LEADING US RETAILER - SQL BUSINESS CASE STUDY

BATCH: MAY 22 BEGINNER BATCH

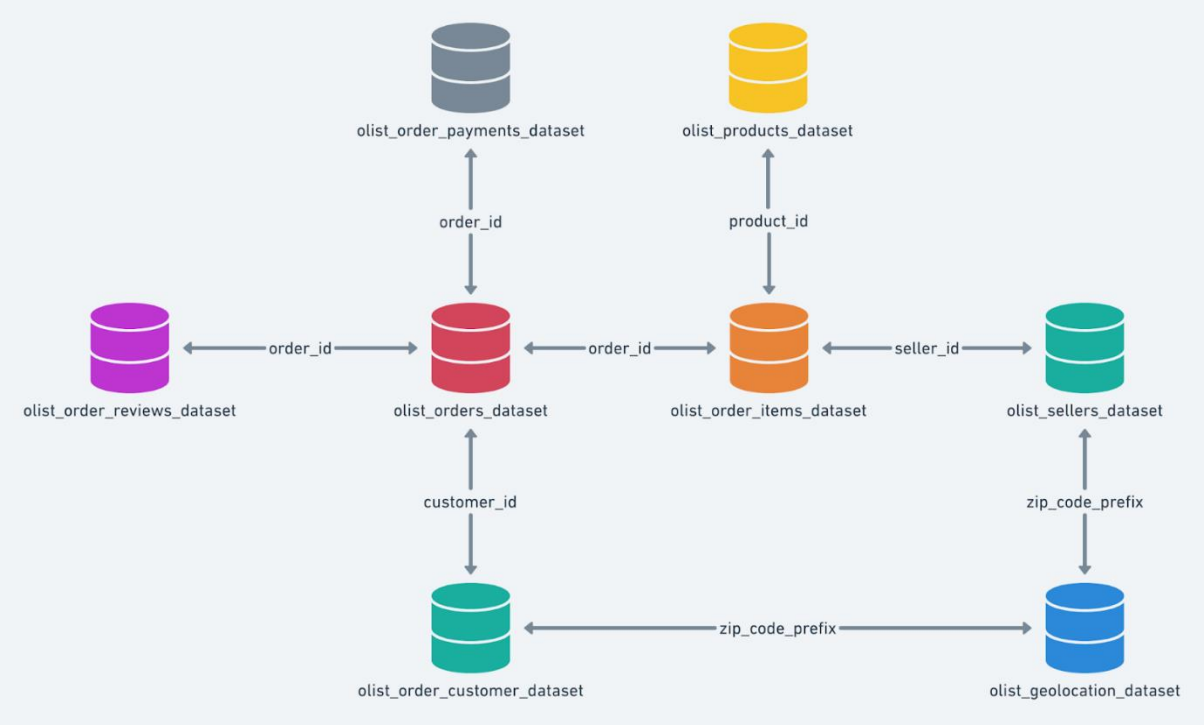
PROBLEM STATEMENT

The dataset is from of America's leading retailers and poses a strong competition for others with its focus on innovation and evolving guest experience.

The goal of this case study is to provide recommendations from the insights uncovered in terms of sales trends across Brazil, purchase patterns, efficiency of delivery network and preferred modes of payments.

The analysis will be done by leveraging the data generated to unearth trends of KPIs over time and identify patterns across states. The platform used for analysis is Google Big Query.

OVERVIEW OF DATASET



EXPLORATORY DATA ANALYSIS

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

1. Data type of columns in a table

customers

Field name	Type
customer_id	STRING
customer_unique_id	STRING
customer_zip_code_prefix	INTEGER
customer_city	STRING
customer_state	STRING

order_items

Field name	Type
order_id	STRING
order_item_id	INTEGER
product_id	STRING
seller_id	STRING
shipping_limit_date	TIMESTAMP
price	FLOAT
freight_value	FLOAT

orders

Field name	Type
order_id	STRING
customer_id	STRING
order_status	STRING
order_purchase_timestamp	TIMESTAMP
order_approved_at	TIMESTAMP
order_delivered_carrier_date	TIMESTAMP
order_delivered_customer_date	TIMESTAMP
order_estimated_delivery_date	TIMESTAMP

payments

Field name	Type
order_id	STRING
payment_sequential	INTEGER
payment_type	STRING
payment_installments	INTEGER
payment_value	FLOAT

sellers

Field name	Type
seller_id	STRING
seller_zip_code_prefix	INTEGER
seller_city	STRING
seller_state	STRING

geolocation

Field name	Type
geolocation_zip_code_prefix	INTEGER
geolocation_lat	FLOAT
geolocation_lng	FLOAT
geolocation_city	STRING
geolocation_state	STRING

products

Field name	Type
product_id	STRING
product_category	STRING
product_name_length	INTEGER
product_description_length	INTEGER
product_photos_qty	INTEGER
product_weight_g	INTEGER
product_length_cm	INTEGER
product_height_cm	INTEGER
product_width_cm	INTEGER

order_reviews

Field name	Type
review_id	STRING
order_id	STRING
review_score	INTEGER
review_comment_title	STRING
review_creation_date	TIMESTAMP
review_answer_timestamp	TIMESTAMP

2. Time period for which the data is given

```
SELECT MIN(order_purchase_timestamp) Min_Date,
MAX(order_purchase_timestamp) Max_Date,
DATETIME_DIFF(MAX( order_purchase_timestamp),MIN(order_purchase_timestamp),DAY)/365 Year
_diff
FROM `Target_data.orders`
```

Row	Min_Date	Max_Date	Year_diff
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC	2.1178082191

- The dataset covers a time period of 2.11 years from 2016-09-04 to 2018-10-17

3. Cities and States covered in the dataset

```
WITH state_table AS

(
SELECT DISTINCT(seller_state) state
FROM `Target_data.sellers`
UNION ALL
SELECT DISTINCT(customer_state) state
FROM `Target_data.customers`
)

SELECT DISTINCT state_table.state
FROM state_table
ORDER BY state_table.state
```

Row	state
1	AC
2	AL
3	AM
4	AP
5	BA
6	CE
7	DF
8	ES
9	GO
10	MA

- The retailer has customers from 27 states in total

```
WITH city_table AS*
(
SELECT DISTINCT(seller_city) city, seller_state state
FROM `Target_data.sellers`
UNION ALL
SELECT DISTINCT(customer_city) city, customer_state state
FROM `Target_data.customers`
)

SELECT DISTINCT(city_table.city),city_table.state
FROM city_table
ORDER BY state
```

Row	city	state
1	rio branco	AC
2	xapuri	AC
3	brasileia	AC
4	porto acre	AC
5	manoel urbano	AC
6	epitaciolandia	AC
7	cruzeiro do sul	AC
8	senador guiomard	AC
9	belem	AL
10	igaci	AL

```
WITH city_table AS
(
SELECT DISTINCT(seller_city) city, seller_state state
FROM `Target_data.sellers`
UNION ALL
SELECT DISTINCT(customer_city) city, customer_state state
FROM `Target_data.customers`
)

SELECT city_table.state,COUNT(DISTINCT(city_table.city)) city_count
FROM city_table
GROUP BY city_table.state
ORDER BY city_count DESC
```

Row	state	city_count
1	MG	750
2	SP	696
3	RS	380
4	PR	372
5	BA	355

- Customers span across 4415 cities in Brazil
- The top 3 states with maximum number of cities covered in the dataset are MG, SP and RS

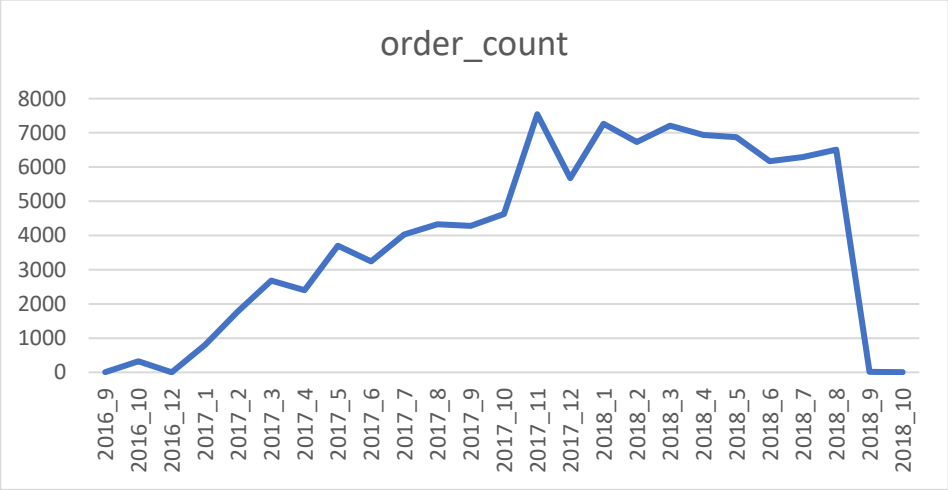
2. In-depth Exploration:

1. 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?

```
With new_table AS
(
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS Year, EXTRACT(MONTH FROM orde
r_purchase_timestamp) AS Month,COUNT(order_id) order_count
FROM `Target_data.orders`
GROUP BY EXTRACT(YEAR FROM order_purchase_timestamp), EXTRACT(MONTH FROM order_pur
chase_timestamp)
)

SELECT *
FROM new_table
ORDER BY Year, Month
```

Row	Year	Month	order_count
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026

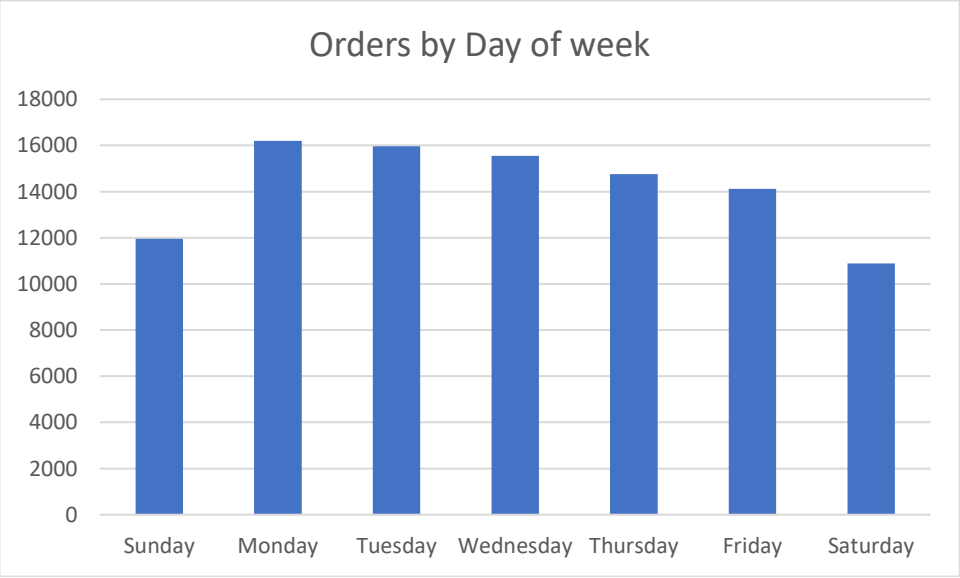


```
WITH day_table AS
(
SELECT EXTRACT(DAYOFWEEK FROM order_purchase_timestamp) AS Day,COUNT(order_id) orde
r_count
FROM `Target_data.orders`
GROUP BY EXTRACT(DAYOFWEEK FROM order_purchase_timestamp)
)

SELECT *
FROM day_table
ORDER BY Day
```

*Day 1 corresponds to Sunday

Row	Day	order_count
1	1	11960
2	2	16196
3	3	15963
4	4	15552
5	5	14761
6	6	14122
7	7	10887



- The month-by-month comparison of number of orders received shows an upward trend with a steep slope from January 2017 till March 2018. There is a small slump till June,2018 and again climbs back up in July,2018 [Assumption: Data is incomplete for months 9 and 10 in 2018]
- The overall scenario is favourable with more orders coming in after each month.
- The highest number of orders were received in November, 2017
- The number of orders received is almost equally distributed across all the weekdays. The number of customers see a slight dip on weekends.

- Since the complete data is only available for 2017, seasonality in terms of number of purchases is inconclusive

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

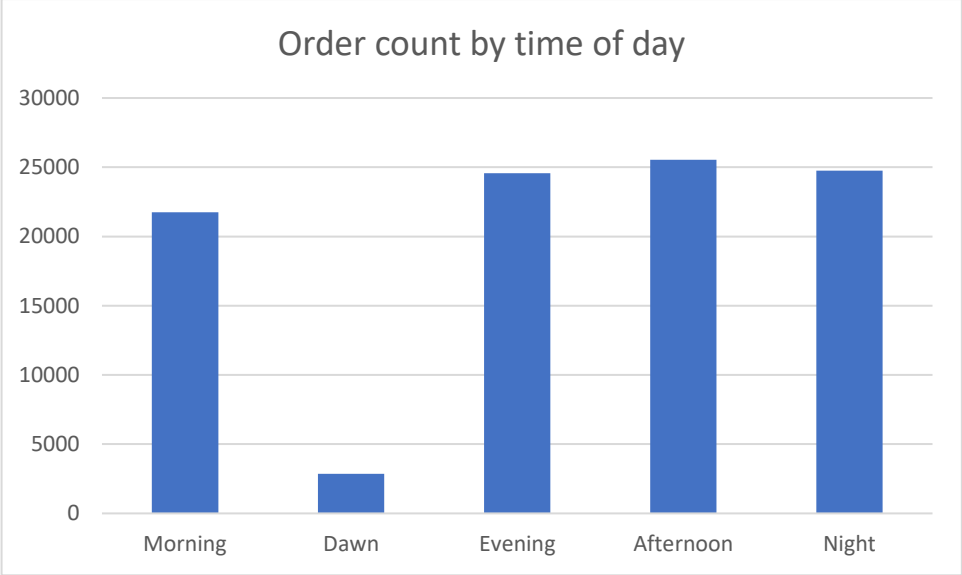
Assuming a day is split as following:

- Dawn – 1AM to 6AM
- Morning 7AM to 11AM
- Afternoon 12PM to 15PM
- Evening 16PM to 19PM
- Night 20PM to 24AM

```
WITH hour_table AS
(
SELECT order_purchase_timestamp, EXTRACT(HOUR FROM order_purchase_timestamp) Hour,
Case WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 1 AND 6
THEN 'Dawn'
WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 7 AND 11
THEN 'Morning'
WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 12 AND 15
THEN 'Afternoon'
WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 16 AND 19
THEN 'Evening'
ELSE 'Night'
END AS Time_of_day
FROM `Target_data.orders`
)

SELECT Time_of_day, COUNT(order_purchase_timestamp) order_count
FROM hour_table
GROUP BY Time_of_day
```

Row	Time_of_day	order_count
1	Morning	21738
2	Dawn	2848
3	Evening	24576
4	Afternoon	25536
5	Night	24743



- They receive a greater number of customers post 7AM and before 12AM. There are very few customers visiting at dawn
- The number of customers visiting themt is almost equally split across the day after 7PM from morning to night with the highest numbers in afternoon. This points to the fact that there are different categories of customers who like to shop in each segment of the day

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

1. Get month on month orders by region, states

```
WITH new_orders AS
(
SELECT customer_state, EXTRACT(YEAR FROM order_purchase_timestamp) Year, EXTRACT(MON
TH FROM order_purchase_timestamp) Month, COUNT(order_id) order_count
FROM `Target_data.orders` o
LEFT JOIN `Target_data.customers` c ON o.customer_id=c.customer_id
GROUP BY customer_state, EXTRACT(YEAR FROM order_purchase_timestamp), EXTRACT(MONTH
 FROM order_purchase_timestamp)
)
SELECT *
FROM new_orders
ORDER BY customer_state, Year, Month
```

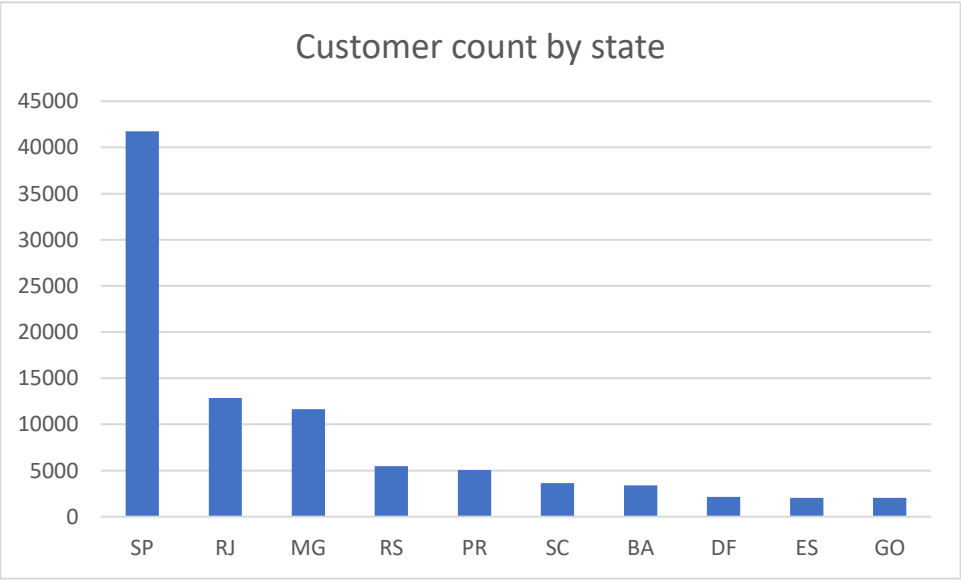
Row	customer_state	Year	Month	order_count
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5
8	AC	2017	8	4

Row	customer_state	Year	Month	order_count
9	AC	2017	9	5
10	AC	2017	10	6

2. How are customers distributed in Brazil?

```
SELECT customer_state, COUNT(customer_id) customer_count
FROM `Target_data.customers`
GROUP BY customer_state
ORDER BY customer_count DESC
```

Row	customer_state	customer_count
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020



- SP is the state in Brazil with the highest number of customers followed by RJ and MG.

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

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

- Cost of orders is taken as price + freight value for each order.

```
WITH year_table AS
(
SELECT EXTRACT(YEAR FROM o.order_purchase_timestamp) Year, SUM(oi.price)+SUM(oi.freight_v
alue) Total_sum
FROM `Target_data.orders` o
LEFT JOIN `Target_data.order_items` oi ON o.order_id=oi.order_id
WHERE EXTRACT(DATE FROM o.order_purchase_timestamp) BETWEEN '2017-01-01' AND '2017-
08-31'
OR EXTRACT(DATE FROM o.order_purchase_timestamp) BETWEEN '2018-01-01' AND '2018-08-31'
GROUP BY EXTRACT(YEAR FROM o.order_purchase_timestamp)
)

SELECT ROUND((Total_sum-
LEAD(Total_sum) OVER(ORDER BY Year DESC))/LEAD(Total_sum) OVER(ORDER BY Year DESC)*
100,2) percent_diff
FROM year_table
ORDER BY Year DESC
LIMIT 1
```

Row	percent_diff
1	139.42

- There is a 139.42% increase in the total cost of orders from 2017 to 2018

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

```
SELECT c.customer_state, ROUND(AVG(oi.price),2) price_avg,ROUND(SUM(oi.price),2) price_sum, R
OUND(AVG(oi.freight_value),2) freight_avg, ROUND(SUM(oi.freight_value),2) freight_sum
FROM `Target_data.customers` c
INNER JOIN `Target_data.orders` o ON c.customer_id=o.customer_id
INNER JOIN `Target_data.order_items` oi ON o.order_id=oi.order_id
GROUP BY c.customer_state
ORDER BY price_avg DESC
```

Row	customer_state	price_avg	price_sum	freight_avg	freight_sum
1	PB	191.48	115268.08	42.72	25719.73
2	AL	180.89	80314.81	35.84	15914.59
3	AC	173.73	15982.95	40.07	3686.75
4	RO	165.97	46140.64	41.07	11417.38
5	PA	165.69	178947.81	35.83	38699.3

- PB, AL and AC are the states with highest average price per order which is an indication of a larger number of high spending customers

5. Analysis on sales, freight, and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

1.1 Difference between purchased and delivered date

```
SELECT order_id,DATETIME_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY
) days_delivered
FROM `Target_data.orders`
```

Row	order_id	days_delivered
1	1950d777989f6a877539f53795b4c3c3	30
2	2c45c33d2f9cb8ff8b1c86cc28c11c30	30
3	65d1e226dfaeb8cdc42f665422522d14	35
4	635c894d068ac37e6e03dc54eccb6189	30
5	3b97562c3aee8bdedcb5c2e45a50d5e1	32
6	68f47f50f04c4cb6774570cfde3a9aa7	29
7	276e9ec344d3bf029ff83a161c6b3ce9	43
8	54e1a3c2b97fb0809da548a59f64c813	40
9	fd04fa4105ee8045f6a0139ca5b49f27	37
10	302bb8109d097a9fc6e9cefc5917d1f3	33

1.2 Difference between purchased and estimated delivery date

```
SELECT order_id,DATETIME_DIFF(order_estimated_delivery_date, order_purchase_timestamp, DAY)
 days_estimated
FROM `Target_data.orders`
```

Row	order_id	days_estimated
1	f88aac7ebccb37f19725a075331ade3e	50
2	790cd37689193dca0d00d2feb6459164	6
3	49db7943d60b6805c3a41f5474772a09	44
4	063b573b88fc80e516aba87df524f809	54
5	a68ce1686d536ca72bd2dadc4b8671e5	56
6	45973912e490866800c0aea8f63099c8	54
7	cda873529ca7ab71f677d5ec11a40304	56
8	ead20687129da8f5d89d831bb0772867	41
9	6f028ccb7d612af251aa442a1fb8b5d0	3
10	8733c8d440c173e524d2fab8025063f4	3

1.3 Difference between estimated delivery date and actual delivery date

```
SELECT order_id,DATETIME_DIFF(order_estimated_delivery_date, order_delivered_customer_date,
DAY) days_estimated
FROM `Target_data.orders`
```

Row	order_id	days_estimated
1	770d331c84e5b214bd9dc70a10b829d0	45
2	1950d777989f6a877539f53795b4c3c3	-12
3	2c45c33d2f9cb8ff8b1c86cc28c11c30	28
4	dabf2b0e35b423f94618bf965fcb7514	44
5	8beb59392e21af5eb9547ae1a9938d06	41

2. Create columns:

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

```
WITH new_orders AS
(
SELECT *,DATETIME_DIFF(order_purchase_timestamp, order_delivered_customer_date, DAY) time_t
o_delivery,
DATETIME_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) diff_estimate
d_delivery
FROM `Target_data.orders`
)
```

Row	order_id	time_to_delivery	diff_estimated_delivery
1	1950d777989f6a877539f53795b4c3c3	-30	-12
2	2c45c33d2f9cb8ff8b1c86cc28c11c30	-30	28
3	65d1e226dfaeb8cdc42f665422522d14	-35	16
4	635c894d068ac37e6e03dc54eccb6189	-30	1
5	3b97562c3aee8bdedcb5c2e45a50d5e1	-32	0

3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
WITH new_orders AS
(
SELECT *,DATETIME_DIFF(order_purchase_timestamp, order_delivered_customer_date, DAY) time_t
o_delivery,
DATETIME_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) diff_estimate
d_delivery
FROM `Target_data.orders`
```

```
)
SELECT c.customer_state, ROUND(AVG(oi.freight_value),2) Mean_freight, ROUND(AVG(ABS(n.time_
to_delivery)),2) Mean_delivery_time,ROUND(AVG(n.diff_estimated_delivery),2) mean_diff_estimated
FROM `Target_data.customers` c
INNER JOIN `Target_data.orders` o ON c.customer_id=o.customer_id
INNER JOIN `Target_data.order_items`oi ON o.order_id=oi.order_id
INNER JOIN new_orders n ON o.order_id=n.order_id
GROUP BY c.customer_state
ORDER BY c.customer_state
```

Row	customer_state	Mean_freight	Mean_delivery_time	mean_diff_estimated
1	AC	40.07	20.33	20.01
2	AL	35.84	23.99	7.98
3	AM	33.21	25.96	18.98
4	AP	34.01	27.75	17.44
5	BA	26.36	18.77	10.12

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

Top 5 states with Highest average freight value

- Using the new_orders CTE created above

```
SELECT c.customer_state, ROUND(AVG(oi.freight_value),2) Mean_freight, ROUND(AVG(ABS(n.time_
to_delivery)),2) Mean_delivery_time,ROUND(AVG(n.diff_estimated_delivery),2) mean_diff_estimated
FROM `Target_data.customers` c
INNER JOIN `Target_data.orders` o ON c.customer_id=o.customer_id
INNER JOIN `Target_data.order_items`oi ON o.order_id=oi.order_id
INNER JOIN new_orders n ON o.order_id=n.order_id
GROUP BY c.customer_state
ORDER BY Mean_freight DESC
LIMIT 5
```

Row	customer_state	Mean_freight	Mean_delivery_time	mean_diff_estimated
1	RR	42.98	27.83	17.43
2	PB	42.72	20.12	12.15
3	RO	41.07	19.28	19.08
4	AC	40.07	20.33	20.01
5	PI	39.15	18.93	10.68

Top 5 states with Lowest average freight value

- Using the new_orders CTE created above

```
SELECT c.customer_state, ROUND(AVG(oi.freight_value),2) Mean_freight, ROUND(AVG(ABS(n.time_
to_delivery)),2) Mean_delivery_time,ROUND(AVG(n.diff_estimated_delivery),2) mean_diff_estimated
FROM `Target_data.customers` c
```

```
INNER JOIN `Target_data.orders` o ON c.customer_id=o.customer_id
INNER JOIN `Target_data.order_items`oi ON o.order_id=oi.order_id
INNER JOIN new_orders n ON o.order_id=n.order_id
GROUP BY c.customer_state
ORDER BY Mean_freight
LIMIT 5
```

Row	customer_state	Mean_freight	Mean_delivery_time	mean_diff_estimated
1	SP	15.15	8.26	10.27
2	PR	20.53	11.48	12.53
3	MG	20.63	11.52	12.4
4	RJ	20.96	14.69	11.14
5	DF	21.04	12.5	11.27

- RR, PB and RO are the states with highest average freight value
- SP, PR and MG are the states with lowest average freight value

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

Top 5 states with highest average time to delivery

```
SELECT c.customer_state, ROUND(AVG(oi.freight_value),2) Mean_freight, ROUND(AVG(ABS(n.time_
to_delivery)),2) Mean_delivery_time,ABS(AVG(n.diff_estimated_delivery),2) mean_diff_estimated
FROM `Target_data.customers` c
INNER JOIN `Target_data.orders` o ON c.customer_id=o.customer_id
INNER JOIN `Target_data.order_items`oi ON o.order_id=oi.order_id
INNER JOIN new_orders n ON o.order_id=n.order_id
GROUP BY c.customer_state
ORDER BY Mean_delivery_time DESC
LIMIT 5
```

Row	customer_state	Mean_freight	Mean_delivery_time	mean_diff_estimated
1	RR	42.98	27.83	17.43
2	AP	34.01	27.75	17.44
3	AM	33.21	25.96	18.98
4	AL	35.84	23.99	7.98
5	PA	35.83	23.3	13.37

Top 5 states with lowest average time to delivery

```
SELECT c.customer_state, ROUND(AVG(oi.freight_value),2) Mean_freight, ROUND(AVG(ABS(n.time_
to_delivery)),2) Mean_delivery_time,ROUND(AVG(n.diff_estimated_delivery),2) mean_diff_estimated
FROM `Target_data.customers` c
INNER JOIN `Target_data.orders` o ON c.customer_id=o.customer_id
INNER JOIN `Target_data.order_items`oi ON o.order_id=oi.order_id
INNER JOIN new_orders n ON o.order_id=n.order_id
GROUP BY c.customer_state
ORDER BY Mean_delivery_time
```

```
LIMIT 5
```

Row	customer_state	Mean_freight	Mean_delivery_time	mean_diff_estimated
1	SP	15.15	8.26	10.27
2	PR	20.53	11.48	12.53
3	MG	20.63	11.52	12.4
4	DF	21.04	12.5	11.27
5	SC	21.47	14.52	10.67

- RR, AP and AM are the states with highest average time to delivery
- SP, PR and MG are the states with lowest average time to delivery

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

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

```
SELECT c.customer_state, ROUND(AVG(oi.freight_value),2) Mean_freight, ROUND(AVG(ABS(n.time_
to_delivery)),2) Mean_delivery_time,ROUND(AVG(n.diff_estimated_delivery),2) mean_diff_estimated
FROM `Target_data.customers` c
INNER JOIN `Target_data.orders` o ON c.customer_id=o.customer_id
INNER JOIN `Target_data.order_items`oi ON o.order_id=oi.order_id
INNER JOIN new_orders n ON o.order_id=n.order_id
GROUP BY c.customer_state
ORDER BY mean_diff_estimated DESC
LIMIT 5
```

Row	customer_state	Mean_freight	Mean_delivery_time	mean_diff_estimated
1	AC	40.07	20.33	20.01
2	RO	41.07	19.28	19.08
3	AM	33.21	25.96	18.98
4	AP	34.01	27.75	17.44
5	RR	42.98	27.83	17.43

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

```
SELECT c.customer_state, ROUND(AVG(oi.freight_value),2) Mean_freight, ROUND(AVG(ABS(n.time_
to_delivery)),2) Mean_delivery_time,ROUND(AVG(n.diff_estimated_delivery),2) mean_diff_estimated
FROM `Target_data.customers` c
INNER JOIN `Target_data.orders` o ON c.customer_id=o.customer_id
INNER JOIN `Target_data.order_items`oi ON o.order_id=oi.order_id
INNER JOIN new_orders n ON o.order_id=n.order_id
GROUP BY c.customer_state
ORDER BY mean_diff_estimated
LIMIT 5
```

Row	customer_state	Mean_freight	Mean_delivery_time	mean_diff_estimated
1	AL	35.84	23.99	7.98
2	MA	38.26	21.2	9.11
3	SE	36.65	20.98	9.17
4	ES	22.06	15.19	9.77
5	BA	26.36	18.77	10.12

- AC, RO and AM are the states where delivery happens much faster than the estimated date
- AL, MA and SE are the states where delivery happens less fast when compared to the estimated date

6. Payment type analysis:

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

```
WITH payment_table AS
(
SELECT p.payment_type, EXTRACT(YEAR FROM o.order_purchase_timestamp) Year, EXTRACT(M
ONTH FROM o.order_purchase_timestamp) Month, COUNT(o.order_id) order_count
FROM `Target_data.orders` o
INNER JOIN `Target_data.payments` p ON o.order_id=p.order_id
GROUP BY p.payment_type, EXTRACT(YEAR FROM o.order_purchase_timestamp), EXTRACT(MON
TH FROM o.order_purchase_timestamp)
)
SELECT *
FROM payment_table
ORDER BY payment_type,Year,Month
```

Row	payment_type	Year	Month	order_count
1	UPI	2016	10	63
2	UPI	2017	1	197
3	UPI	2017	2	398
4	UPI	2017	3	590
5	UPI	2017	4	496
6	UPI	2017	5	772
7	UPI	2017	6	707
8	UPI	2017	7	845
9	UPI	2017	8	938
10	UPI	2017	9	903

```
SELECT payment_type, COUNT(order_id) order_count
FROM `Target_data.payments`
GROUP BY payment_type
ORDER BY order_count DESC
```

Row	payment_type	order_count
1	credit_card	76795
2	UPI	19784
3	voucher	5775
4	debit_card	1529
5	not_defined	3

- Most customers opt for credit card for processing payments followed by UPI

2. Distribution of payment installments and count of orders

```
SELECT payment_installments, COUNT(order_id) order_count
FROM `Target_data.payments`
GROUP BY payment_installments
ORDER BY order_count DESC
```

payment_installments	order_count
1	52546
2	12413
3	10461
4	7098
10	5328
5	5239
8	4268
6	3920
7	1626
9	644

- Most payments are fulfilled in less than 3 installments

INSIGHTS

- The schema consists of 8 tables with details of customers, orders, sellers, payments etc.
- The dataset covers a time period of 2.11 years from 2016-09-04 to 2018-10-17
- The retailer has customers from 27 states in total in Brazil
- Customers span across 4415 cities in Brazil
- The top 3 states with maximum number of cities covered in the dataset are MG, SP and RS
- The month-by-month comparison of number of orders received shows an upward trend with a steep slope from January 2017 till March 2018. There is a small

slump till June,2018 and again climbs back up in July,2018 [Assumption: Data is incomplete for months 9 and 10 in 2018]

- The highest number of orders were received in November, 2017
- The number of orders received is almost equally distributed across all the weekdays. The number of customers see a slight dip on weekends.
- Since the complete data is only available for 2017, seasonality in terms of number of purchases is inconclusive
- They receive a greater number of customers post 7AM and before 12AM. There are very few customers visiting at dawn
- The number of customers visiting them is almost equally split across the day after 7PM from morning to night with the highest numbers in afternoon. This points to the fact that there are different categories of customers who like to shop in each segment of the day
- SP is the state in Brazil with the highest number of customers followed by RJ and MG.
- There is a 139.42% increase in the total cost of orders from 2017 to 2018
- PB, AL and AC are the states with highest average price per order which is an indication of a larger number of high spending customers
- RR, PB and RO are the states with highest average freight value
- SP, PR and MG are the states with lowest average freight value
- RR, AP and AM are the states with highest average time to delivery
- SP, PR and MG are the states with lowest average time to delivery
- AC, RO and AM are the states where delivery happens much faster than the estimated date
- AL, MA and SE are the states where delivery happens less fast when compared to the estimated date
- Most customers opt for credit card for processing payments followed by UPI
- Most payments are fulfilled in less than 3 installments

RECOMMENDATIONS

- The overall scenario is favourable with a solid year on year growth recorded. They can be optimistic about expanding their footprint in Brazil
- November, 2017 saw a sudden spike in number of orders. This can be attributed to the holiday season that follows. Tailored offers during these months can contribute to an increase in footfall
- There are more orders received per weekday than on weekends. Special offers, incentives etc. can be considered to attract more customers on weekends
- In the states with high average price per order, the product catalogue can be customized to include products with higher MRP
- States like RR have a high freight value and time to delivery. The delivery network can be examined to identify bottlenecks, if any. SP, PR and MG have low freight value and time to delivery. The best practices can be identified from the systems here.
- Credit card is the most preferred mode of payment and most customers opt for a single installment. Scrutinizing the availability of EMI options across products should be of priority combined with offers for credit card holder