Business case : Target SQL

Overview : Target is one of the world's most recognized brands and one of America's leading retailers. 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 business case has information of 100k orders from 2016 to 2018 made at Target in Brazil. Its features allows viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers.

1. **Exploratory analysis steps like checking the structure & characteristics of the dataset :**

   a. **Data type of columns in a table (Orders table)**

```
select column_name, data_type

from `target-sql-361409.ecommerce.INFORMATION_SCHEMA.COLUMNS`

where table_name = 'orders';
```

Row	column_name	data_type
1	order_id	STRING
2	customer_id	STRING
3	order_status	STRING
4	order_purchase_timestamp	TIMESTAMP
5	order_approved_at	TIMESTAMP
6	order_delivered_carrier_date	TIMESTAMP
7	order_delivered_customer_date	TIMESTAMP
8	order_estimated_delivery_date	TIMESTAMP

Data type of all the variables are correct.

b. Time period for which the data is given

```
select min(order_purchase_timestamp) as min_time,
max(order_purchase_timestamp) as max_time
from `ecommerce.orders`;
```

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

Data is present from 2016 to 2018.

c. Cities covered in the dataset

```
select distinct geolocation_city as unique_city
from `ecommerce.geolocation`;
```

Row	unique_city
1	aracaju
2	riachuelo
3	nossa senhora do socorro
4	barra dos coqueiros
5	itaporanga d'ajuda
6	sao cristovao
7	são cristóvão
8	santo amaro das brotas
9	pirambu
10	laranjeiras

Total 8011 cities are covered in the dataset.

d. Cities covered in the customer data

```
select distinct customer_city as unique_customer_city

from `ecommerce.customers`;
```

Row	unique_customer_city
1	acu
2	ico
3	ipe
4	ipu
5	ita
6	itu
7	jau
8	luz
9	poa
10	uba

Total 4119 cities are covered in customers dataset.

e. Cities covered in the seller data

```
select distinct seller_city as unique_seller_city

from `ecommerce.sellers`;
```

Row	unique_seller_city
1	rio branco
2	manaus
3	bahia
4	ipira
5	irece
6	ilheus
7	guanambi
8	salvador
9	eunapolis
10	barro alto

Total 611 cities are covered in sellers dataset.

f. State covered in the dataset

```
select distinct geolocation_state as unique_state

from `ecommerce.geolocation`;
```

Row	unique_state
1	SE
2	AL
3	PI
4	AP
5	AM
6	RR
7	AC
8	RO
9	TO
10	BA

Total 27 states are covered in the dataset.

g. States covered in the customer data

```
select distinct customer_state as unique_customer_state

from `ecommerce.customers`;
```

Row	unique_customer_state
1	RN
2	CE
3	RS
4	SC
5	SP
6	MG
7	BA
8	RJ
9	GO
10	MA

Total 27 states are covered in customers dataset.

h. States covered in the seller data

```
select distinct seller_state as unique_seller_state

from `ecommerce.sellers`;
```

Row	unique_seller_state
1	AC
2	AM
3	BA
4	CE
5	DF
6	ES
7	GO
8	MA
9	MG
10	MS

Total 23 states are covered in the sellers dataset.

2. In-depth Exploration:

a. Is there a growing trend on e-commerce in Brazil? (year wise)

```
select order_year, sum(payment_value) as sale, count(distinct order_id) as
total_order
FROM
(select o.order_id,extract(year from o.order_purchase_timestamp) as order_year,
p.payment_value
from `ecommerce.orders` o
JOIN `ecommerce.payments` p
ON o.order_id = p.order_id) x
group by order_year
order by sum(payment_value);
```

Row	order_year	sale	total_order
1	2016	59362.3400...	328
2	2017	7249746.72...	45101
3	2018	8699763.04...	54011

Yes, Total orders and sale both are increasing year wise.

b. Is there a seasonality with peaks at specific months?

```
select order_month, sum(payment_value) as sale, count(distinct order_id) as
total_order
FROM
(select o.order_id,extract(month from o.order_purchase_timestamp) as
order_month, p.payment_value
from `ecommerce.orders` o
JOIN `ecommerce.payments` p
ON o.order_id = p.order_id) x
group by order_month
order by sum(payment_value) desc;
```

Row	order_month	sale	total_order
1	5	1746900.97...	10573
2	8	1696821.64...	10843
3	7	1658923.67...	10318
4	3	1609515.72...	9893
5	4	1578573.51...	9343
6	6	1535156.88...	9412
7	2	1284371.35...	8508
8	1	1253492.22...	8069
9	11	1194882.80...	7544
10	12	878421.100...	5674

Maximum sale is in June month (may be due to summer vacation). Minimum sale is in last quarter (september to december). (September is having very minimum sale)

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

Morning : 4 - 11 hour
Afternoon : 12 - 16 hour
Evening : 17 - 22 hour
Night : 23 - 3 hour

```
select count(distinct order_id) as order_count,
Day_part
FROM
(select order_id,
CASE
WHEN extract(hour from order_purchase_timestamp) >= 4 and extract(hour from
order_purchase_timestamp) <= 11 THEN "Morning"
WHEN extract(hour from order_purchase_timestamp) >= 12 and extract(hour from
order_purchase_timestamp) <= 16 THEN "Afternoon"
WHEN extract(hour from order_purchase_timestamp) >= 17 and extract(hour from
order_purchase_timestamp) <= 22 THEN "Evening"
WHEN extract(hour from order_purchase_timestamp) >= 23 or extract(hour from
order_purchase_timestamp) <= 3 THEN "Night"
END AS Day_part
from `ecommerce.orders`) y
group by Day_part
order by count(distinct order_id);
```

Row	order_count	Day_part
1	8469	Night
2	22634	Morning
3	32211	Afternoon
4	36127	Evening

Customers are mostly purchasing the items in the Evening. In the night, customers are purchasing the items very minimum.

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

a. Get month on month orders by region, states

```
select o.year, o.month, c.customer_state, count(*) as total_order
FROM `ecommerce.customers` c
JOIN
(select customer_id,
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month
FROM `ecommerce.orders`) o
ON c.customer_id = o.customer_id
group by o.year, o.month, c.customer_state
order by o.year, o.month, c.customer_state;
```

Row	year	month	customer_state	total_order
1	2016	9	RR	1
2	2016	9	RS	1
3	2016	9	SP	2
4	2016	10	AL	2
5	2016	10	BA	4
6	2016	10	CE	8
7	2016	10	DF	6
8	2016	10	ES	4
9	2016	10	GO	9
10	2016	10	MA	4

Mostly orders are from SP customer_state in each month.

b. How are customers distributed in Brazil

```
select customer_state, count(*) as customer_count,
round((count(*) / 99441)*100, 1) as percent_of_customer
FROM `ecommerce.customers`
group by customer_state
order by count(*) desc;
```

Row	customer_state	customer_c...	percent_of_...
1	SP	41746	42.0
2	RJ	12852	12.9
3	MG	11635	11.7
4	RS	5466	5.5
5	PR	5045	5.1
6	SC	3637	3.7
7	BA	3380	3.4
8	DF	2140	2.2
9	ES	2033	2.0
10	GO	2020	2.0

Mostly customers are from SP customer_state. (42 % customers are from SP state)

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

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

```
select year, month,

((current_year_order - last_year_order) / last_year_order) * 100 as
percent_increase_order,

((current_year_price - last_year_price) / last_year_price) * 100 as
percent_increase_price,

((current_year_freight - last_year_freight) / last_year_freight) * 100 as
percent_increase_freight

FROM

(select year, month, count(*) as current_year_order, sum(price) as
current_year_price, sum(freight_value) as current_year_freight,
```

```
lag(count(*), 1) over(partition by month order by year) as last_year_order,

lag(sum(price), 1) over(partition by month order by year) as last_year_price,

lag(sum(freight_value), 1) over(partition by month order by year) as
last_year_freight

from

(select o.order_id,i.price, i.freight_value,

extract(year from o.order_purchase_timestamp) as year,

extract(month from o.order_purchase_timestamp) as month

from `ecommerce.orders` o

JOIN

`ecommerce.order_items` i ON o.order_id = i.order_id) x

where year IN (2017, 2018) and (month <= 8)

group by year, month) x

where last_year_order is not null

order by month;
```

Row	year	month	percent_increase_order	percent_increase_price	percent_increase_freight
1	2018	1	759.47643979057591	689.63319551768632	831.94519668019836
2	2018	2	293.23423885187083	241.3539834653252	266.18532182587603
3	2018	3	173.9	162.64950207603766	197.92131572887169
4	2018	4	197.13114754098359	176.90257000006085	210.60150288570694
5	2018	5	91.610251450676984	96.912568458262811	91.293688789327192
6	2018	6	97.543957577449063	99.779952641638118	125.3186439533887
7	2018	7	56.93737552555875	79.8093606452357	87.739299706671034
8	2018	8	47.617107942973526	48.90740428168904	57.717854864308947

Total orders, price and freight value all increased from 2017 to 2018. For January month, there is maximum increment and for August month, there is minimum increment.

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

```
select c.customer_state,

sum(i.price) as price_sum, avg(i.price) as price_mean,

sum(i.freight_value) as freight_sum, avg(i.freight_value) as freight_avg

from `ecommerce.customers` c

JOIN `ecommerce.orders` o ON c.customer_id = o.customer_id

JOIN `ecommerce.order_items` i on o.order_id = i.order_id

group by c.customer_state

order by c.customer_state;
```

Row	customer_state	price_sum	price_mean	freight_sum	freight_avg
1	AC	15982.949999999988	173.72771739130434	3686.7499999999991	40.073369565217405
2	AL	80314.81	180.88921171171171	15914.589999999991	35.843671171171152
3	AM	22356.840000000011	135.49599999999995	5478.8899999999967	33.205393939393936
4	AP	13474.29999999999	164.32073170731707	2788.5000000000009	34.006097560975618
5	BA	511349.99000000674	134.6012082126874	100156.67999999883	26.363958936562248
6	CE	227254.70999999763	153.7582611637348	48351.589999999924	32.714201623815995
7	DF	302603.93999999797	125.77054862842893	50625.499999999811	21.041354945968383
8	ES	275037.30999999633	121.91370124113466	49764.599999999889	22.058776595744682
9	GO	294591.94999999728	126.27173167595369	53114.979999999865	22.766815259322794
10	MA	119648.21999999993	145.20415048543691	31523.770000000033	38.25700242718446

Total price and total freight value are maximum for SP state.

Total price and total freight value are minimum for RR state.

Average price and average freight value are minimum for SP state.

Average price is maximum for PB state and average freight value is maximum for RR state.

5. Analysis on sales, freight and delivery time:

a. Top 5 states with highest average freight value

```
WITH state_order_detail AS

(select x.customer_state,

avg(x.freight_value) as mean_freight_value,

avg(x.time_to_delivery) as avg_time_to_delivery,

avg(x.diff_estimated_delivery) as avg_diff_estimated_delivery

FROM

(select c.customer_state, i.freight_value,

TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
DAY) as time_to_delivery,

TIMESTAMP_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date, DAY) as diff_estimated_delivery

from `ecommerce.orders` o

JOIN `ecommerce.order_items` i ON i.order_id = o.order_id

JOIN `ecommerce.customers` c ON c.customer_id = o.customer_id) x

group by customer_state)

SELECT customer_state, mean_freight_value from state_order_detail order by
mean_freight_value desc LIMIT 5;
```

Row	customer_state	mean_freight_value
1	RR	42.984423076923093
2	PB	42.723803986710941
3	RO	41.069712230215842
4	AC	40.073369565217405
5	PI	39.147970479704767

Average freight value is maximum for RR state and PB state.

b. Top 5 states with lowest average freight value

```
WITH state_order_detail AS

(select x.customer_state,

avg(x.freight_value) as mean_freight_value,

avg(x.time_to_delivery) as avg_time_to_delivery,

avg(x.diff_estimated_delivery) as avg_diff_estimated_delivery

FROM

(select c.customer_state, i.freight_value,

TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
DAY) as time_to_delivery,

TIMESTAMP_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date, DAY) as diff_estimated_delivery

from `ecommerce.orders` o

JOIN `ecommerce.order_items` i ON i.order_id = o.order_id

JOIN `ecommerce.customers` c ON c.customer_id = o.customer_id) x

group by customer_state)

SELECT customer_state, mean_freight_value from state_order_detail order by
mean_freight_value LIMIT 5;
```

Row	customer_state	mean_freight_value
1	SP	15.147275390419248
2	PR	20.531651567944248
3	MG	20.630166806306541
4	RJ	20.96092393168248
5	DF	21.041354945968383

Average freight value is minimum for SP state.

c. Top 5 states with highest average time to delivery

```
WITH state_order_detail AS

(select x.customer_state,

avg(x.freight_value) as mean_freight_value,

avg(x.time_to_delivery) as avg_time_to_delivery,

avg(x.diff_estimated_delivery) as avg_diff_estimated_delivery

FROM

(select c.customer_state, i.freight_value,

TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
DAY) as time_to_delivery,

TIMESTAMP_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date, DAY) as diff_estimated_delivery

from `ecommerce.orders` o

JOIN `ecommerce.order_items` i ON i.order_id = o.order_id

JOIN `ecommerce.customers` c ON c.customer_id = o.customer_id) x

group by customer_state)

SELECT customer_state, avg_time_to_delivery from state_order_detail order by
avg_time_to_delivery desc LIMIT 5;
```

Row	customer_state	avg_time_to_delivery
1	RR	27.826086956521738
2	AP	27.753086419753075
3	AM	25.963190184049076
4	AL	23.992974238875881
5	PA	23.301707779886126

Average delivery time is maximum for RR and AP state.

d. Top 5 states with lowest average time to delivery

```
WITH state_order_detail AS

(select x.customer_state,

avg(x.freight_value) as mean_freight_value,

avg(x.time_to_delivery) as avg_time_to_delivery,

avg(x.diff_estimated_delivery) as avg_diff_estimated_delivery

FROM

(select c.customer_state, i.freight_value,

TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
DAY) as time_to_delivery,

TIMESTAMP_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date, DAY) as diff_estimated_delivery

from `ecommerce.orders` o

JOIN `ecommerce.order_items` i ON i.order_id = o.order_id

JOIN `ecommerce.customers` c ON c.customer_id = o.customer_id) x

group by customer_state)

SELECT customer_state, avg_time_to_delivery from state_order_detail order by
avg_time_to_delivery LIMIT 5;
```

Row	customer_state	avg_time_to_delivery
1	SP	8.25960855241909
2	PR	11.480793060718735
3	MG	11.515522180072811
4	DF	12.501486199575384
5	SC	14.520985846754517

Average delivery time is minimum for SP state.

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

```
WITH state_order_detail AS

(select x.customer_state,

avg(x.freight_value) as mean_freight_value,

avg(x.time_to_delivery) as avg_time_to_delivery,

avg(x.diff_estimated_delivery) as avg_diff_estimated_delivery

FROM

(select c.customer_state, i.freight_value,

TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
DAY) as time_to_delivery,

TIMESTAMP_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date, DAY) as diff_estimated_delivery

from `ecommerce.orders` o

JOIN `ecommerce.order_items` i ON i.order_id = o.order_id

JOIN `ecommerce.customers` c ON c.customer_id = o.customer_id) x

group by customer_state)

SELECT customer_state, avg_diff_estimated_delivery from state_order_detail
order by avg_diff_estimated_delivery desc LIMIT 5;
```

Row	customer_state	avg_diff_estimated_del...
1	AC	20.010989010989018
2	RO	19.080586080586084
3	AM	18.975460122699381
4	AP	17.444444444444443
5	RR	17.434782608695652

Delivery is very fast in AC state.

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

```
WITH state_order_detail AS

(select x.customer_state,

avg(x.freight_value) as mean_freight_value,

avg(x.time_to_delivery) as avg_time_to_delivery,

avg(x.diff_estimated_delivery) as avg_diff_estimated_delivery

FROM

(select c.customer_state, i.freight_value,

TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
DAY) as time_to_delivery,

TIMESTAMP_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date, DAY) as diff_estimated_delivery

from `ecommerce.orders` o

JOIN `ecommerce.order_items` i ON i.order_id = o.order_id

JOIN `ecommerce.customers` c ON c.customer_id = o.customer_id) x

group by customer_state)

SELECT customer_state, avg_diff_estimated_delivery from state_order_detail
order by avg_diff_estimated_delivery LIMIT 5;
```

Row	customer_state	avg_diff_estimated_delivery
1	AL	7.9765807962529349
2	MA	9.1099999999999923
3	SE	9.1653333333333276
4	ES	9.7685393258427116
5	BA	10.119467825142538

Delivery is very slow in AL state.

6. Payment type analysis:

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

```
select year, month, payment_type, count(*) total_order

FROM

(select o.order_id, p.payment_type,

extract(year from o.order_purchase_timestamp) as year,

extract(month from o.order_purchase_timestamp) as month

from `ecommerce.orders` o

JOIN

`ecommerce.payments` p ON o.order_id = p.order_id) x

group by year, month, payment_type

order by year, month, count(*);
```

Row	year	month	payment_type	total_order
1	2016	9	credit_card	3
2	2016	10	debit_card	2
3	2016	10	voucher	23
4	2016	10	UPI	63
5	2016	10	credit_card	254
6	2016	12	credit_card	1
7	2017	1	debit_card	9
8	2017	1	voucher	61
9	2017	1	UPI	197
10	2017	1	credit_card	583

Mostly customers are using credit card for bill payment.

Customers are using debit card very less for bill payment.

b. Distribution of payment installments and count of orders

```
select payment_installments, count(*) as total_order from `ecommerce.payments`

group by payment_installments

order by count(*) desc;
```

Row	payment_in...	total_order
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920
9	7	1626
10	9	644

Mostly customers are preferring 1 installment for bill payments.

7. Product category analysis:

a. Distribution of product category and count of orders

```
select p.product_category, count(*) as total_orders
FROM `ecommerce.orders` o
JOIN `ecommerce.order_items` i ON o.order_id = i.order_id
JOIN `ecommerce.products` p ON i.product_id = p.product_id
group by p.product_category
order by count(*) desc;
```

Row	product_category	total_orders
1	bed table bath	11115
2	HEALTH BEAUTY	9670
3	sport leisure	8641
4	Furniture Decoration	8334
5	computer accessories	7827
6	housewares	6964
7	Watches present	5991
8	telephony	4545
9	Garden tools	4347
10	automotive	4235

Bed table bath, Health beauty, sport leisure, furniture decoration and computer accessories are the top 5 popular product categories.

Recommendations :

(As total orders / sales are increasing continuously, we can say that current system is working well for target.)

- Mostly customers are purchasing the items in evening so target can add new products or start sale/offer at this time.

- Mostly customers are from SP state so target can reduce the difference between estimated delivery and actual delivery date for SP state. (means delivery should be fast in SP state)
- Mostly customers are paying the bills using credit card so target can show more offers on credit card payments.
- Mostly customers are purchasing the items in June, July, August, september months so target can add more products / offers in these months.
- Bed table bath, Health beauty, sport leisure, furniture decoration and computer accessories are the top 5 popular categories for which target can add more products.

Actionable insights :

- There has been an increase in ecommerce in brazil from 2016 to 2019.
- Peak Orders during the month may july and august.
- Brazilian customers tend to buy during the afternoon hours
- Sau paulo has the highest number of e customers in brazil

- The percentage increase in brazil from 2017 to 2018 is 57.8%
- Sp has the lowest avg delivery time among all states
- RR has the highest avg delivery time among all states.
- Credit card is the preferred method of payment followed by upi,voucher and debit card payment
- Most payments were made in a single payment