Business Case: Target SQL

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.

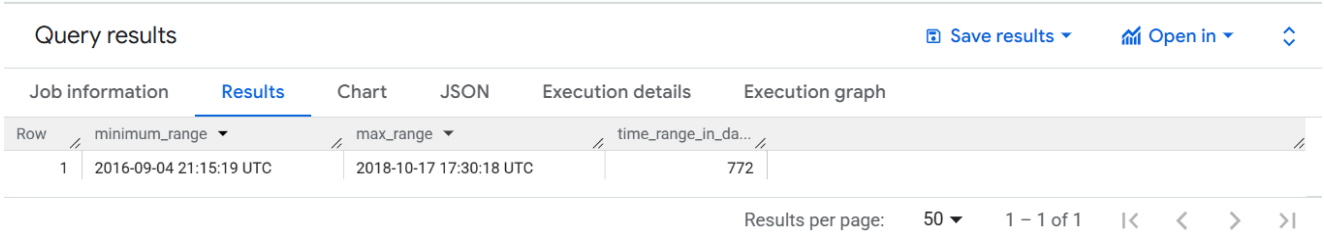
1.Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

```
# 1.Data type of all columns in the "customers" table.
 select column_name, data_type
 from `Target_SQL_business_case.INFORMATION_SCHEMA.COLUMNS`
 where table_name = 'customers'
```

The customers.csv contains following columns/features:

Features	Description
customer_id	ID of the consumer who made the purchase (datatype - STRING)
customer_unique_id	Unique ID of the consumer (datatype - STRING)
customer_zip_code_prefix	Zip Code of consumer's location (datatype - INT
customer_city	Name of the City from where order is made (datatype - STRING)
customer_state	State Code from where order is made (Eg. são paulo - SP) (datatype - STRING)

```
_# 2. Get the time range between which the orders were placed.
select min(order_purchase_timestamp) minimum_range,
 max(order_purchase_timestamp) max_range,
 timestamp_diff(max(order_purchase_timestamp), min(order_purchase_timestamp), day)
time_range_in_days
from `Target_SQL_business_case.orders`
```



Observation:

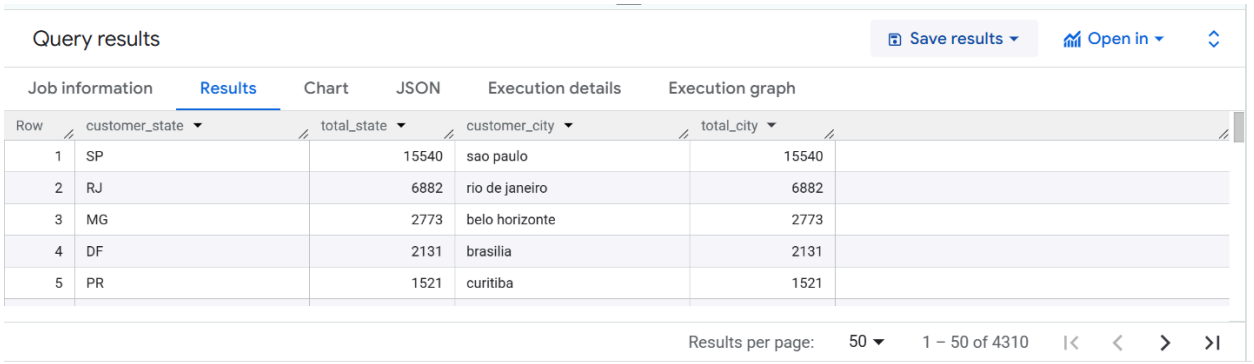
Orders were placed between September 4, 2016, and October 17, 2018.

The first order in this dataset was placed on September 4, 2016, at 21:15:19 UTC, and the last order was placed on October 17, 2018, at 17:30:18 UTC.

This gives us a total coverage of 772 days, meaning the dataset spans over 772 days of order activity.

Recommendations: NA

```
# 3. Count the Cities & States of customers who ordered during the given period.
select c.customer_state, count(distinct c.customer_id) total_state,
c.customer_city, count(distinct c.customer_id) total_city
from `Target_SQL_business_case.customers` c
join `Target_SQL_business_case.orders` o
on c.customer_id = o.customer_id
group by c.customer_state, c.customer_city
order by total_state desc, total_city desc
```

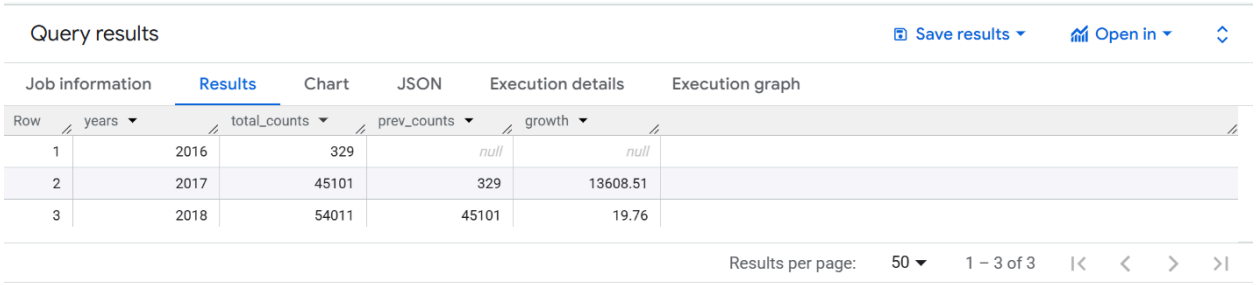


Row	customer_state	total_state	customer_city	total_city
1	SP	15540	sao paulo	15540
2	RJ	6882	rio de janeiro	6882
3	MG	2773	belo horizonte	2773
4	DF	2131	brasilia	2131
5	PR	1521	curitiba	1521

Results per page: 50 1 – 50 of 4310

Insight: Most orders come from São Paulo (SP) and Rio de Janeiro (RJ), indicating a highly concentrated customer base.
Recommendation: Diversify market presence by investing in targeted marketing in emerging cities like Taubaté and Duque de Caxias to reduce regional dependency and capture untapped demand.

```
2. In-depth Exploration:
 # 1. Is there a growing trend in the no. of orders placed over the past years?
select *, round((temp.total_counts - temp.prev_counts)/temp.prev_counts * 100, 2)
growth
from
(select
extract(year from order_purchase_timestamp) years, count(order_id) total_counts,
lag(count(order_id) ) over(order by count(order_id) ) prev_counts
from `Target_SQL_business_case.orders`
group by years) temp
order by years
```



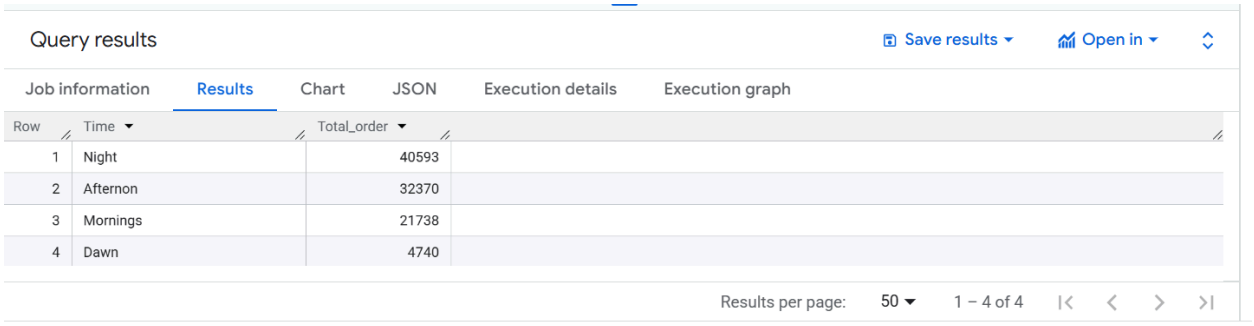
Insight: Orders grew significantly from 2016 to 2017 and continued at a steady pace into 2018.
Recommendation: Proactively scale logistics and support infrastructure to maintain service quality and delivery speed as order volumes increase.

```
# 2. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?
select
format_date('%b', order_purchase_timestamp) months, count(order_id) total_counts
from `Target_SQL_business_case.orders`
group by months
order by total_counts desc
```

Insight: August is a peak sales month, while December shows a notable dip in order volume.

Recommendation: Analyze campaign effectiveness during low-demand periods and consider running targeted promotions in December to balance sales.

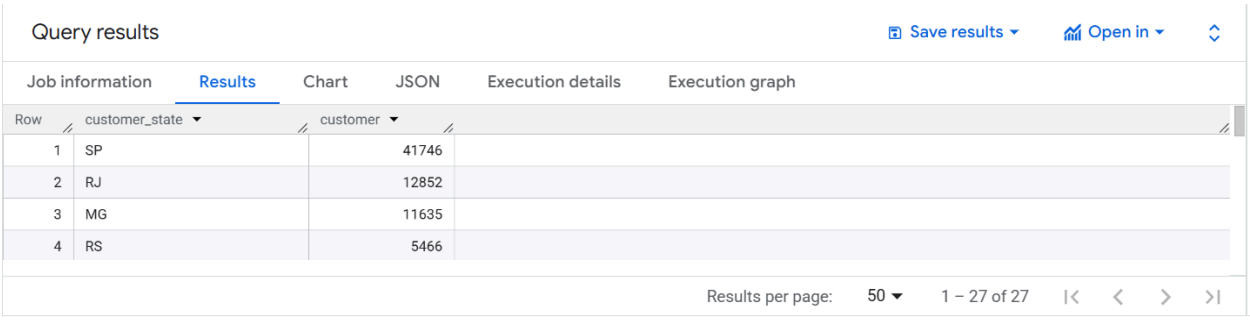
```
# 3. A. During what time of the day, do the Brazilian customers mostly place their orders?

(Dawn, Morning, Afternoon or Night)

#0-6 hrs : Dawn

#7-12 hrs : Mornings

#13-18 hrs : Afternoon

#19-23 hrs : Night

select case when extract(time from order_purchase_timestamp) between '00:00:00' and '06:00:00'

then 'Dawn'

 when extract(time from order_purchase_timestamp) between '07:00:00' and '12:00:00'

then 'Mornings'

 when extract(time from order_purchase_timestamp) between '13:00:00' and '18:00:00'

then 'Afternon'

 else 'Night'



 end Time,

 count(*) Total_order

from `Target_SQL_business_case.orders`

group by Time

order by Total_order desc
```



Insight: A large share of orders are placed between 7 PM and midnight, highlighting customer engagement during evening hours.

Recommendation: Align customer service availability and promotional ads with peak activity hours to improve conversion and support responsiveness.

```
# B.How are the customers distributed across all the states?
select customer_state, count(distinct customer_id) customer
from `Target_SQL_business_case.customers`
group by customer_state
order by 2 desc
```




```
#3.Evolution of E-commerce orders in the Brazil region:
# 1. Get the month on month no. of orders placed in each state.
select extract(month from order_purchase_timestamp ) month, customer_state,
count(order_id) Total_Count
from `Target_SQL_business_case.orders` o
```

```
join `Target_SQL_business_case.customers` c
on c.customer_id = o.customer_id
group by customer_state, extract(month from order_purchase_timestamp )
order by month
```

Query results

Save results | Open in

Job information | Results | Chart | JSON | Execution details | Execution graph

Row	month	customer_state	Total_Count
1	1	SP	3351
2	1	PR	443
3	1	RJ	990
4	1	CE	99
5	1	MG	971
6	1	ES	159
7	1	MA	66
8	1	BA	264

Results per page: 50 | 1 – 50 of 322

```
# 2.How are the customers distributed across all the states?
select distinct customer_state, count(distinct customer_id) count_of_unique_customer
from `Target_SQL_business_case.customers`
group by customer_state
order by 2 desc
```

Query results

Save results | Open in

Job information | Results | Chart | JSON | Execution details | Execution graph

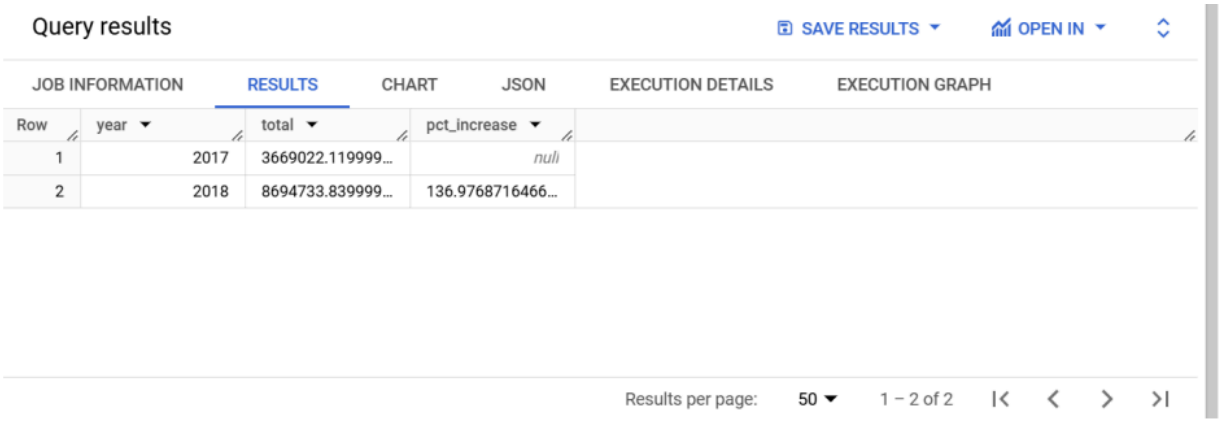
Row	customer_state	count_of_unique_...
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140

Results per page: 50 | 1 – 27 of 27

Insights:

Brazilian state SP have most number of customers

```
# 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).


select *,
(total -lag(total) over(order by year ))/lag(total) over(order by year ) * 100
pct_increase
from
(select extract(year from order_purchase_timestamp) year , sum(p.payment_value) total
from `Target_SQL_business_case.orders` o
join `Target_SQL_business_case.payments` p
 on o.order_id = p
 where extract(year from order_purchase_timestamp) in (2017, 2018)
 and extract(month from order_purchase_timestamp) between 1 and 8
group by extract(year from order_purchase_timestamp)) temp
```



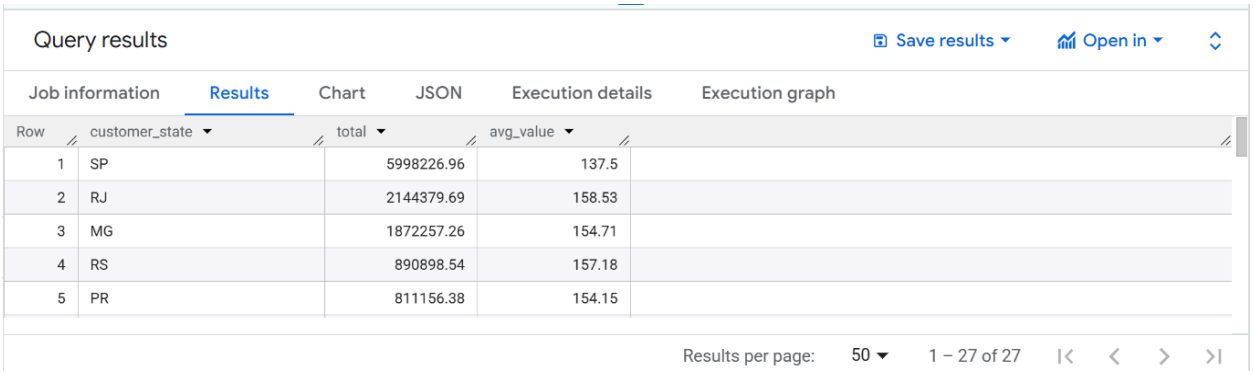
Insight:

The total amount paid by customers almost doubled from January to August in 2017 compared to the same months in 2018.

Recommendation:

Use this strong growth to improve pricing strategies and start loyalty programs to keep customers coming back.

```sql
#4.2 Calculate the Total & Average value of order price for each state.
select c.customer_state, round(sum(p.payment_value),2) total,
round(avg(p.payment_value),2)
avg_value
from `Target_SQL_business_case.orders` o
join `Target_SQL_business_case.payments` p
on o.order_id = p.order_id
join `Target_SQL_business_case.customers` c
 on o.customer_id = c.customer_id
group by c.customer_state
order by 2 desc,3 desc
```



Row	customer_state	total	avg_value
1	SP	5998226.96	137.5
2	RJ	2144379.69	158.53
3	MG	1872257.26	154.71
4	RS	890898.54	157.18
5	PR	811156.38	154.15

Insight:

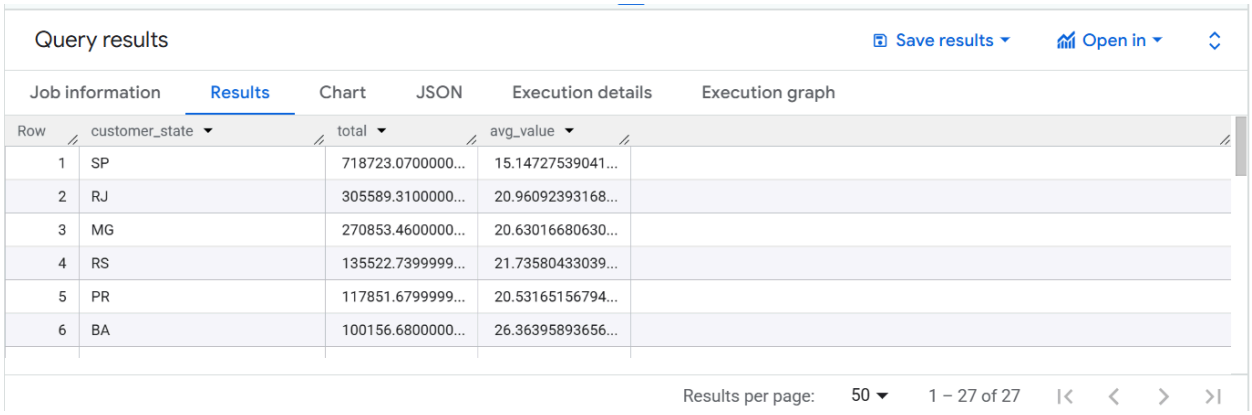
States like **SP (São Paulo)** and **RJ (Rio de Janeiro)** have the highest total order value, showing they are major contributors to overall sales.
However, **PB** and **AC** have the **highest average order value**, meaning customers there tend to spend more per order even if their total volume is lower.

Recommendation:

Focus on **increasing order volume** in high-average-value states like **PB and AC**, where customer spending per order is strong.
Running **targeted promotions or product bundles** in these states could boost total revenue significantly.

```sql
# 4.3 Calculate the Total & Average value of order freight for each state
```

```sql
select c.customer_state, sum(ot.freight_value) total, avg(ot.freight_value) avg_value

from `Target_SQL_business_case.customers` c

join `Target_SQL_business_case.orders` o

 on c.customer_id = o.customer_id

join `Target_SQL_business_case.order_items` ot

 on o.order_id = ot.order_id

group by c.customer_state

order by 2 desc
```



Insights:

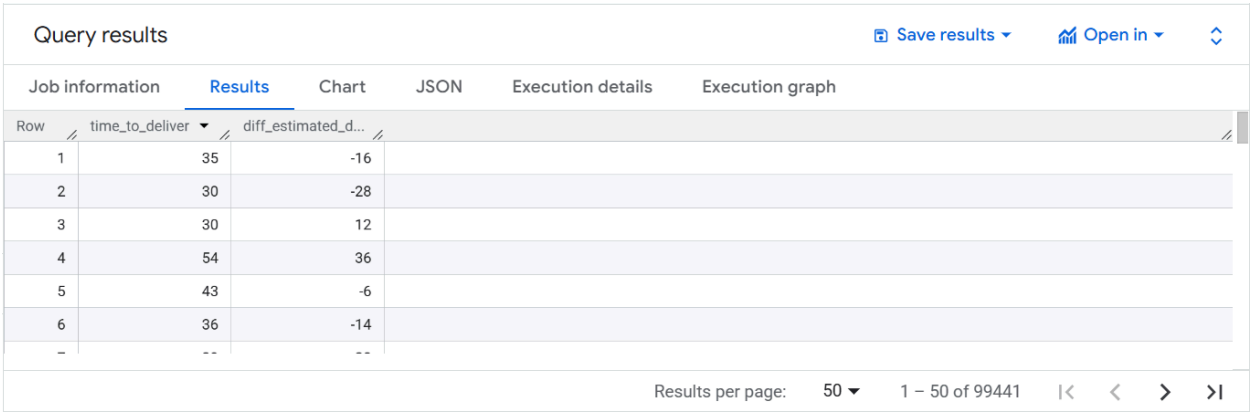
SP, RJ followed by MG have maximum freight charges which means it might cost Target more to ship there.

Recommendation:

Improve logistics or open new delivery hubs in high-freight-cost states to reduce cost and improve delivery speed.

```sql
# 5.Analysis based on sales, freight and delivery time.

/* A 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.

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*/

select datetime_diff(order_delivered_customer_date,order_purchase_timestamp, day)

time_to_deliver,

datetime_diff(order_delivered_customer_date,order_estimated_delivery_date, day )

diff_estimated_delivery

from `Target_SQL_business_case.orders`
```




```
# B. Find out the top 5 states with the highest & lowest average freight value.

 #the top 5 states with lowest average freight value.
```

```sql
 select *

 from

(select customer_state,

avg(freight_value) over(partition by c.customer_id ) lowest_avg_freight

from `Target_SQL_business_case.order_items` ot

join `Target_SQL_business_case.orders` o

  on ot.order_id = o.order_id

join `Target_SQL_business_case.customers` c

  on o.customer_id = c.customer_id

group by c.customer_state , freight_value, c.customer_id

order by 2 ) temp

where lowest_avg_freight != 0.0

limit 5
```


```sql
#highest avg frieght value

select customer_state,

avg(freight_value) over(partition by c.customer_id ) highest_avg_freight

from `Target_SQL_business_case.order_items` ot
```

```
join `Target_SQL_business_case.orders` o
 on ot.order_id = o.order_id
join `Target_SQL_business_case.customers` c
on o.customer_id = c.customer_id
group by c.customer_state , freight_value, c.customer_id
order by 2 desc
limit 5
```

Query results

Save results | Open in

Job information | Results | Chart | JSON | Execution details | Execution graph

Row	customer_state	highest_avg_freight
1	PI	409.68
2	SC	375.28
3	PR	375.28
4	SP	339.59
5	MT	338.3

Results per page: 50 | 1 – 5 of 5

Insights:

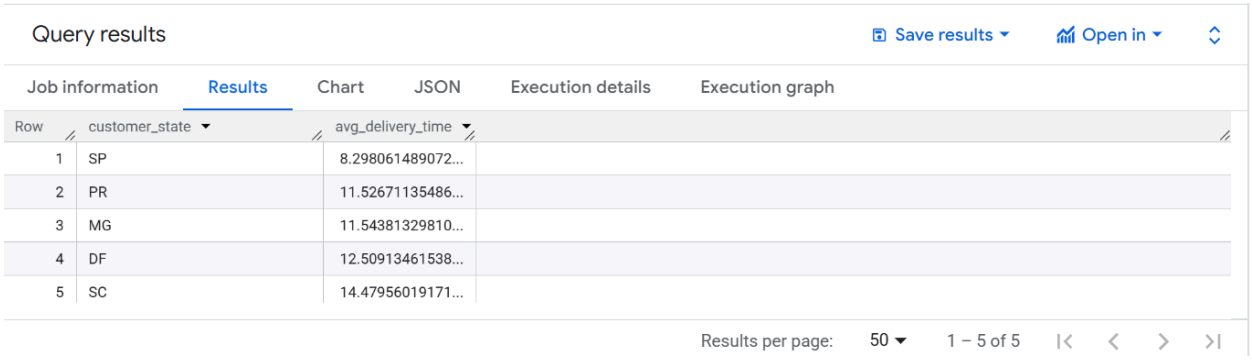
Some states like GO,PR have significantly **lower average freight charges**, indicating that shipping to these locations is more cost-effective.

Recommendations:

Promote free shipping offers in these states with low average freight charges to increase sales

```
# 3. Find out the top 5 states with the highest & lowest average delivery time.
# Find out the top 5 states with the lowest average delivery time.
with delivery_time as (
```

```
 select *, datetime_diff( o.order_delivered_customer_date, o.order_purchase_timestamp,
day)

diff_time

 from `Target_SQL_business_case.orders` o

join `Target_SQL_business_case.customers` c

on o.customer_id = c.customer_id



)

select customer_state, avg(diff_time) avg_delivery_time

from delivery_time

group by 1

order by 2

limit 5
```




```
# Find out the top 5 states with the lowest average delivery time.

with delivery_time as (

 select *, datetime_diff( o.order_delivered_customer_date, o.order_purchase_timestamp,
day)

diff_time
```

```
 from `Target_SQL_business_case.orders` o

join `Target_SQL_business_case.customers` c

on o.customer_id = c.customer_id



)

select customer_state, avg(diff_time)/count(order_id) avg_delivery_time

from delivery_time

group by 1

order by 2

limit 5
```

Insights:

States like RR, AP,AP have high average delivery time indicating they might be far away from target stores. Or they may be in remote location.

States like SP, PR, MG have low average delivery time.

Recommendations:

For States with High Average Delivery Time:

Give Free or Discounted Shipping: This can keep customers happy even if the delivery takes longer.

Improve Delivery Process: Work with better delivery partners or check if routes can be made faster.

For States with Low Average Delivery Time:

Open More Stores or Warehouses: Since delivery is already fast, having more stock nearby can increase sales.

Try New Products Here First: These areas are good for testing new items because customers will receive them quickly and can give feedback sooner.

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

with cte as

(select o.order_id, c.customer_state,

datetime_diff( o.order_delivered_customer_date,o.order_purchase_timestamp, day) order_delivery_time_diff ,

datetime_diff(o.order_estimated_delivery_date, o.order_purchase_timestamp, day) order_estimated_time_diff

from `Target_SQL_business_case.orders` o

join `Target_SQL_business_case.customers` c

on o.customer_id = c.customer_id

where order_delivered_customer_date is not null and order_purchase_timestamp is not null

)

select customer_state,
```

```
round(sum(order_estimated_time_diff-order_delivery_time_diff)/count(order_id))
state_with_fast_delivery

from cte

group by 1

order by 2 desc

limit 5
```

Insights: AC has the fastest order delivery compared to the estimated date as compared to other states.

```
#6. Analysis based on the payments:

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

select extract(month from order_purchase_timestamp ) month,payment_type, count(*)
total_payments

from `Target_SQL_business_case.payments` p

join `Target_SQL_business_case.orders` o

 on p.order_id = o.order_id

group by month, payment_type

order by 1, 3 desc
```

Query results

Save results | Open in

Job information | Results | Chart | JSON | Execution details | Execution graph

Row	month	payment_type	total_payments
1	1	credit_card	6103
2	1	UPI	1715
3	1	voucher	477
4	1	debit_card	118
5	2	credit_card	6609
6	2	UPI	1723
7	2	voucher	424
8	2	debit_card	82

Results per page: 50 | 1 – 50 of 50

Insights:

In Jan and Feb, most of the payments were done using Credit Card

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

select payment_installments, count(distinct order_id) total_orders

from `Target_SQL_business_case.payments`

where payment_value <> 0.0

group by payment_installments

order by payment_installments
```

Recommendations:

- Target more cities beyond SP and RJ.
- Use seasonal insights to time your promotions.
- Match marketing efforts with evening order spikes.
- Improve logistics in high-cost or slow-delivery states.
- Promote high-value bundles in states with high average spend.
- Focus campaigns around popular payment modes.