

Business Case: target SQL

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

a) Data type of columns in a table

USE target;

DESCRIBE products;

	Field	Type	Null	Key	Default	Extra
▶	product_id	text	YES		NULL	
	product category	text	YES		NULL	
	product_name_length	int	YES		NULL	
	product_description_length	int	YES		NULL	
	product_photos_qty	int	YES		NULL	
	product_weight_g	int	YES		NULL	
	product_length_cm	int	YES		NULL	
	product_height_cm	int	YES		NULL	
	product_width_cm	int	YES		NULL	



b) Time period for which the data is given


SELECT

MIN(DATE(order_purchase_timestamp)) AS start_date,

MAX(DATE(order_purchase_timestamp)) AS last_date

FROM orders;

Result Grid   Filter Rows:

	start_date	last_date
	2016-09-04	2018-10-17

c) Cities and States of customers ordered during the given period

SELECT

c.customer_city,

c.customer_state

FROM orders AS o

LEFT JOIN customers AS c

ON o.customer_id=c.customer_id

GROUP BY customer_city,customer_state;

	customer_city	customer_state
▶	vianopolis	GO
	ouro preto	MG
	goiania	GO
	feira de santana	BA
	sao paulo	SP
	cruz das almas	BA
	tocos	RJ
	capelinha	MG
	franca	SP
	guarujá	SP
	januaria	MG
	taubate	SP

2) In-depth Exploration:

- a) Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

```
SELECT
    YEAR(order_purchase_timestamp) as year,
    COUNT(order_id) as order_count
FROM orders
GROUP BY YEAR(order_purchase_timestamp)
ORDER BY year;
```

	year	order_count
▶	2016	329
	2017	45101
	2018	54011

Even though we don't have complete data for the years 2016 and 2018, there is a growing trend of orders on e-commerce.

```
SELECT
    YEAR(order_purchase_timestamp) as year,
    MONTH(order_purchase_timestamp) as month,
    COUNT(order_id) as order_count
FROM orders
```

```
GROUP BY MONTH(order_purchase_timestamp),
YEAR(order_purchase_timestamp)
ORDER BY year, month;
```

	year	month	order_count
▶	2016	9	4
	2016	10	324
	2016	12	1
	2017	1	800
	2017	2	1780
	2017	3	2682
	2017	4	2404
	2017	5	3700
	2017	6	3245
	2017	7	4026
	2017	8	4331
	2017	9	4285

```
SELECT
    YEAR(order_purchase_timestamp) as year,
    MONTH(order_purchase_timestamp) as month,
    COUNT(order_id) as order_count
FROM orders
GROUP BY MONTH(order_purchase_timestamp),
YEAR(order_purchase_timestamp)
ORDER BY order_count DESC limit 5;
```

	year	month	order_count
▶	2017	11	7544
	2018	1	7269
	2018	3	7211
	2018	4	6939
	2018	5	6873

From the above table we can see that the above months have got a higher order count compared to the other months.

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

```

SELECT
    CASE
        WHEN TIME(order_purchase_timestamp) >= '04:00:00' AND
TIME(order_purchase_timestamp) < '06:00:00' THEN 'Dawn'
        WHEN TIME(order_purchase_timestamp) >= '06:00:00' AND
TIME(order_purchase_timestamp) < '12:00:00' THEN 'Morning'
        WHEN TIME(order_purchase_timestamp) >= '12:00:00' AND
TIME(order_purchase_timestamp) < '18:00:00' THEN 'Afternoon'
        ELSE 'Night'
    END AS time_of_the_day,
COUNT(order_id) AS no_of_orders
FROM orders
GROUP BY time_of_the_day
ORDER BY no_of_orders;

```

	time_of_the_day	no_of_orders
►	Night	38446
	Afternoon	38361
	Morning	22240
	Dawn	394

From the above table we can see that Brazilians buy more items in Afternoon and Nights

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

a) Get month on month orders by states

```

SELECT
    c.customer_state,
    YEAR(order_purchase_timestamp) as year,
    MONTH(order_purchase_timestamp) as month,
    COUNT(DISTINCT order_id) as order_count
FROM orders AS o
JOIN customers AS c
ON o.customer_id=c.customer_id
GROUP BY year, month, c.customer_state
ORDER BY year, month;

```

	customer_state	year	month	order_count
▶	RR	2016	9	1
	RS	2016	9	1
	SP	2016	9	2
	AL	2016	10	2
	BA	2016	10	4
	CE	2016	10	8
	DF	2016	10	6
	ES	2016	10	4
	GO	2016	10	9
	MA	2016	10	4
	MG	2016	10	40
	MT	2016	10	3
	PA	2016	10	4

a) Distribution of customers across the states in Brazil.

```

SELECT
    c.customer_state,
    COUNT(DISTINCT o.customer_id) as customer_count
FROM orders AS o
JOIN customers AS c
ON o.customer_id=c.customer_id
GROUP BY c.customer_state;

```

	customer_state	customer_count
▶	AC	81
	AL	413
	AM	148
	AP	68
	BA	3380
	CE	1336
	DF	2140
	ES	2033
	GO	2020
	MA	747
	MG	11635
	MS	715
	MT	907

	customer_state	order_count
▶	RR	46
	AP	68
	AC	81
	AM	148
	RO	253
	TO	280
	SE	350
	AL	413
	RN	485
	PI	495
	PB	536
	MS	715

	customer_state	customer_count
▶	RR	46
	AP	68
	AC	81
	AM	148
	RO	253
	TO	280
	SE	350
	AL	413
	RN	485
	PI	495
	PB	536
	MS	715

The above states have gotten less orders compared to the other states. So, we can devise a promotion campaign to promote the brand's e-commerce business in these states.

4) Impact on Economy: Analyze the money movement 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) - You can use "payment_value" column in payments table

```
SELECT
    YEAR(order_purchase_timestamp) as year,
    ROUND(SUM(p.payment_value),2) AS total_price,
    ROUND(((SUM(p.payment_value)-LAG(SUM(p.payment_value))
OVER(ORDER BY YEAR(order_purchase_timestamp)))/
LAG(SUM(p.payment_value)) OVER(ORDER BY
YEAR(order_purchase_timestamp)))*100,2) AS percentage_increase
FROM orders AS o
LEFT JOIN payments AS p
ON o.order_id=p.order_id
WHERE MONTH(order_purchase_timestamp)<9
GROUP BY year
ORDER BY year;
```

	year	total_price	percentage_increase
▶	2017	3669022.12	NULL
	2018	8694733.84	136.98

There has been 137% growth from 2017 to 2018.

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

```
SELECT
    c.customer_state,
    ROUND(AVG(i.price),2) AS mean_of_price,
    ROUND(SUM(i.price),2) AS total_price,
    ROUND(AVG(i.freight_value),2) AS mean_of_freight_value,
    ROUND(SUM(i.freight_value),2) AS total_freight_value
FROM orders AS o
LEFT JOIN order_items AS i
ON o.order_id=i.order_id
LEFT JOIN customers AS c
ON o.customer_id=c.customer_id
GROUP BY c.customer_state;
```

	customer_state	mean_of_price	total_price	mean_of_freight_value	total_freight_value
►	SP	109.65	5202955.05	15.15	718723.07
	RS	120.34	750304.02	21.74	135522.74
	SC	124.65	520553.34	21.47	89660.26
	MG	120.75	1585308.03	20.63	270853.46
	RJ	125.12	1824092.67	20.96	305589.31
	MT	148.3	156453.53	28.17	29715.43
	PR	119	683083.76	20.53	117851.68
	RO	165.97	46140.64	41.07	11417.38
	MS	142.63	116812.64	23.37	19144.03
	BA	134.6	511349.99	26.36	100156.68
	ES	121.91	275037.31	22.06	49764.6
	PI	160.36	86914.08	39.15	21218.2

5) Analysis on sales, freight and delivery time

- a) Calculate days between purchasing, delivering and estimated delivery
- $\text{time_to_delivery} = \text{order_purchase_timestamp} - \text{order_delivered_customer_date}$
 - $\text{diff_estimated_delivery} = \text{order_estimated_delivery_date} - \text{order_delivered_customer_date}$

```
SELECT
    DISTINCT order_id,
```

```

DATEDIFF(order_delivered_customer_date,order_purchase_timestamp) AS
time_to_delivery,
DATEDIFF(order_estimated_delivery_date,order_delivered_customer_date)
AS diff_estimated_delivery
FROM orders;

```

	order_id	time_to_delivery	diff_estimated_delivery
▶	e481f51cbdc54678b7cc49136f2d6af7	8	8
	53cdb2fc8bc7dce0b6741e2150273451	14	6
	47770eb9100c2d0c44946d9cf07ec65d	9	18
	949d5b44dbf5de918fe9c16f97b45f8a	14	13
	ad21c59c0840e6cb83a9ceb5573f8159	3	10
	a4591c265e18cb1dcee52889e2d8acc3	17	6
	136cce7faa42fdb2cefd53fdc79a6098	HULL	HULL
	6514b8ad8028c9f2cc2374ded245783f	10	12
	76c6e866289321a7c93b82b54852dc33	10	32
	e69bfb5eb88e0ed6a785585b27e16dbf	18	7

c) Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```

SELECT
    c.customer_state,
    ROUND(AVG(freight_value),2) AS mean_freight_value,
    ROUND(AVG(DATEDIFF(o.order_delivered_customer_date,o.order_p
urchase_timestamp)),0) AS avg_time_to_delivery,
    ROUND(AVG(DATEDIFF(o.order_estimated_delivery_date,o.order_de
livered_customer_date)),0) AS avg_diff_estimated_delivery
FROM orders AS o
JOIN customers AS c
ON o.customer_id=c.customer_id
LEFT JOIN order_items AS i
ON o.order_id=i.order_id
GROUP BY c.customer_state;

```


	customer_state	mean_freight_value	avg_time_to_delivery	avg_diff_estimated_delivery
►	SC	21.47	15	12
	SP	15.15	9	11
	RS	21.74	15	14
	MG	20.63	12	13
	RJ	20.96	15	12
	MT	28.17	18	15
	PR	20.53	12	13
	RO	41.07	20	20
	MS	23.37	15	11
	BA	26.36	19	11

e) Top 5 states with highest/lowest average freight value

SELECT

```

    c.customer_state,
    ROUND(AVG(freight_value),2) AS mean_freight_value,
    ROUND(AVG(DATEDIFF(o.order_delivered_customer_date,o.order_purchase_timestamp)),0) AS avg_time_to_delivery,
    ROUND(AVG(DATEDIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date)),0) AS avg_diff_estimated_delivery
FROM orders AS o
JOIN customers AS c
ON o.customer_id=c.customer_id
LEFT JOIN order_items AS i
ON o.order_id=i.order_id
GROUP BY c.customer_state
ORDER BY mean_freight_value DESC LIMIT 5;

```

	customer_state	mean_freight_value	avg_time_to_delivery	avg_diff_estimated_delivery
►	RR	42.98	28	18
	PB	42.72	21	13
	RO	41.07	20	20
	AC	40.07	21	21
	PI	39.15	19	12

The above states have the highest average freight value as well as high delivery time too. Also these states are among the ones that have very low order count. This can be improved by establishing a warehouse(only after considering the market size) that's capable of serving these and neighboring state's populations.

This can reduce the mean freight value and delivery time which might drive more orders.

```
SELECT
    c.customer_state,
    ROUND(AVG(freight_value),2) AS mean_freight_value,
    ROUND(AVG(DATEDIFF(o.order_delivered_customer_date,o.order_purchase_timestamp)),0) AS time_to_delivery,
    ROUND(AVG(DATEDIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date)),0) AS diff_estimated_delivery
FROM orders AS o
JOIN customers AS c
ON o.customer_id=c.customer_id
LEFT JOIN order_items AS i
ON o.order_id=i.order_id
GROUP BY c.customer_state
ORDER BY mean_freight_value LIMIT 5;
```

	customer_state	mean_freight_value	avg_time_to_delivery	avg_diff_estimated_delivery
▶	SP	15.15	9	11
	PR	20.53	12	13
	MG	20.63	12	13
	RJ	20.96	15	12
	DF	21.04	13	12

The above states have the lowest average freight value. These states also seem to have less delivery time. Also observed that these states also have higher order count when compared to other states.

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

```
SELECT
    c.customer_state,
    ROUND(AVG(freight_value),2) AS mean_freight_value,
    ROUND(AVG(DATEDIFF(o.order_delivered_customer_date,o.order_purchase_timestamp)),0) AS avg_time_to_delivery,
    ROUND(AVG(DATEDIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date)),0) AS avg_diff_estimated_delivery
FROM orders AS o
JOIN customers AS c
ON o.customer_id=c.customer_id
LEFT JOIN order_items AS i
```

```

ON o.order_id=i.order_id
GROUP BY c.customer_state
ORDER BY avg_time_to_delivery DESC LIMIT 5;

```

	customer_state	mean_freight_value	avg_time_to_delivery	avg_diff_estimated_delivery
▶	AP	34.01	28	18
	RR	42.98	28	18
	AM	33.21	26	20
	AL	35.84	24	9
	PA	35.83	24	14

The above states have the highest average time to deliver the products. Also these states have some of the highest mean freight values. This can be improved by establishing a warehouse that's capable of serving these state's population which can reduce the mean freight value and delivery time which might drive more orders. Can design some marketing campaigns to promote e-commerce at these locations.

```

SELECT
    c.customer_state,
    ROUND(AVG(freight_value),2) AS mean_freight_value,
    ROUND(AVG(DATEDIFF(o.order_delivered_customer_date,o.order_purchase_timestamp)),0) AS avg_time_to_delivery,
    ROUND(AVG(DATEDIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date)),0) AS avg_diff_estimated_delivery
FROM orders AS o
JOIN customers AS c
ON o.customer_id=c.customer_id
LEFT JOIN order_items AS i
ON o.order_id=i.order_id
GROUP BY c.customer_state
ORDER BY avg_time_to_delivery LIMIT 5;

```

	customer_state	mean_freight_value	avg_time_to_delivery	avg_diff_estimated_delivery
▶	SP	15.15	9	11
	PR	20.53	12	13
	MG	20.63	12	13
	DF	21.04	13	12
	MS	23.37	15	11

The above states have the lowest average time to deliver the products.

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

```
SELECT
    c.customer_state,
    ROUND(AVG(freight_value),2) AS mean_freight_value,
    ROUND(AVG(DATEDIFF(o.order_delivered_customer_date,o.order_purchase_timestamp)),0) AS avg_time_to_delivery,
    ROUND(AVG(DATEDIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date)),0) AS avg_diff_estimated_delivery
FROM orders AS o
JOIN customers AS c
ON o.customer_id=c.customer_id
LEFT JOIN order_items AS i
ON o.order_id=i.order_id
GROUP BY c.customer_state
ORDER BY avg_diff_estimated_delivery DESC LIMIT 5;
```

	customer_state	mean_freight_value	avg_time_to_delivery	avg_diff_estimated_delivery
▶	AC	40.07	21	21
	AM	33.21	26	20
	RO	41.07	20	20
	AP	34.01	28	18
	RR	42.98	28	18

The above are the top 5 states where delivery of products is fastest.

```
SELECT
    c.customer_state,
    ROUND(AVG(freight_value),2) AS mean_freight_value,
    ROUND(AVG(DATEDIFF(o.order_delivered_customer_date,o.order_purchase_timestamp)),0) AS avg_time_to_delivery,
    ROUND(AVG(DATEDIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date)),0) AS avg_diff_estimated_delivery
FROM orders AS o
JOIN customers AS c
ON o.customer_id=c.customer_id
LEFT JOIN order_items AS i
ON o.order_id=i.order_id
GROUP BY c.customer_state
ORDER BY avg_diff_estimated_delivery LIMIT 5;
```

	customer_state	mean_freight_value	avg_time_to_delivery	avg_diff_estimated_delivery
▶	AL	35.84	24	9
	MA	38.26	22	10
	SE	36.65	21	10
	MS	23.37	15	11
	BA	26.36	19	11

The above are the 5 states where the delivery of products is slowest.

6) Payment type analysis:

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

```

SELECT
    YEAR(order_purchase_timestamp) as year,
    MONTH(order_purchase_timestamp) as month,
    p.payment_type,
    COUNT(DISTINCT o.order_id) as order_count
FROM orders AS o
JOIN payments AS p
ON o.order_id=p.order_id
GROUP BY year, month, p.payment_type;

```

	year	month	payment_type	order_count
▶	2016	9	credit_card	3
	2016	10	credit_card	253
	2016	10	debit_card	2
	2016	10	UPI	63
	2016	10	voucher	11
	2016	12	credit_card	1
	2017	1	credit_card	582
	2017	1	debit_card	9
	2017	1	UPI	197
	2017	1	voucher	33
	2017	2	credit_card	1347

b) Count of orders based on the no. of payment installments

```

SELECT
    p.payment_installments,
    COUNT(DISTINCT o.order_id) as order_count
FROM orders AS o

```

JOIN payments AS p
ON o.order_id=p.order_id
GROUP BY p.payment_installments;

	payment_installments	order_count
▶	0	2
	1	49060
	2	12389
	3	10443
	4	7088
	5	5234
	6	3916
	7	1623
	8	4253
	9	644
	10	5315