

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

## A) Data type of columns in a table

Create a new project → Create dataset → Create Table → Then click on schema

## B) Time period for which the data is given

Start from (2016-09-04 21:15:19 UTC)

SELECT

MIN(order\_purchase\_timestamp)

FROM `Target\_Data.orders`

Latest is (2018-10-17 17:30:18 UTC)

SELECT

MAX(order\_purchase\_timestamp)

FROM `Target\_Data.orders`

## C) Cities and States of customers ordered during the given period

For that we need to create a table of geolocation

SELECT

\*

FROM `Target\_Data.geolocation`

The screenshot displays the Google Cloud BigQuery interface. On the left, the 'Explorer' pane shows a project named 'my-project-25578-sql' with a dataset 'Target\_Data' containing tables 'geolocation' and 'orders'. The main editor shows a SQL query: 

```
SELECT * FROM `Target_Data.geolocation`
```

. Below the editor, the 'Query results' section is visible, showing a table with 5 rows and 6 columns: 'geolocation\_zip', 'geolocation\_lat', 'geolocation\_lng', 'geolocation\_city', and 'geolocation\_state'. The results show data for Aracaju, SE. The bottom of the screen shows a Windows taskbar with an 'Activate Windows' watermark.

Row	geolocation_zip	geolocation_lat	geolocation_lng	geolocation_city	geolocation_state
1	49010	-10.910514...	-37.052400...	aracaju	SE
2	49047	-10.9268145	-37.071063...	aracaju	SE
3	49030	-10.970164...	-37.061643...	aracaju	SE
4	49048	-10.940183...	-37.070850...	aracaju	SE
5	49050	-10.927157...	-37.063078...	aracaju	SE

## 2. In-depth Exploration:

- A) 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?

```
SELECT
COUNT(order_id),
EXTRACT(YEAR FROM order_purchase_timestamp)AS YEAR,
EXTRACT(MONTH FROM order_purchase_timestamp)AS MONTH
FROM
`Target_Data.orders`
GROUP BY
EXTRACT(YEAR FROM order_purchase_timestamp),
EXTRACT(MONTH FROM order_purchase_timestamp)
```

The screenshot shows the Google Cloud BigQuery console. On the left, the Explorer pane displays the project hierarchy: my-project-25578-sql > Target\_Data > orders. The main editor shows a SQL query titled 'Untitled 4'. The query is:   
SELECT  
COUNT(order\_id),  
EXTRACT(YEAR FROM order\_purchase\_timestamp)AS YEAR,  
EXTRACT(MONTH FROM order\_purchase\_timestamp)AS MONTH  
FROM  
`Target\_Data.orders`  
GROUP BY  
EXTRACT(YEAR FROM order\_purchase\_timestamp),  
EXTRACT(MONTH FROM order\_purchase\_timestamp)  
The query results are displayed in a table with columns: Row, f0\_, YEAR, MONTH. The results show a growing trend in the number of orders (f0\_) over the years 2016, 2017, and 2018, with a peak in the month of November (MONTH 11) in 2017.

Row	f0_	YEAR	MONTH
17	4331	2017	8
18	6512	2018	8
19	2682	2017	3
20	3700	2017	5
21	324	2016	10

Yes there is a clear growing trend (YOY) as the numbers of order are increasing.

And also seasonality peaks can be observed in the months of November, December and January as these are festive and holiday months .

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

```
SELECT
COUNT(order_id),
EXTRACT(HOUR FROM order_purchase_timestamp) AS TIME
FROM
`Target_Data.orders`
GROUP BY
EXTRACT(HOUR FROM order_purchase_timestamp)
```

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

DISMISS ACTIVATE

Google Cloud My Project 25578 SQL Search (/) for resources, docs, products, and more Search

SANDBOX Set up billing to upgrade to the full BigQuery experience. [Learn more](#) DISMISS UPGRADE

Explorer + ADD

Q Type to search

Viewing workspace resources. SHOW STARRED ONLY

- my-project-25578-sql
  - External connections
  - Target\_Data
    - customers
    - geolocation
    - order\_items
    - order\_reviews
    - orders
    - payments
    - products
    - sellers

Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	id_	TIME
1	6578	11
2	1170	1
3	6150	17
4	6518	13
5	5995	12
6	5769	18
7	6177	10
8	6217	21
9	6454	15
10	5816	22
11	6675	16
12	6193	20
13	2394	0
14	4123	23

Load more

Results per page: 50 1-24 of 24

PERSONAL HISTORY PROJECT HISTORY REFRESH

Most of the orders are placed in afternoon and night time as people of brazil must be having free time at these time intervals.

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

A) Get month on month orders by states

```
SELECT count(order_id),EXTRACT (MONTH FROM order_purchase_timestamp) AS MONTH,customer_state AS state
FROM`Target_Data.orders` AS o left join `Target_Data.customers` AS c ON o.customer_id=c.customer_id
GROUP BY EXTRACT (MONTH FROM order_purchase_timestamp),customer_state
```

SANDBOX Set up billing to upgrade to the full BigQuery experience. [Learn more](#) DISMISS UPGRADE

Explorer + ADD

Q Type to search

Viewing workspace resources. SHOW STARRED ONLY

- my-project-25578-sql
  - External connections
  - Target\_Data
    - customers
    - geolocation
    - order\_items
    - order\_reviews
    - orders
    - payments
    - products
    - sellers

Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

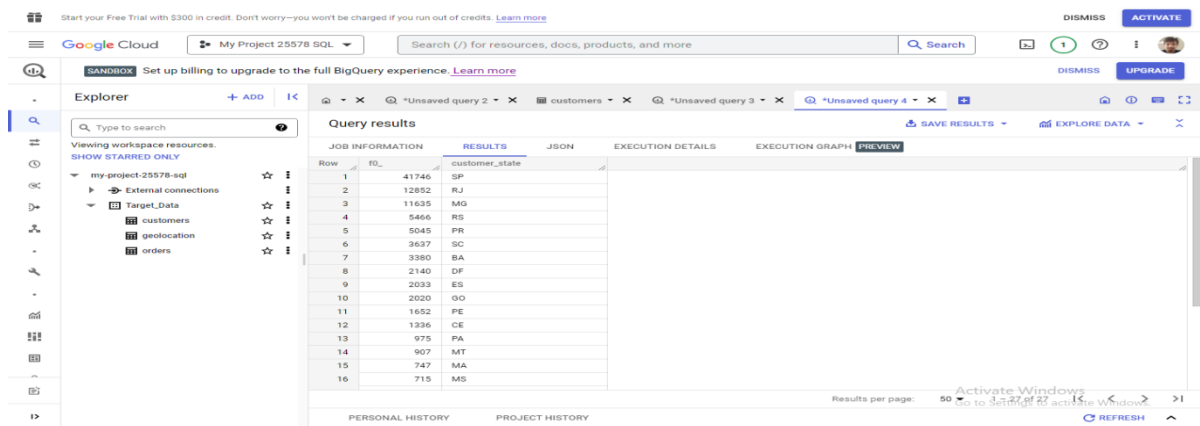
Row	id_	MONTH	state
1	1048	11	RJ
2	283	12	RS
3	2357	12	SP
4	196	2	DF
5	378	11	PR
6	92	4	MT
7	79	7	MA
8	40	7	AL
9	4381	7	SP
10	85	7	MT
11	1111	7	MG
12	1190	5	MG
13	4632	5	SP
14	174	5	PE
15	1908	10	SP
16	990	1	RJ

Results per page: 50 1-50 of 322

PERSONAL HISTORY PROJECT HISTORY REFRESH

## B) Distribution of customers across the states in Brazil

```
SELECT
  COUNT(*),
  customer_state
FROM `Target_Data.customers`
GROUP BY
  customer_state
ORDER BY
  COUNT(*) DESC
```



The screenshot shows the Google Cloud BigQuery interface. On the left, the Explorer pane lists workspace resources, including 'my-project-25578-sql', 'External connections', 'Target\_Data', 'customers', 'geolocation', and 'orders'. The main pane displays the 'Query results' for an unsaved query. The results are shown in a table with columns 'Row', 'f0\_', and 'customer\_state'. The table lists 16 rows of data, sorted by count in descending order. The states and their corresponding counts are: SP (41746), RJ (12852), MG (11635), RS (5466), PR (5045), SC (3637), BA (3380), DF (2140), ES (2033), GO (2020), PE (1652), CE (1336), PA (975), MT (907), MA (742), and MS (715). The interface also includes a top navigation bar with 'Google Cloud' logo, a search bar, and buttons for 'DISMISS' and 'ACTIVATE'. A bottom status bar shows 'Results per page: 50' and a 'REFRESH' button.

Row	f0_	customer_state
1	41746	SP
2	12852	RJ
3	11635	MG
4	5466	RS
5	5045	PR
6	3637	SC
7	3380	BA
8	2140	DF
9	2033	ES
10	2020	GO
11	1652	PE
12	1336	CE
13	975	PA
14	907	MT
15	742	MA
16	715	MS

It seems that most of the customers are located in the richest state of Brazil. As it is obvious that the purchasing power is high of the people of São Paulo and more number of customers is because it has the largest population.

Note that Roraima with lowest GDP contribution is at the bottom when it comes to number of customers.

So it can be concluded that the GDP contribution of a state has direct relation with the number of customers.

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

A) Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use "payment value" column in payments table.

```
with base as (
select
  extract (year from order_purchase_timestamp) as year_,
  sum(payment_value) as cogs from `Target_Data.orders` o
inner join
  `Target_Data.payments` p
on o.order_id=p.order_id
where extract (month from order_purchase_timestamp) between 1 and 8
and extract (year from order_purchase_timestamp) between 2017 and 2018 group by 1
),
```

```
base_2 as (
select *, lag (cogs,1) over (order by year_ asc) as prev_cogs from base) select *, round((c
ogs-prev_cogs)/prev_cogs *100,0) as per_inc from base_2
```

The screenshot shows the Google Cloud BigQuery console. On the left is the Explorer pane showing the project structure. The main area displays a SQL query in 'Untitled 5' and its results in a table.

**Query:**

```
1 with base as (
2 select
3 extract (year from order_purchase_timestamp) as year_,
4 sum(payment_value) as cogs from `Target_Data.orders` o
5 inner join
6 `Target_Data.payments` p
7 on o.order_id=p.order_id
8 where extract (month from order_purchase_timestamp) between 1 and 8
9 and extract (year from order_purchase_timestamp) between 2017 and 2018 group by 1
10 ),
11 base_2 as (
12 select *, lag (cogs,1) over (order by year_ asc) as prev_cogs from base) select *, round((cogs-prev_cogs)/prev_cogs *100,0) as per_inc
13 from base_2
```

**Query results table:**

Row	year_	cogs	prev_cogs	per_inc
1	2018	8694733.83...	3669022.11...	137.0
2	2017	3669022.11...	null	null

## B) Mean & Sum of price and freight value by customer state

```
select c.customer_state ,round(sum(payment_value)) as sum,round(avg(payment_value)) as mean
from `Target_Data.customers` c join `Target_Data.orders` o on c.customer_id = o.customer_id join `Target_
Data.payments` p on o.order_id = p.order_id
where o.order_status = 'delivered'
group by 1
```

The screenshot shows the Google Cloud BigQuery console with a different query in 'Untitled 4'. The results table shows the sum and mean of payment values grouped by customer state.

**Query results table:**

Row	customer_state	sum	mean
1	GO	334234.0	163.0
2	SP	5770266.0	136.0
3	RS	861802.0	155.0
4	BA	591271.0	170.0
5	MG	1819278.0	154.0
6	MT	181442.0	200.0
7	RJ	2055690.0	158.0
8	SC	595208.0	163.0
9	SE	70289.0	204.0
10	PE	309075.0	185.0
11	TO	60007.0	203.0
12	CE	266464.0	199.0
13	PR	781920.0	152.0
14	PA	212028.0	216.0

## 5. Analysis on sales, freight and delivery time

## A) Calculate days between purchasing, delivering and estimated delivery

```
SELECT
customer_id,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day) AS days_to_deliver
FROM `Target_Data.orders`
```

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

DISMISS [ACTIVATE](#)

Google Cloud My Project 25578 SQL Search (/) for resources, docs, products, and more [Search](#)

SANDBOX Set up billing to upgrade to the full BigQuery experience. [Learn more](#) DISMISS [UPGRADE](#)

Explorer + ADD [K](#)

Query results [SAVE RESULTS](#) [EXPLORE DATA](#) [X](#)

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH [PREVIEW](#)

Row	customer_id	days_to_deliver
1	1bccb206de9f0f25adc6871a1...	30
2	de4caa97afa80c8eeac2ff4c8d...	30
3	70fc57eeae292675927697fe0...	35
4	7a34a8e890765ad6f90db76d0...	30
5	065053860347d845788e041c...	32
6	0378e1381c730d4504ebc07d2...	29
7	d33e520a99eb4cfc0d3ef2b6ff...	43
8	a0bc11375dd3dbdd0e0bfcbb...	40
9	8fe0db7abbcac2d788689e91...	37
10	22c0028cdec95ad1808c1fd50...	33
11	dca924c5e55e17bdba2ad42ae...	38
12	1c7a9b908094192a2dfae2819...	36
13	a1fa003a1a17fc47164251e0e...	34
14	f5c36ac199073a62861ebda86...	42

[Load more](#)

Results per page: 50 1 - 50 of 9944 [K](#) [Go to Settings to activate Windows](#) [REFRESH](#)

PERSONAL HISTORY PROJECT HISTORY

## B) Find time\_to\_delivery & diff\_estimated\_delivery. Formula for the same given below:

```
SELECT
customer_id,
DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp, day) AS expected_days_to_deliver
FROM `Target_Data.orders`
```

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

DISMISS [ACTIVATE](#)

Google Cloud My Project 25578 SQL Search (/) for resources, docs, products, and more [Search](#)

SANDBOX Set up billing to upgrade to the full BigQuery experience. [Learn more](#) DISMISS [UPGRADE](#)

Explorer + ADD [K](#)

Query results [SAVE RESULTS](#) [EXPLORE DATA](#) [X](#)

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH [PREVIEW](#) [Expand](#)

Row	customer_id	expected_days_to_deliver
1	b50a0774cd941fa6d114ea6f8...	50
2	53e76dd2ac2339c712daa2fe7...	6
3	9cfff8d557e02418fe939f23fafa...	44
4	285195a5b585842e25bd1ef90...	54
5	d7bed5fac093a4136216072ab...	56
6	912f108a7026f25f99240a5c4c...	54
7	76c74aaff2f3f735f546d9818a...	56
8	b296edf5dacc218b6457fddcb...	41
9	3a0a5f064eaf4a5c0e6030043...	3
10	c561230659c12a017bdb3a60...	3
11	0d0d6e3ef1bb0fc138cc89255...	47
12	d3775d8d2c341f2d6d1e52ec4...	44
13	71ac7ff106160c3ef07f1fe76d...	43
14	1e7569b40852c8c0c06f6ec8d...	45

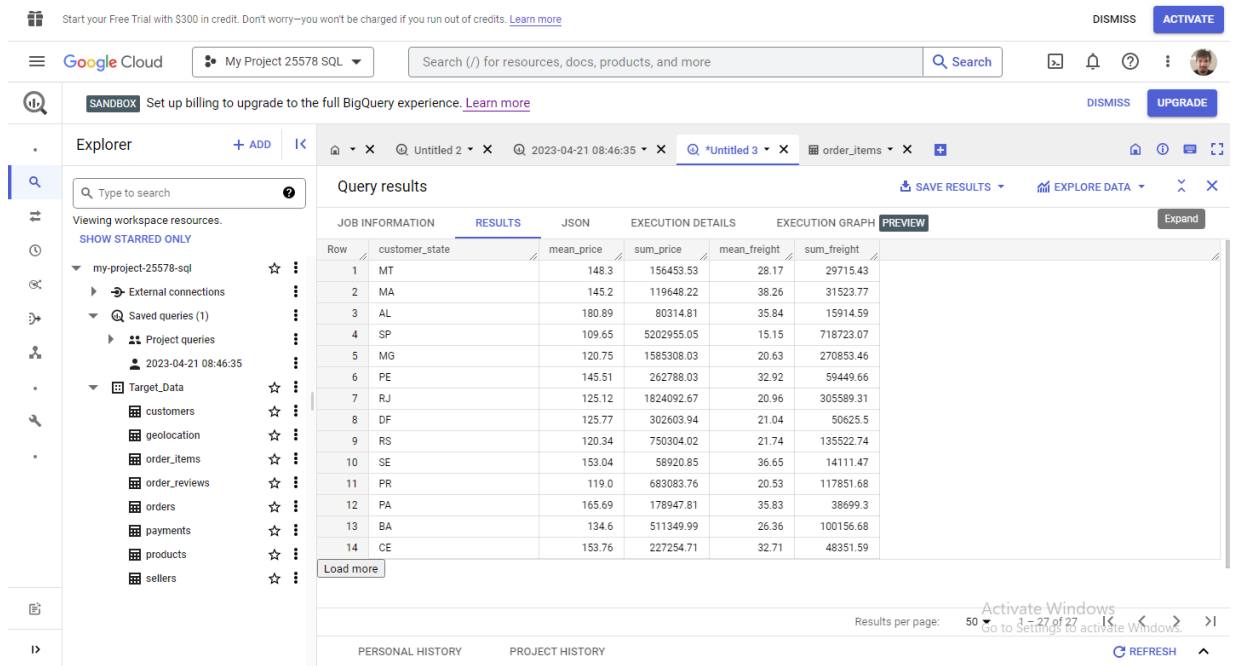
[Load more](#)

Results per page: 50 1 - 50 of 9944 [K](#) [Go to Settings to activate Windows](#) [REFRESH](#)

PERSONAL HISTORY PROJECT HISTORY

C) Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery

```
SELECT c.customer_state,  
  
ROUND(AVG(oi.price), 2) AS mean_price,  
  
ROUND(SUM(oi.price), 2) AS sum_price,  
  
ROUND(AVG(oi.freight_value), 2) AS mean_freight,  
  
ROUND(SUM(oi.freight_value), 2) AS sum_freight  
  
FROM `Target_Data.customers` AS c  
  
JOIN `Target_Data.orders` AS o  
  
ON c.customer_id = o.customer_id  
  
JOIN `Target_Data.order_items` AS oi  
  
ON o.order_id = oi.order_id  
  
GROUP BY c.customer_state
```



The screenshot shows the Google Cloud BigQuery interface. The query results are displayed in a table with the following columns: Row, customer\_state, mean\_price, sum\_price, mean\_freight, and sum\_freight. The results are grouped by customer\_state.

Row	customer_state	mean_price	sum_price	mean_freight	sum_freight
1	MT	148.3	156453.53	28.17	29715.43
2	MA	145.2	119648.22	38.26	31523.77
3	AL	180.89	80314.81	35.84	15914.59
4	SP	109.65	5202955.05	15.15	718723.07
5	MG	120.75	1585308.03	20.63	270853.46
6	PE	145.51	262788.03	32.92	59449.66
7	RJ	125.12	1824092.67	20.96	305589.31
8	DF	125.77	302603.94	21.04	50625.5
9	RS	120.34	750304.02	21.74	135522.74
10	SE	153.04	58920.85	36.65	14111.47
11	PR	119.0	683083.76	20.53	117851.68
12	PA	165.69	178947.81	35.83	38699.3
13	BA	134.6	511349.99	26.36	100156.68
14	CE	153.76	227254.71	32.71	48351.59

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

```
SELECT  
AVG(freight_value),  
c.customer_state,  
o.order_id  
FROM  
`Target_Data.orders` AS o join `Target_Data.order_items` AS oi  
ON o.order_id=oi.order_id join `Target_Data.customers` AS c  
ON o.customer_id = c.customer_id  
GROUP BY  
customer_state,  
order_id  
ORDER BY  
AVG(freight_value) DESC  
LIMIT 5
```

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

Google Cloud My Project 25578 SQL Search (/) for resources, docs, products, and more

SANDBOX Set up billing to upgrade to the full BigQuery experience. [Learn more](#)

Explorer + ADD

Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	f0_	customer_state	order_id
1	409.68	PI	a77e1550db865202c56b19dd...
2	375.28	SC	3fde74c28a3d5d618c00f26d5...
3	375.28	PR	076d1555fb53a89b0ef4d529e...
4	339.59	SP	9f49bd16053df810384e79338...
5	338.3	MT	264a7e199467906c0727394df...

Activate Windows  
Go to Settings to activate Windows.

PERSONAL HISTORY PROJECT HISTORY REFRESH

```

SELECT
AVG(freight_value),
c.customer_state,
o.order_id
FROM
`Target_Data.orders` AS o join `Target_Data.order_items` AS oi
ON o.order_id=oi.order_id join `Target_Data.customers` AS c
ON o.customer_id = c.customer_id
GROUP BY
customer_state,
order_id
ORDER BY
AVG(freight_value)
LIMIT 5

```

SANDBOX Set up billing to upgrade to the full BigQuery experience. [Learn more](#)

Explorer + ADD

Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	f0_	MONTH	state
1	1048	11	RJ
2	283	12	RS
3	2357	12	SP
4	196	2	DF
5	378	11	PR
6	92	4	MT
7	79	7	MA
8	40	7	AL
9	4381	7	SP
10	85	7	MT
11	1111	7	MG
12	1190	5	MG
13	4632	5	SP
14	174	5	PE
15	1908	10	SP
16	990	1	RJ

Results per page: 50 1-50 of 322

PERSONAL HISTORY PROJECT HISTORY REFRESH

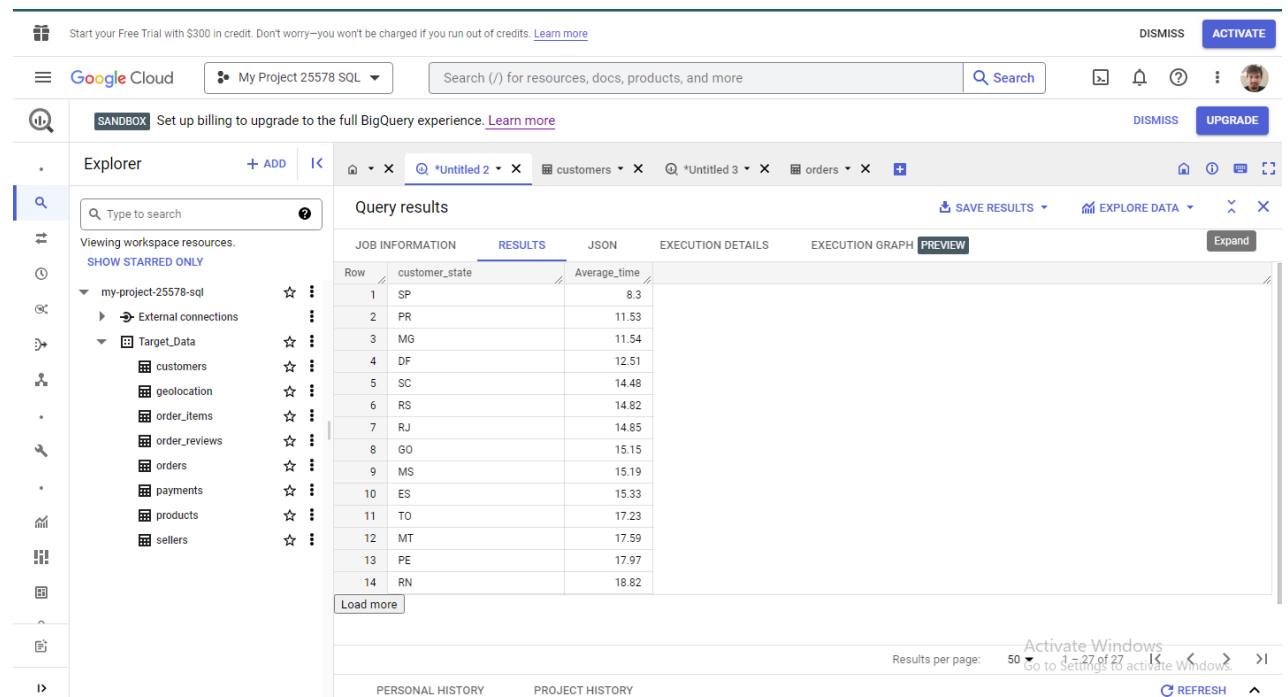
E) Top 5 states with highest/lowest average time to delivery



```

SELECT
customer_state,
ROUND(AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day)),2) Avera
ge_time
FROM
`Target_Data.orders` AS o join `Target_Data.customers` AS c
ON o.customer_id = c.customer_id
GROUP BY
customer_state
ORDER BY
AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day))

```



The screenshot shows the Google Cloud BigQuery interface. On the left is the Explorer pane showing the project structure. The main area displays the 'Query results' for a query named 'Untitled 2'. The results are shown in a table with two columns: 'customer\_state' and 'Average\_time'. The table contains 14 rows of data, sorted by 'Average\_time' in ascending order. At the bottom, there are tabs for 'PERSONAL HISTORY' and 'PROJECT HISTORY', and a 'REFRESH' button.

Row	customer_state	Average_time
1	SP	8.3
2	PR	11.53
3	MG	11.54
4	DF	12.51
5	SC	14.48
6	RS	14.82
7	RJ	14.85
8	GO	15.15
9	MS	15.19
10	ES	15.33
11	TO	17.23
12	MT	17.59
13	PE	17.97
14	RN	18.82

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

```

WITH state_avg_estimated_delivery AS (
SELECT
c.customer_state,
ROUND(AVG(DATE_DIFF(o.order_estimated_delivery_date, o.order_delivered_
customer_date,DAY)), 2) AS mean_diff_estimated_delivery
FROM `Target_Data.orders` AS o
JOIN `Target_Data.customers` AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
HAVING COUNT(*) > 1000 -
- only considering states with more than 1000 orders
)

```

```

(SELECT

"Top 5 States where delivery is really fast" AS title,

customer_state,

mean_diff_estimated_delivery

FROM state_avg_estimated_delivery

ORDER BY mean_diff_estimated_delivery ASC

LIMIT 5)

UNION ALL

(SELECT

"Top 5 States where delivery is not so fast" AS title,

customer_state,

mean_diff_estimated_delivery

FROM state_avg_estimated_delivery

ORDER BY mean_diff_estimated_delivery DESC

LIMIT 5)

```

## 6.Payment type analysis:

### A) Month over Month count of orders for different payment types

```

SELECT COUNT(o.order_id) AS order_count, p.payment_type,
EXTRACT (MONTH FROM order_purchase_timestamp) AS month_on_month
FROM `Target_Data.orders` AS o join `Target_Data.payments` AS p ON o.order_id = p.order_id
GROUP BY payment_type, month_on_month
ORDER BY COUNT(o.order_id)

```

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#) DISMISS ACTIVATE

Google Cloud My Project 25578 SQL Search (/) for resources, docs, products, and more Q Search DISMISS UPGRADE

**Explorer** + ADD Q \*Untitled 2 orders +

Q Type to search

Viewing workspace resources. [SHOW STARRED ONLY](#)

- my-project-25578-sql
  - External connections
  - Target\_Data
    - customers
    - geolocation
    - order\_items
    - order\_reviews
    - orders
    - payments
    - products
    - sellers

**Query results** SAVE RESULTS EXPLORE DATA PREVIEW

Row	order_count	payment_type	month_on_mon
1	1	not_defined	9
2	2	not_defined	8
3	43	debit_card	9
4	54	debit_card	10
5	64	debit_card	12
6	70	debit_card	11
7	81	debit_card	5
8	82	debit_card	2
9	109	debit_card	3
10	118	debit_card	1
11	124	debit_card	4
12	209	debit_card	6
13	264	debit_card	7
14	294	voucher	12
15	302	voucher	9
16	311	debit_card	8

Results per page: 50 1 - 50 of 50 REFRESH

PERSONAL HISTORY PROJECT HISTORY

B) Count of orders based on the no. of payment installments

```
SELECT
COUNT(o.order_id) AS no_of_orders,
p.payment_installments
FROM `Target_Data.orders` AS o join `Target_Data.payments` AS p
ON o.order_id = p.order_id
GROUP BY
p.payment_installments
ORDER BY
COUNT(o.order_id)
```

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#) DISMISS ACTIVATE

Google Cloud My Project 25578 SQL Search (/) for resources, docs, products, and more Q Search DISMISS UPGRADE

**Explorer** + ADD Q \*Untitled 2 payments Q \*Untitled 3 +

Q Type to search

Viewing workspace resources. [SHOW STARRED ONLY](#)

- my-project-25578-sql
  - External connections
  - Target\_Data
    - customers
    - geolocation
    - order\_items
    - order\_reviews
    - orders
    - payments
    - products
    - sellers

**Query results** SAVE RESULTS EXPLORE DATA PREVIEW

Row	no_of_orders	payment_installments
1	1	23
2	1	22
3	2	0
4	3	21
5	5	16
6	8	17
7	15	14
8	16	13
9	17	20
10	18	24
11	23	11
12	27	18
13	74	15
14	133	12

Load more

Results per page: 50 1 - 24 of 24 REFRESH

PERSONAL HISTORY PROJECT HISTORY