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 analysing 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:

Data type of all columns in the "customers" table is shown below and we can also inquire for other tables data type in the same way.
It is essential to identify the data types to perform operations on the tables or we can say to communicate with tables.

target query

RUN SAVE QUERY SHARE SCHEDULE MORE

```
select table_name,column_name,data_type
from target.INFORMATION_SCHEMA.COLUMNS
where table_name='customers'
```

Press Alt+F1 for Accessibility Options.

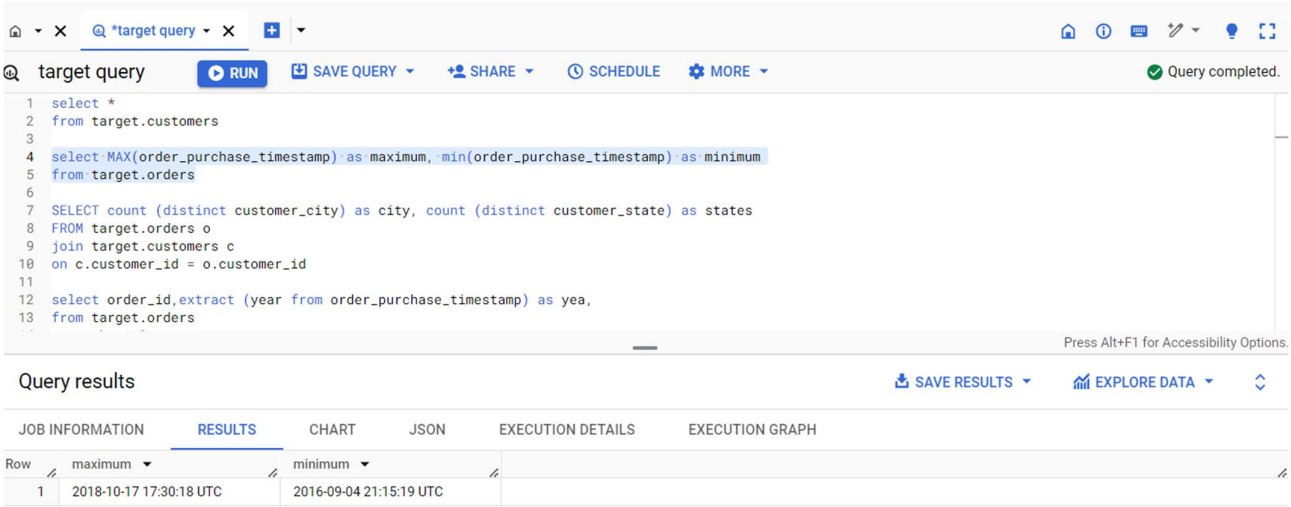
Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS CHART JSON EXECUTION DETAILS EXECUTION GRAPH

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

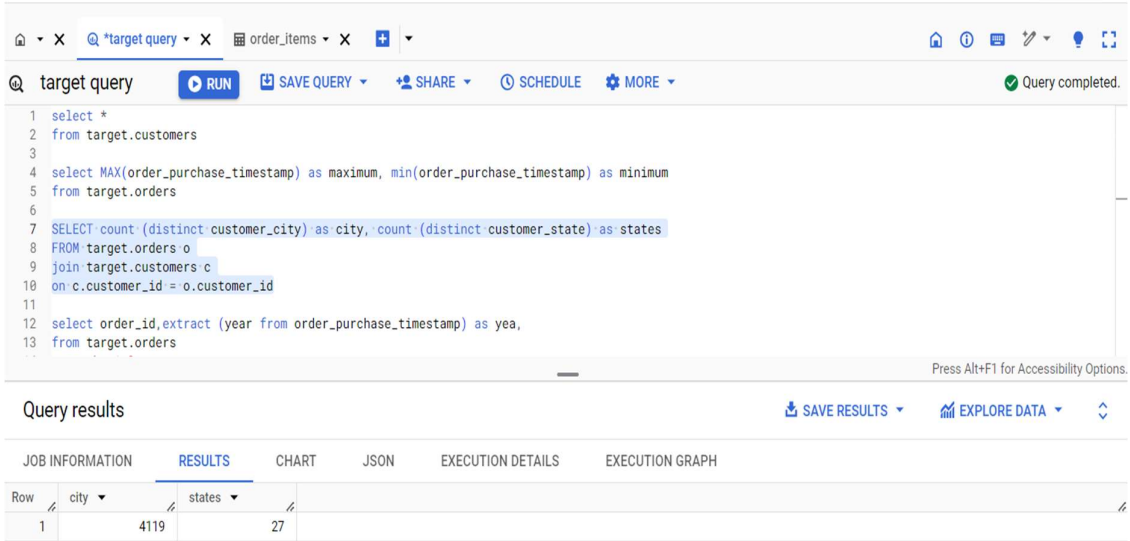
1.2 Get the time range between which the orders were placed.



With the help of **orders table** where the purchase time is given I applied max() and min() function to find the maximum and minimum date when the orders were placed.
The minimum date is 2016-09-04 and the maximum date is 2018-10-17 so the data given is for 2016 to 2018.

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

Cities and state information available in **customers table** and order placed information available in **orders table**, customer_id is the common key in both the tables to join them, we will perform right join as the customer id must be there in the orders table who had made a purchase.

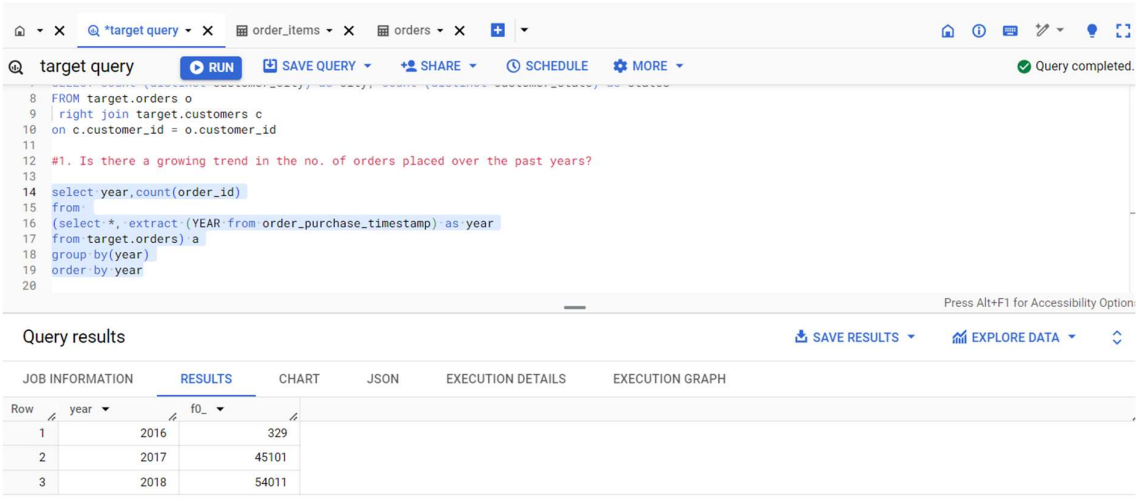


Total number of states are 27 and total no of cities are 4119 from which the orders were placed during the given time period.

2 In-depth Exploration:

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

-Yes with the help of given query we can check the no. of orders are growing over the past years.
Recommendations- as the no. of order grown from 2016 to 2017 rapidly but from 2017 to 2018 the no. of orders are not grown as rapidly as per previous year so need to focus.



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

```
from target.orders) a
group by(year)
order by year


# Can we see some kind of monthly seasonality in terms of the no. of orders being placed?
select year,month,count(order_id) as total_no_of_orders
from
(select *, extract (YEAR from order_purchase_timestamp) as year,extract(month from order_purchase_timestamp ) as month
from target.orders) a
group by year,month
order by year,month
```

Query results

Row	year	month	total_no_of_orders
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682

3.During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night).
- Highest no. of orders placed during the Afternoon.

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


select case when o.time1 >= '0:0:0' and o.time1<= '6:59:59' then 'Dawn'
 when o.time1>= '7:0:0' and o.time1<='12:59:59' then 'Mornings'
 when o.time1>='13:0:0' and o.time1 <= '18:59:59' then 'Afternoon'
 when o.time1>='19:0:0' and o.time1 <= '23:59:59' then 'Night'
 end as cat, count(o.time1) as no_of_orders
from
(select extract (time from order_purchase_timestamp) as time1
from target.orders) o
group by 1
order by 2 desc
```

Query results

Row	cat	no_of_orders
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

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

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

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

select c.customer_state, extract(month from order_purchase_timestamp) as month, count(c.customer_state) as total_no_order
from target.orders o
right join target.customers c
on c.customer_id = o.customer_id
group by 1,2
order by 1,2
```

Query results

Row	customer_state	month	total_no_order
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5

Results per page: 50 1 – 50 of 322

3.3 How are the customers distributed across all the states?

```
# How are the customers distributed across all the states?

select customer_state, count(customer_state) as total_no
from target.customers
group by 1
order by 2 desc
```

Query results

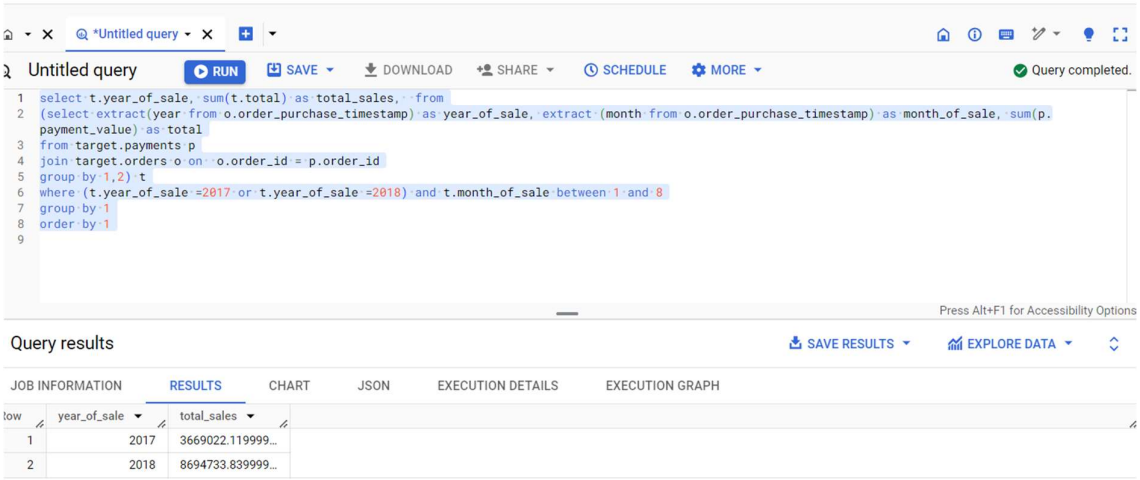
Row	customer_state	total_no
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045

Results per page: 50 1 – 27 of 27

Total no. of customers distributed among 27 states with highest no. of customer from states SP.
Recommendations- Target should focus in other states as well to increase the no. of orders as no of customers are very less in other states as compared to state SP.

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

4.2 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.



From year 2017 to 2018 the sales amount increase 0.42%

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

-As the state information available in **customer table**, order details and price details available in **order_items and orders table** so to extract the information firstly we have joined the tables and then extracted the total no. of orders and avg of order price among all states.

SP, PR, RS, MG, ES are the top 5 states where the average value of order price is highest.

*distributi... ers | *Untitled query | orders | customers

Untitled query — RUN | SAVE | DOWNLOAD | SHARE | SCHEDULE | MORE | Syntax error

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

select c.customer_state, sum(oi.price) as total, avg(oi.price) as average
from target.order_items oi
join target.orders o on o.order_id = oi.order_id
left join target.customers c on c.customer_id = o.customer_id
group by 1
order by 3
```

Query results — SAVE RESULTS

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

Row	customer_state	total	average
1	SP	5202955.050002...	109.6536291597...
2	PR	683083.7600000...	119.0041393728...
3	RS	750304.0200000...	120.3374530874...
4	MG	1585308.029999...	120.7485741488...
5	ES	275037.3099999...	121.9137012411...

Results per page: 50 1

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

*distributi... ers | *Untitled query

Untitled query — RUN | SAVE | DOWNLOAD | SHARE | SCHEDULE | MORE

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

select c.customer_state, sum(oi.freight_value) as total, avg(oi.freight_value) as average
from target.order_items oi
join target.orders o on o.order_id = oi.order_id
left join target.customers c on c.customer_id = o.customer_id
group by 1
order by 3
```

Query results — SAVE RESULTS

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

Row	customer_state	total	average
1	SP	718723.0699999...	15.14727539041...
2	PR	117851.6800000...	20.53165156794...
3	MG	270853.4600000...	20.63016680630...
4	RJ	305589.3100000...	20.96092393168...
5	DF	50625.49999999...	21.04135494596...

Results per page: 50 1 – 2

SP, PR, MG, RJ,DF are the top 5 states where the average value of order FREIGHT is highest.

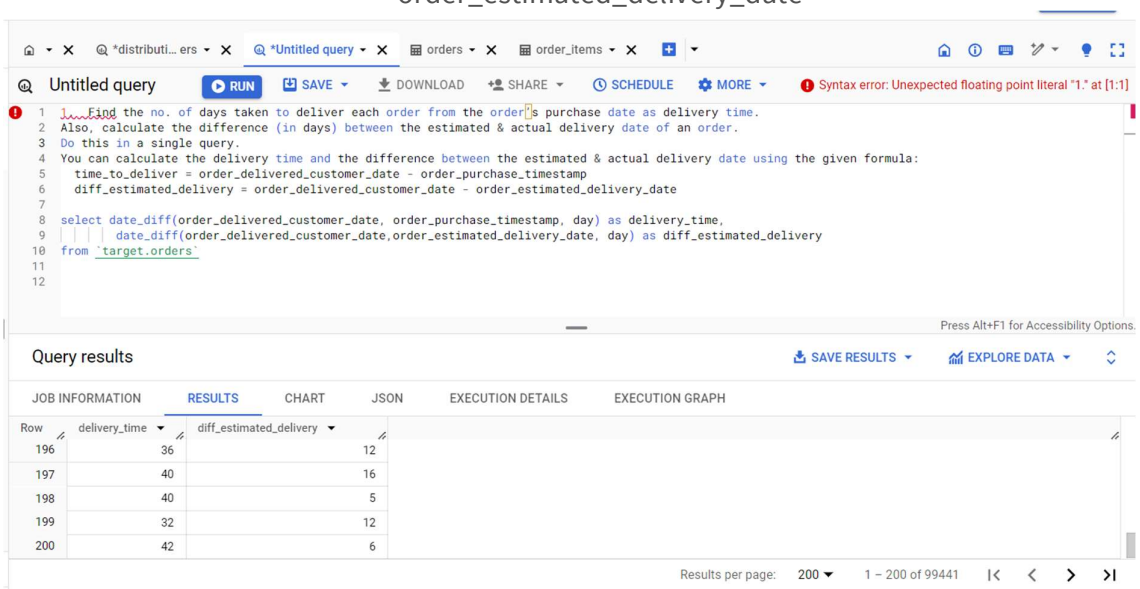
5 **Analysis based on sales, freight and delivery time.**

5.2 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:

5.2.1 **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp

5.2.2 **diff_estimated_delivery** = order_delivered_customer_date - order_estimated_delivery_date



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

TOP 5 states with lowest avg freight value.

*distributi... ers | *Untitled query | freight | *Untitled query

Untitled query

RUN | SAVE | DOWNLOAD | SHARE | SCHEDULE | MORE

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

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

Press Alt

Query results

SAVE RESULTS | EXF

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

Row	customer_state	total	average
1	SP	718723.0699999...	15.14727539041...
2	PR	117851.6800000...	20.53165156794...
3	MG	270853.4600000...	20.63016680630...
4	RJ	305589.3100000...	20.96092393168...
5	DF	50625.49999999...	21.04135494596...

TOP 5 states with heighest avg freight value.

*distributi... ers | *Untitled query | freight | *Untitled query

Untitled query

RUN | SAVE | DOWNLOAD | SHARE | SCHEDULE | MORE | Query completed

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

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

Press Alt+F1 for Accessibility Option

Query results

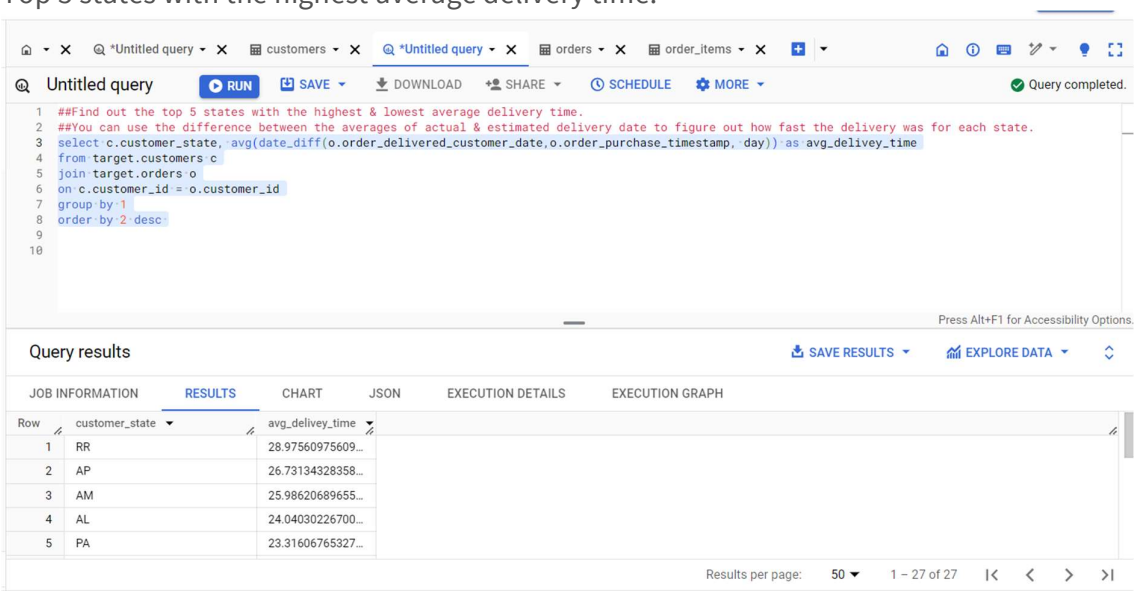
SAVE RESULTS | EXPLORE DATA

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

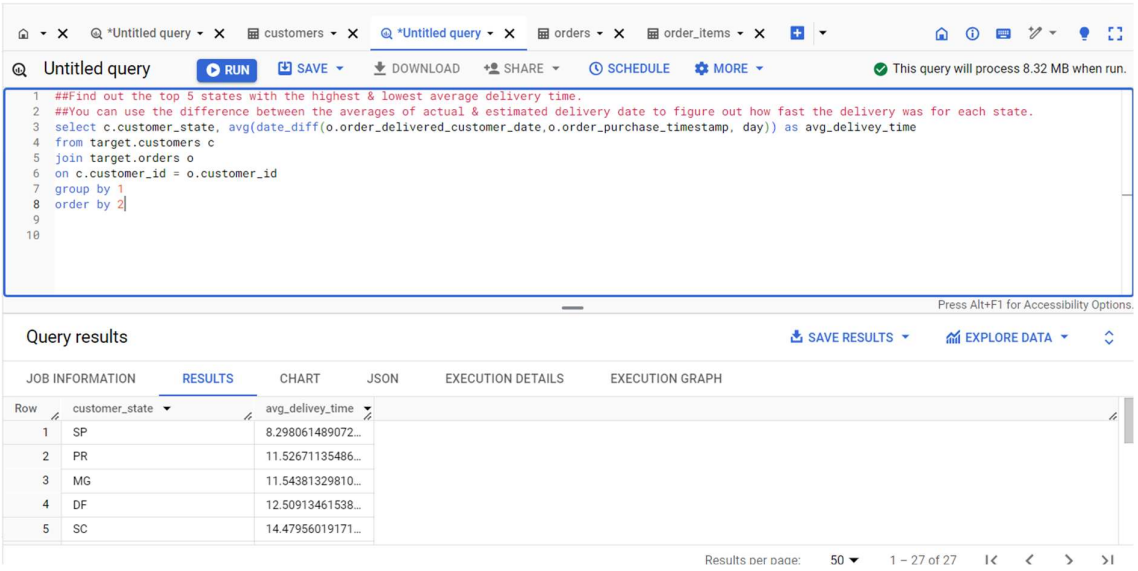
Row	customer_state	total	average
1	RR	2235.190000000...	42.98442307692...
2	PB	25719.73000000...	42.72380398671...
3	RO	11417.38000000...	41.06971223021...
4	AC	3686.750000000...	40.07336956521...
5	PI	21218.2	39.14797047970...

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

Top 5 states with the highest average delivery time.

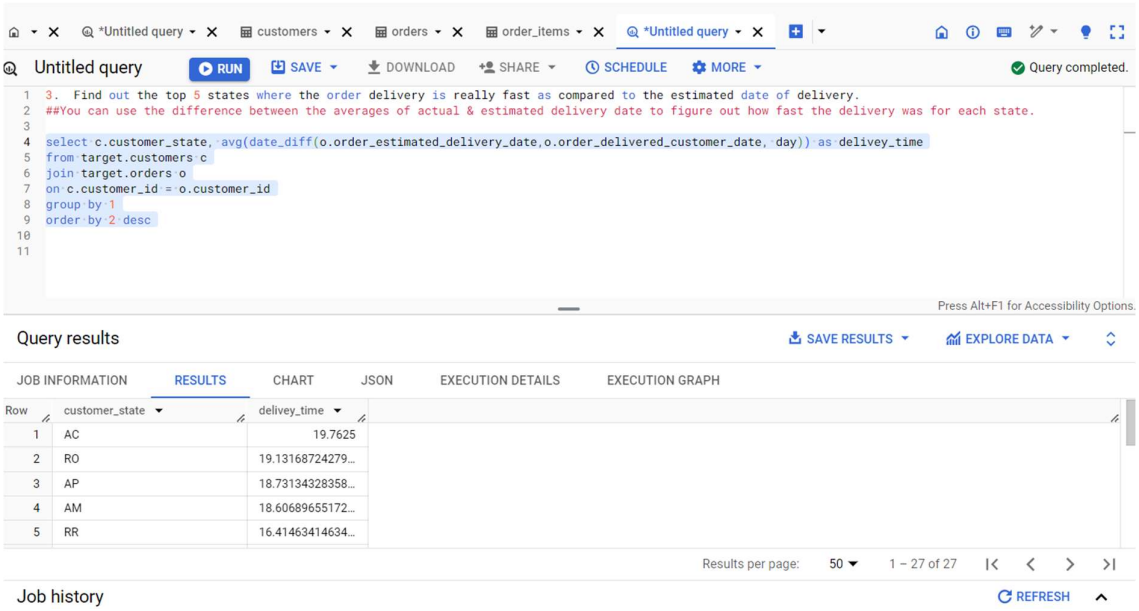


Top 5 states with the lowest average delivery time.



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

You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.



Analysis based on the payments:

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

-I have counted the no. of orders placed using different payment types month on month which shows the payment through upi and credit card is more as compared to debit card and voucher.

*Untitled query | orders | payments | *Untitled query

Untitled query

RUN | SAVE | DOWNLOAD | SHARE | SCHEDULE | MORE

```
select t.months, countif(t.payment_type = 'UPI') AS UPI, countif(t.payment_type = 'voucher') as voucher,countif(t.payment_type = 'credit_card') as
credit_card,countif(t.payment_type = 'debit_card') as debit_card, countif(t.payment_type = 'not_defined') as nd from (
select o.*,p.*,format_date('%Y-%m',o.order_purchase_timestamp) as months
from target.orders o
join target.payments p
on p.order_id = o.order_id) t
GROUP BY 1
```

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	months	UPI	voucher	credit_card	debit_card	nd
1	2016-09	0	0	3	0	0
2	2016-10	63	23	254	2	0
3	2016-12	0	0	1	0	0
4	2017-01	197	61	583	9	0
5	2017-02	398	119	1356	13	0
6	2017-03	590	200	2016	31	0
7	2017-04	496	202	1846	27	0
8	2017-05	772	289	2853	30	0
9	2017-06	707	239	2463	27	0

Results per page: 50 1 – 25 of 25

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

- Maximum no. of orders placed where only 1st instalment is paid.

*Untitled query | orders | payments | *Untitled query | avg deliv... ime | *l

Untitled query

RUN | SAVE | DOWNLOAD | SHARE | SCHEDULE | MORE

```
select p.payment_installments, count(o.order_id) as no_of_orders
from target.orders o
join target.payments p
on p.order_id = o.order_id
group by 1
order by 2 desc
```

Query results

SA

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

Row	payment_installment	no_of_orders
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920
9	7	1626