SQL

Business Case

Context:

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 analysing 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.

Dataset: https://drive.google.com/drive/folders/1TGEc66YKbD443nslRi1bWgVd238gJCnb
The data is available in 8 csv files:

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

The column description for these csv files is given below.

The customers.csv contain following features:

Features	Description
customer_id	ID of the consumer who made the purchase
customer_unique_id	Unique ID of the consumer
customer_zip_code_prefix	Zip Code of consumer's location
customer_city	Name of the City from where order is made
customer_state	State Code from where order is made (Eg. são paulo - SP)

The sellers.csv contains following features:

Features	Description
seller_id	Unique ID of the seller registered
seller_zip_code_prefix	Zip Code of the seller's location
seller_city	Name of the City of the seller
seller_state	State Code (Eg. são paulo - SP)

The order_items.csv contain following features:

Features	Description
order_id	A Unique ID of order made by the consumers
order_item_id	A Unique ID given to each item ordered in the order
product_id	A Unique ID given to each product available on the site
seller_id	Unique ID of the seller registered in Target
shipping_limit_date	The date before which shipping of the ordered product must be completed
price	Actual price of the products ordered
freight_value	Price rate at which a product is delivered from one point to another

The geolocations.csv contain following features:

Features	Description
geolocation_zip_code_prefix	First 5 digits of Zip Code

geolocation_lat	Latitude
geolocation_lng	Longitude
geolocation_city	City
geolocation_state	State

The payments.csv contain following features:

Features	**Description**
order_id	A Unique ID of order made by the consumers.
payment_sequential	Sequences of the payments made in case of EMI
payment_type	Mode of payment used (Eg. Credit Card)
payment_installments	Number of installments in case of EMI purchase.
payment_value	Total amount paid for the purchase order

The orders.csv contain following features:

Features	**Description**
order_id	A Unique ID of order made by the consumers
customer_id	ID of the consumer who made the purchase
order_status	Status of the order made i.e. delivered, shipped, etc.
order_purchase_timestamp	Timestamp of the purchase
order_delivered_carrier_date	Delivery date at which carrier made the delivery
order_delivered_customer_date	Date at which customer got the product
order_estimated_delivery_date	Estimated delivery date of the products

The reviews.csv contain following features:

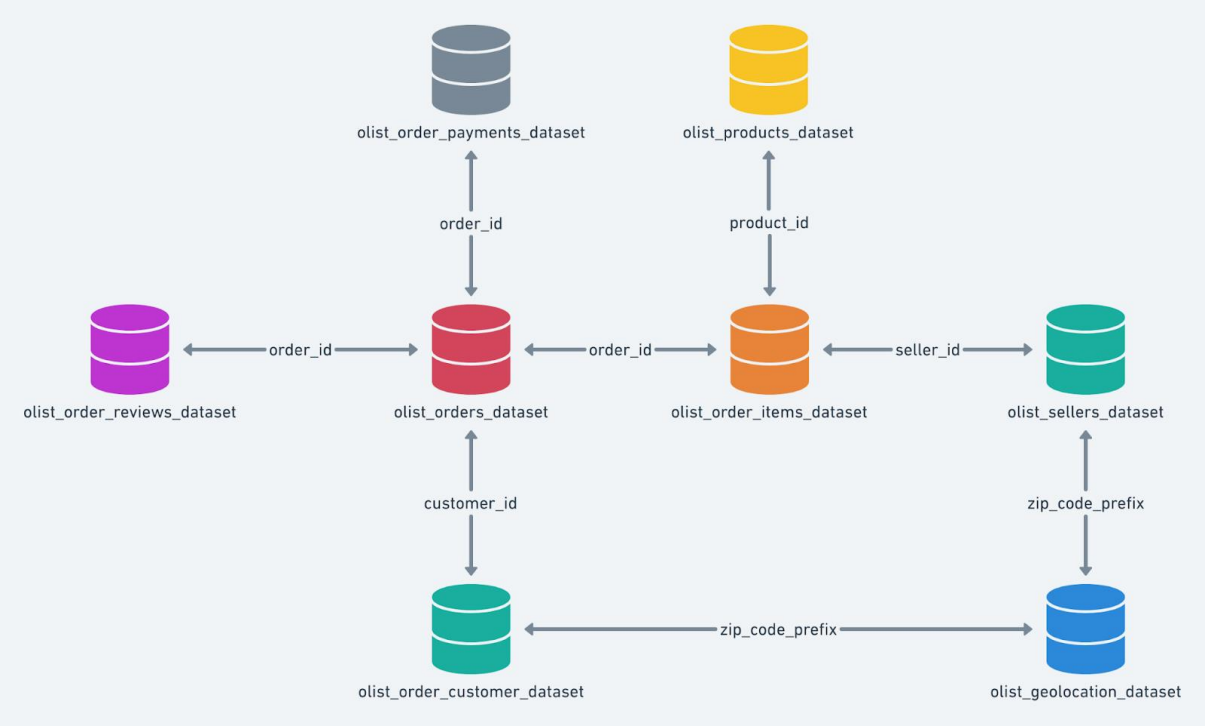
Features	**Description**
review_id	ID of the review given on the product ordered by the order id
order_id	A Unique ID of order made by the consumers
review_score	Review score given by the customer for each order on a scale of 1-5
review_comment_title	Title of the review
review_comment_message	Review comments posted by the consumer for each order
review_creation_date	Timestamp of the review when it is created
review_answer_timestamp	Timestamp of the review answered

The products.csv contain following features:

Features	**Description**
product_id	A Unique identifier for the proposed project.
product_category_name	Name of the product category
product_name_lenght	Length of the string which specifies the name given to the products ordered

product_description_lenght	Length of the description written for each product ordered on the site.
product_photos_qty	Number of photos of each product ordered available on the shopping portal
product_weight_g	Weight of the products ordered in grams
product_length_cm	Length of the products ordered in centimeters
product_height_cm	Height of the products ordered in centimeters
product_width_cm	Width of the product ordered in centimeters

Dataset schema:



Problem Statement:

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

What does 'good' look like?

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

1. Data type of all columns in the “customers” table

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

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

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

```
SELECT MIN(order_purchase_timestamp) AS first_order_placed_date,
MAX(order_purchase_timestamp) AS last_order_placed_date
FROM target_retailer.orders
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

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

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

```
SELECT COUNT (DISTINCT geolocation_city) AS number_of_cities, COUNT(DISTINCT
geolocation_state) AS number_of_states
FROM target_retailer.geolocation
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	number_of_cities	number_of_states
1	8011	27

2. In-depth Exploration:

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

```
SELECT
EXTRACT(year FROM order_purchase_timestamp) AS year,
FORMAT_DATE('%b', order_purchase_timestamp) AS month,
COUNT(order_purchase_timestamp) AS total_orders
FROM target_retailer.orders
GROUP BY month,year
ORDER BY year,month ;
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	year	month	total_orders
1	2016	Dec	1
2	2016	Oct	324
3	2016	Sep	4
4	2017	Apr	2404
5	2017	Aug	4331

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

```
SELECT
 FORMAT_DATE('%b', order_purchase_timestamp) AS month,
 EXTRACT(month FROM order_purchase_timestamp) AS mon,
 COUNT(order_purchase_timestamp) AS Total_orders
 FROM target_retailer.orders
 GROUP BY mon,month
 ORDER BY mon;
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	month	mon	Total_orders
1	Jan	1	8069
2	Feb	2	8508
3	Mar	3	9893
4	Apr	4	9343
5	May	5	10573

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

```
SELECT COUNT(*) AS number_of_orders,
  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 'Mornings'
    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'
  END AS ordering_time
FROM target_retailer.orders
GROUP BY ordering_time
ORDER BY number_of_orders DESC
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	number_of_orders	ordering_time
1	38135	Afternoon
2	28331	Night
3	27733	Mornings
4	5242	Dawn

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

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

```
SELECT
EXTRACT(MONTH from order_purchase_timestamp) as  Month,
EXTRACT(YEAR from order_purchase_timestamp)  as  Year,
COUNT(ord.order_id) AS Number_of_Orders,
cus.customer_state
FROM target_retailer.orders AS ord
JOIN target_retailer.customers AS cus
ON cus.customer_id = ord.customer_id
GROUP BY Month, Year,customer_state
ORDER BY Year, Month,customer_state;
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	Month	Year	Number_of_Orders	customer_state
1	9	2016		RR
2	9	2016	1	RS
3	9	2016	2	SP
4	10	2016	2	AL
5	10	2016	4	BA

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

```
SELECT customer_state, COUNT(DISTINCT customer_id) AS number_of_customers
FROM target_retailer.customers
GROUP BY customer_state
ORDER BY COUNT(DISTINCT customer_id)
LIMIT 10
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	customer_state	number_of_customer
1	RR	46
2	AP	68
3	AC	81
4	AM	148
5	RO	253
6	TO	280
7	SE	350
8	AL	413
9	RN	485

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

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

```
SELECT Year ,orders_cost,
ROUND((orders_cost -lag(orders_cost) over (order by Year ) ) *
100/LAG(orders_cost) OVER (ORDER BY Year ),2) AS per_change_orders_cost
FROM
    (SELECT
       Extract(YEAR FROM order_purchase_timestamp)  AS  Year,
       ROUND(SUM(payment_value),2) AS orders_cost
       FROM target_retailer.orders AS o
       INNER JOIN
       target_retailer.payments AS p
       ON o.order_id= p.order_id
       WHERE EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 AND 8
       GROUP BY Year)
       ORDER BY Year;
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	Year	orders_cost	per_change_orders_c
1	2017	3669022.12	null
2	2018	8694733.84	136.98

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

```
SELECT c.customer_state, ROUND(SUM(oitem.price),2) AS total_price,
ROUND(AVG(oitem.price),2) AS average_price FROM
target_retailer.order_items oitem JOIN target_retailer.orders o
ON oitem.order_id = o.order_id
JOIN target_retailer.customers c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
LIMIT 10
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	customer_state	total_price	average_price
1	SP	5202955.05	109.65
2	RJ	1824092.67	125.12
3	PR	683083.76	119.0
4	SC	520553.34	124.65
5	DF	302603.94	125.77
6	MG	1585308.03	120.75
7	PA	178947.81	165.69
8	BA	511349.99	134.6
9	GO	294591.95	126.27

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

```
SELECT c.customer_state, ROUND(SUM(oitem.freight_value),2) AS
total_freight_value, ROUND(AVG(oitem.freight_value),2) AS
average_freight_value FROM
target_retailer.order_items oitem JOIN target_retailer.orders o
ON oitem.order_id = o.order_id
JOIN target_retailer.customers c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
LIMIT 10
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	customer_state	total_freight_value	average_freight_valu
1	SP	718723.07	15.15
2	RJ	305589.31	20.96
3	PR	117851.68	20.53
4	SC	89660.26	21.47
5	DF	50625.5	21.04
6	MG	270853.46	20.63
7	PA	38699.3	35.83
8	BA	100156.68	26.36
9	GO	53114.98	22.77

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

1. 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.

You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

- time_to_deliver = order_delivered_customer_date - order_purchase_timestamp
- diff_estimated_delivery = order_estimated_delivery_date - order_delivered_customer_date

```
SELECT
order_id,
order_purchase_timestamp,
order_delivered_customer_date ,
order_estimated_delivery_date,
ABS(DATE_DIFF( order_delivered_customer_date,
order_purchase_timestamp,DAY)) AS time_to_deliver,
ABS(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_d
ate, DAY)) AS diff_estimated_delivery
FROM `may23-387016.Target_Project.orders` as o
WHERE order_status='delivered'
LIMIT 10;
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	order_id	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	time_to_deliver	diff_estimated_delivery
1	c158e9806f85a...	2017-04-14 22:06:32 UTC	2017-05-08 11:10:26 UTC	2017-05-18 00:00:00 UTC	23	9
2	b60b53ad0bb7d...	2017-05-10 14:03:27 UTC	2017-05-23 13:12:27 UTC	2017-05-18 00:00:00 UTC	12	5
3	c830f223aae08...	2017-04-22 15:50:30 UTC	2017-05-05 13:27:50 UTC	2017-05-18 00:00:00 UTC	12	12
4	a8aa2cd070eea...	2017-05-09 17:42:45 UTC	2017-05-16 23:22:20 UTC	2017-05-18 00:00:00 UTC	7	1
5	813c55ce9b6ba...	2017-04-26 01:01:39 UTC	2017-05-08 08:54:36 UTC	2017-05-18 00:00:00 UTC	12	9
6	44558a1547e44...	2017-05-10 20:47:02 UTC	2017-05-12 17:00:05 UTC	2017-05-18 00:00:00 UTC	1	5
7	036b791897847...	2017-05-10 15:34:59 UTC	2017-05-17 11:14:40 UTC	2017-05-18 00:00:00 UTC	6	0
8	1aba60c04110b...	2017-04-18 21:20:40 UTC	2017-05-10 11:50:00 UTC	2017-05-18 00:00:00 UTC	21	7
9	0312ecf90786d...	2017-05-10 22:02:40 UTC	2017-05-18 17:09:46 UTC	2017-05-18 00:00:00 UTC	7	0
10	635c894d068ac...	2017-04-15 15:37:38 UTC	2017-05-16 14:49:55 UTC	2017-05-18 00:00:00 UTC	30	1

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

```
WITH state_freight_value
AS (
SELECT
customer_state AS state,
ROUND(AVG(freight_value),2) AS avg_freight_value ,
FROM target_retailer.orders AS o
INNER JOIN
target_retailer.customers AS c
ON o.customer_id= c.customer_id
INNER JOIN
target_retailer.order_items AS oi
ON o.order_id=oi.order_id
GROUP BY customer_state)
```

```
SELECT state,state_fv_rank ,avg_freight_value FROM (
SELECT state,avg_freight_value,
DENSE_RANK() OVER( ORDER BY avg_freight_value DESC ) AS state_fv_rank
FROM state_freight_value ) a
WHERE state_fv_rank<=5


UNION ALL

SELECT state,state_fv_rank ,avg_freight_value FROM (
SELECT state,avg_freight_value,
DENSE_RANK() OVER( ORDER BY avg_freight_value) AS state_fv_rank
FROM state_freight_value ) a
WHERE state_fv_rank<=5
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	state	state_fv_rank	avg_freight_value
1	SP	1	15.15
2	PR	2	20.53
3	RJ	4	20.96
4	DF	5	21.04
5	MG	3	20.63
6	RR	1	42.98
7	PI	5	39.15
8	PB	2	42.72
9	AC	4	40.07
10	RO	3	41.07

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

```
WITH state_delivery_time
AS (
SELECT
customer_state AS state,
ROUND(AVG(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp,DAY)),2) AS avg_delivery_time ,
FROM target_retailer.orders AS o
INNER JOIN
target_retailer.customers AS c
ON o.customer_id= c.customer_id
INNER JOIN
target_retailer.order_items AS oi
ON o.order_id=oi.order_id
GROUP BY customer_state)

SELECT state,state_delivery_time_rank ,avg_delivery_time FROM (
SELECT state,avg_delivery_time,
```

```
DENSE_RANK() OVER( ORDER BY avg_delivery_time DESC ) AS
state_delivery_time_rank
FROM state_delivery_time ) a
WHERE state_delivery_time_rank<=5

UNION ALL

SELECT state,state_delivery_time_rank ,avg_delivery_time FROM (
SELECT state,avg_delivery_time,
DENSE_RANK() OVER(ORDER BY avg_delivery_time) AS
state_delivery_time_rank
FROM state_delivery_time ) a
WHERE state_delivery_time_rank<=5
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	state	state_delivery_time_rank	avg_delivery_time
1	MG	3	11.52
2	DF	4	12.5
3	SP	1	8.26
4	PR	2	11.48
5	SC	5	14.52
6	AP	2	27.75
7	AL	4	23.99
8	AM	3	25.96

4. 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.

6. Analysis based on the payments:

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

```
SELECT Year, Month, payment_type,
COUNT(order_id) as order_count_payment_type
FROM
  (SELECT
EXTRACT(MONTH FROM order_purchase_timestamp) AS  Month,
EXTRACT(YEAR FROM order_purchase_timestamp)  AS  Year,
o.order_id,payment_type
FROM target_retailer.orders AS o
JOIN target_retailer.payments AS p
ON o.order_id = p.order_id) a
```

```
GROUP BY Year, Month, payment_type
ORDER BY Year, Month, payment_type;
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	Year	Month	payment_type	order_count_payment_type
1	2016	9	credit_card	3
2	2016	10	UPI	63
3	2016	10	credit_card	254
4	2016	10	debit_card	2
5	2016	10	voucher	23
6	2016	12	credit_card	1
7	2017	1	UPI	197

2. Find the no. of orders placed on the basis of the payment instalments that have been paid.

```
SELECT payment_installments,COUNT(order_id) AS number_of_orders
FROM target_retailer.payments
WHERE payment_installments = 1
GROUP BY payment_installments
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	payment_installments	number_of_orders
1	1	52546

Actionable Insights and Recommendations

- There's an increasing trend in the number of orders placed month on month
- Delivery and Logistics in the state 'SP' is the most streamlined since the time taken to deliver the products is the least
- 'Afternoon' hours witness the peak traffic since maximum number of customers place orders during this time.
- August - Fall season in the US has observed an increasing trend in the number of orders placed. This could be due to the intake of students in the universities who commence their semester programs
- Revenue has increased by more than double (~136%) from 2017 to 2018 in the months of January to August