TARGET SQL PROJECT

(Using BigQuery)

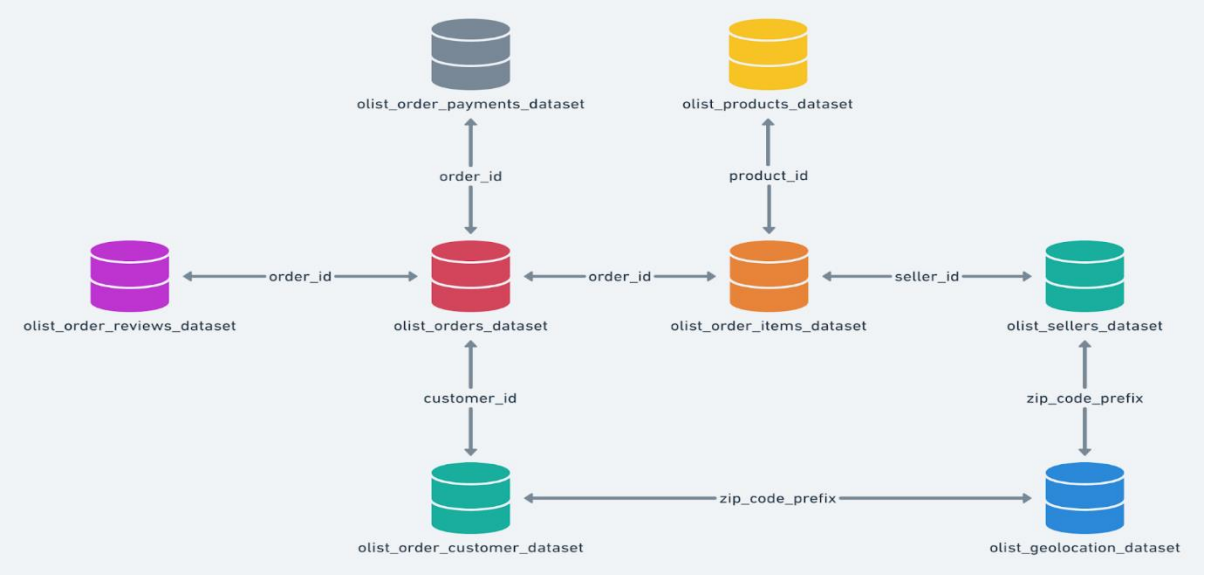
About Target:

Target is a globally renowned brand and a prominent retailer in the United States. 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 particular business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.

By analyzing this extensive dataset, it becomes possible to gain valuable insights into Target's operations in Brazil. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels.

Database Schema Diagram:



Problem Statement:

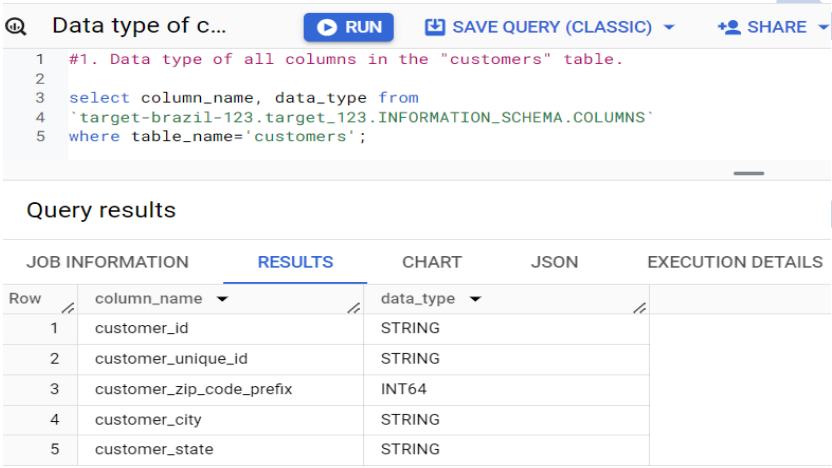
1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**

    1. Data type of all columns in the "customers" table.

    Solution:

    select column_name, data_type from
    `target-brazil-123.target_123.INFORMATION_SCHEMA.COLUMNS`
    where table_name='customers';

    screenshot:



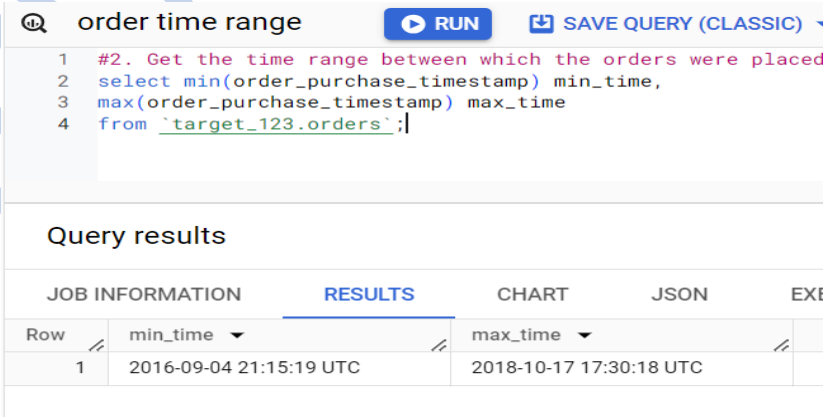
Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

    2. Get the time range between which the orders were placed.

    Solution:

    select min(order_purchase_timestamp) min_time,
    max(order_purchase_timestamp) max_time
    from `target_123.orders`;

    screenshot:



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

3. Count the Cities & States of customers who ordered during the given period.

   Solution:

   select customer_state,
   count(customer_id) customer_counts
   from `target_123.customers`
   group by customer_state
   order by customer_counts desc;

   screenshot:

customer counts across all ...zil RUN

```
# How are the customers distributed across all tl
select customer_state,
count(customer_id) customer_counts
from `target_123.customers`
group by customer_state
order by customer_counts desc;
```

Query results

JOB INFORMATION | RESULTS | CHART | JSO

Row	customer_state	customer_counts
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380

2. **In-depth Exploration:**

   1. Is there a growing trend in the no. of orders placed over the past years?

      Solution:

      select year, count(order_id) counts from
      (select order_id,
      extract(year from order_purchase_timestamp) year
      from `target_123.orders`)t1
      group by t1.year
      order by year, counts;

screenshot:

trend of order... RUN SAVE QUERY (

```
# Is there a growing trend in the no. of orders p
select year, count(order_id) counts from
(select order_id,
extract(year from order_purchase_timestamp) year
from `target_123.orders`)t1
group by t1.year
order by year, counts;
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON

ow	year	counts
1	2016	329
2	2017	45101
3	2018	54011

2. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

Solution:

```
select months, count(order_id) order_count
from
(select
  order_id,
  extract(month from order_purchase_timestamp) months
  from `target_123.orders`)t1
group by months
order by months, order_count;
```

screenshot:

monthly seasonality RUN SAVE QUERY (CLASSI

```
# Q. Can we see some kind of monthly seasonality in terms o
select months, count(order_id) order_count
from
(select
  order_id,
  extract(month from order_purchase_timestamp) months
  from `target_123.orders`)t1
group by months
order by months, order_count;
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXE

ow	months	order_count
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412

3. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

   (0-6 hrs : Dawn, 7-12 hrs : Mornings, 13-18 hrs : Afternoon,19-23 hrs : Night)

   Solution:

```
select time, count(order_id) counts from
(select case when hours>=0 and hours<=6 then '1_Dawn'
when hours>=7 and hours<=12 then '2_Mornings'
when hours>=13 and hours<=18 then '3_Afternoon'
when hours>=19 and hours<=23 then '4_Night'
end as time,
order_id
from
(select
order_id,
extract(hour from order_purchase_timestamp) hours
from `target_123.orders`)t1)t2
group by time
order by time, counts;
```

screenshot:

order timing RUN SAVE QUERY (CLASSIC)

```
/*3.  During what time of the day, do the Brazilian cu
  0-6 hrs : Dawn
  7-12 hrs : Mornings
  13-18 hrs : Afternoon
  19-23 hrs : Night
*/
select time, count(order_id) counts from
(select case when hours>=0 and hours<=6 then '1_Dawn'
when hours>=7 and hours<=12 then '2_Mornings'
when hours>=13 and hours<=18 then '3_Afternoon'
when hours>=19 and hours<=23 then '4_Night'
end as time,
order_id
from
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON

w	time	counts
1	1_Dawn	5242
2	2_Mornings	27733
3	3_Afternoon	38135
4	4_Night	28331

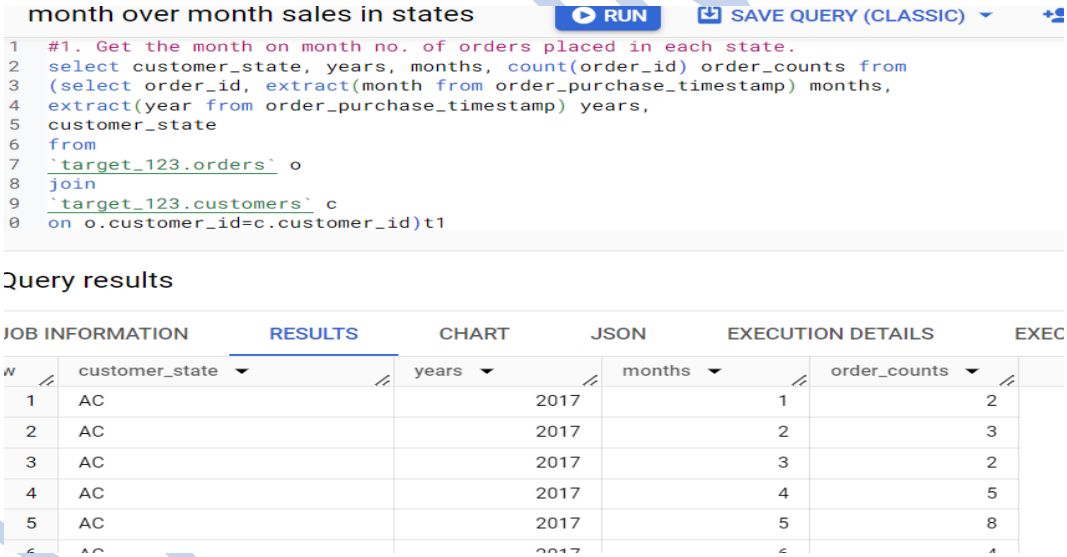
3. **Evolution of E-commerce orders in the Brazil region:**

   1. Get the month on month no. of orders placed in each state.

      Solution:

      select customer_state, years, months, count(order_id) order_counts from
      (select order_id, extract(month from order_purchase_timestamp) months,
      extract(year from order_purchase_timestamp) years,
      customer_state
      from
      `target_123.orders` o
      join
      `target_123.customers` c
      on o.customer_id=c.customer_id)t1
      group by customer_state,years, months
      order by customer_state,years, months;

      Screenshot:



   2. How are the customers distributed across all the states?

      Solution:

      select customer_state,
      count(customer_id) customer_counts
      from `target_123.customers`
      group by customer_state
      order by customer_counts desc;

Screenshot:

customer cou... RUN SAVE QUERY (CLASSIC)

```
# How are the customers distributed across all the states?
select customer_state,
count(customer_id) customer_counts
from `target_123.customers`
group by customer_state
order by customer_counts desc;
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EX

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

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

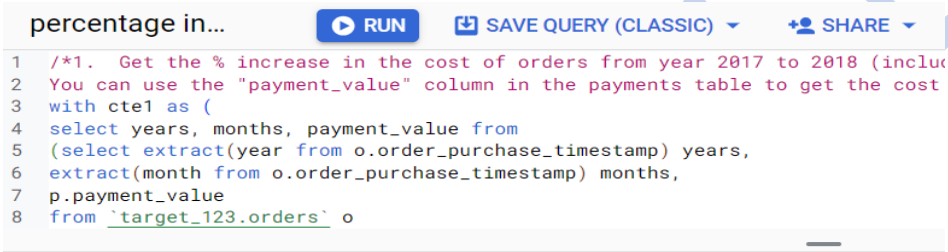
   1. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).
      You can use the "payment_value" column in the payments table to get the cost of orders.

      Solutions:

```
with cte1 as (
select years, months, payment_value from
(select extract(year from o.order_purchase_timestamp) years,
extract(month from o.order_purchase_timestamp) months,
p.payment_value
from `target_123.orders` o
join `target_123.payments` p
on o.order_id = p.order_id)t1
where (months between 1 and 8) and (years between 2017 and 2018)),
cte2 as (
  select months, sum(payment_value) sum_2017 from cte1
  where years=2017
  group by months),
cte3 as (
  select months, sum(payment_value) sum_2018 from cte1
  where years=2018
```

```
 group by months
)
select months, sum_2017, sum_2018, round(sum_2018-sum_2017,2) diff from
(select months, round((sum_2017/total)*100,2) sum_2017,
round((sum_2018/total)*100,2) sum_2018
from
(select cte2.months, sum_2017, sum_2018, (sum_2017+sum_2018) total from
cte2
join cte3
on cte2.months=cte3.months))t1
order by months;
```

Screenshots:



Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS

w	months	sum_2017	sum_2018	diff
1	1	11.05	88.95	77.9
2	2	22.73	77.27	54.54
3	3	27.95	72.05	44.1
4	4	26.47	73.53	47.06
5	5	33.94	66.06	32.12
6	6	33.3	66.7	33.4

2. Calculate the Total & Average value of order price for each state.

Solutions:

```
select customer_state,
round(sum(p.payment_value),2) total_value,
round(avg(p.payment_value),2) avg_value from `target_123.orders` o
join `target_123.customers` c
on o.customer_id=c.customer_id
join `target_123.payments` p
on p.order_id=o.order_id
group by customer_state
order by customer_state;
```

Screenshots:

total & avg val... RUN SAVE QUERY (CLASSIC) SHA

```
#2. Calculate the Total & Average value of order price for each state
select customer_state,
round(sum(p.payment_value),2) total_value,
round(avg(p.payment_value),2) avg_value from `target_123.orders` o
join `target_123.customers` c
on o.customer_id=c.customer_id
join `target_123.payments` p
on p.order_id=o.order_id
group by customer_state
order by customer_state;
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DET

	customer_state	total_value	avg_value
1	AC	19680.62	234.29
2	AL	96962.06	227.08
3	AM	27966.93	181.6
4	AP	16262.8	232.33
5	BA	616645.82	170.82

3. Calculate the Total & Average value of order freight for each state.

Solutions:

select customer_state, round(sum(freight_value),2) total_freight,
round(avg(freight_value),2) avg_freight from `target_123.orders` o
join `target_123.customers` c
on o.customer_id=c.customer_id
join `target_123.order_items` i
on i.order_id=o.order_id
group by customer_state;

Screenshots:

total & Averag... RUN SAVE QUERY (CLASSIC) SHAR

```
#3.  Calculate the Total & Average value of order freight for each st
select customer_state, round(sum(freight_value),2) total_freight,
round(avg(freight_value),2) avg_freight from `target_123.orders` o
join `target_123.customers` c
on o.customer_id=c.customer_id
join `target_123.order_items` i
on i.order_id=o.order_id
group by customer_state;
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETA

	customer_state	total_freight	avg_freight
1	MT	29715.43	28.17
2	MA	31523.77	38.26
3	AL	15914.59	35.84
4	SP	718723.07	15.15
5	MG	270853.46	20.63
6	PE	59449.66	32.92
7	RJ	305589.31	20.96

5. **Analysis based on sales, freight and delivery time.**

   1. Find the no. of days taken to deliver each order from the order's purchase date as delivery time.
      Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
      Do this in a single query.
      You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

      - **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
      - **diff_estimated_delivery** = order_delivered_customer_date - order_estimated_delivery_date

      Solutions:

      select o.order_id,
      date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) time_to_deliver,
      date_diff(o.order_delivered_customer_date,o.order_estimated_delivery_date, day) diff_estimated_delivery
      from `target_123.orders` o;

      Screenshots:

```
select o.order_id, date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) time_to_deliver,
date_diff(o.order_delivered_customer_date,o.order_estimated_delivery_date,day) diff_estimated_delivery
from `target_123.orders` o
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH

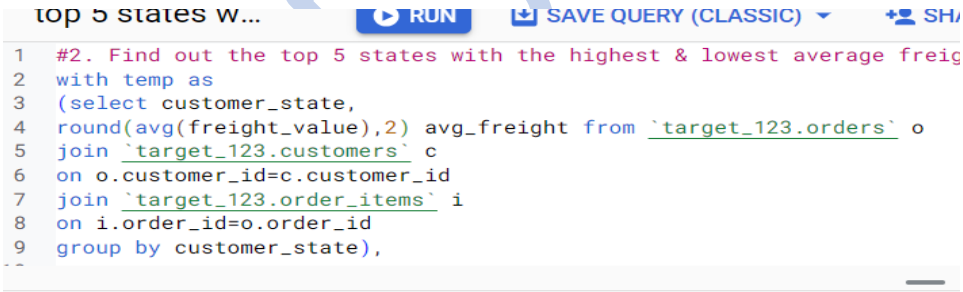
Row	order_id	time_to_deliver	diff_estimated_deliv
1	1950d777989f6a877539f5379...	30	12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	-28
3	65d1e226dfaeb8cdc42f66542...	35	-16
4	635c894d068ac37e6e03dc54e...	30	-1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	-1
7	276e9ec344d3bf029ff83a161c...	43	4

2. Find out the top 5 states with the highest & lowest average freight value.

Solutions:

```
(select customer_state,
round(avg(freight_value),2) avg_freight from `target_123.orders` o
join `target_123.customers` c
on o.customer_id=c.customer_id
join `target_123.order_items` i
on i.order_id=o.order_id
group by customer_state),
cte1 as
(select *, row_number() over(order by avg_freight desc) r1 from temp),
cte2 as
(select *, row_number() over(order by avg_freight) r1 from temp)
select cte1.customer_state as top_5_state, cte2.customer_state as
bottom_5_state from cte1
join cte2
on cte1.r1=cte2.r1
limit 5;
```

Screenshots:



Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DE

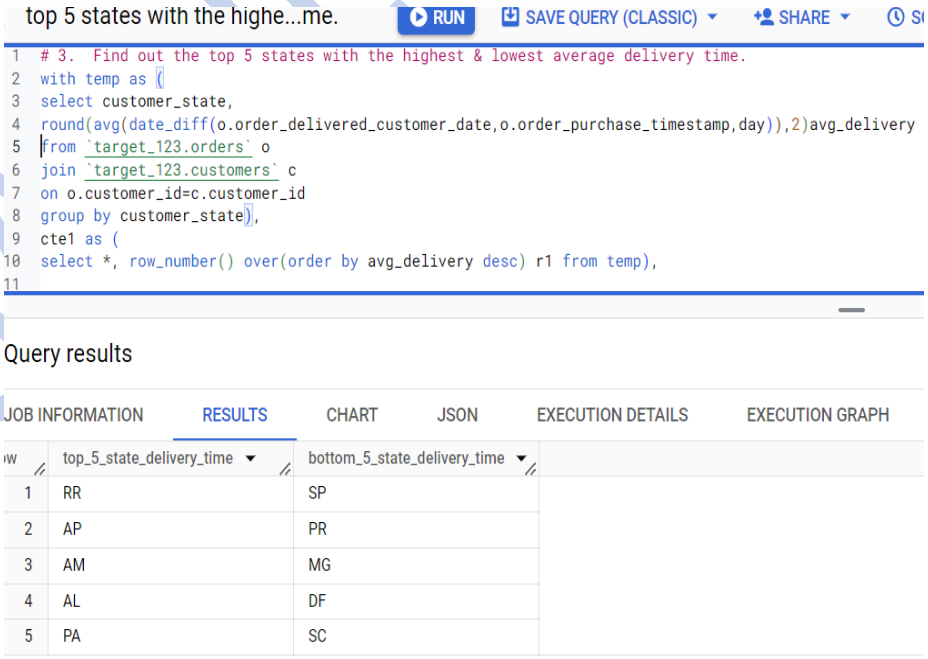
w	top_5_state	bottom_5_state
1	RR	SP
2	PB	PR
3	RO	MG
4	AC	RJ
5	PI	DF

3. Find out the top 5 states with the highest & lowest average delivery time.

Solutions:

```
with temp as (
select customer_state,
round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_time
stamp,day)),2)avg_delivery  from `target_123.orders` o
join `target_123.customers` c
on o.customer_id=c.customer_id
group by customer_state),
cte1 as (
select *, row_number() over(order by avg_delivery desc) r1 from temp),
cte2 as (
select *, row_number() over(order by avg_delivery) r1 from temp
)
select cte1.customer_state as top_5_state_delivery_time, cte2.customer_state
as bottom_5_state_delivery_time
from cte1
join cte2
on cte1.r1=cte2.r1
limit 5;
```

Screenshots:

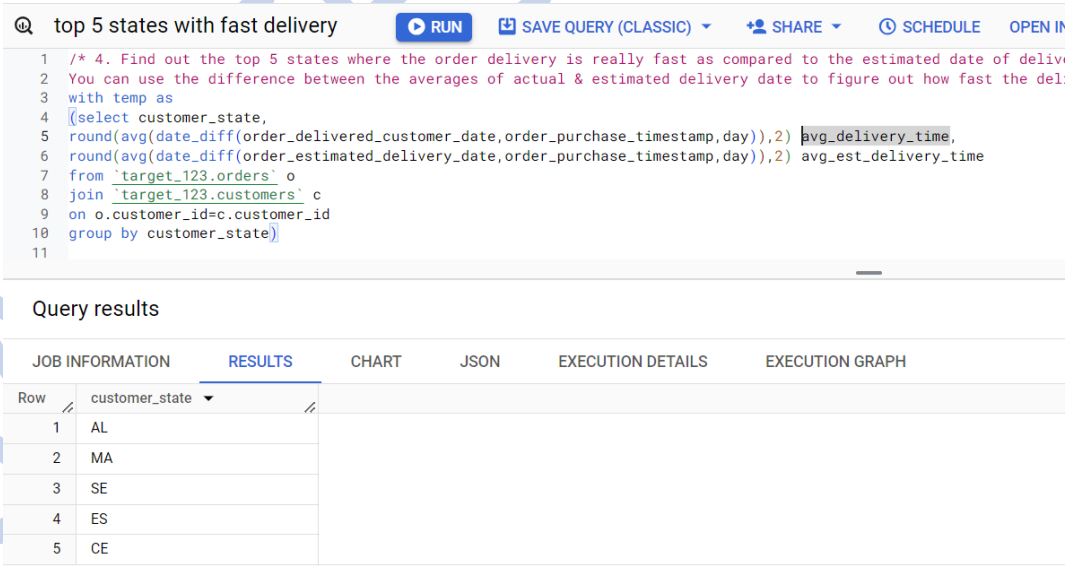


4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

Solutions:

```
with temp as
(select customer_state,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2) avg_delivery_time,
round(avg(date_diff(order_estimated_delivery_date,order_purchase_timestamp,day)),2) avg_est_delivery_time
from `target_123.orders` o
join `target_123.customers` c
on o.customer_id=c.customer_id
group by customer_state)
select customer_state from
(select customer_state,
round(avg_est_delivery_time-avg_delivery_time,2) diff
from temp)t1
order by t1.diff
limit 5;
```

Screenshots:



6. **Analysis based on the payments:**

   1. Find the month on month no. of orders placed using different payment types.

      Solutions:

      select years, months, payment_type, count(*) counts from
      (select payment_type, extract(month from o.order_purchase_timestamp) months,
      extract(year from o.order_purchase_timestamp) years
      from `target_123.payments` p
      join `target_123.orders` o
      on p.order_id=o.order_id)t1
      group by years, months, payment_type
      order by years, months, payment_type;

      Screenshots:

payments types count month ...nth RUN SAVE QUERY (CLASSIC)

```
#1. Find the month on month no. of orders placed using different payment types.
select years, months, payment_type, count(*) counts from
(select payment_type, extract(month from o.order_purchase_timestamp) months,
extract(year from o.order_purchase_timestamp) years
from `target_123.payments` p
join `target_123.orders` o
on p.order_id=o.order_id)t1
group by years, months, payment_type
order by years, months, payment_type;
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXE

w	years	months	payment_type	counts
1	2016	9	credit_card	3
2	2016	10	UPI	63
3	2016	10	credit_card	254
4	2016	10	debit_card	2
5	2016	10	voucher	23
6	2016	12	credit_card	1
7	2017	1	UPI	197
8	2017	1	credit_card	583

2. Find the no. of orders placed on the basis of the payment installments that have been paid.

Solutions:

```
select payment_installments, count(order_id) no_of_orders  from
`target_123.payments`
group by payment_installments
order by payment_installments;
```

Screenshots:

```
# 2.  Find the no. of orders placed on the basis of the payment installments that have

select payment_installments, count(order_id) no_of_orders  from `target_123.payments`
group by payment_installments
order by payment_installments;
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTI

	payment_installment	no_of_orders
1	0	2
2	1	52546
3	2	12413
4	3	10461

Insights:

- Year over year orders increased
- Orders increases in march month year over year
- Orders are high in day time especially in Evening and afternoon
- Orders are high in states (Sao Paulo, Rio de Janeiro, Minas Gerais)
- Most of the orders are paid through credit card
- Most of the payment instalments are one
- Most of the orders delivered before estimated delivery

Recommendations:

1. *Increase the Employees strength in Brazil* because target platform orders are increasing year over year
2. *Increase promotion activities during daytime* because most of orders are done in Afternoon, Evening. Decrease promotion activities in Night time because people are not active at that time.
3. *Increase logistics support in state Sao Paulo, Rio de Janeiro.*
4. Increase payment instalments and promotion offers in non-performing regions. To improve the sales.
5. Provide payment offers in top performing states, to further increase orders.