Business Case: Target SQL

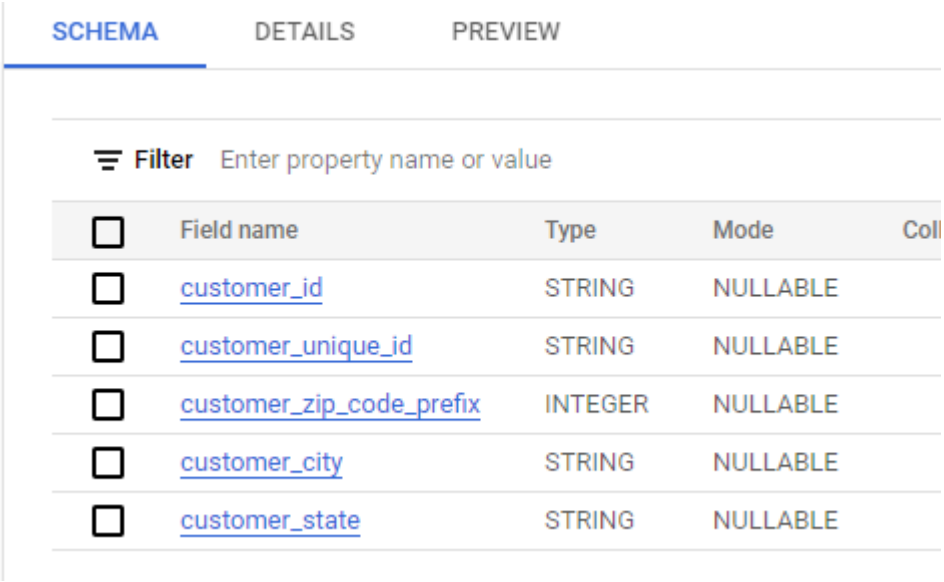
Problem statement: Target is one of the world's most recognized brands and one of America's leading retailers. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

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.

Exploration of dataset:

1. Data type of columns in a table

- **Customer Dataset:**
  These are the columns present with the datatypes.



No. of rows present in this data

Number of rows	99,441

- **Geoloccation Dataset**

SCHEMA | DETAILS | PREVIEW

Filter Enter property name or value

Field name	Type	Mode
geolocation_zip_code_prefix	INTEGER	NULLABLE
geolocation_lat	FLOAT	NULLABLE
geolocation_lng	FLOAT	NULLABLE
geolocation_city	STRING	NULLABLE
geolocation_state	STRING	NULLABLE

Number of rows 1,000,163

- **Order_items table**

SCHEMA | DETAILS | PREVIEW

Filter Enter property name or value

Field name	Type	Mode
order_id	STRING	NULLABLE
order_item_id	INTEGER	NULLABLE
product_id	STRING	NULLABLE
seller_id	STRING	NULLABLE
shipping_limit_date	TIMESTAMP	NULLABLE
price	FLOAT	NULLABLE
freight_value	FLOAT	NULLABLE

Number of rows 112,650

- **Order_reviews**

SCHEMA DETAILS PREVIEW

Filter Enter property name or value

Field name	Type	Mode
review_id	STRING	NULLABLE
order_id	STRING	NULLABLE
review_score	INTEGER	NULLABLE
review_comment_title	STRING	NULLABLE
review_creation_date	TIMESTAMP	NULLABLE
review_answer_timestamp	TIMESTAMP	NULLABLE

Number of rows 99,224

- **Orders table**

SCHEMA DETAILS PREVIEW

Filter Enter property name or value

Field name	Type	Mode
order_id	STRING	NULLABLE
customer_id	STRING	NULLABLE
order_status	STRING	NULLABLE
order_purchase_timestamp	TIMESTAMP	NULLABLE
order_approved_at	TIMESTAMP	NULLABLE
order_delivered_carrier_date	TIMESTAMP	NULLABLE
order_delivered_customer_date	TIMESTAMP	NULLABLE
order_estimated_delivery_date	TIMESTAMP	NULLABLE

Number of rows 99,441

- **Payments table**

SCHEMA DETAILS PREVIEW

Filter Enter property name or value

Field name	Type	Mode
order_id	STRING	NULLABLE
payment_sequential	INTEGER	NULLABLE
payment_type	STRING	NULLABLE
payment_installments	INTEGER	NULLABLE
payment_value	FLOAT	NULLABLE

Number of rows 103,886

- **Products table**

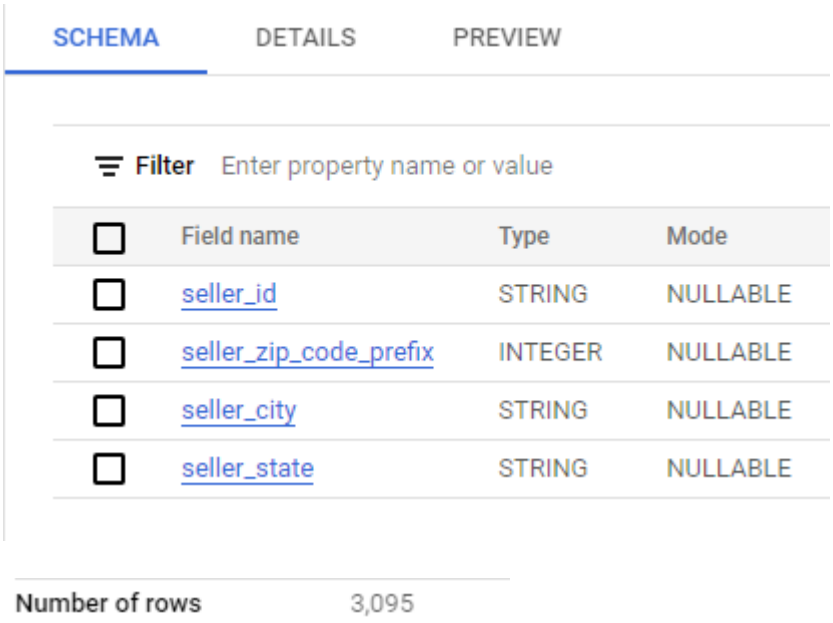
SCHEMA DETAILS PREVIEW

Filter Enter property name or value

Field name	Type	Mode
product_id	STRING	NULLABLE
product_category	STRING	NULLABLE
product_name_length	INTEGER	NULLABLE
product_description_length	INTEGER	NULLABLE
product_photos_qty	INTEGER	NULLABLE
product_weight_g	INTEGER	NULLABLE
product_length_cm	INTEGER	NULLABLE
product_height_cm	INTEGER	NULLABLE
product_width_cm	INTEGER	NULLABLE

Number of rows 32,951

- **Sellers table**

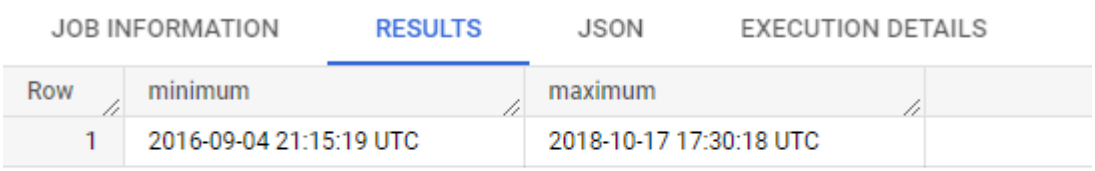


2. Time period for which the data is given.

   Query:

```
-- Time period for which the data is given
-
- select min(order_purchase_timestamp) as minimum,max(order_purchase_timestamp) as maximum
from `Target.orders`;
```

   o/p



3. Cities and States covered in the dataset.

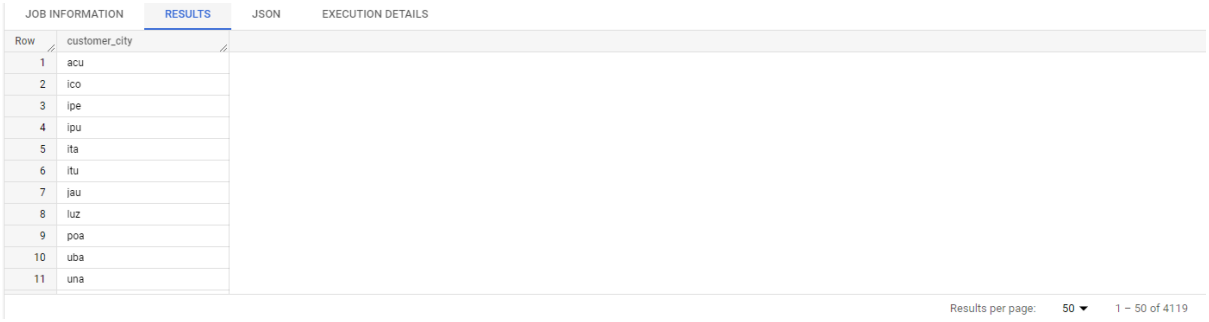
Query:

Cities

```
-- select distinct customer_city
```

```
-- from `Target.customers`;
```

o/p:



JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

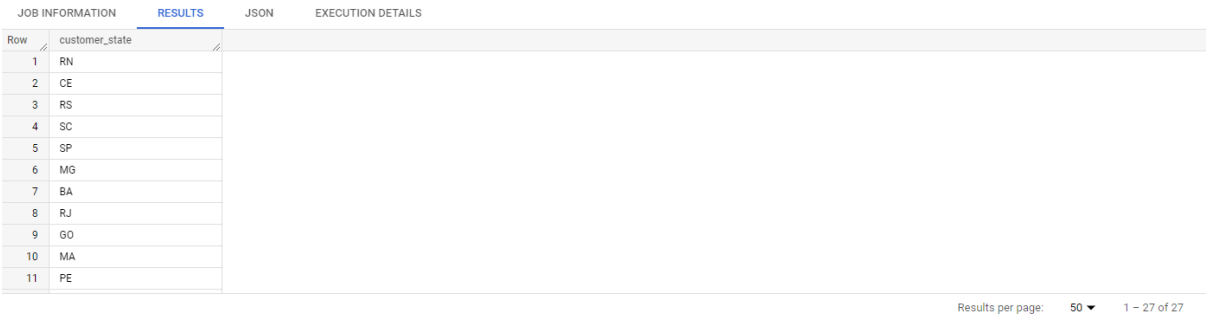
Row	customer_city
1	acu
2	ico
3	ipe
4	ipu
5	ita
6	itu
7	jau
8	luz
9	poa
10	uba
11	una

Results per page: 50 1 – 50 of 4119

States:

```
-- select distinct customer_state
-- from `Target.customers`;
```

o/p:



JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state
1	RN
2	CE
3	RS
4	SC
5	SP
6	MG
7	BA
8	RJ
9	GO
10	MA
11	PE

Results per page: 50 1 – 27 of 27

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:

```
-- select distinct month, count(order_purchase_timestamp) as count,year from(
-
-   select order_purchase_timestamp,extract(month FROM order_purchase_timestamp) as month,e
xtract(year from order_purchase_timestamp)as year
--   from `Target.orders`)
-- group by month,year
-- order by month;
```

o/p:

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	month	count	year
1	1	7269	2018
2	1	800	2017
3	2	6728	2018
4	2	1780	2017
5	3	7211	2018
6	3	2682	2017
7	4	2404	2017
8	4	6939	2018
9	5	6873	2018
10	5	3700	2017
11	6	3245	2017

Results per page: 50 1 – 25 of 25

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

Query:

```
-- with first as (
-- select case when hours in(0,1,2,3,4) then 'Midnight'
-- when hours in(5,6) then 'Dawn'
-- when hours in (7,8,9,10,11) then 'Morning'
-- when hours in (12,13,14,15) then 'Afternoon'
-- when hours in (16,17,18) then 'Evening'
-- else 'Night'
-- end as timings
-- ,hours,count(hours) as count from(
-
- select order_purchase_timestamp,extract(hour FROM order_purchase_timestamp)as hours from
`Target.orders`)
-- group by hours
-- order by hours),

-- second as(
-- select timings,sum(count) as no_of_orders
-- from first
-- group by timings)

-- select * from second
-- order by no_of_orders desc
```

o/p:

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	timings	no_of_orders
1	Night	28331
2	Afternoon	25536
3	Morning	21738
4	Evening	18594
5	Midnight	4552
6	Dawn	690

PERSONAL HISTORY | PROJECT HISTORY

Evolution of E-commerce orders in the Brazil region:

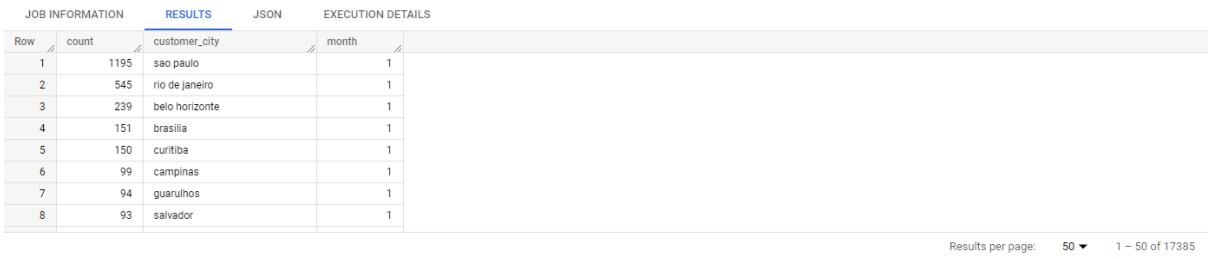
1. Get month on month orders by region, states

   Cities

   Query:

```
-- select count(order_id) as count,customer_city,month from(
-
- select distinct o.order_id,o.customer_id,c.customer_zip_code_prefix,c.customer_city,extra
ct(month from order_purchase_timestamp)as month
-- from `Target.orders` o
-- join `Target.customers` c on o.customer_id = c.customer_id)
-- group by customer_city,month
-- order by month,count(order_id) desc;
```

   o/p:



JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	count	customer_city	month
1	1195	sao paulo	1
2	545	rio de janeiro	1
3	239	belo horizonte	1
4	151	brasilia	1
5	150	curitiba	1
6	99	campinas	1
7	94	guarulhos	1
8	93	salvador	1

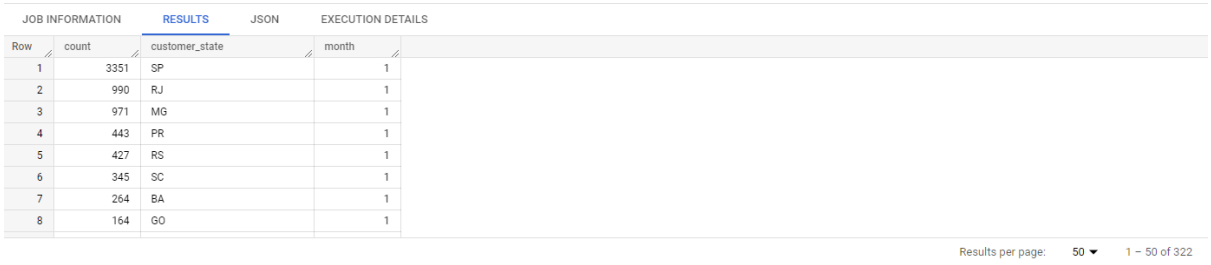
Results per page: 50 1 – 50 of 17385

   States

   Query:

```
-- select count(order_id) as count,customer_state,month from(
-
- select distinct o.order_id,o.customer_id,c.customer_zip_code_prefix,c.customer_state,extr
act(month from order_purchase_timestamp)as month
-- from `Target.orders` o
-- join `Target.customers` c on o.customer_id = c.customer_id)
-- group by customer_state,month
-- order by month,count(order_id) desc;
```

   o/p:



JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	count	customer_state	month
1	3351	SP	1
2	990	RJ	1
3	971	MG	1
4	443	PR	1
5	427	RS	1
6	345	SC	1
7	264	BA	1
8	164	GO	1

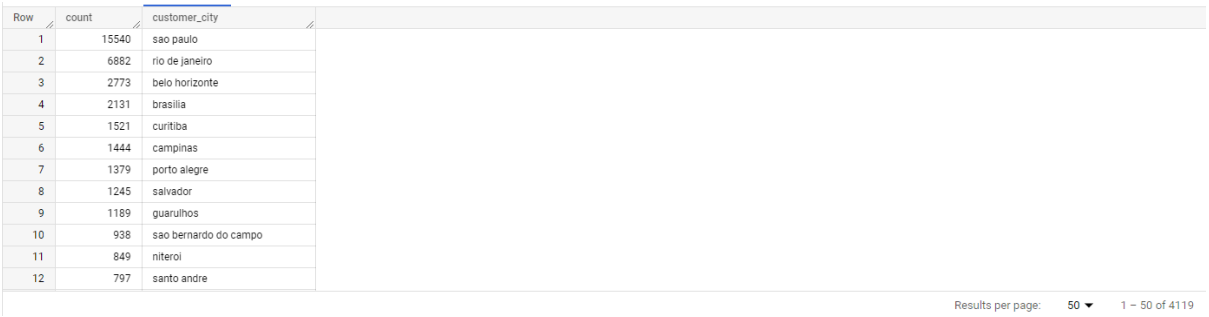
Results per page: 50 1 – 50 of 322

2. How are customers distributed in Brazil

Query:

```
-- with customer_city_table as (
-- select count(customer_id) count,customer_city
-- from `Target.customers`
-- group by customer_city
-- order by count(customer_id) desc)

-- select * from customer_city_table
```

o/p:



Row	count	customer_city
1	15540	sao paulo
2	6882	rio de janeiro
3	2773	belo horizonte
4	2131	brasilia
5	1521	curitiba
6	1444	campinas
7	1379	porto alegre
8	1245	salvador
9	1189	guarulhos
10	938	sao bernardo do campo
11	849	niteroi
12	797	santo andre

Results per page: 50 1 – 50 of 4119

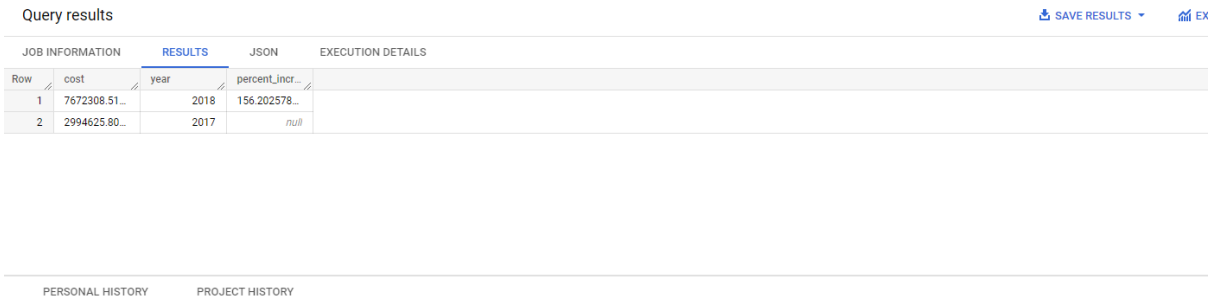
Impact on Economy: Analyze the money movements by e-commerce by looking at order prices, freight and others.

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

Query:

```
-- with percent as(
-- (select sum(p.payment_value) as cost,2017 as year
-- from `Target.orders` o
-- join `Target.payments` p on o.order_id = p.order_id
-- where extract(year from o.order_purchase_timestamp) = 2017
-- and extract(month from o.order_purchase_timestamp) between 1 and 7)
-- union all
-- (select sum(p.payment_value) as cost,2018 as year
-- from `Target.orders` o
-- join `Target.payments` p on o.order_id = p.order_id
-- where extract(year from o.order_purchase_timestamp) = 2018
-- and extract(month from o.order_purchase_timestamp) between 1 and 7)),

-- previous_year as (
-- select cost from percent where year = 2017)

-- select *,
-- case when year = 2018 then round(cost-
(select cost from previous_year),2)/(select cost from previous_year)*100
--   else NULL
-- end as percent_increase_sales
--  from percent;
```

o/p:

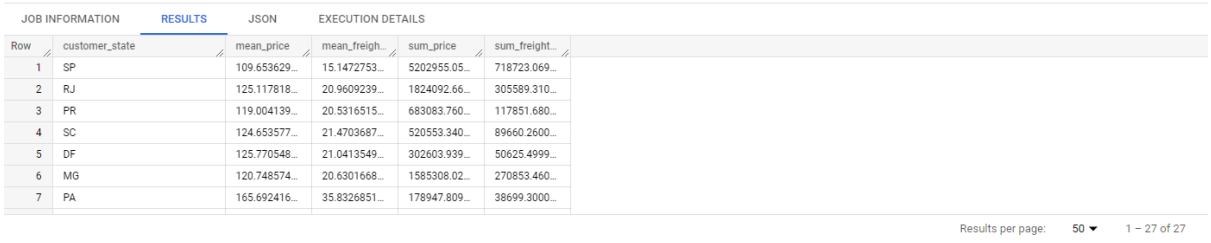


2. Mean & Sum of price and freight value by customer state

Query:

```
-- with fist as(
-- select o.order_id,o.customer_id,oi.price,oi.freight_value
-- from `Target.orders` o
-- join `Target.order_items` oi on o.order_id = oi.order_id
-- ),

-- second as(
-- select c.customer_id,c.customer_state,f.price,f.freight_value
-- from fist f
-- join `Target.customers` c on f.customer_id = c.customer_id
-- ),

-- third as(
-
- select customer_state,avg(price)as mean_price,avg(freight_value)as mean_freight_value,sum
(price)as sum_price,sum(freight_value)as sum_freight_value
-- from second
-- group by customer_state
-- )

-- select * from third;
```

o/p:



Analysis on sales, freight and delivery time.

1. Calculate days between purchasing, delivering and estimated delivery

Query:

```
-- with first as(
-
- select order_purchase_timestamp,order_delivered_customer_date,order_estimated_delivery_da
te
-- from `Target.orders`
-- where order_delivered_customer_date is not null
-- ),

-- second as(
-
- select order_purchase_timestamp,order_delivered_customer_date,order_estimated_delivery_da
te,
-- extract(date from order_purchase_timestamp)as purchase,
-- extract(date from order_delivered_customer_date)as delivered,
-- extract(date from order_estimated_delivery_date)as est_delivery
-- from first
-- ),

-- third as(
-- select purchase,delivered,est_delivery,
-- DATE_DIFF(delivered,purchase,DAY)as diff_pur_del,
-- DATE_DIFF(est_delivery,purchase,DAY)as diff_pur_est,
-- DATE_DIFF(est_delivery,delivered,DAY)as diff_del_est
-- from second
-- )
-- select * from third;
```

o/p:

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	purchase	delivered	est_delivery	diff_pur_del	diff_pur_est	diff_del_est
1	2016-10-07	2016-10-14	2016-11-29	7	53	46
2	2016-10-09	2016-11-09	2016-12-08	31	60	29
3	2016-10-09	2016-10-16	2016-11-30	7	52	45
4	2016-10-08	2016-10-19	2016-11-30	11	53	42
5	2016-10-03	2016-11-08	2016-11-25	36	53	17

Results per page: 50 1 – 50 of 96476

2. Create columns:

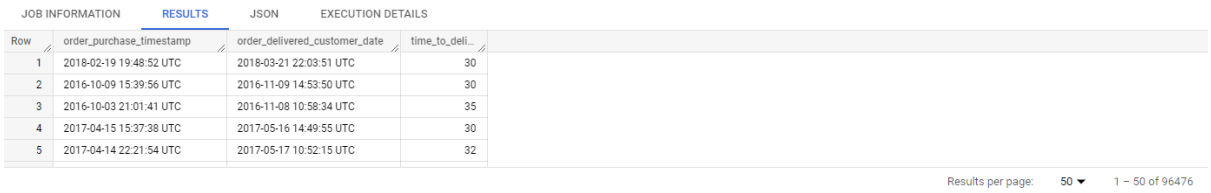
- time_to_delivery = order_purchase_timestamp-order_delivered_customer_date

Query:

```
-- with first as(
-- select order_purchase_timestamp,order_delivered_customer_date,
-- DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)as time_to_delivery
-- from `Target.orders`
-- where order_delivered_customer_date is not null
-- )
```

```
-- select * from first;
```

o/p:



JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	order_purchase_timestamp	order_delivered_customer_date	time_to_deli...
1	2018-02-19 19:48:52 UTC	2018-03-21 22:03:51 UTC	30
2	2016-10-09 15:39:56 UTC	2016-11-09 14:53:50 UTC	30
3	2016-10-03 21:01:41 UTC	2016-11-08 10:58:34 UTC	35
4	2017-04-15 15:37:38 UTC	2017-05-16 14:49:55 UTC	30
5	2017-04-14 22:21:54 UTC	2017-05-17 10:52:15 UTC	32

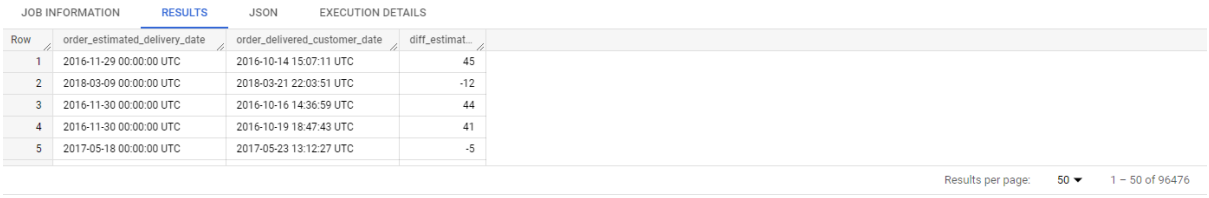
Results per page: 50 1 – 50 of 96476

- diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

  Query:

```
-- select order_estimated_delivery_date,order_delivered_customer_date,
-
- DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY)as diff_estimat
ed_delivery
-- from `Target.orders`
-- where order_delivered_customer_date is not null;
```

  o/p:



JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	order_estimated_delivery_date	order_delivered_customer_date	diff_estimat...
1	2016-11-29 00:00:00 UTC	2016-10-14 15:07:11 UTC	45
2	2018-03-09 00:00:00 UTC	2018-03-21 22:03:51 UTC	-12
3	2016-11-30 00:00:00 UTC	2016-10-16 14:36:59 UTC	44
4	2016-11-30 00:00:00 UTC	2016-10-19 18:47:43 UTC	41
5	2017-05-18 00:00:00 UTC	2017-05-23 13:12:27 UTC	-5

Results per page: 50 1 – 50 of 96476

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

   Query:

```
-- with first as(
-- select  customer_state,
-- avg(freight_value) over(partition by customer_state) as mean_freight_value,
-
- avg(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)) over(partition
 by customer_state) as mean_time_to_delivery,
-
- avg(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY)) over(part
ition by customer_state) as mean_diff_estimated_delivery
-- from `Target.customers` c
-- join `Target.orders` o on c.customer_id = o.customer_id
-- join `Target.order_items` oi on o.order_id = oi.order_id
-- where order_delivered_customer_date is not null),

-- second as(
```

```
-
- select customer_state,mean_freight_value,mean_time_to_delivery,mean_diff_estimated_delive
ry,
-- row_number() over(partition by customer_state) as row
-- from first
-- ),

-- third as(
-- select * from second
-- where row = 1)

-
- select customer_state,mean_freight_value,mean_time_to_delivery,mean_diff_estimated_delive
ry from third;
```

o/p:

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	mean_freigh...	mean_time_...	mean_diff_e...
1	ES	22.0289797...	15.1928089...	9.76853932...
2	GO	22.5628678...	14.9481774...	11.3728590...
3	PB	43.0916894...	20.1194539...	12.1501706...
4	PE	32.6933333...	17.7920962...	12.5521191...
5	RS	21.6142703...	14.7082993...	13.2030001...

Results per page: 50 1 – 27 of 27

4. Sort the data to get the following:

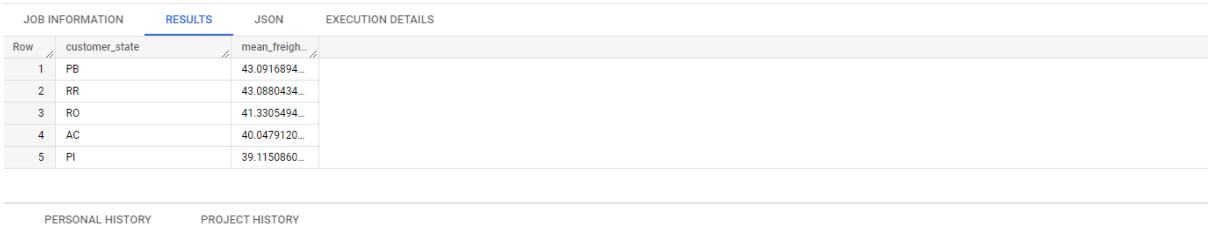
- Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

Query:

```
-- with first as(
-- select  customer_state,
-- avg(freight_value) over(partition by customer_state) as mean_freight_value,
-
- avg(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)) over(partition
 by customer_state) as mean_time_to_delivery,
-
- avg(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY)) over(part
ition by customer_state) as mean_diff_estimated_delivery
-- from `Target.customers` c
-- join `Target.orders` o on c.customer_id = o.customer_id
-- join `Target.order_items` oi on o.order_id = oi.order_id
-- where order_delivered_customer_date is not null),

-- second as(
-
- select customer_state,mean_freight_value,mean_time_to_delivery,mean_diff_estimated_delive
ry,
-- row_number() over(partition by customer_state) as row
-- from first
```

```
-- ),

-- third as(
-- select * from second
-- where row = 1)

-- select customer_state,mean_freight_value
-- from third
-- order by mean_freight_value desc limit 5;
```

o./p:



JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	mean_freigh...
1	PB	43.0916894...
2	RR	43.0880434...
3	RO	41.3305494...
4	AC	40.0479120...
5	PI	39.1150860...

PERSONAL HISTORY | PROJECT HISTORY

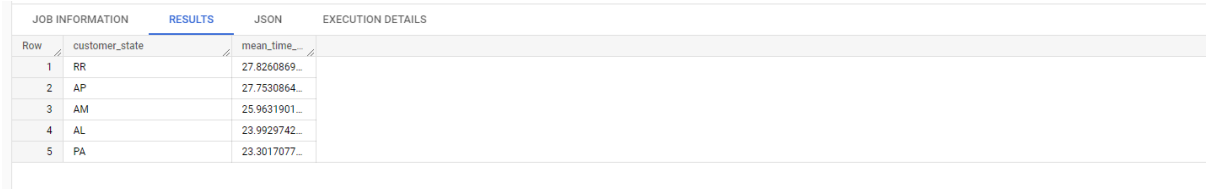
- Top 5 states with highest/lowest average time to delivery

  Query:

  Really Fast:

```
-- with first as(
-- select  customer_state,
-- avg(freight_value) over(partition by customer_state) as mean_freight_value,
-
- avg(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)) over(partition
 by customer_state) as mean_time_to_delivery,
-
- avg(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY)) over(part
ition by customer_state) as mean_diff_estimated_delivery
-- from `Target.customers` c
-- join `Target.orders` o on c.customer_id = o.customer_id
-- join `Target.order_items` oi on o.order_id = oi.order_id
-- where order_delivered_customer_date is not null),

-- second as(
-
- select customer_state,mean_freight_value,mean_time_to_delivery,mean_diff_estimated_delive
ry,
-- row_number() over(partition by customer_state) as row
-- from first
-- ),

-- third as(
-- select * from second
-- where row = 1)

-- select customer_state,mean_time_to_delivery
-- from third
-- order by mean_time_to_delivery desc limit 5;
```

o/p:



Row	customer_state	mean_time_...
1	RR	27.8260869...
2	AP	27.7530864...
3	AM	25.9631901...
4	AL	23.9929742...
5	PA	23.3017077...

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

  Query:

```
-- with first as(
-
- select order_purchase_timestamp,order_delivered_customer_date,order_estimated_delivery_da
te,customer_state
-- from `Target.orders` o
-- join `Target.customers` c on o.customer_id = c.customer_id
-- where order_delivered_customer_date is not null
-- ),

-- second as(
-
- select customer_state,order_purchase_timestamp,order_delivered_customer_date,order_estima
ted_delivery_date,
-- extract(date from order_purchase_timestamp)as purchase,
-- extract(date from order_delivered_customer_date)as delivered,
-- extract(date from order_estimated_delivery_date)as est_delivery
-- from first
-- ),

-- third as(
-- select customer_state,purchase,delivered,est_delivery,
-- DATE_DIFF(delivered,purchase,DAY)as diff_pur_del,
-- DATE_DIFF(est_delivery,purchase,DAY)as diff_pur_est
-- from second
-- ),
-- fourth as(
-- select customer_state,count(customer_state) as count from third
-- where diff_pur_del < 0.25* diff_pur_est
-- group by customer_state
-- order by count(customer_state) desc
-- limit 5
-- )
-- select * from fourth;
```

o/p:

Row	customer_state	count
1	SP	8386
2	RJ	1795
3	MG	1406
4	PR	588
5	RS	463

Really slow:

Query:

```
-- with first as(
-
- select order_purchase_timestamp,order_delivered_customer_date,order_estimated_delivery_da
te,customer_state
-- from `Target.orders` o
-- join `Target.customers` c on o.customer_id = c.customer_id
-- where order_delivered_customer_date is not null
-- ),

-- second as(
-
- select customer_state,order_purchase_timestamp,order_delivered_customer_date,order_estima
ted_delivery_date,
-- extract(date from order_purchase_timestamp)as purchase,
-- extract(date from order_delivered_customer_date)as delivered,
-- extract(date from order_estimated_delivery_date)as est_delivery
-- from first
-- ),

-- third as(
-- select customer_state,purchase,delivered,est_delivery,
-- DATE_DIFF(delivered,purchase,DAY)as diff_pur_del,
-- DATE_DIFF(est_delivery,purchase,DAY)as diff_pur_est
-- from second
-- ),
-- fourth as(
-- select customer_state,count(customer_state) as count from third
-- where diff_pur_del > diff_pur_est
-- group by customer_state
-- order by count(customer_state) desc
-- limit 5
-- )
-- select * from fourth;
```

o/p:

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	count
1	SP	1820
2	RJ	1495
3	MG	520
4	BA	396
5	RS	325

Payment type analysis:

- Month over Month count of orders for different payment types

  Query:

```
-- select * from(
-
- select p.payment_type,extract(month from o.order_purchase_timestamp) as month,extract(yea
r from o.order_purchase_timestamp) as year,count(distinct p.order_id) as count
-- from `Target.payments` p
-- join `Target.orders` o on p.order_id = o.order_id
-
- group by p.payment_type,extract(month from o.order_purchase_timestamp),extract(year from
order_purchase_timestamp))
-- order by year,month
```

o/p:

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	payment_type	month	year	count
1	credit_card	9	2016	3
2	credit_card	10	2016	253
3	voucher	10	2016	11
4	debit_card	10	2016	2
5	UPI	10	2016	63
6	credit_card	12	2016	1
7	voucher	1	2017	33
8	UPI	1	2017	197

Results per page: 50 1 – 50 of 90

- Distribution of payment installments and count of orders

  Query:

```
-- select p.payment_installments,count(o.order_id) as count
-- from `Target.payments` p
-- join `Target.orders` o on p.order_id = o.order_id
-- group by payment_installments
-- order by count(o.order_id) desc;
```

o/p:

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	payment_in...	count
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920

Results per page: 50 1 – 24 of 24

Insights:

1. The data is from September 2016 to October 2018.
2. There's clearly an increase in sales of e-commerce from 2017 to 2018.
3. Can't see seasonality, if we consider from 2017 Jan-Oct and 2018 Jan-Oct we can see the peak is at 2018 Jan but in 2017 Jan there are only 800 orders and in Feb 2017 we get 1780 orders. But in 2017 the max is in Nov. I can't see any seasonality where people tend to buy m ore in particular months.
4. People in Brazil tend to buy more in the time period of 7-11PM i.e Night, followed by Afternoon i.e. 12-3PM.
5. Top 3 categories that are bought more frequently are 'Bed Table Bath', 'Furniture Decoration', 'Health Beauty'.
6. Most orders are coming from 'Sao Paulo' city and the state is SP.
7. Most customers are in Sao Paulo city, followed by Rio de Janeiro and then Belo Horizonte.
8. There is 156.2% increase in the cost of sales from 2017 jan-aug and 2018 jan-aug.
9. The sum of price should be greter than the sum of freight value to generate profits for the company.
10. Here the estimated delivery is showing some abnormal numbers for the delivery while the maximum of orders are being delivered in a short span.
11. There are some orders where it is taking more than estimated delivery too, its ratio is less and might be because of some technical difficulties for these orders.
12. States with highest freight values are PB, RR, RO, AC, and PI.
13. The states with highest average time to delivery are RR, AP, AM, AL, PA.
14. The states where delivery is super-fast are SP, RJ, MG, PR, RS
15. The states where the delivery is really slow are SP, RJ, MG, BA, RS
16. People in Brazil tend to use more credit cards than debit cards for shopping.
17. People are buying more in one installment rather than to go for more installments. Should investigate more in this area.

Recommendations:

1. Need some check up on the estimated time for delivery algorithm, by analysing the data we see that there's significant gap in the actual delivery time and estimated delivery time. This might increase the online sales because consider if you see 50 days estimated delivery time instead of (10+5 buffer) 15 days, would you buy it or go to the store and bring it yourself?
2. Float more offers on the time period of 7-11PM, seems good time for increasing traffic for sales.
3. Provide more offers on are 'Bed Table Bath', 'Furniture Decoration', 'Health Beauty' and increase sales for these categories.
4. Had to go through top highest freight value states as there is no state that is in top state for orders, if we can find a reason for the higher cost of freight value we should optimise the cost (if possible).
5. Provide more offers on credit card purchase as most of the people are buying with these.
6. Provide the above mentioned offers mostly in Sao Paulo, Rio de Janeiro, Belo Horizonte cities as most buying population is in these cities.
7. Need to look further in the states SP, RJ, MG because the highest number of both fastest deliveries and the slowest deliveries occur in these states.