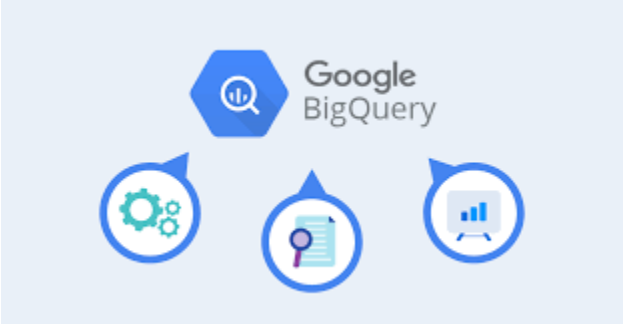
Business Case: Target SQL



About Target SQL:-

● *Target is a globally renowned brand and a prominent retailer in the United States. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can Deliver.*

● *This particular business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.*

● *By analyzing this extensive dataset, it becomes possible to gain valuable insights into Target's operations in Brazil. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels.*

Problem Statement:
Assuming you are a data analyst/ scientist at Target, you have been assigned the task of analyzing the given dataset to extract valuable insights and provide actionable recommendations.

Dataset:

1. ***customers.csv***
2. ***sellers.csv***

3. *order_items.csv*
4. *geolocation.csv*
5. *payments.csv*
6. *reviews.csv*
7. *orders.csv*
8. *products.csv*

What does 'good' look like?

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

A. Data type of all columns in the "customers" table.

```
SELECT
column_name,
data_type,
FROM `target_business_case`.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = "customers"
```

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

Insides:- In this Question I'm checking the data types to related customer then i took my project name with customer column then i did apply where condition because wanted take the customer details. So we can check the details in the table.

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

```
SELECT MIN(order_purchase_timestamp) AS first_order_time,
MAX(order_purchase_timestamp) AS last_order_time
FROM `target_business_case.orders`
```

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

Insides :- In this query im checking the in what time order were placed. I wrote after the select t mentioned the min and max and calling from the with my project name and with that column name i wanted to get the details. So we can check the details in the table.

C. Count the number of Cities and States in our dataset.

```
SELECT COUNT(DISTINCT geolocation_city) AS city_count, COUNT(DISTINCT
geolocation_state) AS state_count
FROM `target_business_case.geolocation`;
```

Row	city_count	state_count
1	8011	27

Insides :- In the query i'm trying the get the details. How many city and states are means total numbers of city and states i took count function then in the From statement i mentioned table name.so we can in the table.

II. In-depth Exploration:

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

```
SELECT
EXTRACT(YEAR FROM order_delivered_customer_date) AS year,
EXTRACT(MONTH FROM order_delivered_customer_date) AS month,
COUNT(*) AS order_count
FROM `target_business_case.orders`
GROUP BY
year, month
ORDER BY
year, month;
```

Row	year	month	order_count
1	null	null	2965
2	2016	10	208
3	2016	11	60
4	2016	12	4
5	2017	1	283
6	2017	2	1351
7	2017	3	2382
8	2017	4	1849
9	2017	5	3751
10	2017	6	3223
11	2017	7	3455
12	2017	8	4302

Insides :- In this query want to get the trend is growing in which year and how many years and in how many months and how many times. So we can check the details as given in table.

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

```
SELECT
  EXTRACT(MONTH FROM order_delivered_customer_date) AS month,
  COUNT(*) AS order_count
FROM
  `target_business_case.orders`
GROUP BY
  month
ORDER BY
  order_count DESC;
```

Row	month	order_count
1	8	12616
2	5	10862
3	6	10052
4	4	9699
5	7	9294
6	3	9207
7	12	7209
8	2	7201

Insides : - In this query wanted to check the seasonality in terms of the orders being placed so here we can check how many months is there and order count in the table.

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

- **0-6 hrs : Dawn**
- **7-12 hrs : Mornings**
- **13-18 hrs : Afternoon**
- **19-23 hrs : Night**

```
SELECT
  CASE
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 0 AND 6 THEN
'Dawn'
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 7 AND 12 THEN
'Morning'
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 13 AND 18 THEN
'Afternoon'
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 19 AND 23 THEN
'Night'
    ELSE 'Other'
  END AS order_time_of_day,
  COUNT(*) AS order_count
FROM

  `target_business_case.orders`

GROUP BY
  order_time_of_day
ORDER BY
  order_count DESC;
```

Row	order_time_of_day	order_count
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

Insides :- i want to categorize the hours of a day into the given time brackets/intervals and find out during which intervals the Brazilian customers usually order the most.so in the table we can check the all details.

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

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

```
SELECT
g.geolocation_state,
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
COUNT(o.order_id) AS order_count
FROM
    `target_business_case.geolocation` AS g
INNER JOIN
    `target_business_case.customers` AS c
ON
    g.geolocation_zip_code_prefix = c.customer_zip_code_prefix
INNER JOIN
    `target_business_case.orders` AS o
ON
    o.customer_id = c.customer_id
GROUP BY
    g.geolocation_state,
    year,
    month
ORDER BY
    g.geolocation_state,
    year,
    Month;
```

Row	geolocation_state	year	month	order_count
1	AC	2017	1	45
2	AC	2017	2	179
3	AC	2017	3	329
4	AC	2017	4	362
5	AC	2017	5	886
6	AC	2017	6	432
7	AC	2017	7	605
8	AC	2017	8	657
9	AC	2017	9	161
10	AC	2017	10	535

Insides :- in want to get the no. of orders placed in each state, in each month by customers with oder count and all the details given in the table.

B. How are the customers distributed across all the states?

```
SELECT
  g.geolocation_state,
  COUNT(DISTINCT c.customer_id) AS unique_customers_count
FROM
  `target_business_case.geolocation` g
INNER JOIN `target_business_case.customers` c
ON g.geolocation_zip_code_prefix=c.customer_zip_code_prefix
GROUP BY
  g.geolocation_state
ORDER BY
  unique_customers_count DESC
```

Row	geolocation_state	unique_customers_c
1	SP	41731
2	RJ	12839
3	MG	11624
4	RS	5473
5	PR	5034
6	SC	3651
7	BA	3371
8	ES	2027
9	GO	2011
10	DF	1974

Insides :- I want to get the no. of unique customers present in each state. And how many uniques customers is there so we can check the as in given table.

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

A. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

```
WITH total_payment AS
(
SELECT EXTRACT(YEAR FROM o.order_purchase_timestamp) AS payment_year,
SUM(p.payment_value) AS summed_payment
FROM `target_business_case.orders` o
INNER JOIN `target_business_case.payments` p
ON o.order_id=p.order_id
WHERE EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 AND 8 AND
EXTRACT(YEAR FROM order_purchase_timestamp) IN (2017, 2018)
GROUP BY payment_year
),
payment_of_2018 AS
(
SELECT summed_payment FROM total_payment WHERE payment_year = 2018
),
payment_of_2017 AS
(
SELECT summed_payment FROM total_payment WHERE payment_year = 2017
)
SELECT DISTINCT(ROUND((((SELECT summed_payment FROM payment_of_2018) -
```

```
(SELECT summed_payment from payment_of_2017))/(SELECT summed_payment FROM
payment_of_2017)) * 100, 2)) AS percentage_increase FROM total_payment
```

Row	percentage_increase
1	136.98

Insides :- payment_value column in the payments table to get the cost of orders. I was trying the check the increasing percents of the cost of orders.

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

```
SELECT
g.geolocation_state AS state,
SUM(o.price) AS total_price,
AVG(o.price) AS average_price
FROM
`target_business_case.sellers` AS s
JOIN
`target_business_case.geolocation` AS g ON s.seller_zip_code_prefix =
g.geolocation_zip_code_prefix
join
`target_business_case.order_items` AS o on o.seller_id = s.seller_id

GROUP BY
state
```

Row	state	total_price	average_price
1	SE	77088.69999999...	190.3424691358...
2	PI	10088.0	210.1666666666...
3	AM	31779.0	392.3333333333...
4	AC	43788.0	267.0
5	RO	628875.6799999...	369.0584976525...
6	BA	23385841.44995...	351.6086278961...
7	CE	740073.6300000...	246.1169371466...
8	DF	4674257.660001...	72.55910679912...
9	ES	3211486.259999...	127.3085808293...
10	GO	4444926.589999...	164.1284465696...
11	MA	3604486.050003...	89.89864197530...
12	MG	253958733.4263...	122.9735643384...
13	MS	724755.2400000...	165.4314631362...
14	MT	961121.8000000...	116.8537142857...
15	PB	852601.9000000...	352.3148347107...

Insides :- I want to fetch the total price and the average price of orders for each state. I did the calculate the total & average value of order price for each state and we can check the tables.

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

```
SELECT
g.geolocation_state AS state,
SUM(o.freight_value) AS total_freight,
AVG(o.freight_value) AS average_freight
FROM
`target_business_case.sellers` AS s
JOIN
`target_business_case.geolocation` AS g ON s.seller_zip_code_prefix =
g.geolocation_zip_code_prefix
join
`target_business_case.order_items` AS o on o.seller_id = s.seller_id

GROUP BY
State
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	state	total_freight	average_freight
1	SE	11798.20000000...	29.13135802469...
2	PI	1773.280000000...	36.94333333333...
3	AM	2208.599999999...	27.26666666666...
4	AC	5385.760000000...	32.84
5	RO	85745.36000000...	50.32004694835...
6	BA	1939324.409999...	29.15794996316...
7	CE	163715.9700000...	54.44495177918...
8	DF	1223546.710000...	18.99327398323...
9	ES	724107.3200000...	28.70480139538...
10	GO	694619.5200000...	25.64875267705...
11	MA	1201987.709999...	29.97849382716...
12	MG	47130618.12004...	22.82189717061...
13	MS	113813.8699999...	25.97897055466...
14	MT	263738.1499999...	32.06542857142...

Insides :- I want to fetch the total freight value and the average freight value of orders for each state. I did the calculate the total &aAverage value of order freight for each state and we can check the tables.

V. Analysis based on sales, freight and delivery time.

A. Find the no. of days taken to deliver each order from the order's purchase date as delivery time. Also, calculate the difference (in days) between the estimated & actual delivery date of an order. Do this in a single query.

```
SELECT
o.order_id,
DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY) AS
delivery_time,
DATE_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date,
DAY) AS diff_estimated_delivery
FROM
`target_business_case.orders` AS o
```

Row	order_id	delivery_time	diff_estimated_delive
1	1950d777989f6a877539f5379...	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28
3	65d1e226dfaeb8cdc42f66542...	35	16
4	635c894d068ac37e6e03dc54e...	30	1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	1
7	276e9ec344d3bf029ff83a161c...	43	-4
8	54e1a3c2b97fb0809da548a59...	40	-4
9	fd04fa4105ee8045f6a0139ca5...	37	-1
10	302bb8109d097a9fc6e9cefc5...	33	-5
11	66057d37308e787052a32828...	38	-6
12	19135c945c554eebfd7576c73...	36	-2

Insides :- The delivery time and the difference between the estimated & actual delivery date checking the given by formula. Finding the no. of days taken to deliver each order from the order's purchase date as delivery time.
Also, calculating the difference (in days) between the estimated & actual delivery date of an order. All details given in the table.

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

```
WITH state_freight AS (
SELECT
g.geolocation_state AS state,
AVG(o.freight_value) AS average_freight
FROM
`target_business_case.geolocation` AS g
JOIN
`target_business_case.sellers` AS s ON
g.geolocation_zip_code_prefix=s.seller_zip_code_prefix
JOIN `target_business_case.order_items` as o ON o.seller_id=s.seller_id
GROUP BY
state
)
SELECT
```

```
state,
average_freight
FROM (
SELECT
state,
average_freight
FROM
State_freight
ORDER BY
average_freight DESC
LIMIT
5
) top_states
UNION ALL
SELECT
state,
average_freight
FROM (
SELECT
state,
average_freight
FROM
state_freight
ORDER BY
average_freight ASC
LIMIT
5
) bottom_states
ORDER BY
average_freight ASC
```

Row	state	average_freight
1	RN	15.931877663574827
2	SP	18.436612450600478
3	RJ	18.9323516999635
4	DF	18.993273983235081
5	PR	22.107301336275114
6	AC	32.84
7	PB	34.694099173553667
8	PI	36.943333333333335
9	RO	50.320046948356847
10	CE	54.4449517791819

Insides :- I want to find the top 5 & the bottom 5 states arranged in increasing order of the average freight value. So Finding out the top 5 states with the highest & lowest average freight value. In the given table we can check the results.

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

```
WITH state_delivery_time AS (
SELECT
g.geolocation_state AS state,
AVG(DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
DAY)) AS average_delivery_time
FROM
`target_business_case.customers` AS c
JOIN
`target_business_case.geolocation` AS g ON c.customer_zip_code_prefix =
g.geolocation_zip_code_prefix
JOIN
`target_business_case.orders` as o on o.customer_id = c.customer_id

GROUP BY
state
)
SELECT
```

```
state,
average_delivery_time
FROM (
SELECT
state,
average_delivery_time
FROM
state_delivery_time
ORDER BY
average_delivery_time DESC
LIMIT
5
) top_states
UNION ALL
SELECT
state,
average_delivery_time
FROM (
SELECT
state,
average_delivery_time
FROM
State_delivery_time
ORDER BY
average_delivery_time ASC
LIMIT
5
) bottom_states
ORDER BY
average_delivery_time ASC
```

Row	state	average_delivery_time
1	SP	8.4705297141904747
2	PR	11.038764047705925
3	MG	11.418626834392896
4	DF	12.496517892339325
5	SC	14.494308328176698
6	PA	22.550239824416419
7	AL	23.143527892713742
8	RR	24.520601336302882
9	AM	24.651196784213447
10	AP	27.991226237727222

Insides :- I want to find the top 5 & the bottom 5 states arranged in increasing order of the average delivery time. So Finding out the top 5 states with the highest & lowest average delivery value. In the given table we can check the results.

D. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

```
WITH state_delivery_speed AS (
SELECT
g.geolocation_state AS state,
AVG(o.order_estimated_delivery_date - o.order_purchase_timestamp) -
AVG(o.order_delivered_customer_date - o.order_purchase_timestamp) AS
delivery_speed
FROM
`target_business_case.customers` AS c
JOIN
`target_business_case.geolocation` AS g ON c.customer_zip_code_prefix =
g.geolocation_zip_code_prefix
INNER JOIN
`target_business_case.orders` AS o ON o.customer_id = c.customer_id
WHERE
o.order_status = 'delivered'
```

```
GROUP BY
state
)
SELECT
state,
delivery_speed
FROM
state_delivery_speed
ORDER BY
delivery_speed DESC
LIMIT
5
```

Row	state	delivery_speed
1	RR	0-0 0 495:11:0.880289532
2	AM	0-0 0 489:12:7.967293988
3	RO	0-0 0 453:54:15.910966288
4	AC	0-0 0 449:21:29.320441254
5	AP	0-0 0 444:13:16.501984541

Insides :- Including the orders. I'm trying to Finding out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
So i'm differentiatng the averages of actual & estimated delivery
date to figure out how fast the delivery was for each state. So we can see the table.

VI. Analysis based on the payments:

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

```
SELECT
EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
p.payment_type,
COUNT(o.order_id) AS order_count
FROM
`target_business_case.orders` AS o
JOIN
`target_business_case.payments` AS p ON o.order_id = p.order_id
```

```
GROUP BY
year, month, p.payment_type
ORDER BY
year, month;
```

Row	year	month	payment_type	order_count
1	2016	9	credit_card	3
2	2016	10	credit_card	254
3	2016	10	UPI	63
4	2016	10	voucher	23
5	2016	10	debit_card	2
6	2016	12	credit_card	1
7	2017	1	credit_card	583
8	2017	1	UPI	197
9	2017	1	voucher	61
10	2017	1	debit_card	9
11	2017	2	credit_card	1356
12	2017	2	UPI	398
13	2017	2	voucher	119
14	2017	2	debit_card	13
15	2017	3	UPI	590
16	2017	3	credit_card	2016
17	2017	3	voucher	200
18	2017	3	debit_card	31
19	2017	4	voucher	202
20	2017	4	credit_card	1846
21	2017	4	UPI	496
22	2017	4	debit_card	27

Insides :- I to count the no. of orders placed using different payment methods in each month over the past years. Finding the on month no. of orders placed with different payment types. So with the help of join i did and we can see the results in the table.

B. Find the no. of orders placed on the basis of the payment installments that have

been paid.

```sql
SELECT
payment_installments,
COUNT(DISTINCT o.order_id) AS order_count
FROM
`target_business_case.orders` AS o
JOIN
`target_business_case.payments` AS p ON o.order_id = p.order_id
WHERE
p.payment_sequential = 1
GROUP BY
payment_installments
ORDER BY
payment_installments
```

Row	payment_installment	order_count
1	1	48236
2	2	12360
3	3	10422
4	4	7066
5	5	5221
6	6	3904
7	7	1619
8	8	4242
9	9	644
10	10	5305
11	11	23
12	12	133
13	13	16
14	14	15

Insides :- I want to count the no. of orders placed based on the no. of payment installments at least one installment has been successfully paid. Finding the no. of orders placed on the basis of the payment installments that have been paid. So i called the tables and mentioned the columns names there and took the obj payment_sequential so in this table we can get the details of payments and orders count.