

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

The screenshot shows the Google Cloud BigQuery console interface. The left sidebar displays the 'Explorer' view with a tree of workspace resources. The 'customers' table is selected under the 'target_project' folder. The main panel shows the 'SCHEMA' tab for the 'customers' table. A filter bar is present above a table listing the columns and their data types.

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
<input type="checkbox"/> customer_id	STRING	NULLABLE				
<input type="checkbox"/> customer_unique_id	STRING	NULLABLE				
<input type="checkbox"/> customer_zip_code_prefix	INTEGER	NULLABLE				
<input type="checkbox"/> customer_city	STRING	NULLABLE				
<input type="checkbox"/> customer_state	STRING	NULLABLE				

Buttons at the bottom: EDIT SCHEMA, VIEW ROW ACCESS POLICIES

The screenshot shows the Google Cloud BigQuery console interface. The left sidebar displays the 'Explorer' view with a tree of workspace resources. The 'geolocation' table is selected under the 'target_project' folder. The main panel shows the 'SCHEMA' tab for the 'geolocation' table. A filter bar is present above a table listing the columns and their data types.

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
<input type="checkbox"/> geolocation_zip_code_prefix	INTEGER	NULLABLE				
<input type="checkbox"/> geolocation_lat	FLOAT	NULLABLE				
<input type="checkbox"/> geolocation_lng	FLOAT	NULLABLE				
<input type="checkbox"/> geolocation_city	STRING	NULLABLE				
<input type="checkbox"/> geolocation_state	STRING	NULLABLE				

Buttons at the bottom: EDIT SCHEMA, VIEW ROW ACCESS POLICIES

Business Case: Target SQL - Problem x BigQuery - SCALER-DSML-SQL - G: x +

console.cloud.google.com/bigquery?_ga=2.102646983.75686585.1679055632-37726303.1679055632&pli=1&project=curious-subject-380912&supportedpurview=project&ws=11m511m41n

Google Cloud SCALER-DSML-SQL Search (/) for resources, docs, products, and more Search

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

Explorer + ADD IK

order_items QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW LINEAGE

Filter Enter property name or value

<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags ?	Description
<input type="checkbox"/>	order_id	STRING	NULLABLE				
<input type="checkbox"/>	order_item_id	INTEGER	NULLABLE				
<input type="checkbox"/>	product_id	STRING	NULLABLE				
<input type="checkbox"/>	seller_id	STRING	NULLABLE				
<input type="checkbox"/>	shipping_limit_date	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	price	FLOAT	NULLABLE				
<input type="checkbox"/>	freight_value	FLOAT	NULLABLE				

EDIT SCHEMA VIEW ROW ACCESS POLICIES

Business Case: Target SQL - Problem x BigQuery - SCALER-DSML-SQL - G: x +

console.cloud.google.com/bigquery?_ga=2.102646983.75686585.1679055632-37726303.1679055632&pli=1&project=curious-subject-380912&supportedpurview=project&ws=11m511m41n

Google Cloud SCALER-DSML-SQL Search (/) for resources, docs, products, and more Search

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

Explorer + ADD IK

order_reviews QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW LINEAGE

Filter Enter property name or value

<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags ?	Description
<input type="checkbox"/>	review_id	STRING	NULLABLE				
<input type="checkbox"/>	order_id	STRING	NULLABLE				
<input type="checkbox"/>	review_score	INTEGER	NULLABLE				
<input type="checkbox"/>	review_comment_title	STRING	NULLABLE				
<input type="checkbox"/>	review_creation_date	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	review_answer_timestamp	TIMESTAMP	NULLABLE				

EDIT SCHEMA VIEW ROW ACCESS POLICIES

Business Case: Target SQL - Problem x BigQuery - SCALER-DSML-SQL - G: x +

console.cloud.google.com/bigquery?_ga=2.102646983.75686585.1679055632-37726303.1679055632&pli=1&project=curious-subject-380912&supportedpurview=project&ws=11m511m414m311sci

Google Cloud SCALER-DSML-SQL Search (/) for resources, docs, products, and more Search

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

Explorer + ADD <

Viewing workspace resources. SHOW STARRED ONLY

curious-subject-380912

- External connections
- emoloyee_schema
- farmers_market
- target_project
 - customers
 - geolocation
 - order_items
 - order_reviews
 - orders
 - payments
 - products
 - sellers

orders QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW LINEAGE

Filter Enter property name or value

<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
<input type="checkbox"/>	order_id	STRING	NULLABLE				
<input type="checkbox"/>	customer_id	STRING	NULLABLE				
<input type="checkbox"/>	order_status	STRING	NULLABLE				
<input type="checkbox"/>	order_purchase_timestamp	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	order_approved_at	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	order_delivered_carrier_date	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	order_delivered_customer_date	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	order_estimated_delivery_date	TIMESTAMP	NULLABLE				

EDIT SCHEMA VIEW ROW ACCESS POLICIES

Business Case: Target SQL - Problem x BigQuery - SCALER-DSML-SQL - G: x +

console.cloud.google.com/bigquery?_ga=2.102646983.75686585.1679055632-37726303.1679055632&pli=1&project=curious-subject-380912&supportedpurview=project&ws=11m511m414m311sci

Google Cloud SCALER-DSML-SQL Search (/) for resources, docs, products, and more Search

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

Explorer + ADD <

Viewing workspace resources. SHOW STARRED ONLY

curious-subject-380912

- External connections
- emoloyee_schema
- farmers_market
- target_project
 - customers
 - geolocation
 - order_items
 - order_reviews
 - orders
 - payments
 - products
 - sellers

payments QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW LINEAGE

Filter Enter property name or value

<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
<input type="checkbox"/>	order_id	STRING	NULLABLE				
<input type="checkbox"/>	payment_sequential	INTEGER	NULLABLE				
<input type="checkbox"/>	payment_type	STRING	NULLABLE				
<input type="checkbox"/>	payment_installments	INTEGER	NULLABLE				
<input type="checkbox"/>	payment_value	FLOAT	NULLABLE				

EDIT SCHEMA VIEW ROW ACCESS POLICIES

Business Case: Target SQL - Problem x BigQuery - SCALER-DSML-SQL - G x +

console.cloud.google.com/bigquery?_ga=2.102646983.75686585.1679055632-37726303.1679055632&pli=1&project=curious-subject-380912&supportedpurview=project&ws=11m511m414m311scurious-sub

Google Cloud SCALER-DSML-SQL Search (/) for resources, docs, products, and more Search

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

Explorer + ADD <

Viewing workspace resources. SHOW STARRED ONLY

- curious-subject-380912
 - External connections
 - emoloyee_schema
 - farmers_market
 - target_project
 - customers
 - geolocation
 - order_items
 - order_reviews
 - orders
 - payments
 - products
 - sellers

products QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW LINEAGE

Filter Enter property name or value

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
product_id	STRING	NULLABLE				
product_category	STRING	NULLABLE				
product_name_length	INTEGER	NULLABLE				
product_description_length	INTEGER	NULLABLE				
product_photos_qty	INTEGER	NULLABLE				
product_weight_g	INTEGER	NULLABLE				
product_length_cm	INTEGER	NULLABLE				
product_height_cm	INTEGER	NULLABLE				
product_width_cm	INTEGER	NULLABLE				

EDIT SCHEMA VIEW ROW ACCESS POLICIES

Business Case: Target SQL - Problem x BigQuery - SCALER-DSML-SQL - G x +

console.cloud.google.com/bigquery?_ga=2.102646983.75686585.1679055632-37726303.1679055632&pli=1&project=curious-subject-380912&supportedpurview=project&ws=11m511m414m311scurious-sub

Google Cloud SCALER-DSML-SQL Search (/) for resources, docs, products, and more Search

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

Explorer + ADD <

Viewing workspace resources. SHOW STARRED ONLY

- curious-subject-380912
 - External connections
 - emoloyee_schema
 - farmers_market
 - target_project
 - customers
 - geolocation
 - order_items
 - order_reviews
 - orders
 - payments
 - products
 - sellers

sellers QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW LINEAGE

Filter Enter property name or value

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
seller_id	STRING	NULLABLE				
seller_zip_code_prefix	INTEGER	NULLABLE				
seller_city	STRING	NULLABLE				
seller_state	STRING	NULLABLE				

EDIT SCHEMA VIEW ROW ACCESS POLICIES

2. Time period for which the data is given

SELECT

MIN(order_purchase_timestamp) AS min_time,

MAX(order_purchase_timestamp) AS max_time

FROM

`target_project.orders`

The screenshot shows the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo and a search bar. The main area is divided into three sections: Explorer, Query Editor, and Query Results.

Explorer: Displays a list of workspace resources under the project 'curious-subject-380912'. The 'target_project' is expanded, showing tables: customers, geolocation, order_items, order_reviews, orders, payments, products, and sellers.

Query Editor: Shows a query in 'Untitled 7' with the following SQL:

```
1 SELECT
2   MIN(order_purchase_timestamp) AS min_time,
3   MAX( order_purchase_timestamp) AS max_time
4 FROM
5   `target_project.orders`
6
```

Query Results: The results are displayed in a table with columns 'min_time' and 'max_time'.

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

Data is given from 2016 to 2018

3. Cities and States of customers ordered during the given period

4. `select distinct customer_state, customer_city`
5. `from `target_project.customers` as c`
6. `join `target_project.orders` as o on c.customer_id =o.customer_id`
7. `where order_status like '%delivered%'`
- 8.

The screenshot shows the Google Cloud BigQuery console interface. On the left, the 'Explorer' pane displays a project named 'curious-subject-380912' with various datasets listed, including 'customers', 'geolocation', 'order_items', 'order_reviews', 'orders', 'payments', 'products', and 'sellers'. The main editor area shows a SQL query in 'Untitled 3' with the following code:

```
1 select distinct customer_state, customer_city
2   from `target_project.customers` as c
3   join `target_project.orders` as o on c.customer_id =o.customer_id
4   where order_status like '%delivered%'
5
```

Below the query editor, the 'Query results' section is visible, showing a table with two columns: 'customer_state' and 'customer_city'. The results are displayed in a table format with 9 rows:

Row	customer_state	customer_city
1	GO	goiania
2	SP	sao paulo
3	RS	entre-juis
4	SP	jundiai
5	BA	ilheus
6	SP	santo andre
7	RS	porto alegre
8	MG	divisa nova
9	MT	culaba

The bottom of the interface shows the 'PERSONAL HISTORY' and 'PROJECT HISTORY' tabs, and a Windows taskbar at the very bottom.

Data about city and state are obtained from customers table and to obtain customer_ids that are ordered , inner joining with orders table made and in the order status delivered are only included

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?

```
select year, month , count(order_id) as count_  
from  
(select order_id,  
extract(year from order_purchase_timestamp ) as year,  
extract (month from order_purchase_timestamp ) as month  
from `target_project.orders`) as tab  
group by year,month  
order by year,count_ desc
```

The screenshot shows the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, a dropdown menu for 'SCALER-DSML-SQL', and a search bar. Below the navigation bar, there's a 'Sandbox' section with a link to 'Set up billing to upgrade to the full BigQuery experience'. The main interface is divided into three panels: Explorer, Query Editor, and Query Results.

Explorer Panel: Displays a tree view of workspace resources. Under the 'curious-subject-380912' project, there are several datasets listed, including 'customers', 'geolocation', 'order_items', 'order_reviews', 'orders', 'payments', 'products', and 'sellers'.

Query Editor Panel: Shows the SQL query being executed. The query is as follows:

```
1  
2  
3 select year, month , count(order_id) as count_  
4 from  
5 (select order_id,  
6 extract(year from order_purchase_timestamp ) as year,  
7 extract (month from order_purchase_timestamp ) as month  
8 from `target_project.orders`) as tab  
9 group by year,month  
10 order by year,count_ desc
```

Query Results Panel: Displays the results of the query in a table format. The table has columns for 'Row', 'year', 'month', and 'count_'. The results show data for the years 2016 and 2017, with counts for each month.

Row	year	month	count_
1	2016	10	324
2	2016	9	4
3	2016	12	1
4	2017	11	7544
5	2017	12	5673
6	2017	10	4631
7	2017	8	4331
8	2017	9	4285
9	2017	7	4026
10	2017	5	3700
11	2017	6	3245
12	2017	3	2682
13	2017	4	2404
14	2017	2	1780

by referring increased order_id count as increased sales , from data it is concluded there is a growing trend on e-commerce in Brazil from 2016 to 2018 .

in 2016 , orders are high during 10th month, followed by decline in orders in 12th month.

in 2017 , orders gradually increased from 1st month to last month , peaked at 11th month followed by 12th month

in 2018 , peak orders are observed in 1st month , sustainable orders are observed upto 8th month followed by rapid decline of order count .

from 2017 to 2018 , high amount of orders are seen during winter times .

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

```
select time , sum(count_)
from
(select( case when hour between 0 and 6 then 'dawn'
when hour between 7 and 12 then 'morning'
when hour between 13 and 18 then 'afternoon' else 'night' end ) as time, count_
from

(select hour, count(order_id) as count_
from

(select order_id,
extract(hour from order_purchase_timestamp ) as hour

from `target_project.orders`)
group by hour
order by hour) tab)
group by time
```


The screenshot shows the Google Cloud BigQuery console interface. On the left is the Explorer pane showing a project named 'curious-subject-380912' with various datasets like 'customers', 'geolocation', 'orders', etc. The main editor shows a SQL query titled 'Untitled' with the following code:

```

1 select time , sum(count_)
2 from
3 (select( case when hour between 0 and 6 then 'dawn'
4 when hour between 7 and 12 then 'morning'
5 when hour between 13 and 18 then 'afternoon' else 'night' end ) as time, count_
6 from
7 from
8 (select hour, count(order_id) as count_
9 from
10 from
11 (select order_id,
12 extract(hour from order_purchase_timestamp ) as hour
13 from `target_project.orders`)
14 group by hour
15 order by hour) tab)

```

Below the query editor, the 'Query results' section is visible, showing a table with 4 rows and 2 columns: 'time' and 'sum(count_)'. The results are as follows:

Row	time	sum(count_)
1	morning	27733
2	dawn	5242
3	afternoon	38135
4	night	28331

Trend suggest that , afternoon orders are high followed by night orders and morning orders and dawn orders.

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

1. Get month on month orders by states
2. `select customer_state, month, count(order_id) as sum`
3. `from`
4. `(select order_id ,customer_state,`
5. `extract (month from order_purchase_timestamp) month`
6. `from `target_project.orders` o join `target_project.customers` c on o.customer_id=c`
7. `where order_status = 'delivered') tab`
8. `group by customer_state,month`
9. `order by customer_state,month`

profound amounts of orders are seen in 5th month from a state 'SP' in brazil

Delivered orders are taken into consideration . group by and order by is performed on both customer_state and month to know, in each state for each month number of orders delivered .

The screenshot shows the Google Cloud BigQuery console. The Explorer on the left lists the project 'curious-subject-380912' and its datasets, including 'target_project' which contains tables like 'customers', 'orders', and 'products'. The main editor shows a query titled 'Untitled' with the following SQL:

```

1 select customer_state, month, count(order_id) as sum
2 from
3 (select order_id, customer_state,
4  extract (month from order_purchase_timestamp ) month
5  from `target_project.orders` o join `target_project.customers` c on o.customer_id=c.customer_id
6  where order_status = 'delivered' ) tab
7 group by customer_state, month
8 order by customer_state, month

```

The 'Query results' tab displays a table with columns: Row, customer_state, month, and sum. The results show data for states AL, AM, and AP across months 9, 10, 11, and 12.

Row	customer_state	month	sum
21	AL	9	19
22	AL	10	29
23	AL	11	26
24	AL	12	12
25	AM	1	12
26	AM	2	16
27	AM	3	13
28	AM	4	19
29	AM	5	17
30	AM	6	8
31	AM	7	23
32	AM	8	9
33	AM	9	9
34	AM	10	3
35	AM	11	10

2. Distribution of customers across the states in Brazil
3. `select customer_state, count(customer_id) as customer_count`
- 4.
5. `from `target_project.customers``
6. `group by customer_state`
7. `order by customer_state`

The screenshot shows the Google Cloud BigQuery console with a new query titled 'Untitled':

```

1 select customer_state, count(customer_id ) as customer_count
2 from `target_project.customers`
3 group by customer_state
4 order by customer_state

```

The 'Query results' tab displays a table with columns: Row, customer_state, and customer_count. The results show the total number of customers for each state, with 'SP' having the highest count at 41746.

Row	customer_state	customer_count
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
12	MS	715
13	MT	907
14	PA	975
15	PB	536

Upon ordering by customer_count it is noticed that, Huge number of customers are from a state called 'SP' WITH WHOOPING 41746 CUSTOMERS

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

```
WITH
cte AS (
SELECT
EXTRACT(year
FROM
order_purchase_timestamp) AS year_,
SUM(payment_value) AS revenue
FROM
`target_project.orders` AS o
INNER JOIN
`target_project.payments` AS p
ON
p.order_id=o.order_id
WHERE
order_status='delivered'
AND EXTRACT(month
FROM
order_purchase_timestamp) BETWEEN 0
AND 8
GROUP BY
year_),
lastRev AS (
SELECT
*,
LAG(revenue) OVER(ORDER BY year_ ASC) AS prev_rev
FROM
cte)
SELECT
(revenue-prev_rev)/prev_rev*100 AS per_inc
FROM
lastRev
```

The screenshot displays the Google Cloud BigQuery console interface. On the left, the 'Explorer' pane shows a project named 'curious-subject-380912' with various datasets listed, 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 WITH
2 cte AS (
3 SELECT
4 EXTRACT(year
5 FROM
6 order_purchase_timestamp) AS year_,
7 SUM(payment_value) AS revenue
8 FROM
```

Below the query editor, the 'Query results' section is visible, showing a table with two columns: 'per_inc' and 'null'. The results are as follows:

Row	per_inc	null
1		
2	143.330718...	

143 % growth of revenue is observed

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

3. **SELECT**

4. `c.customer_state,`

5. `SUM(price) AS sum_price,`

6. `SUM(freight_value) AS sum_freight_value,`

7. `AVG(price) AS mean_price,`

8. `AVG(freight_value) AS mean_freight_value`

9. **FROM**

10. ``target_project.customers` c`

11. **LEFT JOIN**

12. ``target_project.orders` o`

13. **ON**

14. `c.customer_id=o.customer_id`

15. **JOIN**

16. ``target_project.order_items` ot`

17. **ON**

18. `o.order_id=ot.order_id`

19. **GROUP BY**

20. `c.customer_state`

The screenshot displays the Google Cloud BigQuery console interface. On the left, the 'Explorer' pane shows a project named 'curious-subject-380912' with various datasets listed, including 'customers', 'order_items', and 'order_reviews'. The main area shows a SQL query in a text editor, which is a SELECT statement with multiple joins and aggregations. Below the query editor, the 'Query results' section is visible, showing a table with columns for 'customer_state', 'sum_price', 'sum_freight_value', 'mean_price', and 'mean_freight_value'. The results are grouped by customer state, with 13 rows displayed. The bottom of the console shows 'PERSONAL HISTORY' and 'PROJECT HISTORY' tabs.

```
1 SELECT
2   c.customer_state,
3   SUM(price) AS sum_price,
4   SUM(freight_value) AS sum_freight_value,
5   AVG(price) AS mean_price,
6   AVG(freight_value) AS mean_freight_value
7 FROM
8   `target_project.customers` c
9 LEFT JOIN
10  `target_project.orders` o
11 ON
12  c.customer_id=o.customer_id
```

Row	customer_state	sum_price	sum_freight_value	mean_price	mean_freight_value
1	RN	83034.9799...	18860.0999...	156.965935...	35.6523629...
2	CE	227254.709...	48351.5899...	153.758261...	32.7142016...
3	RS	750304.020...	135522.740...	120.337453...	21.7958043...
4	SC	520553.340...	89660.2600...	124.653577...	21.4702687...
5	SP	5202955.05...	718723.069...	109.653629...	15.1472753...
6	MO	1585308.02...	270853.460...	120.748574...	20.6301668...
7	BA	511349.990...	100156.679...	134.601208...	26.3639589...
8	RJ	1824092.66...	305589.310...	125.117818...	20.9609239...
9	GO	294591.949...	53114.9799...	126.271731...	22.7668152...
10	MA	119648.219...	31523.7700...	145.204150...	38.2570024...
11	PE	262788.029...	59449.6599...	145.508322...	32.9178626...
12	PB	115268.080...	25719.7299...	191.475215...	42.7238039...
13	ES	275037.309...	49764.5999...	121.913701...	22.0587765...

sum and mean are performed in select clause and as they are asked according to customer state multiple joins are performed followed by customer state grouping

5. Analysis on sales, freight and delivery time

5.1 Calculate days between purchasing, delivering and estimated delivery

```
SELECT
  order_id,
  DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp, day) AS order_delivered
_date,
  DATE_DIFF(order_estimated_delivery_date,order_purchase_timestamp, day) AS estimated_date,
FROM
  `target_project.orders`
where DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp, day) is not null
```

The screenshot shows the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, a search bar, and tabs for the current session. The left sidebar displays the 'Explorer' view with a tree structure of workspace resources, including 'curious-subject-380912' and its sub-resources like 'external_connections', 'employee_schema', 'farmers_market', 'target_project', 'customers', 'geolocation', 'order_items', 'order_reviews', 'orders', 'payments', 'products', and 'sellers'. The main panel shows a SQL query in the 'Untitled' editor, which is the same query as in the previous block. Below the editor, the 'Query results' section is visible, showing a table with 13 rows and 3 columns: 'order_id', 'order_delivered_date', and 'estimated_date'. The table is sorted by 'order_id' in ascending order. The 'Query results' section also includes tabs for 'RESULTS', 'JSON', 'EXECUTION DETAILS', and 'EXECUTION GRAPH', with 'RESULTS' being the active tab. The 'Query results' section also includes a 'PERSONAL HISTORY' and 'PROJECT HISTORY' section at the bottom.

Row	order_id	order_delivered_date	estimated_date
1	1950d77989f6a877539f5379...	30	17
2	2c45c33d2f9cb8ffbb1c8ecc28...	30	59
3	65d1e226dfaeb8cdc42f66542...	35	52
4	635c894d068ac37e6e03dc54e...	30	32
5	3b97562c3aee8bdedcb5c2e45...	32	33
6	68f47f50f04c4cb6774570cfd...	29	31
7	276e9ec344d3bf029ff83a161c...	43	39
8	54e1a3c2b97fb0809da548a59...	40	36
9	fd04fa4105ee8045fe0139ca5...	37	35
10	302bb8109d097a9fce9c9cfc5...	33	28
11	66057d37308e787052a32828...	38	32
12	19135c945c554eebfdf7576c73...	36	33
13	4493e45e7ca1084efcd38ddeb...	34	33

5.2.

- `time_to_delivery = order_purchase_timestamp - order_delivered_customer_date`
- `diff_estimated_delivery = order_estimated_delivery_date - order_delivered_customer_date`
- `SELECT`
- `order_id,`
- `DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day) time_to_delivery,`
- `DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, day) diff_estimated_delivery`
- `FROM`
- ``target_project.orders``

The screenshot shows the Google Cloud BigQuery console interface. On the left, the Explorer pane displays the project hierarchy for 'curious-subject-380912', including datasets like 'customers', 'geolocation', 'order_items', 'order_reviews', 'orders', 'payments', 'products', and 'sellers'. The main editor shows a SQL query titled 'Untitled 3' with the following code:

```
1 SELECT
2   order_id,
3   DATE_DIFF( order_delivered_customer_date, order_purchase_timestamp, day) time_to_delivery,
4   DATE_DIFF( order_estimated_delivery_date, order_delivered_customer_date, day) diff_estimated_delivery
5 FROM
6   `target_project.orders`
```

The 'Query results' pane shows the output of the query, which includes columns for 'order_id', 'time_to_delivery', and 'diff_estimated_delivery'. The results are displayed in a table with 14 rows.

Row	order_id	time_to_delivery	diff_estimated_delivery
1	1950d777989f6a877539f5379...	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28
3	65d1e226dfaeb8cdc42f66542...	35	16
4	635c894d068ac37e6e03dc54e...	30	1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfe...	29	1
7	276e9ec344d3bf029ff83a161c...	43	-4
8	54e1a3c2b97fb0809da548a59...	40	-4
9	fd04fa4105ee8045f6a0139ca5...	37	-1
10	302bb8109d097a9f6e9cfc5...	33	-5
11	66057d37308e787052a32828...	38	-6
12	19135c945c554aebfd7576c73...	36	-2
13	4493e45e7ca1084efcd38ddeb...	34	0
14	70c77e51e0f179d75a64a6141...	42	-11

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

```

/*
Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery
*/

SELECT
    c.customer_state,
    avg(freight_value) AS mean_freight_value,
    avg(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day)) mean_time
_to_delivery,
    avg( DATE_DIFF( order_estimated_delivery_date, order_delivered_customer_date, day)) mean_d
iff_estimated_delivery

FROM
    `target_project.customers` c
JOIN
    `target_project.orders` o
ON
    c.customer_id=o.customer_id
JOIN
    `target_project.order_items` AS ot
ON
    o.order_id=ot.order_id
GROUP BY
    c.customer_state

```

The screenshot shows the Google Cloud BigQuery console interface. On the left is the Explorer pane showing a project named 'curious-subject-380912' with various datasets like 'employee_schema', 'farmers_market', and 'target_project'. The 'target_project' dataset is expanded, showing tables like 'customers', 'geolocation', 'order_items', 'order_reviews', 'orders', 'payments', 'products', and 'sellers'. The main area displays a SQL query titled 'Untitled 4' which is the same query as shown in the previous block. Below the query editor, the 'Query results' section is visible, showing a table with 7 rows of data grouped by customer state.

Row	customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	MT	28.1662843...	17.5081967...	13.6393442...
2	MA	38.2570024...	21.2037500...	9.109999999...
3	AL	35.8436711...	23.9929742...	7.97658079...
4	SP	15.1472753...	8.25960855...	10.2655943...
5	MG	20.6301668...	11.5155221...	12.3971510...
6	PE	32.9178626...	17.7920962...	12.5521191...
7	RJ	20.9609239...	14.6893821...	11.1444931...

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

(Top 5 states with highest average freight value)

SELECT

```
c.customer_state,  
avg(freight_value) AS mean_freight_value,  
avg(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day)) mean_time  
_to_delivery,  
avg( DATE_DIFF( order_estimated_delivery_date, order_delivered_customer_date, day)) mean_d  
iff_estimated_delivery
```

FROM

```
`target_project.customers` c
```

JOIN

```
`target_project.orders` o
```

ON

```
c.customer_id=o.customer_id
```

JOIN

```
`target_project.order_items` AS ot
```

ON

```
o.order_id=ot.order_id
```

GROUP BY

```
c.customer_state  
order by mean_freight_value desc  
limit 5
```

The screenshot displays the Google Cloud BigQuery console interface. On the left, the 'Explorer' pane shows a project named 'curious-subject-380912' with various datasets like 'customers', 'order_items', and 'orders'. The main editor shows a SQL query titled 'Untitled 4' that calculates the average freight value and delivery time for the top 5 states. Below the query, the 'Query results' section shows a table with 5 rows of data, including customer states (RR, PB, RO, AC, PI) and their corresponding average freight values, mean time to delivery, and mean difference in estimated delivery.

Row	customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	RR	42.9844230...	27.8260869...	17.4347826...
2	PB	42.7238039...	20.1194539...	12.1501706...
3	RO	41.0697122...	19.2820512...	19.0805860...
4	AC	40.0733695...	20.3296703...	20.0109890...
5	PI	39.1479704...	18.9311663...	10.6826003...

(Top 5 states with lowest average freight value)

```
SELECT
  c.customer_state,
  avg(freight_value) AS mean_freight_value,
  avg( DATE_DIFF( order_delivered_customer_date, order_purchase_timestamp, day)) mean_time
_to_delivery,
  avg( DATE_DIFF( order_estimated_delivery_date, order_delivered_customer_date, day)) mean_d
iff_estimated_delivery

FROM
  `target_project.customers` c
JOIN
  `target_project.orders` o
ON
  c.customer_id=o.customer_id
JOIN
  `target_project.order_items` AS ot
ON
  o.order_id=ot.order_id
GROUP BY
  c.customer_state
order by mean_freight_value
limit 5
```

The screenshot displays the Google Cloud BigQuery console interface. On the left, the 'Explorer' pane shows a project named 'curious-subject-380912' with various datasets listed, including 'customers', 'geolocation', 'order_items', 'order_reviews', 'orders', 'payments', 'products', and 'sellers'. The main editor area, titled 'Untitled 4', contains a SQL query that calculates the average freight value, average time to delivery, and average difference in estimated delivery for the top 5 states with the lowest average freight value. The query is as follows:

```
SELECT
  c.customer_state,
  avg(freight_value) AS mean_freight_value,
  avg( DATE_DIFF( order_delivered_customer_date, order_purchase_timestamp, day)) mean_time_to_delivery,
  avg( DATE_DIFF( order_estimated_delivery_date, order_delivered_customer_date, day)) mean_diff_estimated_delivery
FROM
  `target_project.customers` c
JOIN
  `target_project.orders` o
ON
  c.customer_id=o.customer_id
JOIN
  `target_project.order_items` AS ot
ON
  o.order_id=ot.order_id
GROUP BY
  c.customer_state
order by mean_freight_value
limit 5
```

Below the query editor, the 'Query results' section is visible, showing a table with 5 rows of data. The table has 5 columns: 'customer_state', 'mean_freight_value', 'mean_time_to_delivery', 'mean_diff_estimated_delivery', and an unlabeled column. The data is as follows:

Row	customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery	
1	SP	15.1472753...	8.25960855...	10.2655943...	
2	PR	20.5316515...	11.4807930...	12.5389998...	
3	MG	20.6301668...	11.5155221...	12.3971510...	
4	RJ	20.9609239...	14.6893821...	11.1444931...	
5	DF	21.0413549...	12.5014861...	11.2747346...	

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

5.6. Top 5 states with highest/lowest average time to delivery

(Top 5 states with highest average time to delivery)

```
SELECT
    c.customer_state,
    avg(freight_value) AS mean_freight_value,
    avg(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day)) mean_time
_to_delivery,
    avg( DATE_DIFF( order_estimated_delivery_date, order_delivered_customer_date, day)) mean_d
iff_estimated_delivery

FROM
    `target_project.customers` c
JOIN
    `target_project.orders` o
ON
    c.customer_id=o.customer_id
JOIN
    `target_project.order_items` AS ot
ON
    o.order_id=ot.order_id
GROUP BY
    c.customer_state
order by mean_time_to_delivery desc
limit 5
```

The screenshot shows the Google Cloud BigQuery console interface. On the left is the Explorer pane showing the project hierarchy. The main area displays a SQL query in 'Untitled 4' and the resulting 'Query results' table.

Query:

```
SELECT
    c.customer_state,
    avg(freight_value) AS mean_freight_value,
    avg(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day)) mean_time_to_delivery,
    avg( DATE_DIFF( order_estimated_delivery_date, order_delivered_customer_date, day)) mean_diff_estimated_delivery

FROM
    `target_project.customers` c
JOIN
    `target_project.orders` o
ON
    c.customer_id=o.customer_id
JOIN
    `target_project.order_items` AS ot
ON
    o.order_id=ot.order_id
GROUP BY
    c.customer_state
order by mean_time_to_delivery desc
limit 5
```

Query results:

Row	customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	RR	42.9844230...	27.8260869...	17.4347826...
2	AP	34.0060975...	27.7530864...	17.4444444...
3	AM	33.2053939...	25.9631901...	18.9754601...
4	AL	35.8436711...	23.9929742...	7.97658079...
5	PA	35.8326851...	23.3017077...	13.3747628...

(Top 5 states with lowest average time to delivery)

```
SELECT
    c.customer_state,
    avg(freight_value) AS mean_freight_value,
    avg(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day)) mean_time
_to_delivery,
    avg( DATE_DIFF( order_estimated_delivery_date, order_delivered_customer_date, day)) mean_d
iff_estimated_delivery

FROM
    `target_project.customers` c
JOIN
    `target_project.orders` o
ON
    c.customer_id=o.customer_id
JOIN
    `target_project.order_items` AS ot
ON
    o.order_id=ot.order_id
GROUP BY
    c.customer_state
order by mean_time_to_delivery
limit 5
```

The screenshot shows the Google Cloud BigQuery console interface. On the left is the Explorer pane showing a project named 'curious-subject-380912' with various datasets like 'customers', 'orders', and 'order_items'. The main area displays a SQL query in 'Untitled 4' and its results. The query is a SELECT statement that calculates the average freight value and average time to delivery for the top 5 states with the lowest average time to delivery. The results are shown in a table with columns: Row, customer_state, mean_freight_value, mean_time_to_delivery, and mean_diff_estimated_delivery. The results are sorted by mean_time_to_delivery in ascending order.

Row	customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	SP	15.1472753...	8.25960855...	10.2655943...
2	PR	20.5316515...	11.4807930...	12.5389998...
3	MG	20.6301668...	11.5155221...	12.3971510...
4	DF	21.0413549...	12.5014861...	11.2747346...
5	SC	21.4703687...	14.5209858...	10.6688628...

5.7. Top 5 states where delivery is really fast/ not so fast compared to estimated date

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

SELECT

```
c.customer_state,  
avg(freight_value) AS mean_freight_value,  
avg(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day)) mean_time  
_to_delivery,  
avg( DATE_DIFF( order_estimated_delivery_date, order_delivered_customer_date, day)) mean_d  
iff_estimated_delivery
```

FROM

```
`target_project.customers` c
```

JOIN

```
`target_project.orders` o
```

ON

```
c.customer_id=o.customer_id
```

JOIN

```
`target_project.order_items` AS ot
```

ON

```
o.order_id=ot.order_id
```

GROUP BY

```
c.customer_state
```

```
order by mean_diff_estimated_delivery desc
```

```
limit 5
```

The screenshot shows the Google Cloud BigQuery console interface. On the left is the Explorer pane showing the project hierarchy. The main area displays a SQL query in the editor, and below it, the 'Query results' section shows a table with 5 rows of data.

Query results

Row	customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	AC	40.0733695...	20.3296703...	20.0109890...
2	RO	41.0697122...	19.2820512...	19.0805860...
3	AM	33.2053939...	25.9631901...	18.9754601...
4	AP	34.0060975...	27.7530864...	17.4444444...
5	RR	42.9844230...	27.8260869...	17.4347826...

Top 5 states where delivery is really fast compared to estimated date

```
SELECT
    c.customer_state,
    avg(freight_value) AS mean_freight_value,
    avg(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day)) mean_time
_to_delivery,
    avg( DATE_DIFF( order_estimated_delivery_date, order_delivered_customer_date, day)) mean_d
iff_estimated_delivery

FROM
    `target_project.customers` c
JOIN
    `target_project.orders` o
ON
    c.customer_id=o.customer_id
JOIN
    `target_project.order_items` AS ot
ON
    o.order_id=ot.order_id
GROUP BY
    c.customer_state
order by mean_diff_estimated_delivery
limit 5
```

The screenshot shows the Google Cloud BigQuery console interface. On the left is the Explorer pane showing the project hierarchy for 'curious-subject-380912', including datasets like 'customers', 'order_items', and 'orders'. The main editor displays a SQL query (Untitled 4) that calculates the average time to delivery and the difference between estimated and actual delivery dates, grouped by customer state. Below the query editor, the 'Query results' section is visible, showing a table with 5 rows of data. The columns are 'customer_state', 'mean_freight_value', 'mean_time_to_delivery', and 'mean_diff_estimated_delivery'. The results are ordered by 'mean_diff_estimated_delivery' in descending order.

Row	customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	AL	35.8436711...	23.9929742...	7.97658079...
2	MA	38.2570024...	21.2037500...	9.10999999...
3	SE	36.6531688...	20.9786666...	9.16533333...
4	ES	22.0587765...	15.1928089...	9.76853932...
5	BA	26.3639589...	18.7746402...	10.1194678...

6. Payment type analysis:

1. Month over Month count of orders for different payment types

```
2. SELECT
3.   payment_type,
4.   month_,
5.   COUNT(order_id) order_id_count
6. FROM (
7.   SELECT
8.     p.order_id,
9.     p.payment_type,
10.    EXTRACT(month
11.    FROM
12.      o.order_purchase_timestamp) month_
13. FROM
14.   `target_project.payments` p
15. JOIN
16.   `target_project.orders` o
17. ON
18.   p.order_id=o.order_id) tab
19. GROUP BY
20.   payment_type,
21.   month_
22. ORDER BY
23.   payment_type,
24.   month_
```

The screenshot displays the Google Cloud BigQuery console interface. On the left, the 'Explorer' pane shows a project named 'curious-subject-380912' with various datasets listed, including 'orders' and 'payments'. The main editor area contains a SQL query titled 'Untitled 5' that performs a month-over-month count of orders for different payment types. The query results are displayed in a table with columns: 'payment_type', 'month_', and 'order_id_count'. The results show data for the month of January (month_ = 1) for the payment type 'UPI', with order counts ranging from 1715 to 2077. The interface also includes tabs for 'JOB INFORMATION', 'RESULTS', 'JSON', 'EXECUTION DETAILS', and 'EXECUTION GRAPH'.

Row	payment_type	month_	order_id_count
1	UPI	1	1715
2	UPI	2	1723
3	UPI	3	1942
4	UPI	4	1783
5	UPI	5	2035
6	UPI	6	1807
7	UPI	7	2074
8	UPI	8	2077

Upon sorting data through ordercounts it is established that, Huge number of orders are placed in 5th month through creditcards

2. Count of orders based on the no. of payment instalments

1. `SELECT`
2. `payment_installments,`
3. `COUNT(order_id) order_count_per_installment`
4. `FROM`
5. ``target_project.payments``
6. `GROUP BY`
7. `payment_installments`

The screenshot shows the Google Cloud BigQuery console interface. On the left is the Explorer pane showing the project hierarchy: curious-subject-380912 > target_project > payments. The main editor shows a SQL query in 'Untitled 6' with the following code:

```
1 SELECT
2   payment_installments,
3   COUNT(order_id) order_count_per_installment
4 FROM
5   `target_project.payments`
6 GROUP BY
7   payment_installments
```

Below the query editor, the 'Query results' section is visible, showing a table with 9 rows. The table has two columns: 'payment_installments' and 'order_count_per_installment'. The first row shows 0 installments with a count of 2. The second row shows 1 installment with a count of 52546. The third row shows 2 installments with a count of 12413. The fourth row shows 3 installments with a count of 10461. The fifth row shows 4 installments with a count of 7098. The sixth row shows 5 installments with a count of 5239. The seventh row shows 6 installments with a count of 3920. The eighth row shows 7 installments with a count of 1626. The ninth row shows 8 installments with a count of 4268.

Row	payment_installments	order_count_per_installment
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

Order count is huge for 1st payment_installment
18 orders took most number of payment instalments that are 24