

**PROJECT:**

**Business Case: Target  
SQL**

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

--1)Data type of columns in a table

```
SELECT column_name, data_type
FROM shopping.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'customers';
```

The screenshot displays the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, the current project 'Target SQL Project', and a search bar. Below the navigation bar, a sidebar on the left shows the 'Explorer' view with a tree structure of resources. The main area is divided into two panes: the top pane shows the SQL query editor with a query to check the data types of columns in the 'customers' table, and the bottom pane shows the 'Query results' table.

**Query:**

```
# 1.Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset
--1)Data type of columns in a table
SELECT column_name, data_type
FROM shopping.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'customers';
```

**Query results:**

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)Time period for which the data is given

```
SELECT  
MIN(order_purchase_timestamp) AS first_order, MAX(order_purchase_timestamp) AS last_order  
FROM `shopping.orders` ;
```

The screenshot displays the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, the current project 'Target SQL Project', and a search bar. Below this, a 'SANDBOX' banner prompts the user to set up billing. The left sidebar, titled 'Explorer', shows a tree view of workspace resources, including the 'target-sql-project-382307' and its various datasets like 'customers', 'geolocation', and 'orders'. The main editor area contains a SQL query with line numbers 9 through 22. The query is split into two parts: the first part (lines 10-13) calculates the first and last order timestamps from the 'shopping.orders' table, and the second part (lines 18-20) counts the number of distinct cities and states from the 'shopping.geolocation' table. Below the query editor, the 'Query results' section is visible, showing a table with two columns: 'first\_order' and 'last\_order'. The first row of data shows the first order on 2016-09-04 and the last order on 2018-10-17. The bottom of the console features tabs for 'PERSONAL HISTORY' and 'PROJECT HISTORY', along with a 'REFRESH' button.

Business Case: Target SQL Intro - X Query results - BigQuery - Target SQL - Prob: X +

console.cloud.google.com/bigquery?\_ga=2.18835833.1366719014.1679202918-1191789870.1679202918&project=target-sql-project-382307&supportedpurview=project&ws=11m51m...

Google Cloud Target SQL Project 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

Viewing workspace resources. SHOW STARRED ONLY

target-sql-project-382307

External connections

Saved queries (1)

Project queries

target sql

shopping

customers

geolocation

order items

order reviews

orders

payments

products

sellers

9 --2)Time period for which the data is given  
10  
11 SELECT  
12 MIN(order\_purchase\_timestamp) AS first\_order, MAX(order\_purchase\_timestamp) AS last\_order  
13 FROM `shopping.orders` ;  
14  
15  
16 --3) Cities and States of customers ordered during the given period  
17  
18 SELECT COUNT(DISTINCT(geolocation\_city)) as city\_count,  
19 COUNT(DISTINCT(geolocation\_state)) as state\_count  
20 FROM `shopping.geolocation`;  
21  
22 # 2.In-depth Exploration:

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

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

PERSONAL HISTORY PROJECT HISTORY REFRESH

--3) Cities and States of customers ordered during the given period

```
SELECT COUNT(DISTINCT(geolocation_city)) as city_count,  
COUNT(DISTINCT(geolocation_state)) as state_count  
FROM `shopping.geolocation`;
```

The screenshot displays the Google Cloud BigQuery console interface. The top navigation bar shows the Google Cloud logo, the project name 'Target SQL Project', and a search bar. Below this, the 'Sandbox' section indicates the user is in a sandbox environment. The left sidebar contains an 'Explorer' panel with a search bar and a list of workspace resources, including 'target-sql-project-382307' and 'shopping'. The main editor area shows a SQL query with line numbers 9 through 22. The query is:   
9 -- 2) Time period for which the data is given  
10  
11 SELECT  
12 MIN(order\_purchase\_timestamp) AS first\_order, MAX(order\_purchase\_timestamp) AS last\_order  
13 FROM `shopping.orders`;  
14  
15  
16 -- 3) Cities and States of customers ordered during the given period  
17  
18 SELECT COUNT(DISTINCT(geolocation\_city)) as city\_count,  
19 COUNT(DISTINCT(geolocation\_state)) as state\_count  
20 FROM `shopping.geolocation`;  
21  
22 # 2. In-depth Exploration:  
The 'Query results' section at the bottom shows a table with two columns: 'city\_count' and 'state\_count'. The first row of data shows 8011 for city\_count and 27 for state\_count. The interface also includes buttons for 'RUN', 'SAVE', 'SHARE', 'SCHEDULE', and 'MORE' at the top of the editor, and a 'REFRESH' button at the bottom right.

```
9 -- 2) Time period for which the data is given  
10  
11 SELECT  
12 MIN(order_purchase_timestamp) AS first_order, MAX(order_purchase_timestamp) AS last_order  
13 FROM `shopping.orders`;  
14  
15  
16 -- 3) Cities and States of customers ordered during the given period  
17  
18 SELECT COUNT(DISTINCT(geolocation_city)) as city_count,  
19 COUNT(DISTINCT(geolocation_state)) as state_count  
20 FROM `shopping.geolocation`;  
21  
22 # 2. In-depth Exploration:
```

Row	city_count	state_count
1	8011	27

## # 2.In-depth Exploration:

-- 1) 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?

--METHOD 1:

```
SELECT EXTRACT(year from order_purchase_timestamp) as year,  
EXTRACT(month from order_purchase_timestamp) as month,  
COUNT(1) AS num_orders  
FROM `shopping.orders`  
GROUP BY year,month  
ORDER BY year, month;
```

--METHOD 2:

```
SELECT DATE(order_purchase_timestamp) as year,  
DATE(order_purchase_timestamp) as month,  
COUNT(1) AS num_orders  
FROM `shopping.orders`  
GROUP BY year,month  
ORDER BY year, month;
```

The screenshot displays the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, the current project 'Target SQL Project', and a search bar. Below the navigation bar, the 'Explorer' panel on the left shows the project hierarchy, including 'target-sql-project-382307' and its sub-projects like 'shopping', 'customers', 'geolocation', etc. The main editor area shows a SQL query with two methods for extracting year and month from the 'order\_purchase\_timestamp' column. The 'Query results' panel at the bottom displays the output of the query, showing a table with columns 'year', 'month', and 'num\_orders'. The results are paginated, showing rows 1 through 11.

Query results

Row	year	month	num_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
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331

```

-
- 2)What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Nigh?
SELECT
COUNT(DISTINCT order_id) as Order_count,
Day_part
FROM
(SELECT
order_id,
CASE
WHEN EXTRACT(hour FROM order_purchase_timestamp) >= 5 AND EXTRACT (hour
FROM order_purchase_timestamp) <= 7 THEN 'Dawn'
WHEN EXTRACT(hour FROM order_purchase_timestamp) >= 7 AND EXTRACT (hour
FROM order_purchase_timestamp) <= 12 THEN 'Morning'
WHEN EXTRACT(hour FROM order_purchase_timestamp) >= 12 AND EXTRACT (hour
FROM order_purchase_timestamp) <= 17 THEN 'Afternoon'
ELSE 'Night'
END as Day_Part
FROM `shopping.orders`) Y
GROUP BY Day_Part
ORDER BY COUNT(DISTINCT order_id);

```

Business Case: Target SQL Intro x Query results – BigQuery – Target x Business Case: Target SQL - Prob x +

console.cloud.google.com/bigquery?\_ga=2.18835833.1366719014.1679202918-1191789870.1679202918&project=target-sql-project-382307&supportedpurview=project&ws=11m511m...

(1) Kalpesh Patil | Li... AZ-900 Exam – Fre... My Essex for Applic... Home | Scaler Acad... India Young Profess... SQL workspace – Bi... SQL-Library - Goog...

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

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

Explorer + ADD IK

Type to search

Viewing workspace resources. SHOW STARRED ONLY

- target-sql-project-382307
  - External connections
  - Saved queries (1)
    - Project queries
    - target sql
  - shopping
    - customers
    - geolocation
    - order items
    - order reviews
    - orders
    - payments
    - products
    - sellers

Query editor: targetsql

```

37 GROUP BY year, month
38 ORDER BY year, month;
39
40 -- 2)What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?
41 SELECT
42 COUNT(DISTINCT order_id) as Order_count,
43 Day_part
44 FROM
45 (SELECT
46 order_id,
47 CASE
48 WHEN EXTRACT(hour FROM order_purchase_timestamp) >= 5 AND EXTRACT (hour
49 FROM order_purchase_timestamp) <= 7 THEN 'Dawn'
50 WHEN EXTRACT(hour FROM order_purchase_timestamp) >= 7 AND EXTRACT (hour
51 FROM order_purchase_timestamp) <= 12 THEN 'Morning'
52 WHEN EXTRACT(hour FROM order_purchase_timestamp) >= 12 AND EXTRACT (hour
53 FROM order_purchase_timestamp) <= 17 THEN 'Afternoon'

```

Query results

SAVE RESULTS EXPLORE DATA

Row	Order_count	Day_part
1	1921	Dawn
2	26502	Morning
3	32366	Afternoon
4	38652	Night

PERSONAL HISTORY PROJECT HISTORY REFRESH

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

-- 1) Get month on month orders by states

```
SELECT
EXTRACT(month from order_purchase_timestamp) month,
count(*) total_orders
FROM `shopping.orders`
GROUP BY month
ORDER BY month;
```

The screenshot shows the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, the current project 'Target SQL Project', and a search bar. Below the navigation bar, there's a 'SANDBOX' banner with a 'DISMISS' button and an 'UPGRADE' button. The main interface is divided into three sections: Explorer, Editor, and Query results.

**Explorer:** Displays a tree view of the workspace resources. The 'target-sql-project-382307' is expanded, showing 'External connections', 'Saved queries (1)', 'Project queries', and 'target-sql'. Under 'target-sql', the 'shopping' dataset is expanded, listing tables: customers, geolocation, order items, order reviews, orders, payments, products, and sellers.

**Editor:** Contains the SQL query being executed. The query is:

```

61 # 3. evolution of e-commerce orders in the Brazil region:
62
63 -- 1) Get month on month orders by states
64
65 SELECT
66 EXTRACT(month from order_purchase_timestamp) month,
67 count(*) total_orders
68 FROM `shopping.orders`
69 GROUP BY month
70 ORDER BY month;

```

The query is highlighted in blue. Above the editor, there are buttons for 'RUN', 'SAVE', 'SHARE', 'SCHEDULE', and 'MORE'. A status bar at the top right of the editor says 'Query completed.'

**Query results:** Displays the results of the query in a table format. The table has two columns: 'month' and 'total\_orders'. The results are as follows:

Row	month	total_orders
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305

Below the table, there are tabs for 'JOB INFORMATION', 'RESULTS', 'JSON', 'EXECUTION DETAILS', 'EXECUTION GRAPH', and 'PREVIEW'. The 'RESULTS' tab is selected. At the bottom right, it says 'Results per page: 50' and '1 - 12 of 12'. There are also buttons for 'SAVE RESULTS', 'EXPLORE DATA', and 'REFRESH'.



-- 2) Distribution of customers across the states in Brazil

```
SELECT customer_state,  
COUNT(*) AS customers  
FROM `shopping.customers`  
GROUP BY customer_state  
ORDER BY customer_state;
```

The screenshot displays the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, the current project 'Target SQL Project', and a search bar. Below this, the 'Sandbox' section indicates the user is using a limited version of BigQuery. The left sidebar shows the 'Explorer' view with a tree of workspace resources, including 'target-sql-project-382307' and its sub-resources like 'customers', 'geolocation', 'order items', 'order reviews', 'orders', 'payments', 'products', and 'sellers'. The main editor area shows a SQL query being executed. The query results are displayed in a table format, showing the distribution of customers across different states in Brazil. The table has two columns: 'customer\_state' and 'customers'. The results are ordered by 'customer\_state'.

Query results

Row	customer_state	customers
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747
11	MG	11635

Results per page: 50 1 - 27 of 27



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

--

1) 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

```
SELECT *,
((total_payment - lag(total_payment) over(order by year, month))/(lag(total_payment)
)over(order by year,month)))*100 as Raise
FROM
(SELECT
EXTRACT(year from order_purchase_timestamp)as year,
EXTRACT(month from order_purchase_timestamp)as month,
round(sum(payment_value),2) as total_payment
FROM `shopping.orders` as o join `shopping.payments` as p on o.order_id = p.order_id
GROUP BY year, month
ORDER BY year, month )t1
WHERE year in (2017,2018) and month not in (9,10,11,12)
ORDER BY year, month;
```

The screenshot shows the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, the project name 'Target SQL Project', and a search bar. Below the navigation bar, there's a 'Sandbox' section with a 'Set up billing to upgrade to the full BigQuery experience' link. The main area is divided into three panes: Explorer, Editor, and Query results.

**Explorer:** Shows a tree view of workspace resources. Under 'target-sql-project-382307', there are 'External connections', 'Saved queries (1)', 'Project queries', and a folder named 'shopping'. The 'shopping' folder is expanded, showing sub-folders like 'customers', 'geolocation', 'order items', 'order reviews', 'orders', 'payments', 'products', and 'sellers'.

**Editor:** Contains the SQL query being executed. The query is: 

```
SELECT *,
((total_payment - lag(total_payment) over(order by year, month))/(lag(total_payment)over(order by year,month)))*100 as Raise
FROM
(SELECT
EXTRACT(year from order_purchase_timestamp)as year,
EXTRACT(month from order_purchase_timestamp)as month,
round(sum(payment_value),2) as total_payment
FROM `shopping.orders` as o join `shopping.payments` as p on o.order_id = p.order_id
GROUP BY year, month
ORDER BY year, month )t1
WHERE year in (2017,2018) and month not in (9,10,11,12)
ORDER BY year, month;
```

**Query results:** Shows the results of the query in a table format. The table has columns: Row, year, month, total\_payment, and Raise. The results are as follows:

Row	year	month	total_payment	Raise
1	2017	1	138488.04	null
2	2017	2	291908.01	110.782107...
3	2017	3	449863.6	54.1114270...
4	2017	4	417788.03	-7.13006564...
5	2017	5	592918.82	41.9185753...
6	2017	6	511276.38	-13.7695814...
7	2017	7	592382.92	15.8635413...
8	2017	8	674396.32	13.8446598...
9	2018	1	1115004.18	65.3336690...

At the bottom of the results pane, there's a 'Results per page' dropdown set to 50, and a '1 - 16 of 16' indicator. There are also navigation buttons for 'PERSONAL HISTORY', 'PROJECT HISTORY', and a 'REFRESH' button.

--2) Mean & Sum of price and freight value by customer state

```
SELECT s.seller_state,  
AVG(price) AS mean_price,  
SUM(price) AS total_price,  
AVG(freight_value) AS mean_freight,  
SUM(freight_value) AS total_freight  
FROM `shopping.order_items` AS o  
JOIN `shopping.sellers` AS s  
ON o.seller_id = s.seller_id  
GROUP BY s.seller_state;
```

The screenshot displays the Google Cloud BigQuery console interface. At the top, there's a navigation bar with the Google Cloud logo and a search bar. Below this, a sidebar on the left shows the 'Explorer' view with a tree structure of workspace resources, including 'target-sql-project-382307' and its sub-items like 'customers', 'geolocation', 'order\_items', 'order\_reviews', 'orders', 'payments', 'products', and 'sellers'. The main area is divided into two sections: the top section shows the SQL query being executed, and the bottom section displays the 'Query results' table. The query is a SELECT statement that calculates the mean and sum of price and freight values for each seller state. The results table has 9 rows, one for each state (SP, MG, PR, SC, RS, DF, ES, RJ, GO), and 6 columns: 'seller\_state', 'mean\_price', 'total\_price', 'mean\_freight', and 'total\_freight'. The bottom of the console shows 'PERSONAL HISTORY' and 'PROJECT HISTORY' tabs, along with a 'REFRESH' button.

Query results

Row	seller_state	mean_price	total_price	mean_freight	total_freight
1	SP	108.951684...	8753396.21...	18.4522126...	1482487.66...
2	MG	114.598928...	1011564.74...	24.0846335...	212595.060...
3	PR	145.529605...	1261887.20...	22.7209687...	197013.520...
4	SC	155.196581...	632426.070...	26.1465177...	106547.060...
5	RS	172.150768...	378559.540...	26.0314188...	57243.0899...
6	DF	108.731345...	97749.4799...	20.5718121...	18494.0600...
7	ES	128.197876...	47689.6100...	32.7180913...	12171.1300...
8	RJ	175.173146...	843984.220...	19.4748650...	93829.8999...
9	GO	127.690788...	66399.2100...	24.1644230...	12565.4999...

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

--1) Calculate days between purchasing, delivering and estimated delivery

```
SELECT
  order_id,
  DATE_DIFF(order_delivered_carrier_date,order_purchase_timestamp,DAY) AS days_between_purchase,
  DATE_DIFF(order_estimated_delivery_date,order_delivered_carrier_date,DAY) AS days_between_delivering,
  DATE_DIFF(order_estimated_delivery_date,order_purchase_timestamp,DAY) AS days_between_estimate,
FROM `shopping.orders` ;
```

The screenshot displays the Google Cloud BigQuery console interface. The top navigation bar shows the 'Target SQL Project' and a search bar. The left sidebar contains an 'Explorer' panel with a tree view of the project structure, including 'customers', 'geolocation', 'order items', 'order reviews', 'orders', 'payments', 'products', and 'sellers'. The main area shows a SQL query editor with the following code:

```
--1) Calculate days between purchasing, delivering and estimated delivery
SELECT
  order_id,
  DATE_DIFF(order_delivered_carrier_date,order_purchase_timestamp,DAY) AS days_between_purchase,
  DATE_DIFF(order_estimated_delivery_date,order_delivered_carrier_date,DAY) AS days_between_delivering,
  DATE_DIFF(order_estimated_delivery_date,order_purchase_timestamp,DAY) AS days_between_estimate,
FROM `shopping.orders` ;
```

Below the query editor, the 'Query results' section is visible, showing a table with 9 rows and 5 columns: 'order\_id', 'days\_between\_p', 'days\_between\_d', 'days\_between\_e', and an unlabeled column. The results are as follows:

Row	order_id	days_between_p	days_between_d	days_between_e	
1	f88aac7ebccb37f19725a0753...	9	41	50	
2	790cd37689193dca0d00d2feb...	2	3	6	
3	49db7943d60b6805c3a41f547...	6	37	44	
4	063b573b88fc90e516aba87df...	22	31	54	
5	a68ce1686d536ca72bd2dad4...	33	23	56	
6	45973912e490866800c0aea8f...	18	35	54	
7	cda873529ca7ab71f677d5ec1...	39	16	56	
8	ead20687129da8f5d89d831bb...	1	40	41	
9	6f028ccb7d612af251aa442a1f...	1	2	3	

The bottom of the console shows 'Results per page: 50' and '1 - 50 of 99441' rows. There are also buttons for 'SAVE RESULTS', 'EXPLORE DATA', and 'REFRESH'.

```
--
2) Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:
--time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
--diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

select
date_diff(order_delivered_customer_date,order_purchase_timestamp,DAY) as time_to_delivery ,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,DAY) as diff_estimated_delivery
from
`shopping.orders`
as e1 join
`shopping.order_items`
as e2 on e1.order_id=e2.order_id
join `shopping.customers`
as c on e1.customer_id = c.customer_id;
```

The screenshot shows the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, the project name 'Target SQL Project', and a search bar. Below the navigation bar, there's a sidebar with an 'Explorer' view showing the project structure, including tables like 'customers', 'geolocation', 'order\_items', 'order\_reviews', 'orders', 'payments', 'products', and 'sellers'. The main area displays a SQL query in the editor, which is the same query provided in the text block above. The query has been executed, and the results are shown in a table below the editor. The table has three columns: 'time\_to\_delivery' and 'diff\_estimated\_delivery'. The results show 10 rows of data.

**Query results**

Row	time_to_delivery	diff_estimated_delivery
1	30	-12
2	30	28
3	35	16
4	30	1
5	32	0
6	32	0
7	29	1
8	43	-4
9	40	-4
10	37	-1

Results per page: 50 | 1 - 50 of 112650 | REFRESH

--

3) Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimate  
d\_delivery

```
select c.customer_state, count(e1.order_id) as count_orders,  
avg(freight_value) as avg_freight,  
avg(date_diff(order_purchase_timestamp,order_delivered_customer_date,DAY)) as time_  
to_delivery ,  
avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,DAY)) as  
diff_estimated_delivery  
from  
`shopping.orders`  
as e1 join  
`shopping.order_items`  
as e2 on e1.order_id=e2.order_id  
join `shopping.customers`  
as c on e1.customer_id = c.customer_id  
where order_delivered_customer_date is not null  
group by c.customer_state ;
```

The screenshot displays the Google Cloud BigQuery console interface. At the top, there's a navigation bar with the Google Cloud logo and a search bar. Below this, the 'Explorer' panel on the left shows a tree view of workspace resources, including 'target-sql-project-382307' and its sub-entities like 'customers', 'geolocation', 'order\_items', 'order\_reviews', 'orders', 'payments', 'products', and 'sellers'. The main area is divided into two sections: the top section shows the SQL query being executed, and the bottom section displays the 'Query results' table. The query is a SELECT statement grouping data by customer state and calculating various metrics. The results table has 11 rows, each representing a different state (RJ, MG, SC, SP, GO, RS, BA, MT, SE, PE, TO) with columns for customer\_state, count\_orders, avg\_freight, time\_to\_delivery, and diff\_estimated\_delivery. The bottom of the console shows a 'PERSONAL HISTORY' and 'PROJECT HISTORY' section with a 'REFRESH' button.

Row	customer_state	count_orders	avg_freight	time_to_delivery	diff_estimated_delivery
1	RJ	14146	20.9097843...	-14.6893821...	11.1444931...
2	MG	12917	20.6258372...	-11.5155221...	12.3971510...
3	SC	4098	21.5066276...	-14.5209858...	10.6688628...
4	SP	46443	15.1149940...	-8.25960855...	10.2655943...
5	GO	2277	22.5628678...	-14.9481774...	11.3728590...
6	RS	6133	21.6142703...	-14.7082993...	13.2030001...
7	BA	3683	26.4875563...	-18.7746402...	10.1194678...
8	MT	1037	27.9969141...	-17.5081967...	13.6393442...
9	SE	375	36.5731733...	-20.9786666...	9.16533333...
10	PE	1746	32.6933333...	-17.7920962...	12.5521191...
11	TO	310	37.4350322...	-17.0032258...	11.4612903...



# Sort the data to get the following:

--

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

--HIGHEST AVG FREIGHT VALUE

```
SELECT b.seller_state,  
AVG(freight_value) AS highest_avg_freight_value  
FROM `shopping.order_items` AS a  
JOIN `shopping.sellers` AS b ON a.seller_id = b.seller_id  
GROUP BY b.seller_state  
ORDER BY highest_avg_freight_value DESC  
LIMIT 5;
```

The screenshot shows the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, the project name 'Target SQL Project', and a search bar. The main area is divided into three sections: Explorer, Editor, and Query results.

**Explorer:** Displays a tree view of the project's resources, including 'target-sql-project-382307', 'External connections', 'Saved queries (1)', 'Project queries', and 'target sql'. The 'shopping' dataset is expanded, showing tables like 'customers', 'geolocation', 'order\_items', 'order\_reviews', 'orders', 'payments', 'products', and 'sellers'.

**Editor:** Contains the SQL query being executed. The query is as follows:

```
-- Sort the data to get the following:  
--5.Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5  
--HIGHEST AVG FREIGHT VALUE  
SELECT b.seller_state,  
AVG(freight_value) AS highest_avg_freight_value  
FROM `shopping.order_items` AS a  
JOIN `shopping.sellers` AS b ON a.seller_id = b.seller_id  
GROUP BY b.seller_state  
ORDER BY highest_avg_freight_value DESC  
LIMIT 5;  
-- LOWEST AVG FREIGHT VALUE  
SELECT b.seller_state,
```

**Query results:** Displays the results of the query in a table format. The table has two columns: 'seller\_state' and 'highest\_avg\_freight\_value'. The results are as follows:

Row	seller_state	highest_avg_freight_value
1	RO	50.9128571...
2	CE	46.3811702...
3	PB	39.1881578...
4	PI	36.9433333...
5	AC	32.84

The bottom of the console shows tabs for 'PERSONAL HISTORY' and 'PROJECT HISTORY', along with a 'REFRESH' button.

-- LOWEST AVG FREIGHT VALUE

```
SELECT b.seller_state,  
AVG(freight_value) AS highest_avg_freight_value  
FROM `shopping.order_items` AS a  
JOIN `shopping.sellers` AS b ON a.seller_id = b.seller_id  
GROUP BY b.seller_state  
ORDER BY highest_avg_freight_value ASC  
LIMIT 5;
```

The screenshot shows the Google Cloud BigQuery console interface. The Explorer panel on the left lists the project 'target-sql-project-382307' and its datasets: 'customers', 'geolocation', 'order items', 'order reviews', 'orders', 'payments', 'products', and 'sellers'. The main editor displays a SQL query that filters for the lowest average freight value and limits the results to 5 rows. The 'Query results' panel shows the following data:

Row	customer_state	highest_avg_freight
1	SP	18.4522126...
2	PA	19.3887499...
3	RJ	19.4748650...
4	DF	20.5718131...
5	PR	22.7209687...

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

--HIGHEST TIME TO DELIVERY

```
SELECT b.customer_state,
ROUND(AVG(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)),2)
as highest_avg_delivery
FROM `shopping.orders` AS a
JOIN `shopping.customers` AS b
ON b.customer_id = a.customer_id
GROUP BY b.customer_state
ORDER BY highest_avg_delivery desc
LIMIT 5;
```

The screenshot shows the Google Cloud BigQuery console interface. The Explorer panel on the left lists the project 'target-sql-project-382307' and its datasets: 'customers', 'geolocation', 'order items', 'order reviews', 'orders', 'payments', 'products', and 'sellers'. The main editor displays a SQL query that calculates the average time to delivery for each customer state and limits the results to 5 rows. The 'Query results' panel shows the following data:

Row	customer_state	highest_avg_delivery
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32



-- LOWEST TIME TO DELIVERY

```
SELECT b.customer_state,  
ROUND(AVG(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)),2)  
as lowest_avg_delivery  
FROM `shopping.orders` AS a  
JOIN `shopping.customers` AS b  
ON b.customer_id = a.customer_id  
GROUP BY b.customer_state  
ORDER BY lowest_avg_delivery ASC  
LIMIT 5;
```

The screenshot shows the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, the current project 'Target SQL Project', and a search bar. Below the navigation bar, the 'Explorer' panel on the left shows the project hierarchy, including 'target-sql-project-382307' and its sub-projects 'customers', 'geolocation', 'order items', 'order reviews', 'orders', 'payments', 'products', and 'sellers'. The main editor area displays a SQL query to find the top 5 states with the lowest average delivery time. The query is as follows:

```
-- LOWEST TIME TO DELIVERY  
SELECT b.customer_state,  
ROUND(AVG(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)),2) as lowest_avg_delivery  
FROM `shopping.orders` AS a  
JOIN `shopping.customers` AS b  
ON b.customer_id = a.customer_id  
GROUP BY b.customer_state  
ORDER BY lowest_avg_delivery ASC  
LIMIT 5;
```

Below the query editor, the 'Query results' section shows the execution details and the resulting data. The 'RESULTS' tab is selected, displaying a table with 5 rows of data. The table has two columns: 'customer\_state' and 'lowest\_avg\_delivery'. The data is as follows:

Row	customer_state	lowest_avg_delivery
1	SP	8.3
2	PR	11.53
3	MG	11.54
4	DF	12.51
5	SC	14.48

The bottom of the console shows the 'PERSONAL HISTORY' and 'PROJECT HISTORY' tabs, along with a 'REFRESH' button.

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

--FAST DELIVERY

```
SELECT b.customer_state,  
ROUND(MAX(DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date,DAY  
)),2) as fast_delivery  
FROM `shopping.orders` AS a  
JOIN `shopping.customers` AS b  
ON b.customer_id = a.customer_id  
GROUP BY b.customer_state  
ORDER BY fast_delivery desc  
LIMIT 5;
```

The screenshot shows the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, the project name 'Target SQL Project', and a search bar. Below the navigation bar, the 'Explorer' panel on the left shows the project structure, including 'target-sql-project-382307' and its sub-projects 'shopping', 'customers', 'geolocation', 'order items', 'order reviews', 'orders', 'payments', 'products', and 'sellers'. The main editor area displays a SQL query in a code editor with syntax highlighting. The query is the same as the one provided in the previous blocks. Below the code editor, the 'Query results' section shows a table with 5 rows of data. The table has two columns: 'customer\_state' and 'fast\_delivery'. The results are as follows:

Row	customer_state	fast_delivery
1	RJ	188.0
2	ES	181.0
3	SP	175.0
4	SE	166.0
5	PA	165.0

At the bottom of the console, there are tabs for 'PERSONAL HISTORY' and 'PROJECT HISTORY', and a 'REFRESH' button.

--NOT FAST

```
SELECT b.customer_state,  
ROUND(MIN(DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date,DAY  
)),2) as slow_delivery  
FROM `shopping.orders` AS a  
JOIN `shopping.customers` AS b  
ON b.customer_id = a.customer_id  
GROUP BY b.customer_state  
ORDER BY slow_delivery ASC  
LIMIT 5;
```

The screenshot displays the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, the 'Target SQL Project' dropdown, a search bar, and user profile information. Below the navigation bar, the 'Explorer' sidebar on the left shows the project hierarchy: 'target-sql-project-382307' containing 'External connections', 'Saved queries (1)', 'Project queries', and a 'shopping' dataset with tables like 'customers', 'geolocation', 'order items', 'order reviews', 'orders', 'payments', 'products', and 'sellers'. The main editor area shows a SQL query with line numbers 209 to 221. The query is identical to the one in the previous block. Below the editor, the 'Query results' section is active, showing a table with 5 rows of data. The table has columns 'customer\_state' and 'slow\_delivery'. The results are: SP (-146.0), MA (-139.0), RS (-134.0), RJ (-108.0), and MG (-77.0). The bottom of the console shows 'PERSONAL HISTORY' and 'PROJECT HISTORY' tabs, with a 'REFRESH' button.

Query results

Row	customer_state	slow_delivery
1	SP	-146.0
2	MA	-139.0
3	RS	-134.0
4	RJ	-108.0
5	MG	-77.0

## #6. Payment type analysis:

-- 1) Month over Month count of orders for different payment types

```
SELECT p.payment_type,  
EXTRACT(month from order_purchase_timestamp) month,  
count(*) total_orders  
FROM `shopping.orders` as o  
JOIN `shopping.payments` as p  
ON o.order_id=p.order_id  
GROUP BY month,p.payment_type  
ORDER BY month,p.payment_type;
```

The screenshot displays the Google Cloud BigQuery console interface. At the top, there's a navigation bar with the Google Cloud logo and a search bar. Below this, the 'Explorer' panel on the left shows a tree view of the 'target-sql-project-382307' resources, including 'shopping' and its sub-resources like 'customers', 'geolocation', 'order items', 'order reviews', 'orders', 'payments', 'products', and 'sellers'. The main editor area shows a SQL query being executed. The query is:   
`SELECT p.payment_type,  
EXTRACT(month from order_purchase_timestamp) month,  
count(*) total_orders  
FROM `shopping.orders` as o  
JOIN `shopping.payments` as p  
ON o.order_id=p.order_id  
GROUP BY month,p.payment_type  
ORDER BY month,p.payment_type;`  
The 'Query results' panel at the bottom shows the output of the query. It includes a table with columns 'payment\_type', 'month', and 'total\_orders'. The table contains 12 rows of data. At the bottom right, there are controls for 'Results per page' (set to 50) and a 'REFRESH' button.

Row	payment_type	month	total_orders
1	UPI	11	1509
2	credit_card	12	4378
3	UPI	2	1723
4	credit_card	11	5897
5	voucher	4	572
6	credit_card	7	7841
7	UPI	7	2074
8	credit_card	5	8350
9	credit_card	10	3778
10	credit_card	1	6103
11	credit_card	6	7276
12	credit_card	9	2286

--2) Count of orders based on the no. of payment installments

```
SELECT p.payment_installments,  
count(*) total_orders  
FROM `shopping.orders` as o  
JOIN `shopping.payments` as p  
ON o.order_id=p.order_id  
GROUP BY p.payment_installments  
ORDER BY p.payment_installments;
```

The screenshot shows the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, the 'Target SQL Project' dropdown, a search bar, and utility icons. A 'SANDBOX' banner with a 'DISMISS' button and an 'UPGRADE' button is visible. The left sidebar contains an 'Explorer' panel with a search bar and a tree view of workspace resources, including 'target-sql-project-382307' and its sub-items like 'customers', 'geolocation', 'order items', 'order reviews', 'orders', 'payments', 'products', and 'sellers'. The main editor area displays a SQL query in the 'Editor' tab, with a 'RUN' button and options for 'SAVE', 'SHARE', 'SCHEDULE', and 'MORE'. The query is:   
`SELECT p.payment_installments,  
count(*) total_orders  
FROM `shopping.orders` as o  
JOIN `shopping.payments` as p  
ON o.order_id=p.order_id  
GROUP BY p.payment_installments  
ORDER BY p.payment_installments;`  
The 'Query results' section shows a table with 12 rows and 2 columns: 'payment\_installments' and 'total\_orders'. The table is displayed in a 'PREVIEW' tab. The bottom of the interface shows 'PERSONAL HISTORY' and 'PROJECT HISTORY' tabs, along with a 'REFRESH' button and pagination information: 'Results per page: 50' and '1 - 24 of 24'.

Query results

Row	payment_installments	total_orders
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
11	10	5328
12	11	22