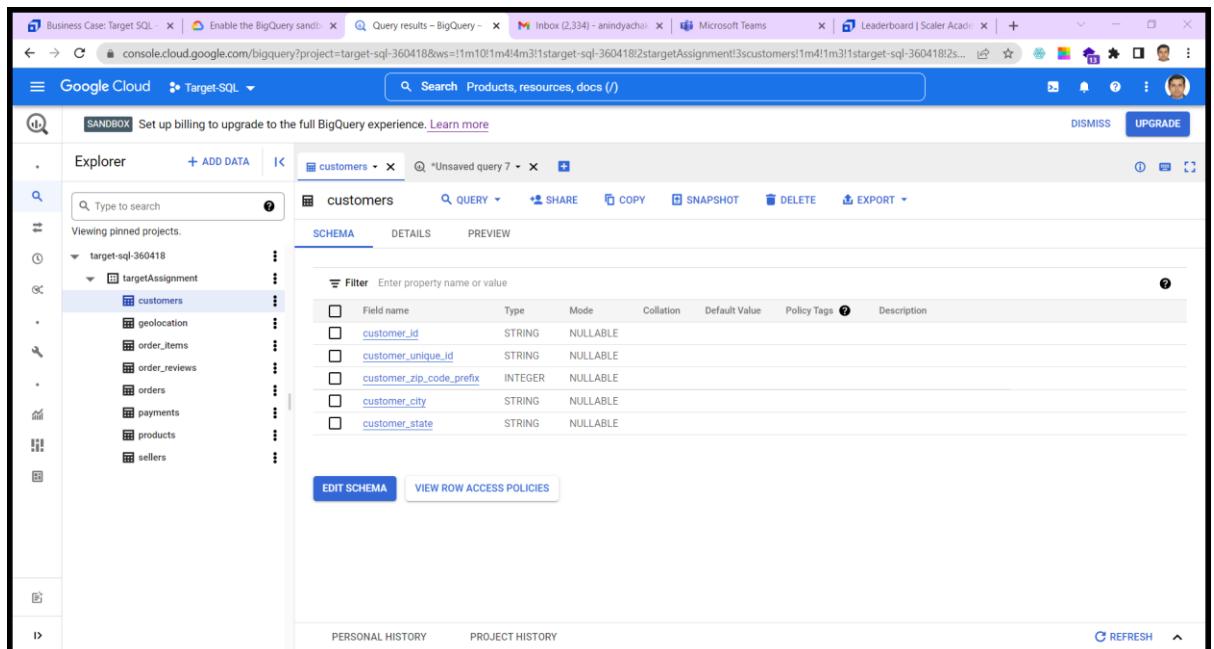


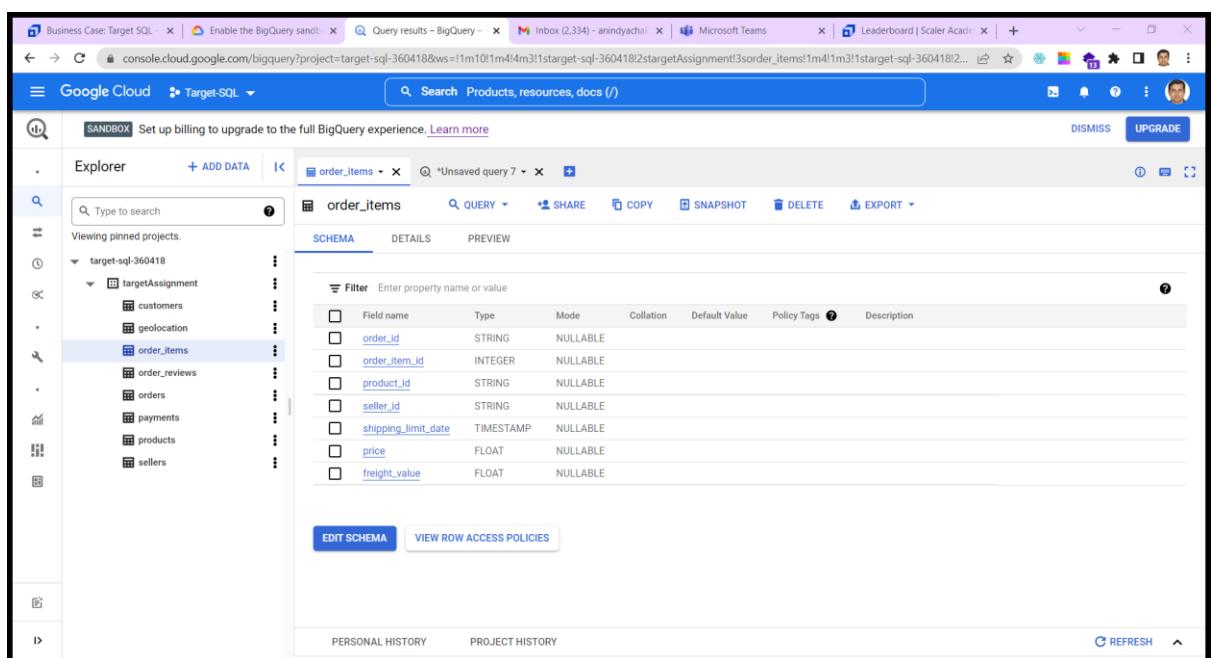
Business Case: Target SQL

- Initial exploration of dataset like checking the characteristics of data
 - Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset
 - Data type of columns in a table



The screenshot shows the Google Cloud BigQuery interface for the 'customers' table in the 'targetAssignment' dataset of project 'target-sql-360418'. The 'SCHEMA' tab is selected, displaying the following columns:

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
customer_id	STRING	NULLABLE				
customer_unique_id	STRING	NULLABLE				
customer_zip_code_prefix	INTEGER	NULLABLE				
customer_city	STRING	NULLABLE				
customer_state	STRING	NULLABLE				



The screenshot shows the Google Cloud BigQuery interface for the 'order_items' table in the 'targetAssignment' dataset of project 'target-sql-360418'. The 'SCHEMA' tab is selected, displaying the following columns:

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
order_id	STRING	NULLABLE				
order_item_id	INTEGER	NULLABLE				
product_id	STRING	NULLABLE				
seller_id	STRING	NULLABLE				
shipping_limit_date	TIMESTAMP	NULLABLE				
price	FLOAT	NULLABLE				
freight_value	FLOAT	NULLABLE				

The screenshot shows the Google Cloud BigQuery schema editor for the `order_reviews` table. The table has the following schema:

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
<code>review_id</code>	STRING	NULLABLE				
<code>order_id</code>	STRING	NULLABLE				
<code>review_score</code>	INTEGER	NULLABLE				
<code>review_comment_title</code>	STRING	NULLABLE				
<code>review_creation_date</code>	TIMESTAMP	NULLABLE				
<code>review_answer_timestamp</code>	TIMESTAMP	NULLABLE				

Below the schema, there are buttons for `EDIT SCHEMA` and `VIEW ROW ACCESS POLICIES`.

The screenshot shows the Google Cloud BigQuery schema editor for the `orders` table. The table has the following schema:

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

Below the schema, there are buttons for `EDIT SCHEMA` and `VIEW ROW ACCESS POLICIES`.

The screenshot shows the Google Cloud BigQuery interface. The left sidebar displays the 'Explorer' section with a tree view of datasets and tables. Under the 'target-sql-360418' dataset, the 'targetAssignment' table is expanded, showing sub-tables like 'customers', 'geolocation', 'order_items', etc., and the 'payments' table is selected. The main panel shows the schema for the 'payments' table. The schema consists of the following fields:

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
order_id	STRING	NULLABLE				
payment_sequential	INTEGER	NULLABLE				
payment_type	STRING	NULLABLE				
payment_installments	INTEGER	NULLABLE				
payment_value	FLOAT	NULLABLE				

Below the schema, there are 'EDIT SCHEMA' and 'VIEW ROW ACCESS POLICIES' buttons.

This screenshot shows the Google Cloud BigQuery interface, similar to the previous one but with a different table selected. The 'payments' table from the first screenshot has been deselected, and the 'products' table under the 'targetAssignment' table in the 'target-sql-360418' dataset is now selected. The schema for the 'products' table is displayed:

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				

Below the schema, there are 'EDIT SCHEMA' and 'VIEW ROW ACCESS POLICIES' buttons.

The screenshot shows the Google Cloud BigQuery interface. The left sidebar displays the 'target-sql-360418' project with its tables: customers, geolocation, order_items, order_reviews, orders, payments, products, and sellers. The 'sellers' table is selected. The main panel shows the schema for the 'sellers' table:

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				

Buttons at the bottom include 'EDIT SCHEMA' and 'VIEW ROW ACCESS POLICIES'.

The screenshot shows the Google Cloud BigQuery interface. The left sidebar displays the 'target-sql-360418' project with its tables: customers, geolocation, order_items, order_reviews, orders, payments, products, and sellers. The 'geolocation' table is selected. The main panel shows the schema for the 'geolocation' table:

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
geolocation_zip_code_prefix	INTEGER	NULLABLE				
geolocation_lat	FLOAT	NULLABLE				
geolocation_lng	FLOAT	NULLABLE				
geolocation_city	STRING	NULLABLE				
geolocation_state	STRING	NULLABLE				

Buttons at the bottom include 'EDIT SCHEMA' and 'VIEW ROW ACCESS POLICIES'.

- Time period for which the data is given

- ✓ Orders – From **2016-09-04 21:15:19 UTC** to **2018-11-12 00:00:00 UTC**
- ✓ Order_reviews - From **2016-09-19 00:15:34 UTC** to **2020-04-09 22:35:08 UTC**

The screenshot shows the Google Cloud BigQuery interface. On the left, the Explorer sidebar lists datasets like 'target-sql-360418' and tables such as 'order_items', 'orders', and 'payments'. The main area displays an 'Unsaved query' with the following SQL code:

```

1 SELECT
2   MIN(shipping_limit_date) AS min_shipping_limit_date,
3   MAX(shipping_limit_date) AS max_shipping_limit_date
4   FROM `target-sql-360418.targetAssignment.order_items`;
5

```

The 'Query results' section shows a single row of data:

Row	min_shipping_limit_date	max_shipping_limit_date
1	2016-09-19 00:15:34 UTC	2020-04-09 22:35:08 UTC

This screenshot shows the same Google Cloud BigQuery interface after running the previous query. The status bar indicates 'Query completed.' The results table now includes more columns and data:

Row	min_order_purchase_timestamp	max_order_purchase_timestamp	min_order_approved_at	max_order_approved_at	min_order_delivered_carrier_date	max_order_delivered_customer_date
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC	2016-09-15 12:16:38 UTC	2018-09-03 17:40:06 UTC	2016-10-08 10:34:01 UTC	2018-09-11 19:48:21 UTC

■ Cities and States covered in the dataset

The screenshot shows the Google Cloud BigQuery interface. The left sidebar has an 'Explorer' section pinned to the top, showing projects like 'target-sql-360418'. Below it are tables: 'customers', 'geolocation', 'order_items', 'order_reviews', 'orders', 'payments', 'products', and 'sellers'. The main area shows a query editor with the following SQL code:

```

1 SELECT DISTINCT customer_city FROM `target-sql-360418.targetAssignment.customers`;
2 SELECT DISTINCT customer_state FROM `target-sql-360418.targetAssignment.customers`;
3
4 SELECT DISTINCT seller_city FROM `target-sql-360418.targetAssignment.sellers`;
5 SELECT DISTINCT seller_state FROM `target-sql-360418.targetAssignment.sellers`;
6
7

```

The results pane shows a table of seller states:

seller_state
AC
AM
BA
CE
DF
ES

- **In-depth Exploration**

- Is there a growing trend on e-commerce in Brazil?
 - Yes, there is a growing trend year on year since 2017 onwards.

The screenshot shows the Google Cloud BigQuery interface. The left sidebar has an 'Explorer' section pinned to the top, showing projects like 'target-sql-360418'. Below it are tables: 'customers', 'geolocation', 'order_items', 'order_reviews', 'orders', 'payments', 'products', and 'sellers'. The main area shows a query editor with the following SQL code:

```

1 SELECT
2   EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
3   COUNT(*) AS orders_by_year,
4   FROM `target-sql-360418.targetAssignment.orders`
5   WHERE order_status = 'delivered'
6   GROUP BY 1
7   ORDER BY year;
8
9 SELECT
10   EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
11   EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
12   COUNT(*) AS orders_by_year_month,
13   FROM `target-sql-360418.targetAssignment.orders`
14   WHERE order_status = 'delivered'
15   GROUP BY 1,2
16   ORDER BY year,month;

```

The results pane shows two tables: one for total orders by year and one for total orders by year and month.

year	orders_by_year
2016	267
2017	43428
2018	52783

year	month	orders_by_year_month
2016	1	1
2016	2	1
2016	3	1
2016	4	1
2016	5	1
2016	6	1
2016	7	1
2016	8	1
2016	9	1
2016	10	1
2016	11	1
2016	12	1
2017	1	1
2017	2	1
2017	3	1
2017	4	1
2017	5	1
2017	6	1
2017	7	1
2017	8	1
2017	9	1
2017	10	1
2017	11	1
2017	12	1
2018	1	1
2018	2	1
2018	3	1
2018	4	1
2018	5	1
2018	6	1
2018	7	1
2018	8	1
2018	9	1
2018	10	1
2018	11	1
2018	12	1

- How can we describe a complete scenario?
 - In year 2016, it was just the beginning it seems
 - In year 2017, it started showing **almost growing** trends every month on month, till month of November
 - There was slight decline in month of December 2017

- In year 2018 January onwards, it showed again consistent orders of 6K + per month for the rest of year

```

8
9   SELECT
10    EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
11    EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
12    COUNT(*) as orders_by_year_month,
13  FROM `target-sql-360418.targetAssignment.orders`
14 WHERE order_status = 'delivered'
15 GROUP BY 1,2
16 ORDER BY year,month;
  
```

Row	year	month	orders_by_year_month
14	2017	11	7289
15	2017	12	5513
16	2018	1	7069
17	2018	2	6555
18	2018	3	7003
19	2018	4	6798
20	2018	5	6749
21	2018	6	6099
22	2018	7	6159

- Can we see some seasonality with peaks at specific months?
 - More orders in months of January, March, May etc. i.e., in the first half of the year
 - In later half of the year i.e. in months of July to August (based on data availability in year 2018) , there are consistent good volume of orders
- What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?
 - Most of the purchases are during Morning, Evening and Afternoon.
 - There are few purchases in Early morning as well i.e., between 0th hour and 5th Hour

The screenshot shows the Google Cloud BigQuery interface. On the left, the Explorer sidebar lists various projects and queries, including 'Growing-Orders-By-Month'. The main area displays a SQL query:

```

18 SELECT
19   CASE
20     WHEN x.hour = 6 THEN "DAWN"
21     WHEN x.hour >= 7 AND x.hour <=12 THEN "MORNING"
22     WHEN x.hour >= 12 AND x.hour <=17 THEN "AFTERNOON"
23     WHEN x.hour > 17 AND x.hour <=21 THEN "EVENING"
24     ELSE "NIGHT"
25   END AS purchase_hour_category,
26   COUNT(*) as orders_by_purchase_hour_category
27 FROM (
28   SELECT
29     EXTRACT(HOUR FROM order_purchase_timestamp) AS hour
30     FROM `target-sql-360418.targetAssignment.orders`
31   ) AS x
32   GROUP BY 1
33 ORDER BY orders_by_purchase_hour_category;

```

The 'Query results' section shows the output:

purchase_hour_category	orders_by_purchase_hour_category
DAWN	502
NIGHT	14679
MORNING	21738
EVENING	24161
AFTERNOON	38361

- Evolution of E-commerce orders in the Brazil region

- Get month on month orders by region, states

The screenshot shows the Google Cloud BigQuery interface. On the left, the Explorer sidebar lists various projects and queries, including 'Get month on month orders by region, states'. The main area displays a complex SQL query:

```

1 --Get month on month orders by region, states
2
3 SELECT
4   DISTINCT customer_city,
5   customer_state,
6   EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
7   COUNT(order_id) OVER(PARTITION BY EXTRACT(MONTH FROM order_purchase_timestamp),customer_city,customer_state) AS month_on_month_orders
8   FROM `target-sql-360418.targetAssignment.orders` o
9   JOIN `target-sql-360418.targetAssignment.customers` c ON o.customer_id = c.customer_id

```

The 'Query results' section shows the output:

customer_city	customer_state	month	month_on_month_orders
boa vista	RR	1	768
brasilia	DF	1	974
macapa	AP	1	3349
manaus	AM	1	3418
rio branco	AC	1	4774
santana	AP	1	6620

- How are customers distributed in Brazil

- Major customers are from Sao Paulo, Rio De Janeiro, Belo Horizonte

The screenshot shows the Google Cloud BigQuery interface. The left sidebar lists projects and saved queries, including 'target-sql-360418'. The main area displays a query for 'Customers distributed in ...' with the following SQL code:

```

1 SELECT
2   DISTINCT customer_state,
3   c.customer_city,
4   COUNT(o.customer_id) OVER(PARTITION BY customer_state,c.customer_city) AS customers_by_state_city
5   FROM `target-sql-360418.targetAssignment.orders` o
6   JOIN `target-sql-360418.targetAssignment.customers` c ON o.customer_id = c.customer_id
7   ORDER BY customers_by_state_city DESC
8

```

The results table shows the following data:

customer_state	customer_city	customers_by_state_city
SP	sao paulo	15540
RJ	rio de janeiro	6882
MG	belo horizonte	2773
DF	brasilia	2131
PR	curitiba	1521
ES	campos	1444

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

- Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only)

The screenshot shows the Google Cloud BigQuery interface. The left sidebar lists projects and saved queries, including 'target-sql-360418'. The main area displays a query for 'Increase-in-cost-of-orders-from-2017-2018' with the following SQL code:

```

2 SELECT product_price_2017.product_id,
3   product_price_2017.year,
4   product_price_2017.price AS product_price_2017,
5   product_price_2018.price AS product_price_2018,
6   ROUND((product_price_2018.price - product_price_2017.price)/product_price_2018.price,2) AS
product_price_change_percentage_2018_to_2017
7
8 FROM
9 (SELECT
10   DISTINCT product_id,
11   EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
12   price
13   FROM `target-sql-360418.targetAssignment.orders` o
14   JOIN `target-sql-360418.targetAssignment.order_items` oi ON o.order_id = oi.order_id
15   WHERE EXTRACT(YEAR FROM order_purchase_timestamp) = 2017
16   AND EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 AND 8) AS product_price_2017
17
18 JOIN
19 (SELECT
20   DISTINCT product_id,
21   EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
22   price
23   FROM `target-sql-360418.targetAssignment.orders` o
24   JOIN `target-sql-360418.targetAssignment.order_items` oi ON o.order_id = oi.order_id
25   WHERE EXTRACT(YEAR FROM order_purchase_timestamp) = 2018
26   AND EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 AND 8) AS product_price_2018
27
28 ON product_price_2017.product_id = product_price_2018.product_id
29
30 ORDER BY product_price_change_percentage_2018_to_2017

```

The screenshot shows the Google Cloud BigQuery interface. The left sidebar has an 'Explorer' section with pinned projects like 'Growing-Orders-By-Month' and 'Increase-in-cost-of-orders-from-2017-2018'. The main area shows a query editor with the following SQL code:

```

2 SELECT product_price_2017.product_id,
3       product_price_2017.year,
4       product_price_2017.price AS product_price_2017,
5       product_price_2018.price AS product_price_2018,
6       ROUND((product_price_2018.price - product_price_2017.price)/product_price_2018.price,2) AS
7       product_price_change_percentage_2018_to_2017
8   FROM (
9     SELECT DISTINCT product_id,
10           EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
11           price
12     FROM `target-sql-360418.targetAssignment_orders` o
13   JOIN `target-sql-360418.targetAssignment_order_items` oi ON o.order_id = oi.order_id
14   WHERE EXTRACT(YEAR FROM order_purchase_timestamp) = 2017

```

The results table shows 4898 rows of data with columns: product_id, year, product_price_2017, product_price_2018, and product_price_change_percentage_2018_to_2017.

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

The screenshot shows the Google Cloud BigQuery interface. The left sidebar has an 'Explorer' section with pinned projects like 'Growing-Orders-By-Month' and 'Impact on Economy'. The main area shows a query editor with the following SQL code:

```

52 SELECT
53   c.customer_state, ROUND(AVG(price),2) AS mean_price,ROUND(AVG(freight_value),2) AS mean_freight_value,
54   ROUND(SUM(price),2) AS total_price,ROUND(SUM(freight_value),2) AS total_freight_value
55   FROM `target-sql-360418.targetAssignment_orders` o
56   JOIN `target-sql-360418.targetAssignment_order_items` oi ON o.order_id = oi.order_id
57   JOIN `target-sql-360418.targetAssignment_customers` c ON o.customer_id = c.customer_id
58   GROUP BY c.customer_state;

```

The results table shows 10 rows of data with columns: customer_state, mean_price, mean_freight_value, total_price, and total_freight_value.

- Analysis on sales, freight and delivery time

- Calculate days between purchasing, delivering and estimated delivery

The screenshot shows the Google Cloud BigQuery interface. On the left, the Explorer sidebar lists projects and datasets, including 'target-sql-360418' and its sub-datasets like 'targetAssignment'. In the center, a query editor window displays a SQL query:

```

1 SELECT
2   order_id,
3   DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp), DAY) AS days_between_purchasing_delivering,
4   DATE_DIFF(DATE(order_estimated_delivery_date), DATE(order_purchase_timestamp), DAY) AS days_between_purchasing_estimated_delivering
5
6   FROM `target-sql-360418.targetAssignment.orders`
7 WHERE order_status = 'delivered';

```

The 'RESULTS' tab is selected, showing the query results in a table:

Row	order_id	days_between_purchasing_delivering	days_between_purchasing_estimated_delivering
1	635c894d068a37e6e03dc54e...	31	33
2	3b97562c3ae8bdedcb5c2e45...	33	34
3	8f847f50f04c4cb6774570fde...	30	32
4	2769e9e344d3bf029ff83a161c...	44	40
5	54e1a3c2b97fb0809da548a59...	41	37
6	f1nfafaa101fcaea8n45fkw119a+a5...	97	96

At the bottom right of the results table, it says 'Results per page: 50 1 - 50 of 96478'.

- Create columns:

- `time_to_delivery = order_purchase_timestamp - order_delivered_customer_date`
- `diff_estimated_delivery = order_estimated_delivery_date - order_delivered_customer_date`

The screenshot shows the Google Cloud BigQuery interface. On the left, the Explorer sidebar lists projects and saved queries. In the center, a query editor window displays a SQL query and its results. The query is:

```

1 SELECT
2   order_id,
3   DATE_DIFF(order_delivered_customer_date, DATE(order_purchase_timestamp),DAY) AS time_to_delivery ,
4   DATE_DIFF(order_estimated_delivery_date, DATE(order_delivered_customer_date),DAY) AS diff_estimated_delivery
5
6   FROM `target-sql-360418.targetAssignment.orders`
7 WHERE order_status = 'delivered';
8

```

The results table has columns: Row, order_id, time_to_delivery, and diff_estimated_delivery. The data is:

Row	order_id	time_to_delivery	diff_estimated_delivery
1	635c894d068a37e603dc54...	31	2
2	3b97562c3ee8bdedcb5c2e45...	33	1
3	68f4750f04c4cb774570cd...	30	2
4	2769e344dbfb29f83a161c...	44	-4
5	54e1a3c2b97fb0809da548a59...	41	-4
6	f9bfafad156eef8a46a119a...	97	-1

On the right, there are tabs for 'SAVE RESULTS' and 'EXPLORE DATA'.

- Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

The screenshot shows the Google Cloud BigQuery interface. The query results page displays a grouped query result. The query is:

```

10 SELECT
11   customer_state,
12   ROUND(AVG(freight_value),2) AS mean_freight_value,
13   DATE_DIFF(order_delivered_customer_date, DATE(order_purchase_timestamp),DAY) AS time_to_delivery ,
14   DATE_DIFF(order_estimated_delivery_date, DATE(order_delivered_customer_date),DAY) AS diff_estimated_delivery,
15   FROM `target-sql-360418.targetAssignment.orders` o
16   JOIN `target-sql-360418.targetAssignment.customers` c ON o.customer_id = c.customer_id
17   JOIN `target-sql-360418.targetAssignment.order_items` oi ON o.order_id = oi.order_id
18 WHERE order_status = 'delivered'
19 GROUP BY customer_state,DATE_DIFF(order_delivered_customer_date, DATE(order_purchase_timestamp),DAY),DATE_DIFF(DATE(order_estimated_delivery_date),
20 ,DATE(order_delivered_customer_date),DAY);

```

The results table has columns: Row, customer_state, mean_freight_value, time_to_delivery, and diff_estimated_delivery. The data is:

Row	customer_state	mean_freight_value	time_to_delivery	diff_estimated_delivery
1	GO	21.01	24	10
2	SP	16.32	13	-5
3	RS	24.69	13	13
4	SP	13.43	7	2
5	SP	17.12	12	10
6	SP	10.61	2	6
7	SP	13.37	7	1
8	RA	25.09	22	8

On the right, there are tabs for 'SAVE RESULTS' and 'EXPLORE DATA'.

- Sort the data to get the following:
 - Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5?

DSML Adv: SQL - 04 | Scaler Academy: Acc... | Query results - BigQ... | Query syntax | BigQ... | Business Case: Target... | bigquery create table... | Inbox (2,523) - amino... | +

console.cloud.google.com/bigquery?project=target-sql-360418&ws=1m851m4!4m3!1starget-sql-360418!2stargetAssignment!3order_items!1m4!4m3!1starget-sql-360418!2...

Google Cloud Target-SQL

SANDBOX Set up billing to upgrade to the full BigQuery experience. Learn more

Explorer + ADD DATA

Type to search

Viewing pinned projects.

- Analysis on sales, freight and delivery time
- Customers distributed in Brazil
- Get month on month orders by region, states
- Growing-Orders-By-Month
- Impact on Economy
- Increase-in-cost-of-orders-from-2017-2018
- Increase-in-cost-of-orders-from-2017-2018-usin...
- MOM-Orders-by-region-states
- Month-on-Month-Growth
- Payment type analysis
- Top 5 states where delivery is really fast/ not so...
- Top 5 states with highest/lowest average freight...
- Top 5 states with highest/lowest average time to...
- targetAssignment**
- customers
- geolocation
- order_items
- order_reviews

RUN SAVE SHARE SCHEDULE MORE

```

1 -- Top 5 states with highest average freight value - sort in desc/asc limit 5 (UNION issue) what approach is expected ?
2 SELECT
3   customer_state,
4   mean_freight_value,
5   time_to_delivery ,
6   diff_estimated_delivery
7   FROM
8   (SELECT
9     customer_state,
10    ROUND(AVG(freight_value),2) AS mean_freight_value,
11    DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp),DAY) AS time_to_delivery ,
12    DATE_DIFF(DATE(order_estimated_delivery_date), DATE(order_delivered_customer_date),DAY) AS diff_estimated_delivery
13   FROM `target-sql-360418.targetAssignment.orders` o
14   JOIN `target-sql-360418.targetAssignment.customers` c ON o.customer_id = c.customer_id
15   JOIN `target-sql-360418.targetAssignment.order_items` o1 ON o.order_id = o1.order_id
16   WHERE order_status = 'delivered'
17   GROUP BY customer_state,DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp),DAY),DATE_DIFF(DATE(order_estimated_delivery_date), DATE(order_delivered_customer_date),DAY)) x1
18   ORDER BY mean.freight_value DESC LIMIT 5;
19 
```

Press Alt+F1 for

Query results

JOB INFORMATION				RESULTS		JSON		EXECUTION DETAILS	
Row	customer_state	mean_freight_value	time_to_delivery						diff_estimated_delivery
1	PI	409.68						11	
2	MT	338.3						32	
3	ES	321.88						27	

PERSONAL HISTORY PROJECT HISTORY

DSML Adv: SQL - 04 | Scaler Academy: Acc... | Query results - BigQ... | Query syntax | BigQ... | Business Case: Target... | bigquery create table... | Inbox (2,523) - amino... | +

console.cloud.google.com/bigquery?project=target-sql-360418&ws=1m851m4!4m3!1starget-sql-360418!2stargetAssignment!3order_items!1m4!4m3!1starget-sql-360418!2...

Google Cloud Target-SQL

SANDBOX Set up billing to upgrade to the full BigQuery experience. Learn more

Explorer + ADD DATA

Type to search

Viewing pinned projects.

- Analysis on sales, freight and delivery time
- Customers distributed in Brazil
- Get month on month orders by region, states
- Growing-Orders-By-Month
- Impact on Economy
- Increase-in-cost-of-orders-from-2017-2018
- Increase-in-cost-of-orders-from-2017-2018-usin...
- MOM-Orders-by-region-states
- Month-on-Month-Growth
- Payment type analysis
- Top 5 states where delivery is really fast/ not so...
- Top 5 states with highest/lowest average freight...
- Top 5 states with highest/lowest average time to...
- targetAssignment**
- customers
- geolocation
- order_items
- order_reviews

RUN SAVE SHARE SCHEDULE MORE

```

19 -- Top 5 states with lowest average freight value - sort in desc/asc limit 5 (UNION issue) what approach is expected ?
20 SELECT
21   customer_state,
22   mean_freight_value,
23   time_to_delivery ,
24   diff_estimated_delivery
25   FROM
26   (SELECT
27     customer_state,
28     ROUND(AVG(freight_value),2) AS mean_freight_value,
29     DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp),DAY) AS time_to_delivery ,
30     DATE_DIFF(DATE(order_estimated_delivery_date), DATE(order_delivered_customer_date),DAY) AS diff_estimated_delivery,
31   FROM `target-sql-360418.targetAssignment.orders` o
32   JOIN `target-sql-360418.targetAssignment.customers` c ON o.customer_id = c.customer_id
33   JOIN `target-sql-360418.targetAssignment.order_items` o1 ON o.order_id = o1.order_id
34   WHERE order_status = 'delivered'
35   GROUP BY customer_state,DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp),DAY),DATE_DIFF(DATE(order_estimated_delivery_date), DATE(order_delivered_customer_date),DAY)) x2
36   ORDER BY mean.freight_value ASC LIMIT 5;
37 
```

Press Alt+F1 for

Query results

JOB INFORMATION				RESULTS		JSON		EXECUTION DETAILS	
Row	customer_state	mean_freight_value	time_to_delivery						diff_estimated_delivery
1	AP	0.0						21	
2	GO	0.0						16	
3	CE	0.0						30	

PERSONAL HISTORY PROJECT HISTORY

Sandbox Set up billing to upgrade to the full BigQuery experience. Learn more

Explorer

Type to search

Viewing pinned projects.

- Growing-Orders-By-Month
- Impact on Economy
- Increase-in-cost-of-orders-from-2017-2018
- Increase-in-cost-of-orders-from-2017-2018-using-order...
- MOM-Orders-by-region-states
- Month-on-Month-Growth
- Payment type analysis
- Top 5 states where delivery is really fast/ not so fast c...
- Top 5 states with highest/lowest average freight value
- Top 5 states with highest/lowest average time to delivery
- targetAssignment**
- customers
- geolocation
- order_items
- order_reviews
- orders
- payments
- products

RUN **SAVE** **SHARE** **SCHEDULE** **MORE**

```

1 -- Top 5 states with highest average freight value - sort in desc/asc limit 5 (UNION issue) what approach is expected
2 (SELECT
3     customer_state,
4     mean_freight_value,
5     time_to_delivery ,
6     diff_estimated_delivery
7   FROM
8   (SELECT
9       customer_state,
10      ROUND(AVG(freight_value),2) AS mean_freight_value,
11      DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp),DAY) AS time_to_delivery ,
12      DATE_DIFF(DATE(order_estimated_delivery_date), DATE(order_delivered_customer_date),DAY) AS diff_estimated_delivery
13    FROM `target-sql-360418.targetAssignment.orders` o
14    JOIN `target-sql-360418.targetAssignment.customers` c ON o.customer_id = c.customer_id
15    WHERE order_status = 'delivered'
16    GROUP BY customer_state,DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp),DAY),DATE_DIFF(DATE(order_estimated_delivery_date), DATE(order_delivered_customer_date),DAY)) x1
17    ORDER BY mean_freight_value DESC LIMIT 5)
18 UNION ALL
19 -- Top 5 states with lowest average freight value - sort in desc/asc limit 5 (UNION issue) what approach is expected
20 (SELECT
21     customer_state,
22     mean_freight_value,
23     time_to_delivery ,
24     diff_estimated_delivery
25   FROM
26   (SELECT
27       customer_state,
28

```

Query results

PERSONAL HISTORY PROJECT HISTORY

■ Top 5 states with highest/lowest average time to delivery

Sandbox Set up billing to upgrade to the full BigQuery experience. Learn more

Explorer

Type to search

Viewing pinned projects.

- Project queries
- Analysis on sales, freig...
- Customers distributed ...
- Growing-Orders-By-Mo...
- Impact on Economy
- MOM-Orders-by-regio...
- Month-on-Month-Grow...
- Payment type analysis
- Top 5 states with high...
- Top 5 states with high...
- targetAssignment**
- customers
- geolocation
- order_items
- order_reviews
- orders
- payments
- products

RUN **SAVE** **SHARE** **SCHEDULE** **MORE**

```

2 SELECT
3     customer_state,
4     ROUND(AVG(DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp),DAY)),2) AS avg_time_to_delivery ,
5   FROM `target-sql-360418.targetAssignment.orders` o
6   JOIN `target-sql-360418.targetAssignment.customers` c ON o.customer_id = c.customer_id
7   WHERE order_status = 'delivered'
8   GROUP BY customer_state
9   ORDER BY avg_time_to_delivery DESC
10  LIMIT 5;
11
12 SELECT
13     customer_state,
14     ROUND(AVG(DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp),DAY)),2) AS avg_time_to_delivery ,
15   FROM `target-sql-360418.targetAssignment.orders` o
16   JOIN `target-sql-360418.targetAssignment.customers` c ON o.customer_id = c.customer_id
17   WHERE order_status = 'delivered'
18   GROUP BY customer_state
19   ORDER BY avg_time_to_delivery ASC
20  LIMIT 5;
21

```

Query results

JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS
Row	customer_state	avg_time_to...	
1	RR	29.34	
2	AP	27.18	
3	AM	26.36	

PERSONAL HISTORY PROJECT HISTORY

Google Cloud Target-SQL

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

Explorer + ADD DATA

Type to search

Viewing pinned projects.

- Growing-Orders-By-Month
- Impact on Economy
- Increase-in-cost-of-orders-from-2017-2018
- Increase-in-cost-of-orders-from-2017-2018-using-order
- MOM-Orders-by-region-states
- Month-on-Month-Growth
- Payment type analysis
- Top 5 states where delivery is really fast/ not so fast c...
- Top 5 states with highest/lowest average freight value
- Top 5 states with highest/lowest average time to deliv...

targetAssignment

- customers
- geolocation
- order_items
- order_reviews
- orders
- payments
- products
- all

Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

Row	customer_state	avg_time_to_delivery
1	RR	29.34
2	AP	27.18
3	AM	26.36

PERSONAL HISTORY PROJECT HISTORY

DISMISS UPGR...

```

1 -- Top 5 states with highest/lowest average time to delivery
2 (SELECT
3   customer_state,
4   ROUND(AVG(DATE_DIFF(order_delivered_customer_date), DATE(order_purchase_timestamp),DAY)),2) AS avg_time_to_delivery
5   FROM `target-sql-360418.targetAssignment.orders` o
6   JOIN `target-sql-360418.targetAssignment.customers` c ON o.customer_id = c.customer_id
7   WHERE order_status = 'delivered'
8   GROUP BY customer_state
9   ORDER BY avg_time_to_delivery DESC
10  LIMIT 5)
11 UNION ALL
12 (SELECT
13   customer_state,
14   ROUND(AVG(DATE_DIFF(order_delivered_customer_date), DATE(order_purchase_timestamp),DAY)),2) AS avg_time_to_delivery
15   FROM `target-sql-360418.targetAssignment.orders` o
16   JOIN `target-sql-360418.targetAssignment.customers` c ON o.customer_id = c.customer_id
17   WHERE order_status = 'delivered'
18   GROUP BY customer_state
19   ORDER BY avg_time_to_delivery ASC
20  LIMIT 5)

```

Press Alt+F1 for Accessibility...

SAVE RESULTS EXPLORE DATA

REFRESH

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

Google Cloud Target-SQL

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

Explorer + ADD DATA

Type to search

Viewing pinned projects.

- Project queries
- Analysis on sales, freig...
- Customers distributed ...
- Growing-Orders-By-Mo...
- Impact on Economy
- MOM-Orders-by-regio...
- Month-on-Month-Gro...
- Payment type analysis
- Top 5 states where del...
- Top 5 states with high...

targetAssignment

- customers
- geolocation
- order_items
- order_reviews
- orders
- payments
- all

Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

Row	customer_state	time_to_delivery
1	RJ	0
2	RN	1
3	RJ	1
4	DF	1

PERSONAL HISTORY PROJECT HISTORY

DISMISS UPGR...

```

1 -- Top 5 states where delivery is really fast/ not so fast compared to estimated date
2 SELECT
3   DISTINCT customer_state,
4   DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp),DAY) AS time_to_delivery ,
5   FROM `target-sql-360418.targetAssignment.orders` o
6   JOIN `target-sql-360418.targetAssignment.customers` c ON o.customer_id = c.customer_id
7   WHERE order_status = 'delivered' AND DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp),DAY) IS NOT NULL
8   ORDER BY time_to_delivery ASC
9  LIMIT 5;
10
11 SELECT
12   DISTINCT customer_state,
13   DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp),DAY) AS time_to_delivery ,
14   FROM `target-sql-360418.targetAssignment.orders` o
15   JOIN `target-sql-360418.targetAssignment.customers` c ON o.customer_id = c.customer_id
16   WHERE order_status = 'delivered' AND DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp),DAY) IS NOT NULL
17   ORDER BY time_to_delivery DESC
18  LIMIT 5;

```

Press Alt+F1 for Accessibility...

SAVE RESULTS EXPLORE DATA

REFRESH

```

1 --Top 5 states where delivery is really fast compared to estimated date
2 (SELECT
3   | DISTINCT customer_state,
4   | DATE_DIFF(order.delivered_customer_date), DATE(order.purchase_timestamp).DAY AS time_to_delivery ,
5   | FROM `target-sql-360418.targetAssignment.orders` o
6   | JOIN `target-sql-360418.targetAssignment.customers` c ON o.customer_id = c.customer_id
7   | WHERE order_status = "delivered" AND DATE_DIFF(DATE(order.delivered_customer_date), DATE(order.purchase_timestamp).DAY) IS NOT NULL
8   | ORDER BY time_to_delivery ASC
9   | LIMIT 5)
10 UNION ALL
11 --Top 5 states where delivery is not so fast compared to estimated date
12 (SELECT
13   | DISTINCT customer_state,
14   | DATE_DIFF(order.delivered_customer_date), DATE(order.purchase_timestamp).DAY AS time_to_delivery ,
15   | FROM `target-sql-360418.targetAssignment.orders` o
16   | JOIN `target-sql-360418.targetAssignment.customers` c ON o.customer_id = c.customer_id
17   | WHERE order_status = "delivered" AND DATE_DIFF(DATE(order.delivered_customer_date), DATE(order.purchase_timestamp).DAY) IS NOT NULL
18   | ORDER BY time_to_delivery DESC
19   | LIMIT 5)

```

Query results

customer_state	time_to_delivery
RJ	0
RN	1
RJ	1
DN	1

- **Payment type analysis**

- Month over Month count of orders for different payment types

```

1 SELECT
2   | DISTINCT payment_type,EXTRACT(MONTH FROM order.purchase_timestamp) AS month,
3   | COUNT(*) OVER(PARTITION BY payment_type,EXTRACT(MONTH FROM order.purchase_timestamp)) AS counts
4   | FROM `target-sql-360418.targetAssignment.orders` o
5   | JOIN `target-sql-360418.targetAssignment.payments` p ON o.order_id = p.order_id
6   | ORDER BY month,counts;
7
8

```

Query results

payment_type	month	counts
debit_card	1	118
voucher	1	477
UPI	1	1715
credit_card	1	6103
debit_card	2	82
voucher	2	424
UPI	2	1723
credit_card	2	6609
debit_card	3	109

- Distribution of payment instalments and count of orders

The screenshot shows the Google Cloud BigQuery interface. On the left, there's a sidebar with pinned projects: Analysis on sales, freight, Customers distributed in..., Growing-Orders-By-Month, Impact on Economy, MOM-Orders-by-region-st..., Month-on-Month-Growth, Payment type analysis, targetAssignment (customers, geolocation, order_items, order_reviews, orders, payments), products, and sellers. The main area has tabs for geolocation, Editor, *Analysis on sales, freight ..., orders, Editor 3, and Payment type analysis. A query is running, and the results are displayed in a table:

Row	payment_in...	counts
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	539

- **Actionable Insights**

- There is a growing trend of e-commerce from 2016 to 2018
- There is consistent order volume across the months
- Highest number of orders placed during afternoon (highest), Evening and morning (in the order of sequence)
- There are few orders placed during Dawn, it will be interesting to see if orders placed during Dawn are from specific customers, region or for specific product categories
- Major orders are placed from city Sao Paulo, Rio De Janeiro, Belo Horizonte
- Actual delivery is faster than estimated time of delivery
- TOP 5 states which has highest freight values, have time to delivery of max 33 days
- RR, AP, AM, AL and PA are top 5 states where average delivery time is highest
- SP, PR, MG, DF, SC are top 5 states where average delivery time is lowest
- Delivery time is fastest in states such as RJ, RN, DF
- Delivery time is slowest in states such as ES, RJ, PA, PI etc.
- Most of the payments are via credit card (highest), UPI and voucher(lowest) (in order of their sequence)

- **Recommendations**

- There should more promotions in the regions of fastest delivery with committed delivery time to raise sales (e.g., states SP, PR, MG, DF, SC etc.)
- More promotions during Afternoon and Evening as there are most of the orders being placed or early morning sales for customers to boost sales during off hours
- Season based promotions during festive season such December, January or in November (during Nation Holidays) or in other regional Carnival days
- Special promotions during Father's Day (in August), mothers' day (in May)

- Product based marketing can be another option based on customer profile or customer geo locations.
- Wallet based cashbacks can be offered for customers to minimize credit /UPI transaction costs.
- As there is a growing trend, social media marketing channels can be targeted or referral system can be introduced to avail discounts on purchases.