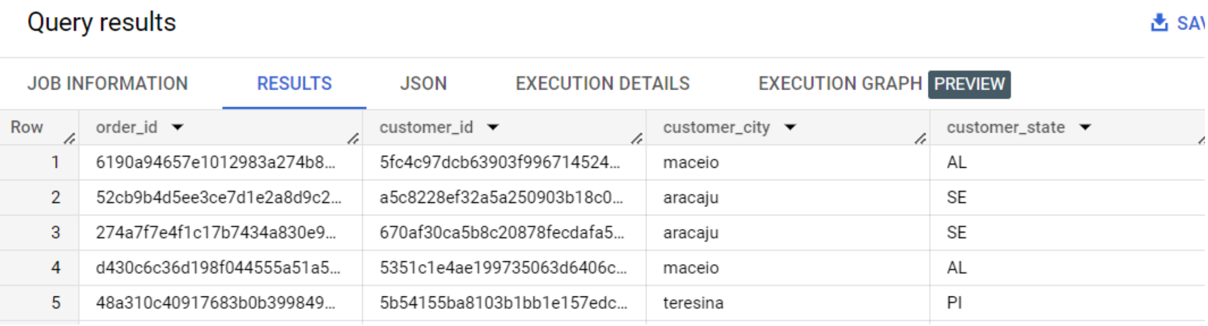
Insights:

```
Start period is 4th Sep 2016
End period is 17 Oct 2018
```

3. Cities and States of customers ordered during the given period

Query :

```
select
  o.order_id,
  o.customer_id,
  c.customer_city,
  c.customer_state,
from
  `project_target_sql.orders` o join `project_target_sql.customer` c ON
o.customer_id = c.customer_id;
```



Query results

Row	order_id	customer_id	customer_city	customer_state
1	6190a94657e1012983a274b8...	5fc4c97dcb63903f996714524...	maceio	AL
2	52cb9b4d5ee3ce7d1e2a8d9c2...	a5c8228ef32a5a250903b18c0...	aracaju	SE
3	274a7f7e4f1c17b7434a830e9...	670af30ca5b8c20878fecdafa5...	aracaju	SE
4	d430c6c36d198f044555a51a5...	5351c1e4ae199735063d6406c...	maceio	AL
5	48a310c40917683b0b399849...	5b54155ba8103b1bb1e157edc...	teresina	PI

Insights:

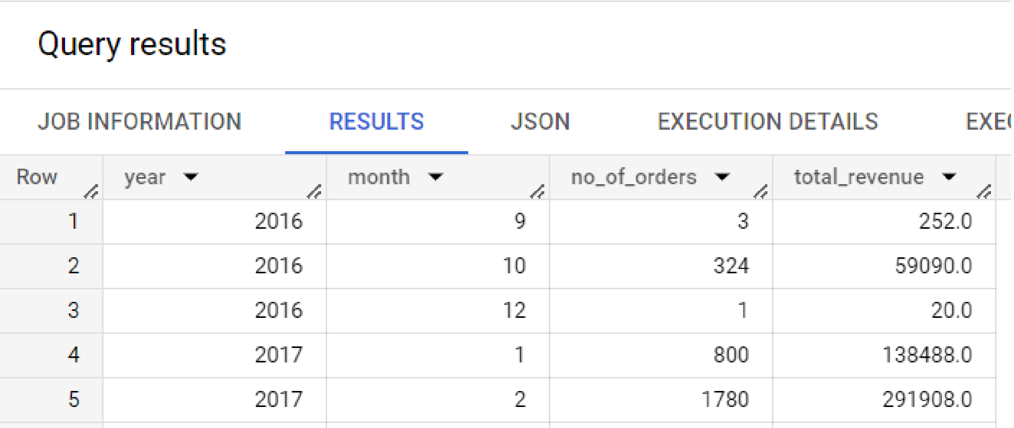
```
sao paulo city  has highest orders
SP state has highest orders
```

2. In-depth Exploration:

1. 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?

Query :

```
with CTE as (
select
  o.order_id,
  extract(month from o.order_purchase_timestamp) as month,
  extract(year from o.order_purchase_timestamp) as year,
  p.payment_value
from
  `project_target_sql.orders` o join `project_target_sql.payments` p on o.order_id
= p.order_id)
select
  year,month,
  count(distinct order_id) as no_of_orders,
  round(sum(payment_value)) as total_revenue
from
  CTE
group by year,month
order by year,month;
```



Row	year	month	no_of_orders	total_revenue
1	2016	9	3	252.0
2	2016	10	324	59090.0
3	2016	12	1	20.0
4	2017	1	800	138488.0
5	2017	2	1780	291908.0

Insights:

The growing trend in the number of orders is observed from Dec 2016 till Nov 2017 but unable to maintain growth after that.

The number of orders slightly decreased after March 2018 but maintained it above 6000 per month till August 2018.

Orders drastically reduced after November 2018.

Peak is observed in November 2017.

No specific seasonality is observed.

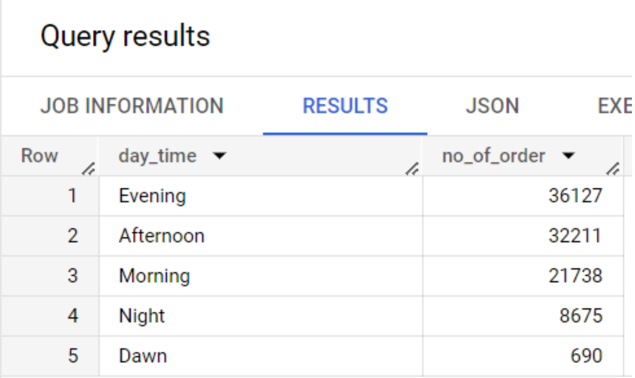
2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

By considering

Dawn	- 5:00:00 to 6:59:59
Morning	- 7:00:00 to 11:59:59
Afternoon	- 12:00:00 to 16:59:59
Evening	- 17:00:00 to 22:59:59
Night	- 23:00:00 to 4:59:59

Query :

```
with time_CTE as(
with CTE as (
select
  order_id,
  extract(time from order_purchase_timestamp) as time
from `project_target_sql.orders`)
select
  case
    when time between '5:00:00' and '6:59:59' then 'Dawn'
    when time between '7:00:00' and '11:59:59' then 'Morning'
    when time between '12:00:00'and '16:59:59' then 'Afternoon'
    when time between '17:00:00' and '22:59:59' then 'Evening'
    else 'Night'
  end as day_time,
  order_id
from CTE)
select
  day_time,
  count(distinct order_id) as no_of_order
from time_CTE
group by day_time
order by no_of_order desc;
```



Row	day_time	no_of_order
1	Evening	36127
2	Afternoon	32211
3	Morning	21738
4	Night	8675
5	Dawn	690

Insights:

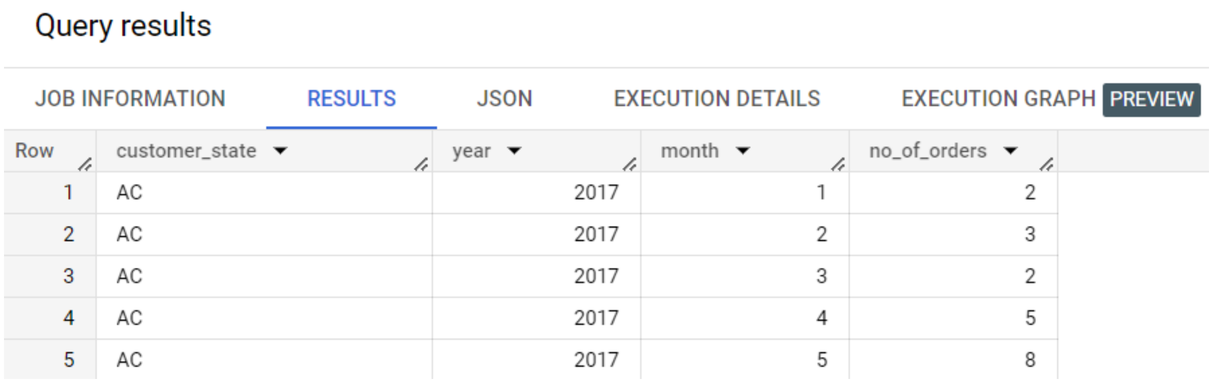
Afternoon and Evening is peak time when Brazilian customers tend to buy.
Maximum orders are observed in Evening.

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

1. Get month on month orders by states

Query :

```
with order_CTE as (
select
  o.order_id,
  extract(month from o.order_purchase_timestamp) as month,
  extract(year from o.order_purchase_timestamp) as year,
  c.customer_state
from
  `project_target_sql.orders` o join `project_target_sql.customer` c on
o.customer_id = c.customer_id)
select
  customer_state,
  year,
  month,
  count(order_id) as no_of_orders
from order_CTE
group by customer_state,year,month
order by customer_state,year,month;
```



Row	customer_state	year	month	no_of_orders
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8

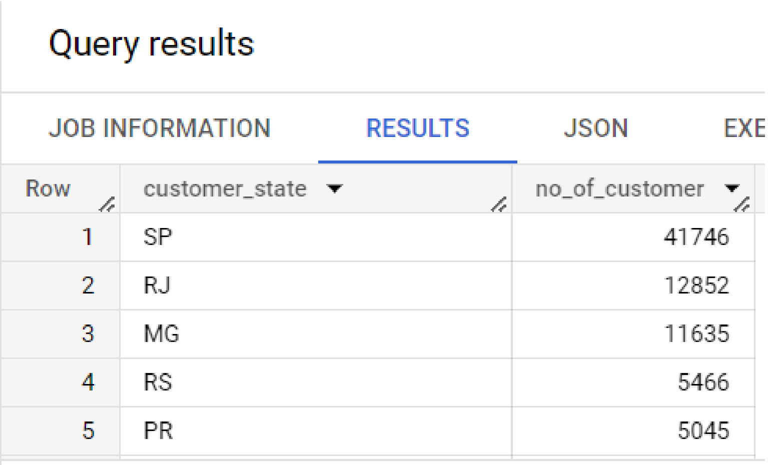
Insights:

Maximum orders are by state SP in the month of August 2018.

2. Distribution of customers across the states in Brazil

Query :

```
select
  customer_state,
  count(customer_unique_id) as no_of_customer
from
`project_target_sql.customer`
group by customer_state
order by no_of_customer desc;
```



Row	customer_state	no_of_customer
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045

Insights:

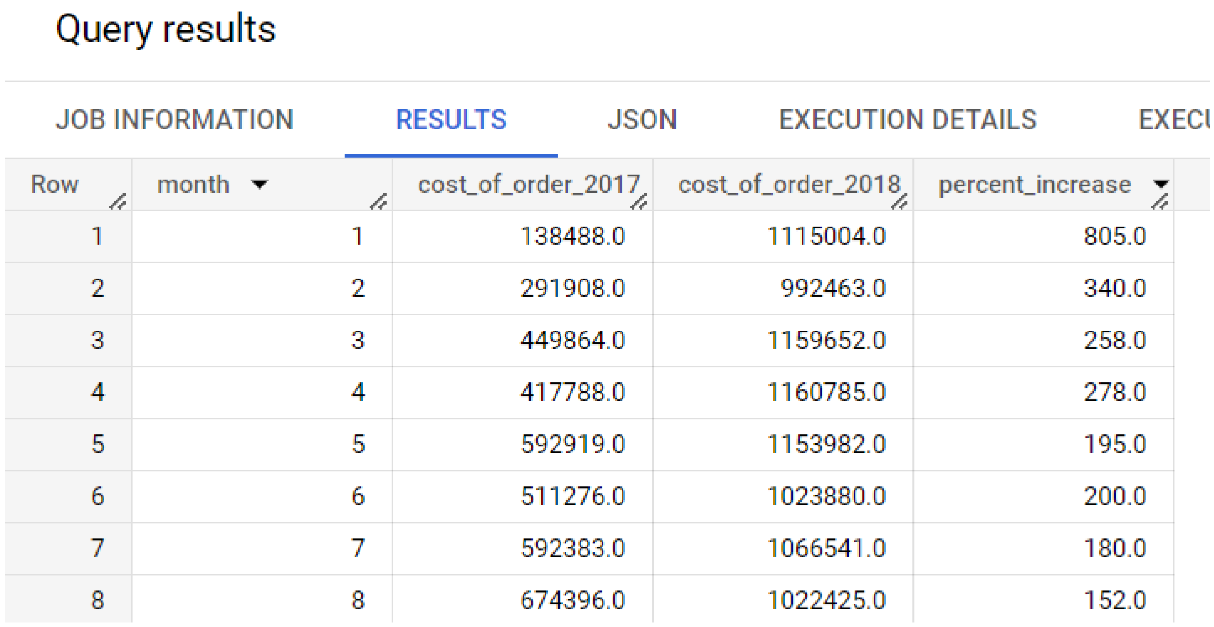
SP has maximum customers.

RR has minimum customers.

4. **Impact on Economy: Analyse the money movement by e-commerce by looking at order prices, freight and others.**
    1. **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**

Query :

```
select
 Y17.month,
 y17.total_revenue as cost_of_order_2017,
 y18.total_revenue as cost_of_order_2018,
 round((y18.total_revenue/y17.total_revenue) *100) as percent_increase
from
(with CTE as (
  select
    o.order_id,
    extract(month from o.order_purchase_timestamp) as month,
    extract(year from o.order_purchase_timestamp) as year,
    p.payment_value
  from
    `project_target_sql.orders` o join `project_target_sql.payments` p on
o.order_id = p.order_id
  where o.order_purchase_timestamp between '2017-01-01 00:00:01 UTC' and
'2017-08-31 23:59:59 UTC')
  select month, year, round(sum(payment_value)) as total_revenue
  from  CTE
  group by month,year
  order by year,month) y17 join (with CTE as (
  select
  o.order_id,
    extract(month from o.order_purchase_timestamp) as month,
    extract(year from o.order_purchase_timestamp) as year,
    p.payment_value
  from
    `project_target_sql.orders` o join `project_target_sql.payments` p on
o.order_id = p.order_id
  where o.order_purchase_timestamp between '2018-01-01 00:00:01 UTC' and
'2018-08-31 23:59:59 UTC')
  select
    month, year,
    round(sum(payment_value)) as total_revenue
  from
    CTE
  group by month,year
  order by year,month) y18 ON y17.month = y18.month
  order by y17.month;
```



Row	month	cost_of_order_2017	cost_of_order_2018	percent_increase
1	1	138488.0	1115004.0	805.0
2	2	291908.0	992463.0	340.0
3	3	449864.0	1159652.0	258.0
4	4	417788.0	1160785.0	278.0
5	5	592919.0	1153982.0	195.0
6	6	511276.0	1023880.0	200.0
7	7	592383.0	1066541.0	180.0
8	8	674396.0	1022425.0	152.0

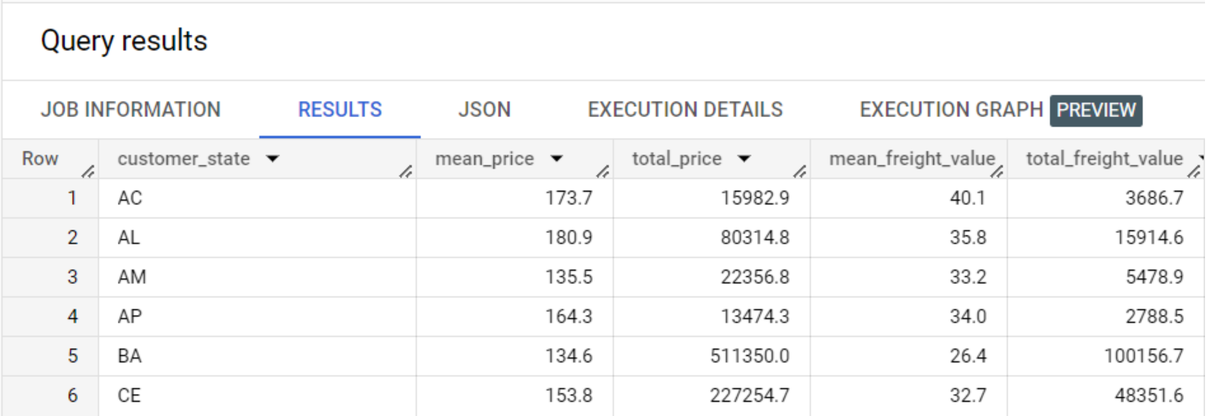
Insights:

```
January has the highest percent increase in 2018 over 2017.
August has the lowest percent increase in 2018 over 2017.
Every month has an increasing trend.
```

2. Mean & Sum of price and freight value by customer state

Query :

```
select
  c.customer_state,
  round(avg(oi.price),1) as mean_price,
  round(sum(oi.price),1) as total_price,
  round(avg(oi.freight_value),1) as mean_freight_value,
  round(sum(oi.freight_value),1) as total_freight_value
from
  `project_target_sql.orders` o
    join `project_target_sql.order_items` oi on o.order_id = oi.order_id
    join `project_target_sql.customer` c on o.customer_id = c.customer_id
group by c.customer_state
order by c.customer_state;
```



Row	customer_state	mean_price	total_price	mean_freight_value	total_freight_value
1	AC	173.7	15982.9	40.1	3686.7
2	AL	180.9	80314.8	35.8	15914.6
3	AM	135.5	22356.8	33.2	5478.9
4	AP	164.3	13474.3	34.0	2788.5
5	BA	134.6	511350.0	26.4	100156.7
6	CE	153.8	227254.7	32.7	48351.6

Insights:

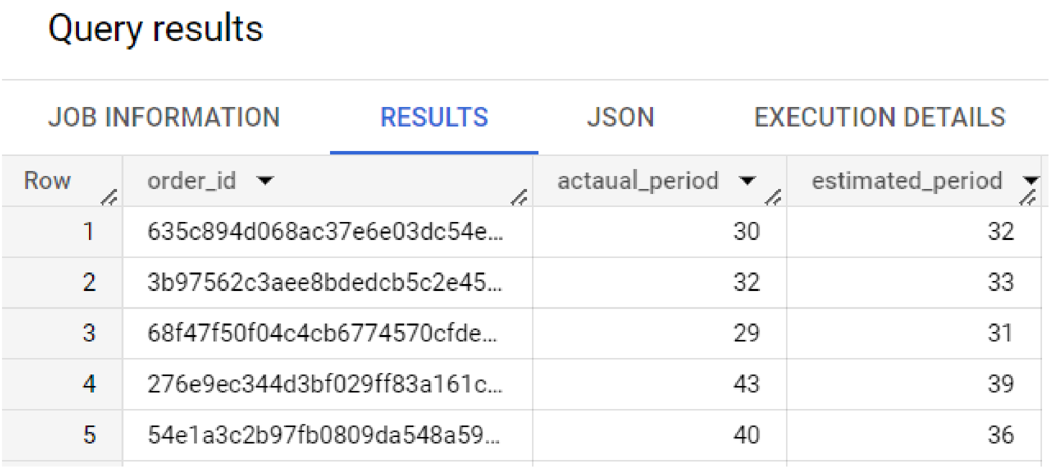
SP has maximum total price and total freight value whereas minimum mean price and mean freight value.

5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

Query :

```
select
  order_id,
  date_diff(order_delivered_customer_date, order_purchase_timestamp,day) as
actaual_period,
  date_diff(order_estimated_delivery_date, order_purchase_timestamp,day) as
estimated_period
from
  `project_target_sql.orders`
where
  order_status = 'delivered'and order_delivered_customer_date is not null;
```



Query results

Row	order_id	actaual_period	estimated_period
1	635c894d068ac37e6e03dc54e...	30	32
2	3b97562c3aee8bdedcb5c2e45...	32	33
3	68f47f50f04c4cb6774570cfde...	29	31
4	276e9ec344d3bf029ff83a161c...	43	39
5	54e1a3c2b97fb0809da548a59...	40	36

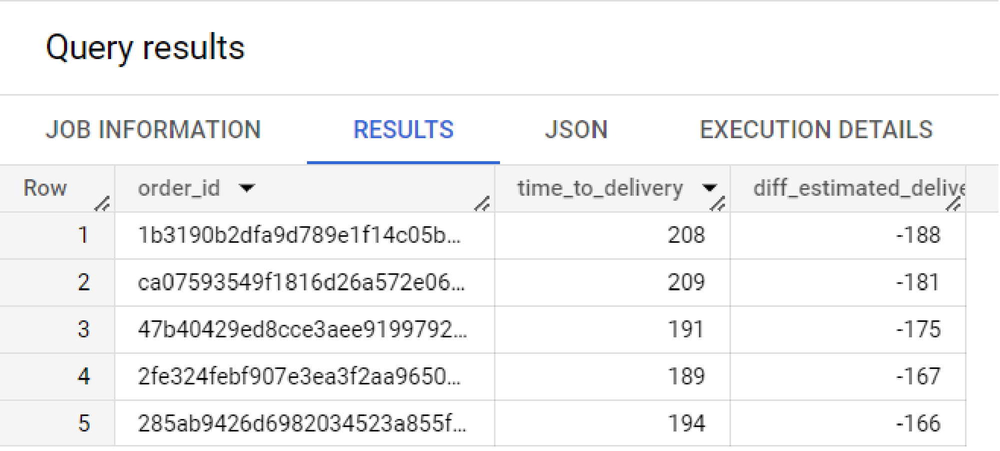
Insights:

Maximum Delivery days needed are 209 whereas fastest delivery is same day delivery.
Estimated delivery period is 2 to 155 days.

2. **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**

Query :

```
select
  order_id,
  date_diff(order_delivered_customer_date, order_purchase_timestamp,day) as
time_to_delivery,
  date_diff(order_estimated_delivery_date, order_delivered_customer_date,day) as
diff_estimated_delivery
from
  `project_target_sql.orders`
where
  order_status = 'delivered'and order_delivered_customer_date is not null
order by
  diff_estimated_delivery;
```



Row	order_id	time_to_delivery	diff_estimated_delive
1	1b3190b2dfa9d789e1f14c05b...	208	-188
2	ca07593549f1816d26a572e06...	209	-181
3	47b40429ed8cce3aee9199792...	191	-175
4	2fe324febf907e3ea3f2aa9650...	189	-167
5	285ab9426d6982034523a855f...	194	-166

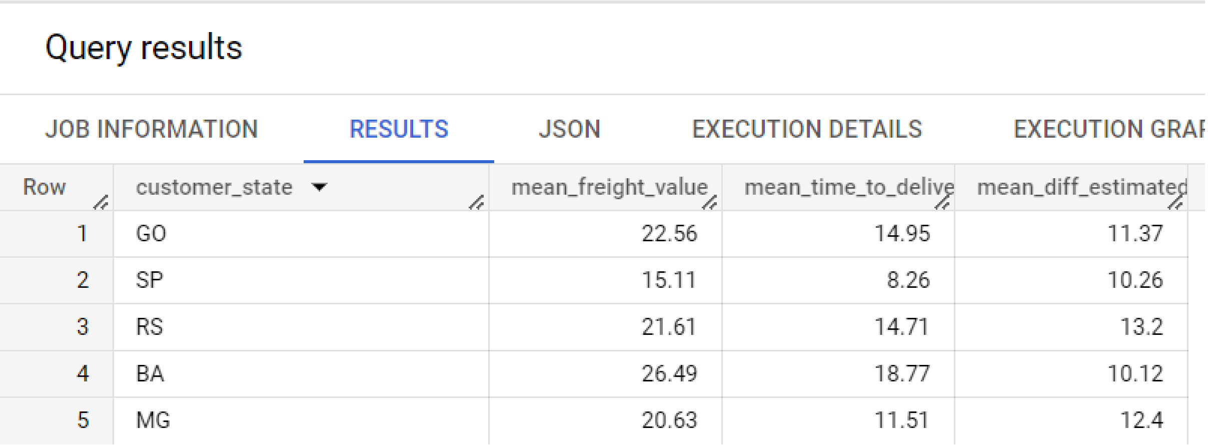
Insights:

Maximum delay period is 188 days.

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

Query :

```
select
  c.customer_state,
  round(avg(oi.freight_value),2) as mean_freight_value,
  round(avg(date_diff(o.order_delivered_customer_date,
o.order_purchase_timestamp,day)),2) as mean_time_to_delivery,
  round(avg(date_diff(o.order_estimated_delivery_date,
o.order_delivered_customer_date,day)),2) as mean_diff_estimated_delivery
from
  `project_target_sql.orders` o
    join `project_target_sql.customer` c on o.customer_id = c.customer_id
    join `project_target_sql.order_items` oi on o.order_id = oi.order_id
where
  o.order_status = 'delivered'and o.order_delivered_customer_date is not null
group by
  c.customer_state;
```



Query results

Row	customer_state	mean_freight_value	mean_time_to_delive	mean_diff_estimated
1	GO	22.56	14.95	11.37
2	SP	15.11	8.26	10.26
3	RS	21.61	14.71	13.2
4	BA	26.49	18.77	10.12
5	MG	20.63	11.51	12.4

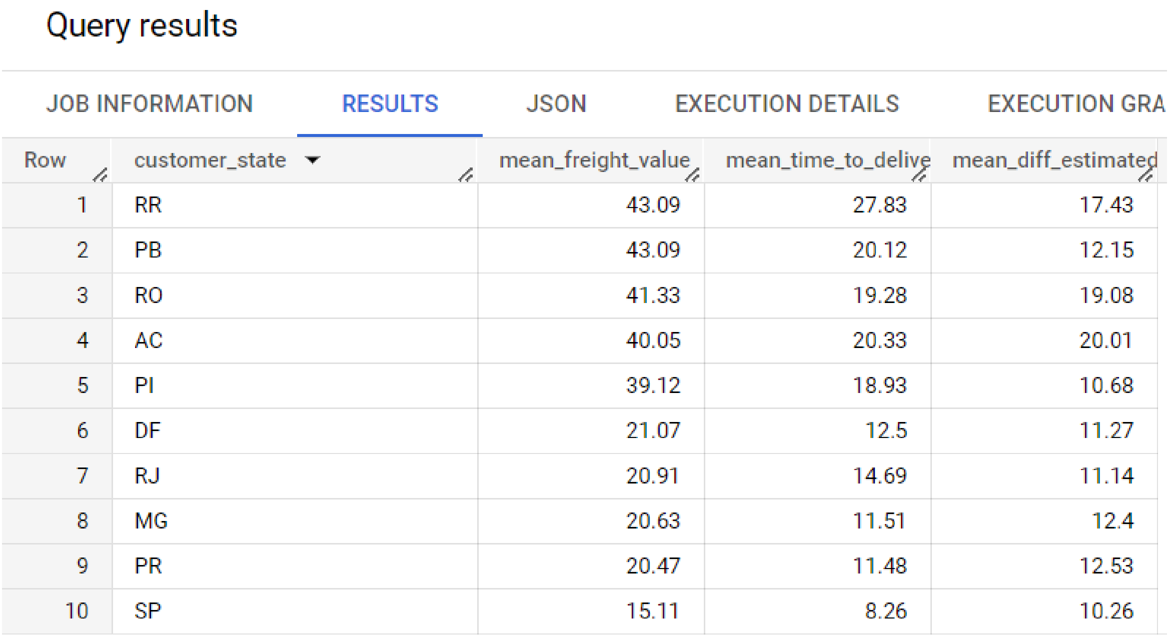
Insights:

All mean time differences are positive indicating that most of the deliveries are before estimated time.

4. Sort the data to get the following:
    - Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

Query :

```
select * from
(select
  c.customer_state,
  round(avg(oi.freight_value),2) as mean_freight_value,
  round(avg(date_diff(o.order_delivered_customer_date,
o.order_purchase_timestamp,day)),2) as mean_time_to_delivery,
  round(avg(date_diff(o.order_estimated_delivery_date,
o.order_delivered_customer_date,day)),2) as mean_diff_estimated_delivery
from
  `project_target_sql.orders` o
    join `project_target_sql.customer` c on o.customer_id = c.customer_id
    join `project_target_sql.order_items` oi on o.order_id = oi.order_id
where
  o.order_status = 'delivered'and o.order_delivered_customer_date is not null
group by
  c.customer_state
order by
  mean_freight_value desc
limit 5) top
UNION all
select * from
(select
  c.customer_state,
  round(avg(oi.freight_value),2) as mean_freight_value,
  round(avg(date_diff(o.order_delivered_customer_date,
o.order_purchase_timestamp,day)),2) as mean_time_to_delivery,
  round(avg(date_diff(o.order_estimated_delivery_date,
o.order_delivered_customer_date,day)),2) as mean_diff_estimated_delivery
from
  `project_target_sql.orders` o
    join `project_target_sql.customer` c on o.customer_id = c.customer_id
    join `project_target_sql.order_items` oi on o.order_id = oi.order_id
where
  o.order_status = 'delivered'and o.order_delivered_customer_date is not null
group by
  c.customer_state
order by
  mean_freight_value
limit 5) bottom order by mean_freight_value desc;
```



Row	customer_state	mean_freight_value	mean_time_to_delive	mean_diff_estimated
1	RR	43.09	27.83	17.43
2	PB	43.09	20.12	12.15
3	RO	41.33	19.28	19.08
4	AC	40.05	20.33	20.01
5	PI	39.12	18.93	10.68
6	DF	21.07	12.5	11.27
7	RJ	20.91	14.69	11.14
8	MG	20.63	11.51	12.4
9	PR	20.47	11.48	12.53
10	SP	15.11	8.26	10.26

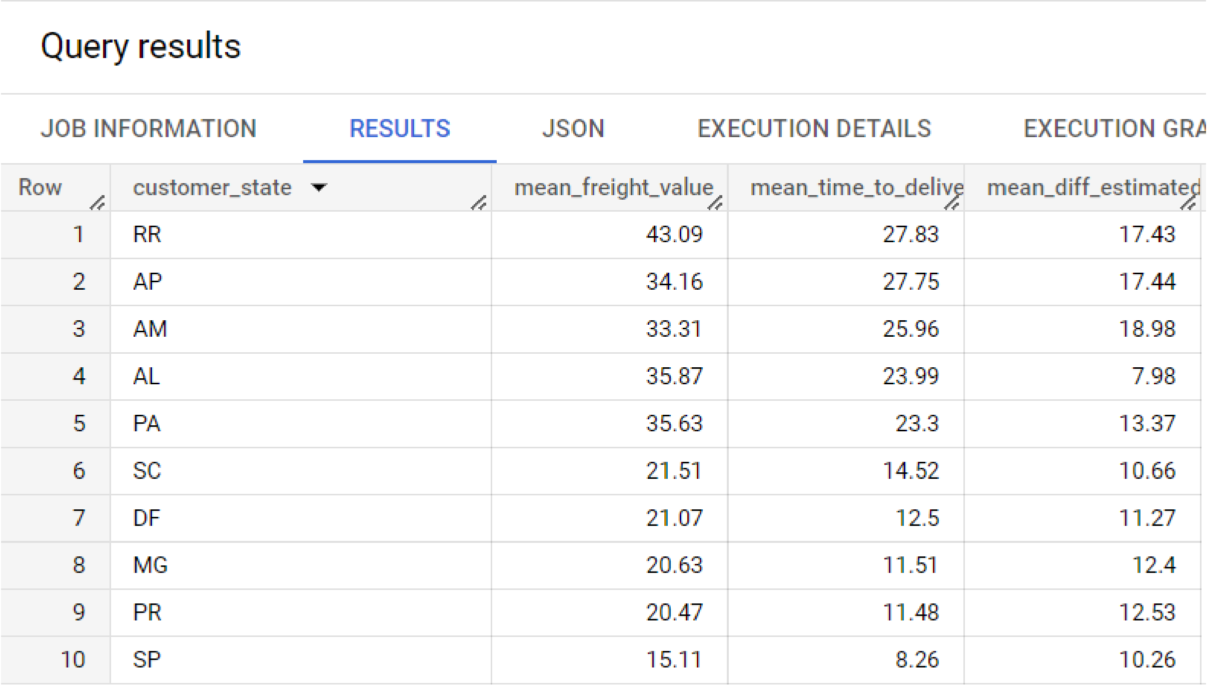
Insights:

RR, PB, RO, AC, PI are top 5 states for max average freight value.
SP, PR, MG, RJ, DF are having lowest average freight value

5. Top 5 states with highest/lowest average time to delivery

Query :

```
select * from
(select
  c.customer_state,
  round(avg(oi.freight_value),2) as mean_freight_value,
  round(avg(date_diff(o.order_delivered_customer_date,
o.order_purchase_timestamp,day)),2) as mean_time_to_delivery,
  round(avg(date_diff(o.order_estimated_delivery_date,
o.order_delivered_customer_date,day)),2) as mean_diff_estimated_delivery
from
  `project_target_sql.orders` o
    join `project_target_sql.customer` c on o.customer_id = c.customer_id
    join `project_target_sql.order_items` oi on o.order_id = oi.order_id
where
  o.order_status = 'delivered'and o.order_delivered_customer_date is not null
group by
  c.customer_state
order by
  mean_time_to_delivery desc
limit 5) top
UNION all
select * from
(select
  c.customer_state,
  round(avg(oi.freight_value),2) as mean_freight_value,
  round(avg(date_diff(o.order_delivered_customer_date,
o.order_purchase_timestamp,day)),2) as mean_time_to_delivery,
  round(avg(date_diff(o.order_estimated_delivery_date,
o.order_delivered_customer_date,day)),2) as mean_diff_estimated_delivery
from
  `project_target_sql.orders` o
    join `project_target_sql.customer` c on o.customer_id = c.customer_id
    join `project_target_sql.order_items` oi on o.order_id = oi.order_id
where
  o.order_status = 'delivered'and o.order_delivered_customer_date is not null
group by
  c.customer_state
order by
  mean_time_to_delivery
limit 5) bottom order by mean_time_to_delivery desc;
```



Row	customer_state	mean_freight_value	mean_time_to_delive	mean_diff_estimated
1	RR	43.09	27.83	17.43
2	AP	34.16	27.75	17.44
3	AM	33.31	25.96	18.98
4	AL	35.87	23.99	7.98
5	PA	35.63	23.3	13.37
6	SC	21.51	14.52	10.66
7	DF	21.07	12.5	11.27
8	MG	20.63	11.51	12.4
9	PR	20.47	11.48	12.53
10	SP	15.11	8.26	10.26

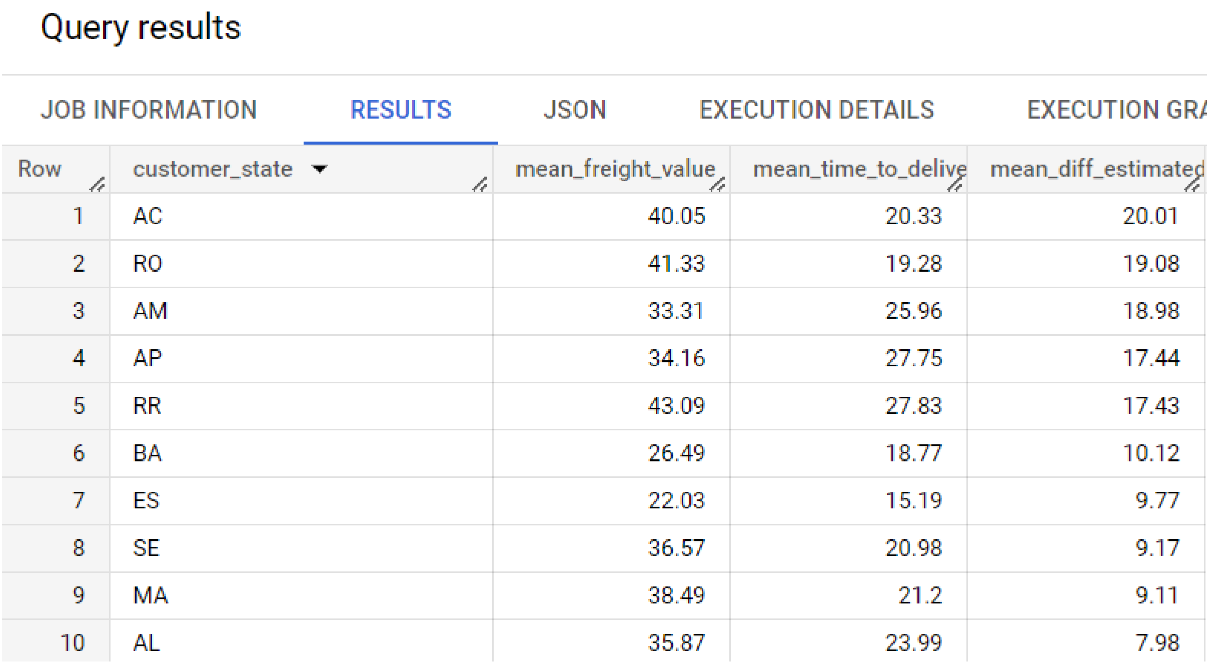
Insights:

```
RR, AP, AM, AL, PA are top 5 states for highest average time to delivery.
SP, PR, MG, DF, SC are having the lowest average time to delivery.
Lowest time to delivery in SP also has the lowest mean freight value.
```

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

Query :

```
select * from
(select
  c.customer_state,
  round(avg(oi.freight_value),2) as mean_freight_value,
  round(avg(date_diff(o.order_delivered_customer_date,
o.order_purchase_timestamp,day)),2) as mean_time_to_delivery,
  round(avg(date_diff(o.order_estimated_delivery_date,
o.order_delivered_customer_date,day)),2) as mean_diff_estimated_delivery
from
  `project_target_sql.orders` o
    join `project_target_sql.customer` c on o.customer_id = c.customer_id
    join `project_target_sql.order_items` oi on o.order_id = oi.order_id
where
  o.order_status = 'delivered'and o.order_delivered_customer_date is not null
group by
  c.customer_state
order by
  mean_diff_estimated_delivery desc
limit 5) top
UNION all
select * from
(select
  c.customer_state,
  round(avg(oi.freight_value),2) as mean_freight_value,
  round(avg(date_diff(o.order_delivered_customer_date,
o.order_purchase_timestamp,day)),2) as mean_time_to_delivery,
  round(avg(date_diff(o.order_estimated_delivery_date,
o.order_delivered_customer_date,day)),2) as mean_diff_estimated_delivery
from
  `project_target_sql.orders` o
    join `project_target_sql.customer` c on o.customer_id = c.customer_id
    join `project_target_sql.order_items` oi on o.order_id = oi.order_id
where
  o.order_status = 'delivered'and o.order_delivered_customer_date is not null
group by
  c.customer_state
order by
  mean_diff_estimated_delivery
limit 5) bottom order by mean_diff_estimated_delivery desc;
```



Query results

Row	customer_state	mean_freight_value	mean_time_to_delive	mean_diff_estimated
1	AC	40.05	20.33	20.01
2	RO	41.33	19.28	19.08
3	AM	33.31	25.96	18.98
4	AP	34.16	27.75	17.44
5	RR	43.09	27.83	17.43
6	BA	26.49	18.77	10.12
7	ES	22.03	15.19	9.77
8	SE	36.57	20.98	9.17
9	MA	38.49	21.2	9.11
10	AL	35.87	23.99	7.98

Insights:

AP, RO, AM, AP, RR are top 5 states for highest positive difference in estimated and actual delivery. i.e. it provides the fastest delivery way before estimated time.

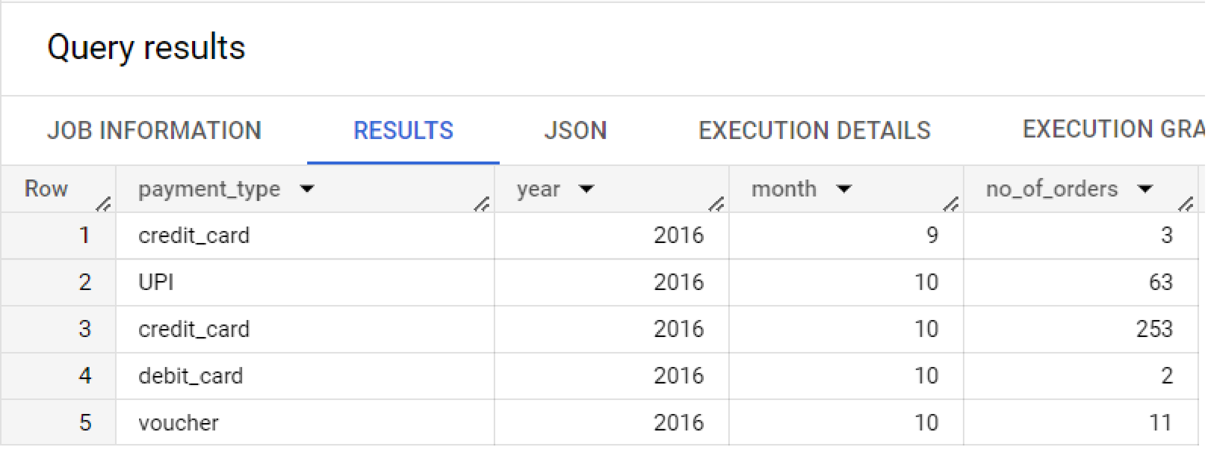
AL, MA, SE, ES, BA have the lowest positive difference in estimated and actual delivery.

6. Payment type analysis:

1. Month over Month count of orders for different payment types

Query :

```
with CTE as (
select
  o.order_id,
  p.payment_type,
  extract(month from o.order_purchase_timestamp) as month,
  extract(year from o.order_purchase_timestamp) as year,
  p.payment_value
from
  `project_target_sql.orders` o join `project_target_sql.payments` p on o.order_id
= p.order_id)
select
  payment_type,year,month,count(distinct order_id) as no_of_orders
from
  CTE
group by year,month,payment_type
order by year,month,payment_type;
```



Row	payment_type	year	month	no_of_orders
1	credit_card	2016	9	3
2	UPI	2016	10	63
3	credit_card	2016	10	253
4	debit_card	2016	10	2
5	voucher	2016	10	11

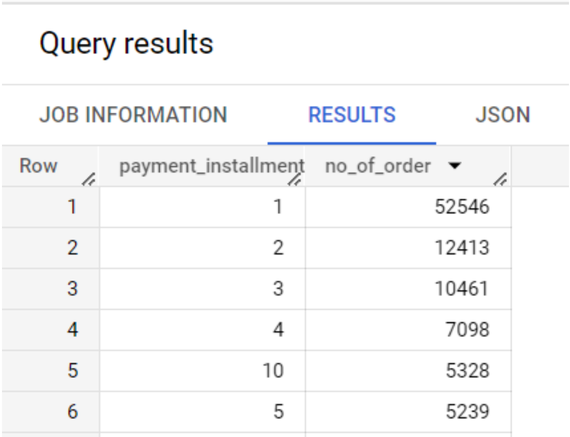
Insights:

Credit card is the most preferred mode of payment and maximum payment orders 5867 in Dec 2017 by credit card.

2. Count of orders based on the no. of payment installments

Query :

```
select payment_installments, count(order_id) as no_of_order
from `project_target_sql.payments`
group by payment_installments
order by no_of_order desc;
```



Query results

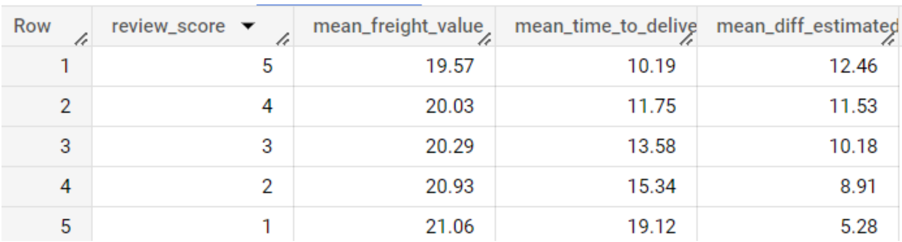
Row	payment_installment	no_of_order
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239

Insights:

```
Highest number of users use one time instalment.
1,2,3,4 and 10 are the most preferred instalments.
For higher amounts, higher instalments are chosen.
```

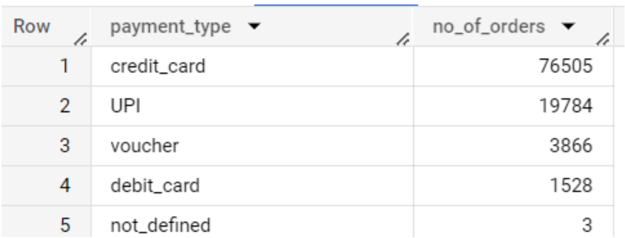
Overall recommendation

1. Order delivery is important for customer satisfaction and customers give better ratings when delivery time is less and product received way before estimated delivery date.



Row	review_score	mean_freight_value	mean_time_to_delive	mean_diff_estimated
1	5	19.57	10.19	12.46
2	4	20.03	11.75	11.53
3	3	20.29	13.58	10.18
4	2	20.93	15.34	8.91
5	1	21.06	19.12	5.28

2. Afternoon and Evening is the Peak time when customers tend to buy.
3. With increase in cost, customer instalment increases.
4. Credit cards are the most preferred mode of payment.



Row	payment_type	no_of_orders
1	credit_card	76505
2	UPI	19784
3	voucher	3866
4	debit_card	1528
5	not_defined	3

5. After April 2018, the number of orders are decreasing which needs to be focused.
6. SP state has the highest number of customer leads to lowest average freight value, In contrast RR state has lowest customer leads to highest average freight value.
7. Improving vise versa can help to reduce average freight value and increase the count of customers.