TARGET CASE STUDY :

BUSINESS Insights:

- Most of the sales were done in October and November in 2017.
- Most of the orders were ordered in between 12 and 6 pm
- There was 136 percent in cost of orders from 2017 to 2018
- Most of the orders delivered within 30 days
- Top 5 states with highest/lowest average freight value are RR,PB,RO,AC,PI
- Top 5 states with highest/lowest average time to delivery are RR,AP,AM,AL,PA.

RECOMMENDATIONS:

- Company should increase their sales in other months other than November by giving offers.
- Company should make their deliver reached to customers more faster.
- Company should focus more on the least avg freight value states.

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

Ans) CUSTOMERS

```
select table_name,column_name,data_type from `target-sql-casestudy-scaler.target.INFORMATION_SCHEMA.COLUMNS` where table_name='customers'
```

RUN SAVE SHARE SCHEDULE MORE

```
select table_name,column_name,data_type from `target-sql-casestudy-scaler.target.INFORMATION_SCHEMA.COLUMNS` where table_name='customers'
```

Press Alt+F1

Query results SAVE RESULTS EXPLOR

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

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

GEOLOCATION

```
select table_name,column_name,data_type from `target-sql-casestudy-scaler.target.INFORMATION_SCHEMA.COLUMNS` where table_name='geolocation'
```

Query results SAVE RESULTS EXPLORE DA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

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

ORDER_ITEMS

```
select table_name,column_name,data_type from `target-sql-casestudy-scaler.target.INFORMATION_SCHEMA.COLUMNS` where table_name='order_items'
```

RUN SAVE SHARE SCHEDULE MORE

```
select table_name,column_name,data_type from `target-sql-casestudy-scaler.target.INFORMATION_SCHEMA.COLUMNS` where table_name='order_items'
```

Press Alt+F1 for

Query results SAVE RESULTS EXPLORE I

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

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

ORDER_REVIEWS

```
select table_name,column_name,data_type from `target-sql-casestudy-scaler.target.INFORMATION_SCHEMA.COLUMNS` where table_name='order_reviews'
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

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

ORDERS

```
select table_name,column_name,data_type from `target-sql-casestudy-scaler.target.INFORMATION_SCHEMA.COLUMNS` where table_name='orders'
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

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

PAYMENTS

```
select table_name,column_name,data_type from `target-sql-casestudy-scaler.target.INFORMATION_SCHEMA.COLUMNS` where table_name='payments'
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

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

PRODUCTS

```
select table_name,column_name,data_type from `target-sql-casestudy-scaler.target.INFORMATION_SCHEMA.COLUMNS` where table_name='products'
```

RUN SAVE SHARE SCHEDULE MORE This query will pro

```
1 select table_name,column_name,data_type from `target-sql-casestudy-scaler.target.INFORMATION_SCHEMA.COLUMNS` where table_name='products'
```

Press Alt+F1

Query results SAVE RESULTS EXPLO

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

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

SELLERS

```
select table_name,column_name,data_type from `target-sql-casestudy-scaler.target.INFORMATION_SCHEMA.COLUMNS` where table_name='sellers'
```

```
1 select table_name,column_name,data_type from `target-sql-casestudy-scaler.target.INFORMATION_SCHEMA.COLUMNS` where table_name='sellers'
```

Press Alt+F1

Query results SAVE RESULTS EXPLO

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

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

ii) Time period for which the data is given

```
select min(order_purchase_timestamp) as min, max(order_purchase_timestamp) as max from `target.orders`
```

RUN SAVE SHARE SCHEDULE MORE

```
1 select min(order_purchase_timestamp) as min, max(order_purchase_timestamp) as max from `target.orders`
```

Processing location: us-central1

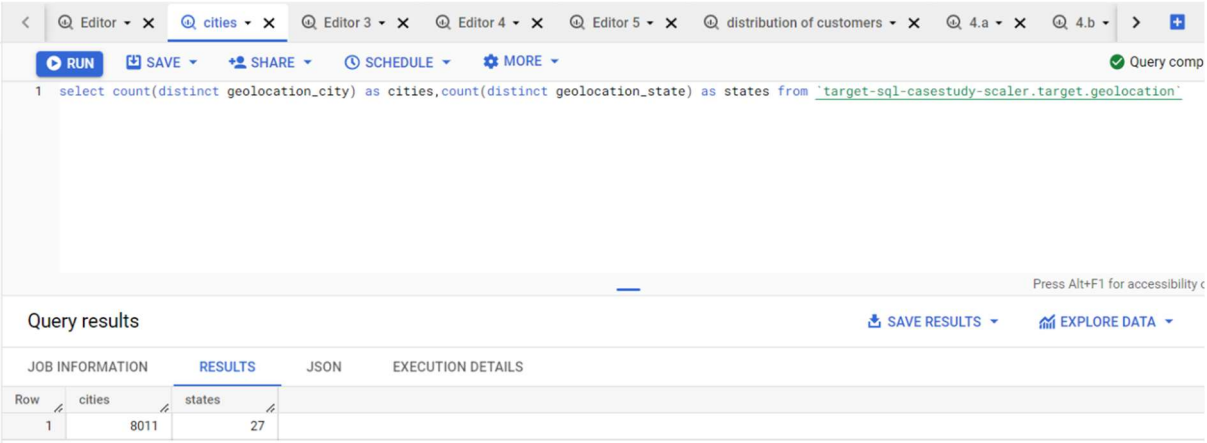
Query results SAVE RE

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

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

iii)Cities and States covered in the dataset

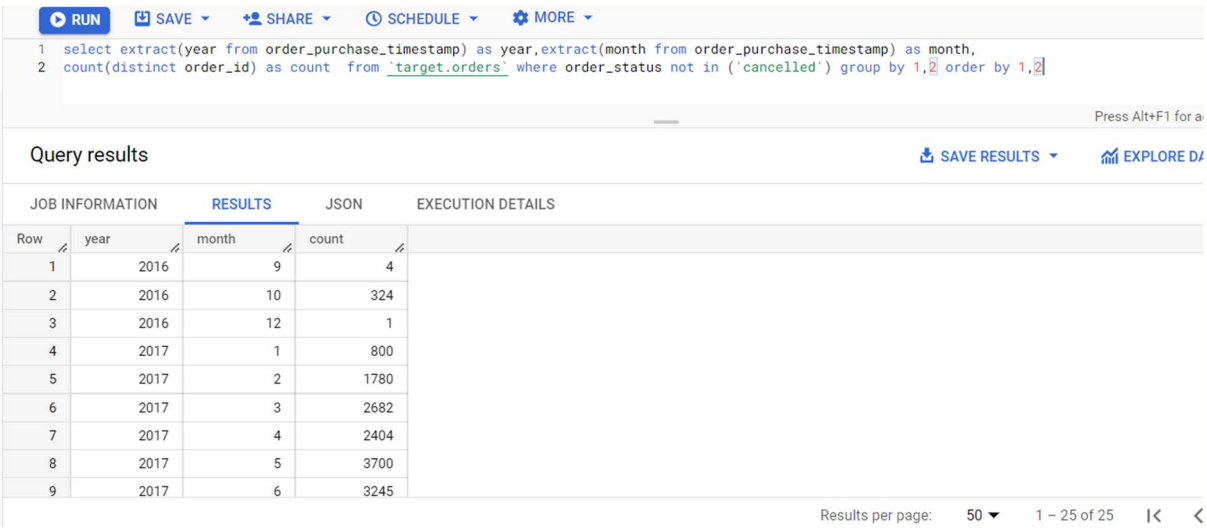
```
select count(distinct geolocation_city) as cities,count(distinct geolocation_state) as stat
es from `target-sql-casestudy-scaler.target.geolocation`
```



1) In-depth Exploration:

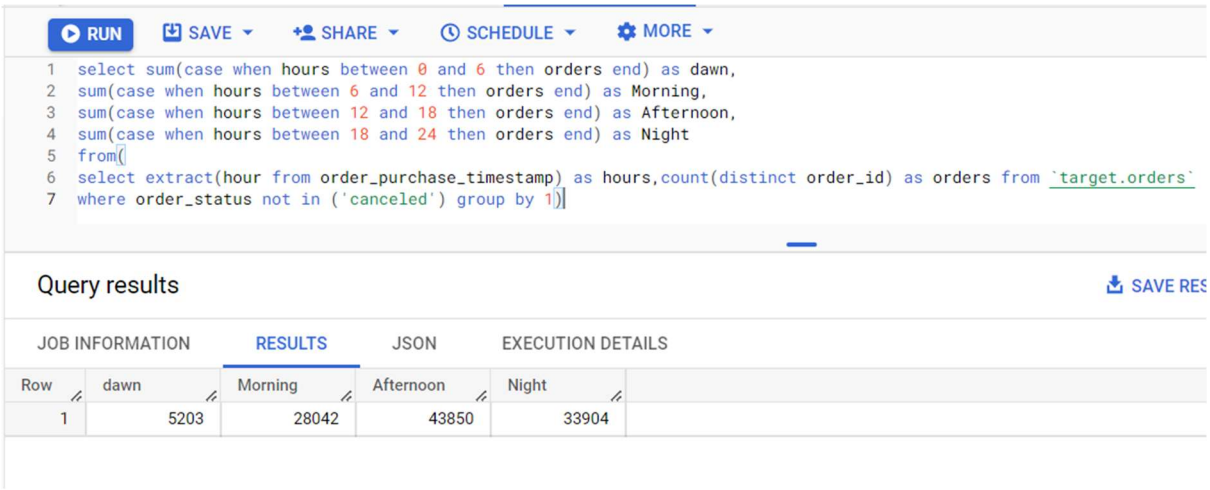
1. i) Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

```
select extract(year from order_purchase_timestamp) as year,extract(month from order_purchas
e_timestamp) as month,count(distinct order_id) as count  from `target.orders` where order_s
tatus not in ('cancelled') group by 1,2 order by 1,2
```



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

```
select sum(case when hours between 0 and 6 then orders end) as dawn,
sum(case when hours between 6 and 12 then orders end) as Morning,
sum(case when hours between 12 and 18 then orders end) as Afternoon,
sum(case when hours between 18 and 24 then orders end) as Night
from(
select extract(hour from order_purchase_timestamp) as hours,count(distinct order_id) as ord
ers from `target.orders`
where order_status not in ('canceled') group by 1)
```

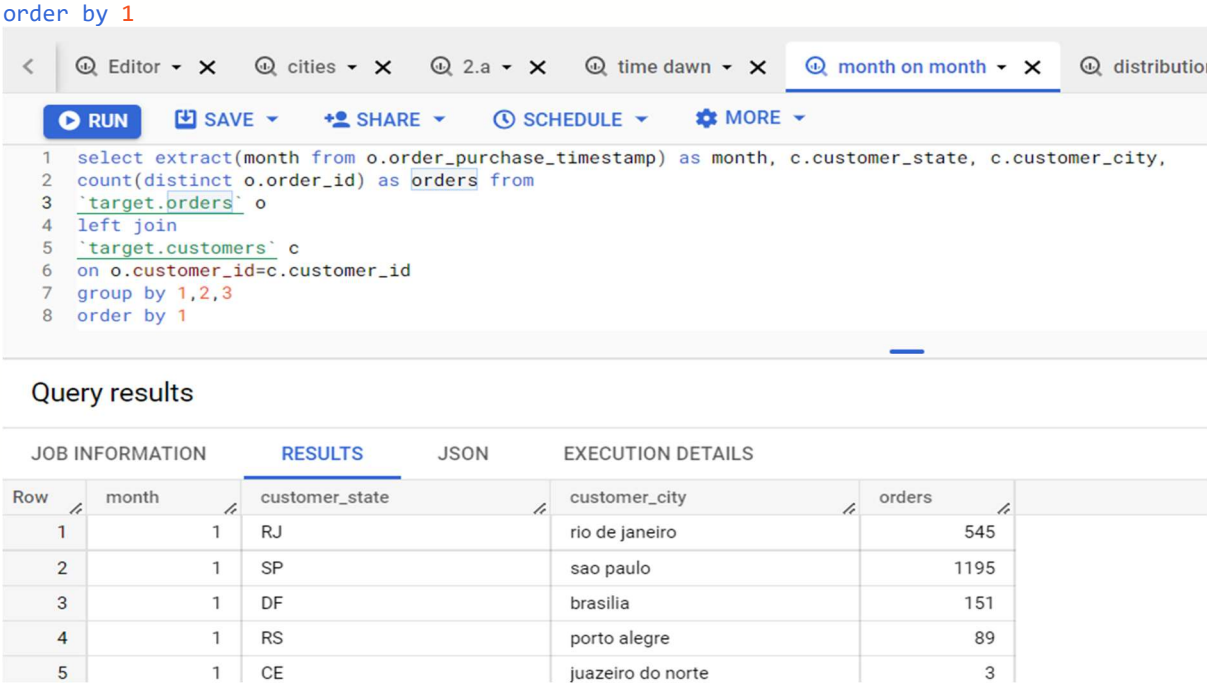


Row	dawn	Morning	Afternoon	Night
1	5203	28042	43850	33904

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

i) Get month on month orders by region, states

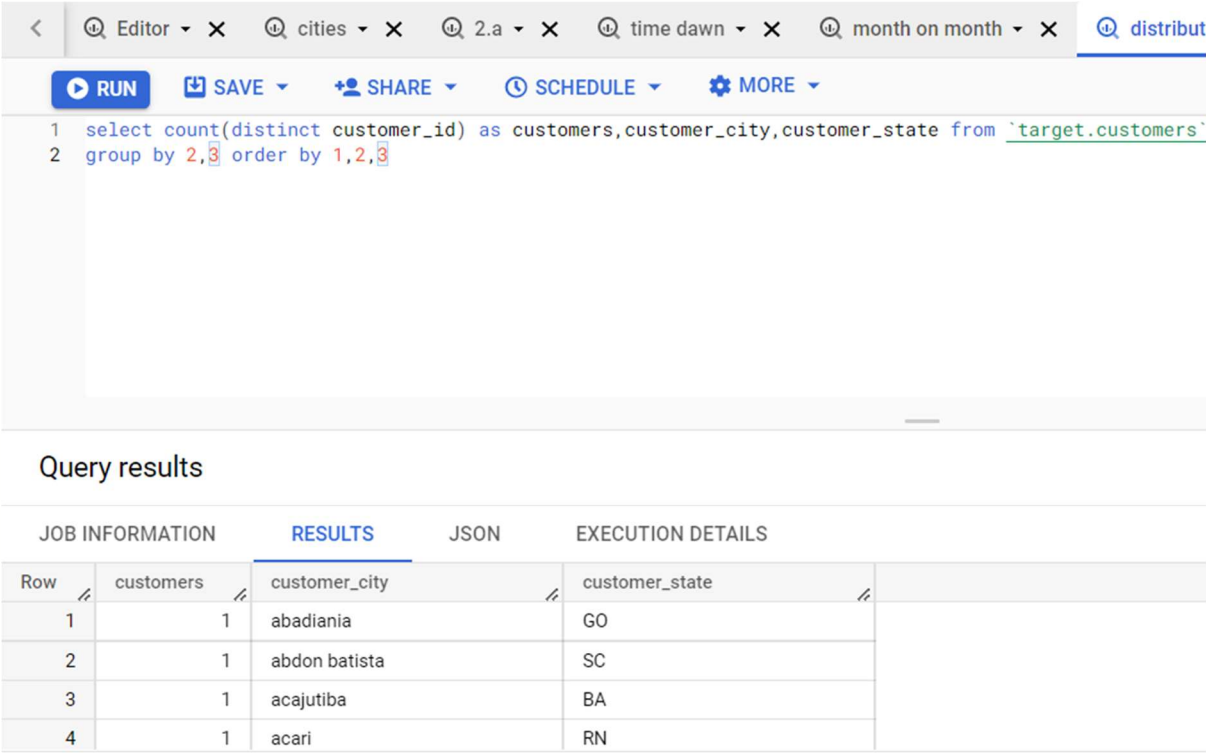
```
select extract(month from o.order_purchase_timestamp) as month, c.customer_s
tate, c.customer_city,count(distinct o.order_id) as orders from
`target.orders` o
left join
`target.customers` c on o.customer_id=c.customer_id group by 1,2,3
order by 1
```



Row	month	customer_state	customer_city	orders
1	1	RJ	rio de janeiro	545
2	1	SP	sao paulo	1195
3	1	DF	brasilia	151
4	1	RS	porto alegre	89
5	1	CE	juazeiro do norte	3

ii) How are customers distributed in Brazil

```
select count(distinct customer_id) as customers,customer_city,customer_state
 from `target.customers`
group by 2,3 order by 1,2,3
```



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

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

```
with base_1
as (select extract (year from o.order_purchase_timestamp) as year,
extract (month from o.order_purchase_timestamp) as month,
sum(payment_value) as payment
from `target.payments` p
left join
`target.orders` o
on p.order_id = o.order_id
where extract (month from o.order_purchase_timestamp) between 1 and 8
and extract (year from o.order_purchase_timestamp) between 2017 and 2018
group by 1,2
order by 1,2
),
base_2 as (
select year, sum(payment) as payment from base_1 group by 1 order by 1),
base_3 as (select *, lead(payment,1) over (order by year asc) as leader from
 base_2)select *, round((leader-payment)/payment *100,2) from base_3
```

```
with base_1
as (select extract (year from o.order_purchase_timestamp) as year,
extract (month from o.order_purchase_timestamp) as month,
sum(payment_value) as payment
from `target.payments` p
left join
`target.orders` o
on p.order_id = o.order_id
where extract (month from o.order_purchase_timestamp) between 1 and 8
and extract (year from o.order_purchase_timestamp) between 2017 and 2018
group by 1,2
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	year	payment	leader	f0_
1	2017	3669022.12...	8694733.83...	136.98
2	2018	8694733.83...	null	null

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

```
select c.customer_state,avg(oi.price),sum(oi.freight_value) from `target.customers` c left
join `target.orders` o
on c.customer_id=o.customer_id left join `target.order_items` oi on o.order_id=oi.order_id
group by c.customer_state
```

```
select c.customer_state,avg(oi.price),sum(oi.freight_value) from `target.customers` c left join `target.orders` o
on c.customer_id=o.customer_id left join `target.order_items` oi on o.order_id=oi.order_id
group by c.customer_state
```

Query results

SAVE RES

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	f0_	f1_
1	RN	156.965935...	18860.0999...
2	CE	153.758261...	48351.5899...
3	RS	120.337453...	135522.740...
4	SC	124.653577...	89660.2600...

5)Analysis on sales, freight and delivery time

i) Calculate days between purchasing, delivering and estimated delivery

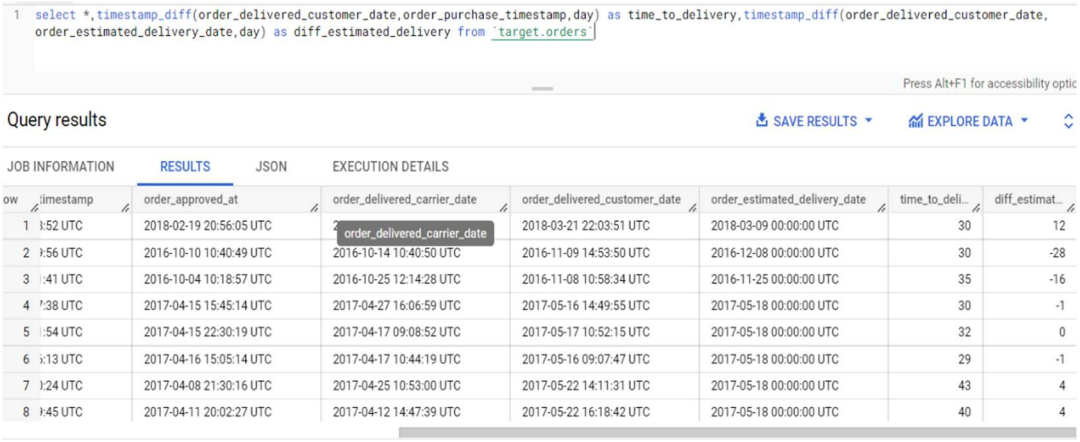
```
select timestamp_diff(order_delivered_customer_date,order_purchase_timestamp
,day) as diff_purchase_delivered,timestamp_diff(order_estimated_delivery_dat
e,order_delivered_customer_date,day) as diff_delivered_est_delivered
from `target.orders`
```

Row	diff_purchas...	diff_delivere...
1	30	-12
2	30	28
3	35	16
4	30	1

ii) Create columns:

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

```
select *,timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day)
 as time_to_delivery,timestamp_diff(order_delivered_customer_date,order_estimated_d
elivery_date,day) as diff_estimated_delivery from `target.orders`
```



ow	timestamp	order_approved_at	order_delivered_carrier_date	order_delivered_customer_date	order_estimated_delivery_date	time_to_deli...	diff_estimat...
1	52 UTC	2018-02-19 20:56:05 UTC		2018-03-21 22:03:51 UTC	2018-03-09 00:00:00 UTC	30	12
2	56 UTC	2016-10-10 10:40:49 UTC	2016-10-14 10:40:50 UTC	2016-11-09 14:53:50 UTC	2016-12-08 00:00:00 UTC	30	-28
3	41 UTC	2016-10-04 10:18:57 UTC	2016-10-25 12:14:28 UTC	2016-11-08 10:58:34 UTC	2016-11-25 00:00:00 UTC	35	-16
4	38 UTC	2017-04-15 15:45:14 UTC	2017-04-27 16:06:59 UTC	2017-05-16 14:49:55 UTC	2017-05-18 00:00:00 UTC	30	-1
5	54 UTC	2017-04-15 22:30:19 UTC	2017-04-17 09:08:52 UTC	2017-05-17 10:52:15 UTC	2017-05-18 00:00:00 UTC	32	0
6	13 UTC	2017-04-16 15:05:14 UTC	2017-04-17 10:44:19 UTC	2017-05-16 09:07:47 UTC	2017-05-18 00:00:00 UTC	29	-1
7	24 UTC	2017-04-08 21:30:16 UTC	2017-04-25 10:53:00 UTC	2017-05-22 14:11:31 UTC	2017-05-18 00:00:00 UTC	43	4
8	45 UTC	2017-04-11 20:02:27 UTC	2017-04-12 14:47:39 UTC	2017-05-22 16:18:42 UTC	2017-05-18 00:00:00 UTC	40	4

Sort the data to get the following:

1. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
2. Top 5 states with highest/lowest average time to delivery
3. Top 5 states where delivery is really fast/ not so fast compared to estimated date

1)

```
select c.customer_state,avg(oi.freight_value) from `target.customers` c left join `target.o
rders` o on c.customer_id=o.customer_id
left join `target.order_items` oi on o.order_id=oi.order_id
group by c.customer_state
order by 2 desc limit 5
```

```
1 select c.customer_state,avg(oi.freight_value) from `target.customers` c left join `target.orders` o on c.customer_id=o.customer_id
2 left join `target.order_items` oi on o.order_id=oi.order_id
3 group by c.customer_state
4 order by 2 desc limit 5
```

Press

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	f0_
1	RR	42.9844230...
2	PB	42.7238039...
3	RO	41.0697122...
4	AC	40.0733695...
5	PI	39.1479704...

2)

```
select c.customer_state,avg(timestamp_diff(o.order_delivered_customer_date,o.order_purchase
_timestamp,day)) as avg_time_to_delivery from `target.customers` c
left join `target.orders` o on c.customer_id=o.customer_id
group by c.customer_state
order by avg(timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)
) desc limit 5
```

```
1 select c.customer_state,avg(timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)) as avg_time_to_delivery from `target.customers` c
2 left join `target.orders` o on c.customer_id=o.customer_id
3 group by c.customer_state
4 order by avg(timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)) desc limit 5
```

Press Alt+F1 for accessib

Query results

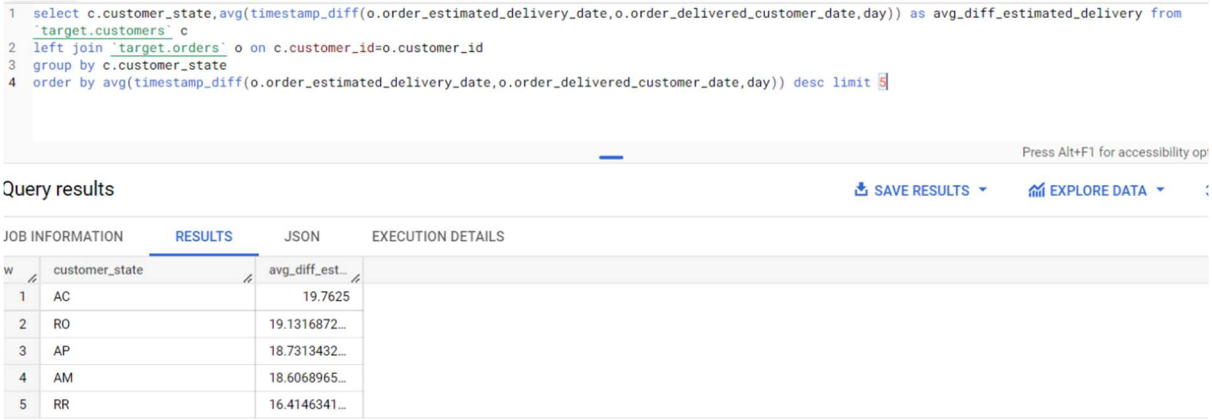
SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	avg_time_to...
1	RR	28.9756097...
2	AP	26.7313432...
3	AM	25.9862068...
4	AL	24.0403022...
5	PA	23.3160676...

3)

```
select c.customer_state,avg(timestamp_diff(o.order_estimated_delivery_date,o.order_delivere
d_customer_date,day)) as avg_diff_estimated_delivery from `target.customers` c
left join `target.orders` o on c.customer_id=o.customer_id
group by c.customer_state
order by avg(timestamp_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date
,day)) desc limit 5
```



w	customer_state	avg_diff_est...
1	AC	19.7625
2	RO	19.1316872...
3	AP	18.7313432...
4	AM	18.6068965...
5	RR	16.4146341...

6) Payment type analysis:

1. Month over Month count of orders for different payment types
2. Distribution of payment installments and count of orders

1)

```
with x as (select o.order_id,extract (month from o.order_purchase_timestamp) as month,
from `target.payments` p
left join
`target.orders` o
on p.order_id = o.order_id)
select month,count(distinct order_id) from x group by x.month order by 1
```

```
with x as (select o.order_id,extract (month from o.order_purchase_timestamp) as month,
from `target.payments` p
left join
`target.orders` o
on p.order_id = o.order_id)
select month,count(distinct order_id) from x group by x.month order by 1
```

uery results

OB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

	month	f0_
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318

2)

```
select p.payment_installments,count(distinct p.order_id) as count,extract(year from o.order
_purchase_timestamp) as year,extract(month from o.order_purchase_timestamp) as month,
count(distinct p.order_id) as count  from `target.orders` o right join `target.payments` p
on o.order_id=p.order_id where o.order_status not in ('cancelled') group by 1,3,4 order by
1,2,3,4
```

```
select p.payment_installments,count(distinct p.order_id) as count,extract(year from o.order_purchase_timestamp) as year,extract(month from o.
order_purchase_timestamp) as month,
count(distinct p.order_id) as count  from `target.orders` o right join `target.payments` p on o.order_id=p.order_id where o.order_status not in
('cancelled') group by 1,3,4 order by 1,2,3,4
```

Press Alt+F1 for acces

uery results SAVE RESULTS EXPLORE DATA

OB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

	payment_in...	count	year	month	count_1
1	0	1	2018	4	1
2	0	1	2018	5	1
3	1	1	2016	9	1
4	1	1	2016	12	1
5	1	4	2018	10	4
6	1	16	2018	9	16
7	1	127	2016	10	127