DSML-NOV23-SQL
Dec 25, 2023 1:45 AM

TARGET
SQL Business case

By Vishal Gopalkrishna
vishal.luna545@gmail.com *+91 97380 33193*

INTRODUCTION

Target is a multi- chain retailer with offline and online presence across multiple countries. Here we study the transactions in Brazil particularly between the years 2016 and 2018. We analyse various business aspects such as orders, buyers, sellers, geo locations, pricing, product attributes, reviews, etc.

We conduct an analysis by answering a few fundamental business Problems/questions that enable us to gain valuable insights into Target's business performance and operations in Brazil.

Problems:

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

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

Field name	Type
customer_id	STRING
customer_unique_id	STRING
customer_zip_code_prefix	INTEGER
customer_city	STRING
customer_state	STRING

Observation: One out of the 6 fields in the customers table is an integer field - customer_zip_code_prefix.

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

```
SELECT
 min(order_purchase_timestamp) as min_time_range,
 max(order_purchase_timestamp) as max_time_range
FROM
 `Target.orders`
LIMIT
 10
```

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

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

Observation: The orders in the dataset were placed in the time range between the above time stamps.

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

```
SELECT
 count(distinct(customer_city)) as no_of_cities,
 count(distinct(customer_state)) as no_of_states
FROM
 `Target.customers`
LIMIT
 10
```

Query results

JOB INFORMATION | RESULTS | CHA

Row	no_of_cities	no_of_states
1	4119	27

Observation: The customers are from 4119 different cities across 27 states.

2. In-depth Exploration:

2.1. Is there a growing trend in the number of orders placed over the past years?

```
SELECT
*,
 CONCAT(ROUND(((next_order-no_of_orders)*100/no_of_orders), 1),"%") AS
Growth
FROM (
 SELECT
  *
```

```
  LEAD(no_of_orders) OVER (ORDER BY yearmonth) AS next_order
FROM (
  SELECT
  FORMAT_DATE("%y-%m", order_purchase_timestamp) AS yearmonth,
    COUNT(order_id) AS no_of_orders
  FROM
    `Target.orders`
  GROUP BY
    Yearmonth
        ORDER BY
     yearmonth ) AS T
ORDER BY
T.yearmonth)
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTIO

Row	yearmonth	no_of_orders	next_order	Growth
1	16-09	4	324	8000%
2	16-10	324	1	-99.7%
3	16-12	1	800	79900%
4	17-01	800	1780	122.5%
5	17-02	1780	2682	50.7%
6	17-03	2682	2404	-10.4%
7	17-04	2404	3700	53.9%
8	17-05	3700	3245	-12.3%
9	17-06	3245	4026	24.1%
10	17-07	4026	4331	7.6%

Observation: We observe that, the overall growth during the 2 years has certainly been positive. Although there are ups and downs in the growth rate

over these months. These growth rates going back and forth suggest seasonality and/or other external factors that may be at play.
Also, the data points towards the beginning and end of the time range seem to be inconsistent with the rest of the data suggesting potential loss/missing of Data.
Finally, taking into account the status of the orders will give us a clearer picture of the orders as some of them can be placed and later cancelled which may not reflect the true business performance.

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

Observation: It is hard to conclude Seasonality with just 2 years of data. We observe that during the months Dec-Jan of 2016 and 2017, there has been a significant growth in orders. The orders go up during halloween and black-friday during 2017, but data during this period for 2016 and 2018 seem missing.
Albeit, these data points are not sufficient to make conclusive forecasting for the coming business seasons nor to highlight seasonality.

2.3. 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

```
select TT.time_of_day, count(TT.order_id) as no_of_orders

from (select *,

case

when timestamp between "00:00:00" and "07:00:00" then "Dawn"

when timestamp between "07:00:00" and "13:00:00" then "Mornings"

when timestamp between "13:00:00" and "19:00:00" then "Afternoons"

else "Night"

end as time_of_day
```

```
from
(select order_id,
time(order_purchase_timestamp) as timestamp
from `Target.orders` ) T) as TT
group by 1
order by 2 desc
```

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	time_of_day	no_of_orders
1	Afternoons	38133
2	Night	28331
3	Mornings	27735
4	Dawn	5242

Observation: We observe that, Brazilian customers place most orders during the Afternoons, followed by Night and Mornings. This suggests a compelling reason to run marketing campaigns and offers during this period.

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

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

```
select *, concat(round(((no_of_orders-next_order)/no_of_orders)*100, 1), "%") as growth
from (select *,
lag(no_of_orders) over (partition by customer_state order by customer_state, yearmonth) as next_order
from (select customer_state, yearmonth, count(order_id) as no_of_orders
from (select O.order_id, O.customer_id, C.customer_state,
format_date("%Y-%m",O.order_purchase_timestamp) as yearmonth
```

```
from `Target.orders` as O

inner join `Target.customers` as C

on O.customer_id = C.customer_id

) as T

group by 1, 2

order by 1, 2) as TT

order by 1, 2) as TTT
```

Query results

SAVE RESULTS

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

Row	customer_state	yearmonth	no_of_orders	next_order	growth
1	AC	2017-01	2	*null*	*null*
2	AC	2017-02	3	2	33.3%
3	AC	2017-03	2	3	-50%
4	AC	2017-04	5	2	60%
5	AC	2017-05	8	5	37.5%
6	AC	2017-06	4	8	-100%
7	AC	2017-07	5	4	20%
8	AC	2017-08	4	5	-25%
9	AC	2017-09	5	4	20%
10	AC	2017-10	6	5	16.7%

Observation:

No significant observation.

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

```
select customer_state, count(distinct(customer_id)) as no_of_customers

from (select O.order_id, O.customer_id, C.customer_state

from `Target.orders` as O

inner join `Target.customers` as C

on O.customer_id = C.customer_id) T

group by 1

order by 2 desc
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETA

Row	customer_state	no_of_customers
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

Observation: The top three states with most customers are from SP, RJ, and MG. Further city-level break down and business operations analysis may reveal interesting data elements to construct a compelling business plan.

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

4.1. 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.

```
select *,

concat(round(((next_year_pay-sum_pay_val)*100/sum_pay_val), 1), "%") as
increase_percent

from (select *,

lead(TT.sum_pay_val, 12) over (order by yearmonth) as next_year_pay

from (select yearmonth, round(sum(payment_value), 2) as sum_pay_val

from (select O.order_id,

format_date("%Y-%m", O.order_purchase_timestamp) as yearmonth, P.payment_value

from `Target.orders` as O

inner join  `Target.payments` as P

on O.order_id = P.order_id) as T
```

```
group by 1

order by 1) as TT

where yearmonth between "2017-01" and "2018-08"

order by 1) as TTT

where yearmonth < "2017-09"
```

Query results

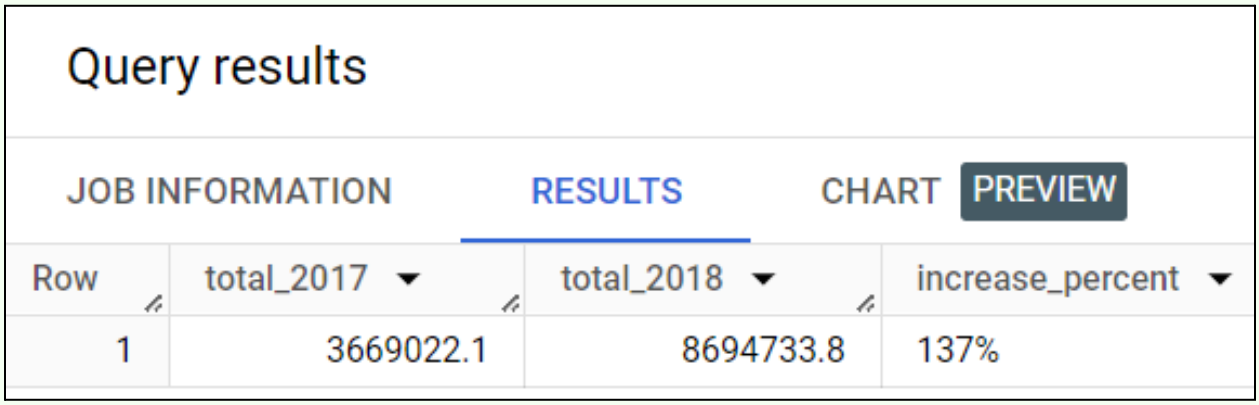
JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS

Row	yearmonth	sum_pay_val	next_year_pay	increase_percent
1	2017-01	138488.04	1115004.18	705.1%
2	2017-02	291908.01	992463.34	240%
3	2017-03	449863.6	1159652.12	157.8%
4	2017-04	417788.03	1160785.48	177.8%
5	2017-05	592918.82	1153982.15	94.6%
6	2017-06	511276.38	1023880.5	100.3%
7	2017-07	592382.92	1066540.75	80%
8	2017-08	674396.32	1022425.32	51.6%

Next: For the total percentage increase from 2017 to 2018 only on the months 01-08 in both years:

```
with base as(

  select *,

concat(round(((next_year_pay-sum_pay_val)*100/sum_pay_val), 1), "%") as
increase_percent

from (select *,

lead(TT.sum_pay_val, 12) over (order by yearmonth) as next_year_pay

from (select yearmonth, round(sum(payment_value), 2) as sum_pay_val

from (select O.order_id,

format_date("%Y-%m", O.order_purchase_timestamp) as yearmonth, P.payment_value

from `Target.orders` as O

inner join  `Target.payments` as P
```

```
on O.order_id = P.order_id) as T

group by 1

order by 1) as TT

where yearmonth between "2017-01" and "2018-08"

order by 1) as TTT

where yearmonth < "2017-09"

)

select round(sum(base.sum_pay_val), 1) total_2017,

round(sum(base.next_year_pay), 1) as total_2018,

concat(round(((sum(base.next_year_pay)-sum(base.sum_pay_val))*100/sum(base.sum_p
ay_val)), 1), "%") as increase_percent

from base
```



Observation: The percentage increase in cost of orders from 2017 to 2018 is calculated to be 137%. Understanding the factors contributing to this growth can help the business appropriate its budget allocation, pricing strategies, etc. It also helps them identify increase in demand for certain products which enables them to make strategic decision according to the market trends.

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

```
select customer_state, round(avg(payment_value), 1) as avg_order_price,
round(sum(payment_value), 1) as total_order_price

from (select T.*, P.payment_value

from (select O.order_id, O.customer_id, O.order_purchase_timestamp,
C.customer_state

from `Target.orders` as O

inner join `Target.customers` as C

on O.customer_id = C.customer_id) as T

inner join `Target.payments` as P

on T.order_id = P.order_id) as TT

group by 1

order by 2 desc
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

Row	customer_state	avg_order_price	total_order_price
1	PB	248.3	141545.7
2	AC	234.3	19680.6
3	RO	233.2	60866.2
4	AP	232.3	16262.8
5	AL	227.1	96962.1
6	RR	218.8	10064.6
7	PA	215.9	218295.8
8	SE	208.4	75246.3
9	PI	207.1	108524.0
10	TO	204.3	61485.3

Observation: The total price of orders is highest in the states SP, RJ and MG which are also the top three states for the number of customers. However, the top 3 states for the average order price are PB, AC and RO. This may indicate that the customers here are not frequent shoppers hence they may

buy high value products a few times as compared to SP, RJ and MG where there are more people placing more orders of less value.

But this is not a conclusion, a product level analysis may reveal more features of customers behavious across the states.

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

```
select customer_state, round(avg(TT.freight_sum), 1) as avg_freight_val,
round(sum(TT.freight_sum), 1) as total_freight_val

from (select T.*, F.freight_sum

from (select O.order_id, O.customer_id, O.order_purchase_timestamp,
C.customer_state

from `Target.orders` as O

inner join `Target.customers` as C

on O.customer_id = C.customer_id) as T

inner join (select order_id,

round(sum(freight_value), 2) as freight_sum

from `Target.order_items`

group by 1) as F

on T.order_id = F.order_id) as TT

group by 1

order by 3 desc
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

Row	customer_state	avg_freight_val	total_freight_val
1	SP	17.4	718723.1
2	RJ	23.9	305589.3
3	MG	23.5	270853.5
4	RS	24.9	135522.7
5	PR	23.6	117851.7
6	BA	29.8	100156.7
7	SC	24.8	89660.3
8	PE	36.1	59449.7
9	GO	26.5	53115.0
10	DF	23.8	50625.5

Observation: The top 3 states for Total Order freight value is as expected the top 3 states in number of customers. But their average freight value is significantly lower than other states, indicating a strong frequent buyer customer bas in these states for Target.

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

5.1. 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.

```
select order_id,

date_diff(order_delivered_customer_date, order_purchase_timestamp, day) as time_to_deliver,

date_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery

from `Target.orders`
```

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW | JSON | E

Row	order_id	time_to_deliver	diff_estimated_delivery
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
8	54e1a3c2b97fb0809da548a59...	40	-4
9	fd04fa4105ee8045f6a0139ca5...	37	-1
10	302bb8109d097a9fc6e9cefc5...	33	-5

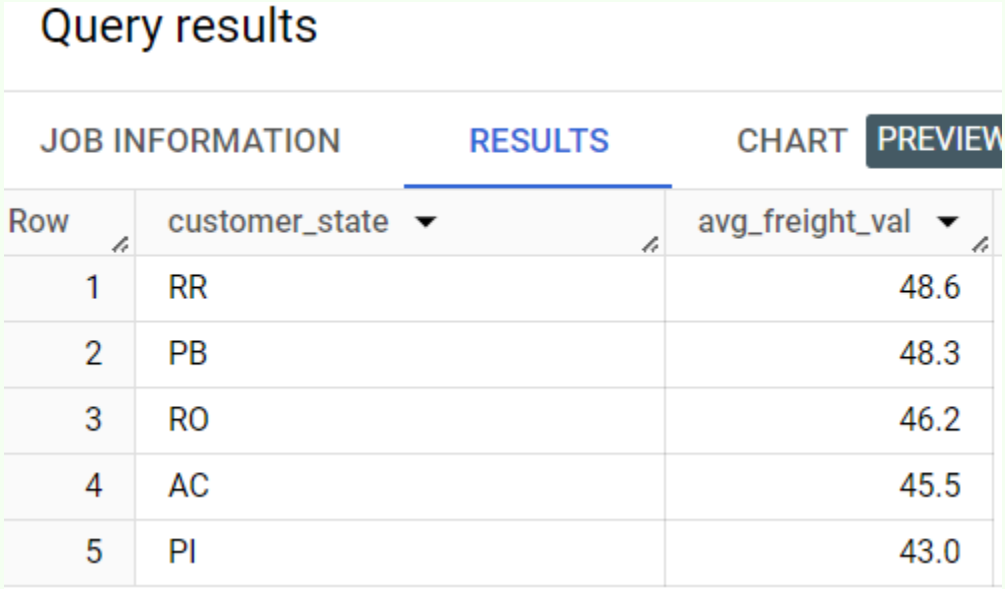
Observation: The negative number in the diff_estimated_delivery field represent the orders that more days to deliver than the estimated date. Similarly, a positive number in this field denotes the "early" deliveries and 0 denotes staying exactly on target with the delivery estimations.

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

Top Five States with Highest Average Freight Value:

```
select customer_state, round(avg(TT.freight_sum), 1) as avg_freight_val

from (select T.*, F.freight_sum

from (select O.order_id, O.customer_id, O.order_purchase_timestamp,
C.customer_state

from `Target.orders` as O

inner join `Target.customers` as C

on O.customer_id = C.customer_id) as T

inner join (select order_id,

round(sum(freight_value), 2) as freight_sum
```

```
from `Target.order_items`

group by 1) as F

on T.order_id = F.order_id) as TT

group by 1

order by 2 desc

limit 5
```



Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

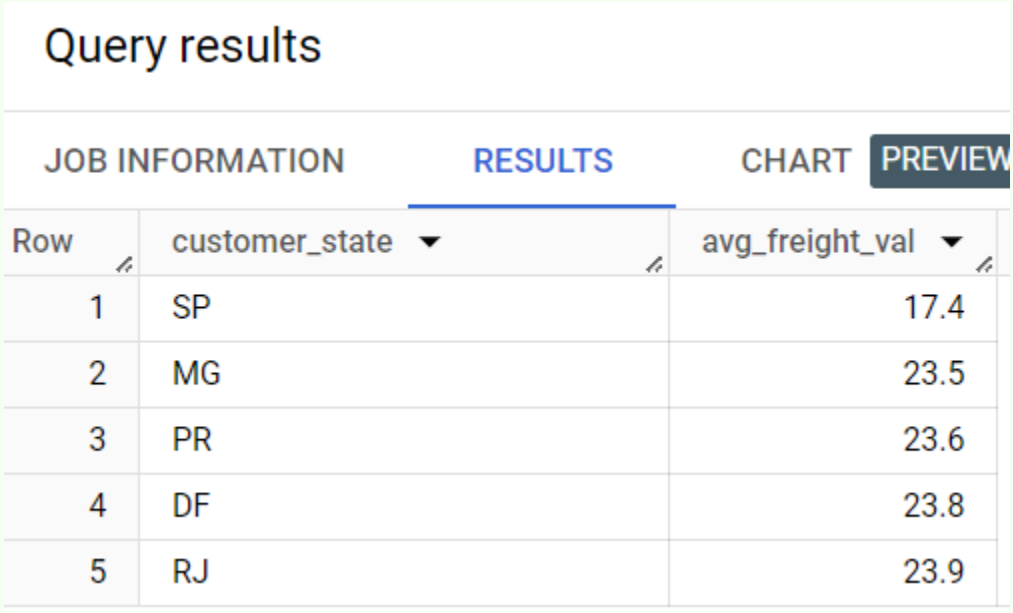
Row	customer_state	avg_freight_val
1	RR	48.6
2	PB	48.3
3	RO	46.2
4	AC	45.5
5	PI	43.0

Observation: The top 5 states RR, PB, RO, AC and PI.

Top Five States with Lowest Average Freight Value:

```
select customer_state, round(avg(TT.freight_sum), 1) as avg_freight_val

from (select T.*, F.freight_sum

from (select O.order_id, O.customer_id, O.order_purchase_timestamp,
C.customer_state

from `Target.orders` as O

inner join `Target.customers` as C

on O.customer_id = C.customer_id) as T

inner join (select order_id,

round(sum(freight_value), 2) as freight_sum
```

```
from `Target.order_items`

group by 1) as F

on T.order_id = F.order_id) as TT

group by 1

order by 2 asc

limit 5
```



Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	customer_state	avg_freight_val
1	SP	17.4
2	MG	23.5
3	PR	23.6
4	DF	23.8
5	RJ	23.9

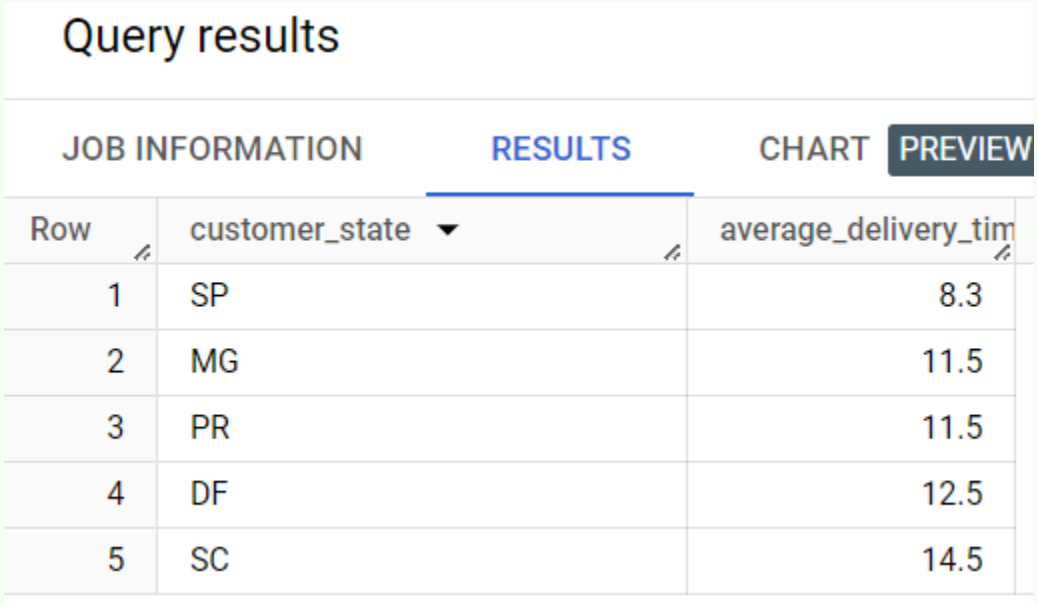
Observation: The bottom 5 states are SP, MG, PR, DF and RJ.

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

Lowest:

```
select customer_state, round(avg(T.time_to_deliver), 1) as average_delivery_time

from (select O.order_id, O.customer_id,

date_diff(O.order_delivered_customer_date, O.order_purchase_timestamp, day) as time_to_deliver,

C.customer_state

from `Target.orders` as O

inner join `Target.customers` as C
```

```
on O.customer_id = C.customer_id) as T

group by 1

order by 2

limit 5
```



Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

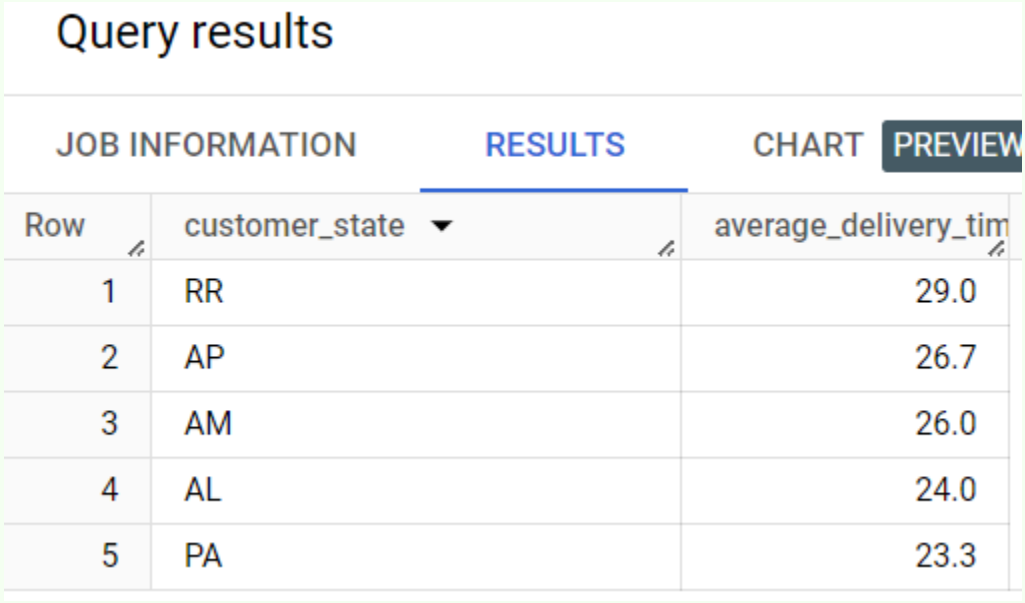
Row	customer_state	average_delivery_tim
1	SP	8.3
2	MG	11.5
3	PR	11.5
4	DF	12.5
5	SC	14.5

Observation: The bottom 5 states with average delivery time are SP, MG, PR, DF and SC.

Highest:

```
select customer_state, round(avg(T.time_to_deliver), 1) as average_delivery_time

from (select O.order_id, O.customer_id,

date_diff(O.order_delivered_customer_date, O.order_purchase_timestamp, day) as time_to_deliver,

C.customer_state

from `Target.orders` as O

inner join `Target.customers` as C

on O.customer_id = C.customer_id) as T

group by 1

order by 2 desc
```

```
limit 5
```



Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	customer_state	average_delivery_tim
1	RR	29.0
2	AP	26.7
3	AM	26.0
4	AL	24.0
5	PA	23.3

Observation: The top 5 states with highest average delivery time are RR, AP. AM, AL, PA.

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

```
select TT.*, round(TT.average_estimated_delivery_time-TT.average_delivery_time, 1) as diff_of_avg

from (select customer_state, round(avg(T.time_to_deliver), 1) as average_delivery_time,

round(avg(T.estimated_time_to_deliver), 1) as average_estimated_delivery_time

from (select O.order_id, O.customer_id,

date_diff(O.order_delivered_customer_date, O.order_purchase_timestamp, day) as time_to_deliver,

date_diff(O.order_estimated_delivery_date, O.order_purchase_timestamp, day) as estimated_time_to_deliver,

C.customer_state

from `Target.orders` as O

inner join `Target.customers` as C

on O.customer_id = C.customer_id) as T
```

```
group by 1) TT

order by 4 desc

limit 5
```

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW | JSON | EXECUTION DE

Row	customer_state	average_delivery_time	average_estimated_delivery_time	diff_of_avg
1	AC	20.6	40.8	20.2
2	RO	18.9	38.4	19.5
3	AP	26.7	45.7	19.0
4	AM	26.0	44.8	18.8
5	RR	29.0	46.2	17.2

Observation: AC, RO, AP, AM and RR are the top 5 states where the order delivery is really fast as compared to the estimated date of delivery. An interesting point to consider is that the technique use to calculate the estimation for delivery time can be flawed.

6. Analysis based on payments:

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

```
select *, concat(round(((no_of_orders-next_order)/no_of_orders)*100, 2), "%") as growth

from (select *,

lag(no_of_orders) over (partition by payment_type order by payment_type, yearmonth) as next_order

from (select payment_type, yearmonth, count(order_id) as no_of_orders

from (select O.order_id, O.customer_id, P.payment_type,

format_date("%Y-%m",O.order_purchase_timestamp) as yearmonth

from `Target.orders` as O

inner join `Target.payments` as P

on O.order_id = P.order_id
```

```
) as T

group by 1, 2

order by 1, 2) as TT

order by 1, 2) as TTT
```

Query results SAVE RESUL

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

Row	payment_type	yearmonth	no_of_orders	next_order	growth
1	UPI	2016-10	63	*null*	*null*
2	UPI	2017-01	197	63	68.02%
3	UPI	2017-02	398	197	50.5%
4	UPI	2017-03	590	398	32.54%
5	UPI	2017-04	496	590	-18.95%
6	UPI	2017-05	772	496	35.75%
7	UPI	2017-06	707	772	-9.19%
8	UPI	2017-07	845	707	16.33%
9	UPI	2017-08	938	845	9.91%
10	UPI	2017-09	903	938	-3.88%

Observation: Above is the month on month no. of orders placed using different payment types.

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

```
select P.payment_installments, count(O.order_id)

from `Target.orders` as O

inner join `Target.payments` as P

on O.order_id = P.order_id

group by 1

order by 1
```

Query results

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

Row	payment_installments	order_count
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644

Observation: Above is the breakdown of the no. of orders placed on the basis of the payment installments that have been paid.

Actionable Insights and Recomendations

1. Address the inconsistencies/incompletness of the data at the beginning and end of the timeframe.
2. Investigate deeper into the Top 3 states with the largest customer base - SP, RJ and MG. Collect demographical data to understand the customer behaviors and preferences for tailored marketing campaigns.
3. Understand the contributing factors for the increase in order costs from 2017 to 2018. It could be due to product pricing changes, increased demand for higher-priced items, or other economic factors.
4. Explore reasons behind faster deliveries in specific states and assess if replicable strategies can be implemented across other regions.
5. Use payment type preferences to run promotional offers, discounts, and cashbacks to hold and grow the customer base.

The end.