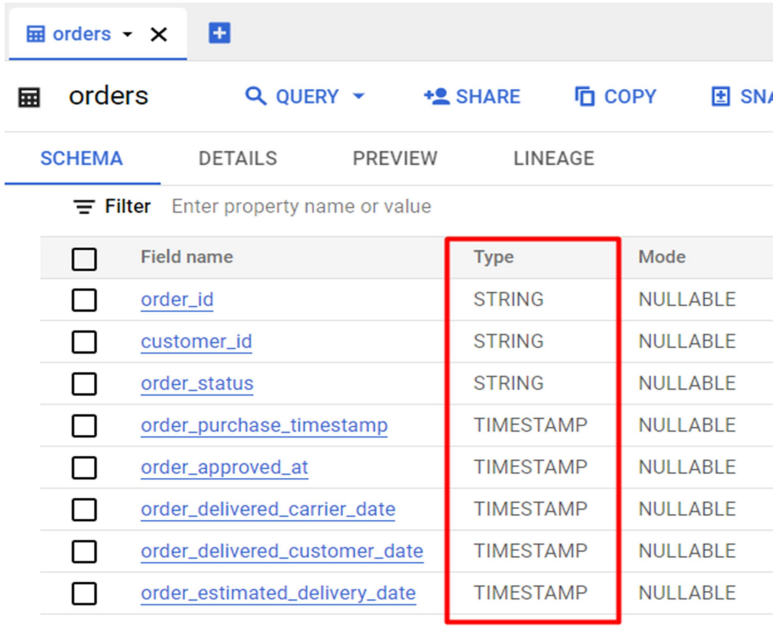
BUISNESS CASE STUDY SQL

Q1) Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset

1.1. Data type of columns in a table



1.2. Time period for which the data is given

```
SELECT  MIN(order_purchase_timestamp)
MIN_TIMESTAMP,MAX(order_purchase_timestamp)MAX_TIMESTAMP
FROM `target_dataset.orders`
```

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

1.3. Cities and States of customers ordered during the given period

```
SELECT DISTINCT
  c.customer_city City,c.customer_state State
FROM `target_dataset.orders` o, `target_dataset.customers` c
WHERE o.customer_id = c.customer_id
ORDER BY 1,2
```

Row	City	State
1	abadia dos dourados	MG
2	abadiania	GO
3	abaete	MG
4	abaetetuba	PA
5	abaiara	CE
6	abaira	BA
7	abare	BA
8	abatia	PR
9	abdon batista	SC
10	abelardo luz	SC

2. In-depth Exploration:

2.1. 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?

Ans. Yes, e-commerce is overall growing in Brazil, it can be clearly seen in the yearly level data attached below. We can see at the starting of every year at the month of January the number of orders and sales goes up and at December sales and the number of orders goes down. But we don't have enough data to state that this seasonality will always be true.

YEARLY LEVEL DATA:

```
SELECT
    EXTRACT(YEAR FROM o.order_purchase_timestamp) year,
    ROUND(SUM(p.payment_value),2) total_sales, COUNT(o.order_id)
number_of_orders
 FROM `target_dataset.orders` o, `target_dataset.payments` p
 WHERE o.order_id = p.order_id
 GROUP BY EXTRACT(YEAR FROM o.order_purchase_timestamp)
```

Row	year	total_sales	number_of_orders
1	2017	7249746.73	47525
2	2018	8699763.05	56015
3	2016	59362.34	346

QUARTERLY LEVEL DATA:

```
SELECT year,quarter, total_sales,ROUND(100*(total_sales-LAG(total_sales,1)
OVER(ORDER BY year,quarter))/total_sales,2) pct_increase_in_sales,
        count_of_orders, ROUND(100*(count_of_orders - LAG(count_of_orders,1)
OVER(ORDER BY year,quarter))/count_of_orders,2) pct_increase_in_orders
FROM
(SELECT year,quarter, ROUND(sum(payment_value),2) total_sales,
count(order_id) count_of_orders
FROM
(SELECT YEAR, CASE WHEN month BETWEEN 1 AND 3 THEN 'Q1'WHEN month BETWEEN 4
AND 6 THEN 'Q2'
              WHEN month BETWEEN 7 AND 9 THEN 'Q3' ELSE 'Q4' END
quarter,order_id,payment_value
FROM
(
  SELECT
  EXTRACT(YEAR FROM o.order_purchase_timestamp) year,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) month,
  p.payment_value, o.order_id
  FROM `target_dataset.orders` o, `target_dataset.payments` p
  WHERE o.order_id = p.order_id
) )
GROUP BY year,quarter
)ORDER BY 1,2
```

Row	year	quarter	total_sales	pct_increase_in	count_of_orders	pct_increase_in
1	2016	Q3	252.24	*null*	3	*null*
2	2016	Q4	59110.1	99.57	343	99.13
3	2017	Q1	880259.65	93.28	5573	93.85
4	2017	Q2	1521983.23	42.16	9951	44.0
5	2017	Q3	1994541.69	23.69	13383	25.64
6	2017	Q4	2852962.16	30.09	18618	28.12
7	2018	Q1	3267119.64	12.68	22027	15.48
8	2018	Q2	3338648.13	2.14	20763	-6.09
9	2018	Q3	2093405.61	-59.48	13221	-57.05
10	2018	Q4	589.67	-354913.08	4	-330425.0

MONTHLY LEVEL DATA:

```
SELECT year,month, total_sales, ROUND(100*(total_sales-LAG(total_sales,1)
OVER(ORDER BY year,month))/total_sales,2) pct_increase_in_sales,
                  count_of_orders, ROUND(100*(count_of_orders -
LAG(count_of_orders,1) OVER(ORDER BY year,month))/count_of_orders,2)
pct_increase_in_orders
FROM
(SELECT year,month, ROUND(sum(payment_value),2) total_sales, count(order_id)
count_of_orders
FROM
(
  SELECT
```

```
  EXTRACT(YEAR FROM o.order_purchase_timestamp) year,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) month,
  p.payment_value, o.order_id
  FROM `target_dataset.orders` o, `target_dataset.payments` p
  WHERE o.order_id = p.order_id
)
GROUP BY year,month
)ORDER BY 1,2
```

Row	year	month	total_sales	pct_increase_in	count_of_orders	pct_increase_in
1	2016	9	252.24	*null*	3	*null*
2	2016	10	59090.48	99.57	342	99.12
3	2016	12	19.62	-301074.72	1	-34100.0
4	2017	1	138488.04	99.99	850	99.88
5	2017	2	291908.01	52.56	1886	54.93
6	2017	3	449863.6	35.11	2837	33.52
7	2017	4	417788.03	-7.68	2571	-10.35
8	2017	5	592918.82	29.54	3944	34.81
9	2017	6	511276.38	-15.97	3436	-14.78
10	2017	7	592382.92	13.69	4317	20.41

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

Ans. Brazilian Customers tend to buy mostly at Afternoon.

```
SELECT phases, COUNT(order_id) number_of_orders, ROUND(SUM(sales),2)
total_sales
FROM
(SELECT CASE WHEN hour BETWEEN 0 AND 6 THEN 'Dawn' WHEN hour BETWEEN 7 AND 12
THEN 'Morning'
        WHEN hour BETWEEN 13 AND 18 THEN 'Afternoon' ELSE 'Night' END phases,
sales, order_id
FROM
(SELECT
  EXTRACT(hour FROM o.order_purchase_timestamp) hour,
  p.payment_value sales, o.order_id
  FROM `target_dataset.orders` o, `target_dataset.payments` p
  WHERE o.order_id = p.order_id
))
GROUP BY phases
```

Row	phases	number_of_orders	total_sales
1	Morning	28950	4469481.39
2	Dawn	5506	781003.11
3	Afternoon	39691	6252719.87
4	Night	29739	4505667.75

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

3.1. Get month on month orders by states

```
SELECT state,year,month,no_of_orders, SUM(no_of_orders) OVER(PARTITION BY
state ORDER BY year,month) Cum_orders
FROM
(SELECT
    COUNT(order_id) no_of_orders,
    state,
    EXTRACT(YEAR FROM order_purchase_timestamp) YEAR,
    EXTRACT(MONTH FROM order_purchase_timestamp) MONTH
FROM
  (  SELECT o.order_id,o.order_purchase_timestamp, c.customer_state state
    FROM `target_dataset.orders` o, `target_dataset.customers` c
    WHERE o.customer_id = c.customer_id
  )
 GROUP BY  state,EXTRACT(YEAR FROM order_purchase_timestamp),
           EXTRACT(MONTH FROM order_purchase_timestamp)
)
ORDER BY state,year,cast(month AS INT)
```

Row	state	year	month	no_of_orders	Cum_orders
1	AC	2017	1	2	2
2	AC	2017	2	3	5
3	AC	2017	3	2	7
4	AC	2017	4	5	12
5	AC	2017	5	8	20
6	AC	2017	6	4	24
7	AC	2017	7	5	29
8	AC	2017	8	4	33
9	AC	2017	9	5	38
10	AC	2017	10	6	44

3.2. Distribution of customers across the states in Brazil

```
SELECT count(customer_id) no_of_customers,customer_state
FROM `target_dataset.customers`
GROUP BY customer_state
ORDER BY 1
```

Row	no_of_customers	customer_state
1	46	RR
2	68	AP
3	81	AC
4	148	AM
5	253	RO
6	280	TO
7	350	SE
8	413	AL
9	485	RN
10	495	PI

Observations:

a. State SP has the highest number of customers followed by RJ, MG, RS, PR.
b. State RR has least number of customers followed by AP, AC, AM, RO

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

4.1 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,month,total_cost,
IFNULL(ROUND((total_cost - LAG(total_cost,1) OVER(ORDER BY year,month)
)/LAG(total_cost,1) OVER(ORDER BY year,month),2),0)
pct_increase_in_cost_of_orders
FROM
        (
        SELECT ROUND(SUM(p.payment_value),2) total_cost,
                EXTRACT(YEAR FROM o.order_purchase_timestamp) year,
                EXTRACT(MONTH FROM o.order_purchase_timestamp) month
        FROM  `target_dataset.orders` o, `target_dataset.payments` p
        WHERE o.order_id = p.order_id
        GROUP BY EXTRACT(YEAR FROM o.order_purchase_timestamp),EXTRACT(MONTH
FROM o.order_purchase_timestamp)
        )
WHERE year IN (2017,2018) AND month BETWEEN 1 AND 8
ORDER BY year,month
```

Row	year	month	total_cost	pct_increase_in_cost_of_orders
1	2017	1	138488.04	0.0
2	2017	2	291908.01	1.11
3	2017	3	449863.6	0.54
4	2017	4	417788.03	-0.07
5	2017	5	592918.82	0.42
6	2017	6	511276.38	-0.14
7	2017	7	592382.92	0.16
8	2017	8	674396.32	0.14
9	2018	1	1115004.18	0.65
10	2018	2	992463.34	-0.11

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

```
SELECT c.customer_state, ROUND(AVG(oi.price),2) Mean_of_Price,
ROUND(SUM(oi.price),2) Sum_of_Price,
        ROUND(AVG(oi.freight_value),2) Mean_of_Freight_value,
ROUND(SUM(oi.freight_value),2) Sum_of_Freight_value,
FROM `target_dataset.orders` o, `target_dataset.order_items` oi,
`target_dataset.customers` c
WHERE o.order_Id = oi.order_id AND o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER by 1
```

Row	customer_state	Mean_of_Price	Sum_of_Price	Mean_of_Freight_value	Sum_of_Freight_value
1	AC	173.73	15982.95	40.07	3686.75
2	AL	180.89	80314.81	35.84	15914.59
3	AM	135.5	22356.84	33.21	5478.89
4	AP	164.32	13474.3	34.01	2788.5
5	BA	134.6	511349.99	26.36	100156.68
6	CE	153.76	227254.71	32.71	48351.59
7	DF	125.77	302603.94	21.04	50625.5
8	ES	121.91	275037.31	22.06	49764.6
9	GO	126.27	294591.95	22.77	53114.98
10	MA	145.2	119648.22	38.26	31523.77

Observations:

a. Average spent on every order is more for people of state PB than any other state i.e. around 191 denominations in their currency.

b. Considering Sum of Price of all the products state SP ranks the highest. Its average order is around 110.

c. Also found RR has the highest average of freight value to average of price ratio.

5. Analysis on sales, freight and delivery time

5.1. Calculate days between purchasing, delivering and estimated delivery

Considering order_delivered_carrier_date to be the delivery date.

```
SELECT order_id,
      DATE_DIFF(order_delivered_carrier_date, order_purchase_timestamp,DAY)
Days_between_purchase__delivery,
      DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp,DAY)
Days_between_purchase_estimated_delivery,
      DATE_DIFF(order_estimated_delivery_date, order_delivered_carrier_date
,DAY)Days_between_estimated_delivery_actual_delivery
FROM `target_dataset.orders`
ORDER BY 1
```

Row	order_id	Days_between_purchase__delivery	Days_between_purchase_estimated_delivery	Days_between_estimated_delivery_actual_delivery
1	00010242fe8c5a6d1ba2dd792...	6	15	9
2	00018f77f2f0320c557190d7a1...	8	18	10
3	000229ec398224ef6ca0657da...	1	21	19
4	00024acbcdf0a6daa1e931b03...	2	11	9
5	00042b26cf59d7ce69dfabb4e...	11	40	28
6	00048cc3ae777c65dbb7d2a06...	1	21	19
7	00054e8431b9d7675808bcb8...	1	24	22
8	000576fe39319847cbb9d288c...	1	20	19
9	0005a1a1728c9d785b8e2b08...	8	9	0
10	0005f50442cb953dcd1d21e1f...	1	20	19

Observations: Considering order_delivered_carrier_date as the date the order got delivered,
a. Around 1.8% of the order_delivered_carrier_date is null/blank
b. Including these 1.8% records around 98% of order were delivered on time i.e. on or before estimated delivery date.

5.2. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

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

Changes: As per the above formula **time_to_delivery** will always be negative as order_delivered_customer_date will be always greater than order_purchase_timestamp. Below query

and results are based on considering
time_to_delivery = order_delivered_customer_date – order_purchase_timestamp

```
SELECT order_id , COUNT(ORDER_ID) OVER() TOTAL_ROWS,
      DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)
time_to_delivery ,
      DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date
,DAY) diff_estimated_delivery
FROM `target_dataset.orders`
ORDER BY 1
```

Row	order_id	time_to_delivery	diff_estimated_delivery
1	00010242fe8c5a6d1ba2dd792...	7	8
2	00018f77f2f0320c557190d7a1...	16	2
3	000229ec398224ef6ca0657da...	7	13
4	00024acbcdf0a6daa1e931b03...	6	5
5	00042b26cf59d7ce69dfabb4e...	25	15
6	00048cc3ae777c65dbb7d2a06...	6	14
7	00054e8431b9d7675808bcb8...	8	16
8	000576fe39319847cbb9d288c...	5	15
9	0005a1a1728c9d785b8e2b08...	9	0
10	0005f50442cb953dcd1d21e1f...	2	18

Observations:

a. Around 3% of the order_delivered_customer_date are blank/null.

b. Including these records around 87.5% of the orders delivered were delivered before the estimated delivered date.

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

```
SELECT c.customer_state ,
      ROUND(AVG(oi.freight_value),2) mean_of_freight_value,
      ROUND(AVG(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp,DAY)),2) mean_of_time_to_delivery ,
      ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date ,DAY)),2) mean_of_diff_estimated_delivery
FROM `target_dataset.orders` o, `target_dataset.order_items` oi,
`target_dataset.customers` c
WHERE o.order_id = oi.order_id AND o.customer_id = c.customer_id
GROUP BY c.customer_state
```

Row	customer_state	mean_of_freight_value	mean_of_time_to_delivery	mean_of_diff_estimated_delivery
1	MT	28.17	17.51	13.64
2	MA	38.26	21.2	9.11
3	AL	35.84	23.99	7.98
4	SP	15.15	8.26	10.27
5	MG	20.63	11.52	12.4
6	PE	32.92	17.79	12.55
7	RJ	20.96	14.69	11.14
8	DF	21.04	12.5	11.27
9	RS	21.74	14.71	13.2
10	SE	36.65	20.98	9.17

5.4. Sort the data to get the following:

```
SELECT c.customer_state ,
      ROUND(AVG(oi.freight_value),2) mean_of_freight_value,
      ROUND(AVG(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp,DAY)),2) mean_of_time_to_delivery ,
      ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date ,DAY)),2) mean_of_diff_estimated_delivery
FROM `target_dataset.orders` o, `target_dataset.order_items` oi,
`target_dataset.customers` c
WHERE o.order_id = oi.order_id AND o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY c.customer_state
```

Row	customer_state	mean_of_freight_value	mean_of_time_to_delivery	mean_of_diff_estimated_delivery
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
6	CE	32.71	20.54	10.26
7	DF	21.04	12.5	11.27
8	ES	22.06	15.19	9.77
9	GO	22.77	14.95	11.37
10	MA	38.26	21.2	9.11

5.5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

```
SELECT c.customer_state ,
      ROUND(AVG(oi.freight_value),2) mean_of_freight_value,
      ROUND(AVG(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp,DAY)),2) mean_of_time_to_delivery ,
      ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date ,DAY)),2) mean_of_diff_estimated_delivery
FROM `target_dataset.orders` o, `target_dataset.order_items` oi,
`target_dataset.customers` c
WHERE o.order_id = oi.order_id AND o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY 2
LIMIT 5
```

Row	customer_state	mean_of_freight_value	mean_of_time_to_delivery	mean_of_diff_estimated_delivery
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

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

```
SELECT c.customer_state ,
      ROUND(AVG(oi.freight_value),2) mean_of_freight_value,
      ROUND(AVG(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp,DAY)),2) mean_of_time_to_delivery ,
      ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date ,DAY)),2) mean_of_diff_estimated_delivery
FROM `target_dataset.orders` o, `target_dataset.order_items` oi,
`target_dataset.customers` c
WHERE o.order_id = oi.order_id AND o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY 3
LIMIT 5
```

Row	customer_state	mean_of_freight_value	mean_of_time_to_delivery	mean_of_diff_estimated_delivery
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

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

```sql
SELECT c.customer_state ,
      ROUND(AVG(oi.freight_value),2) mean_of_freight_value,
      ROUND(AVG(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp,DAY)),2) mean_of_time_to_delivery ,
      ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date ,DAY)),2) mean_of_diff_estimated_delivery
FROM `target_dataset.orders` o, `target_dataset.order_items` oi,
`target_dataset.customers` c
WHERE o.order_id = oi.order_id AND o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY 4
LIMIT 5
```

Row	customer_state	mean_of_freight_value	mean_of_time_to_delivery	mean_of_diff_estimated_delivery
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

Observations:

a. RR, AP, AM, AL, PA are the states who get late deliveries(mean of time to delivery).

b. SP, PR, MG, DF, SC are the states who get earliest deliveries(mean of time to delivery).

c. RR, PB, RO, AC, PI has the highest amount of freight values.

By c and a we can assume that state RR has some transportation difficulties as it's freight values are high as well as time taken to deliver to the state is also high.

6. Payment type analysis

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

```sql
SELECT year,month,payment_type,count_of_orders
FROM
(
  SELECT
    EXTRACT(YEAR FROM o.order_purchase_timestamp) year,
    EXTRACT(month FROM o.order_purchase_timestamp) month,
    p.payment_type,
    COUNT(p.order_id) count_of_orders
  FROM
    `target_dataset.payments` p,
    `target_dataset.orders` o
  WHERE o.order_id = p.order_id
  GROUP BY EXTRACT(YEAR FROM o.order_purchase_timestamp) ,
      EXTRACT(month FROM o.order_purchase_timestamp),
      p.payment_type
```

```
)
ORDER BY 1,2,3
```

Row	year	month	payment_type	count_of_orders
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
8	2017	1	credit_card	583
9	2017	1	debit_card	9
10	2017	1	voucher	61

6.2. Count of orders based on the no. of payment instalments

```
SELECT  payment_installments, COUNT(order_id) count_of_orders
FROM `target_dataset.payments`
GROUP BY payment_installments
```

Row	payment_installments	count_of_orders
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644

Observations:

a. 1, 2, 3, 4, 10 are the most preferred payment_installments.

b. Most preferred payment type is credit card then UPI followed by debit card.

Insights:

- State SP has the highest number of customers
- State SP has the highest number of customers followed by RJ, MG, RS, PR.
- State SP has the highest number of sellers
- Average spent on every order is more for people of state PB than any other state i.e. around 191 denominations in their currency.
- State RR has least number of customers followed by AP, AC, AM, RO
- Also found RR has the highest average of freight value to average of price ratio.

- Considering order_delivered_carrier_date as the date the order got delivered,
    - Around 1.8% of the order_delivered_carrier_date is null/blank
    - Including these 1.8% records around 98% of orders were delivered on time i.e. on or before estimated delivery date.
- Considering order_delivered_customer_date as the date the order got delivered,
    - Around 3% of the order_delivered_customer_date are blank/null.
    - Including these records around 87.5% of the orders delivered were delivered before the estimated delivered date.

- RR, AP, AM, AL, PA are the states who get late deliveries (mean of time to delivery).
- SP, PR, MG, DF, SC are the states who get earliest deliveries (mean of time to delivery).
- RR, PB, RO, AC, PI has the highest amount of freight values.
- We can assume that state RR has some transportation difficulties as its freight values are high as well as time taken to deliver to the state is also high.
- 1, 2, 3, 4, 10 are the most preferred payment instalments.
- Most preferred payment type is credit card then UPI followed by debit card.

- Product Photos Qty not always plays important role for purchasing the product, as even if photos qty is less customers still buy that high number of products.
- High frequency products i.e. products that are ordered more than 50 times has an average rating of more than 3 rating.
- Bed Table Bath is the most bought product category followed by Health Beauty, Sport Leisure, Furniture Decorations and Computer accessories.
- Customers tend to buy least items from these product categories (Art, PCS, Signalization and Safety, Construction and Tools and Christmas articles).

- There isn't much difference in average of weight or volume of items delivered to every state, attaching below top ten and bottom ten states based on average of weight and volume of items.

Row	customer_state	AVG_VOLUME_CM3	AVG_WEIGHT_G
1	TO	17654.84	2642.52
2	SE	18680.92	2441.6
3	PI	15695.54	2388.7
4	RO	14787.85	2379.68
5	MS	16639.02	2374.04
6	MT	15980.24	2314.47
7	RJ	15676.07	2270.79
8	PA	16140.18	2253.97
9	BA	15498.65	2223.75
10	PB	16381.05	2209.81

18	PE	14148.47	1948.8
19	MA	14982.6	1946.32
20	AL	14285.09	1943.73
21	GO	14529.6	1924.58
22	AC	12723.43	1915.07
23	AP	11631.3	1850.65
24	RN	13387.88	1783.72
25	RR	14541.88	1760.77
26	DF	12860.02	1700.25
27	AM	11920.9	1538.47

Recommendations:

- Target can tie up with any credit card company(promote a credit card company or run advertisements for that company) which would likely make customers to go for this credit card as most of the customers prefer credit card payment.
- It should offer schemes like NO COST EMI options up to 1-12 payment instalments, which the customer prefers the most.
- Orders from state RR could focus more on the logistics aspect of business as its takes highest time to deliver and freight values are also high.
- Could recommend the customers of state PB some luxurious items/ premium range items.
- Can offer customers of state SP to have various delivery options (fastest delivery and normal delivery).
- Target can run sales during 1300 to 1800 hours as customers buy more between these hours.
- Target can put items on these product categories (Bed Table Bath, Health Beauty, Sport Leisure, Furniture Decorations and Computer accessories) for frequent sales as customers tend to buy more of these product categories.
- For the least bought product categories Target could do a clearance sale