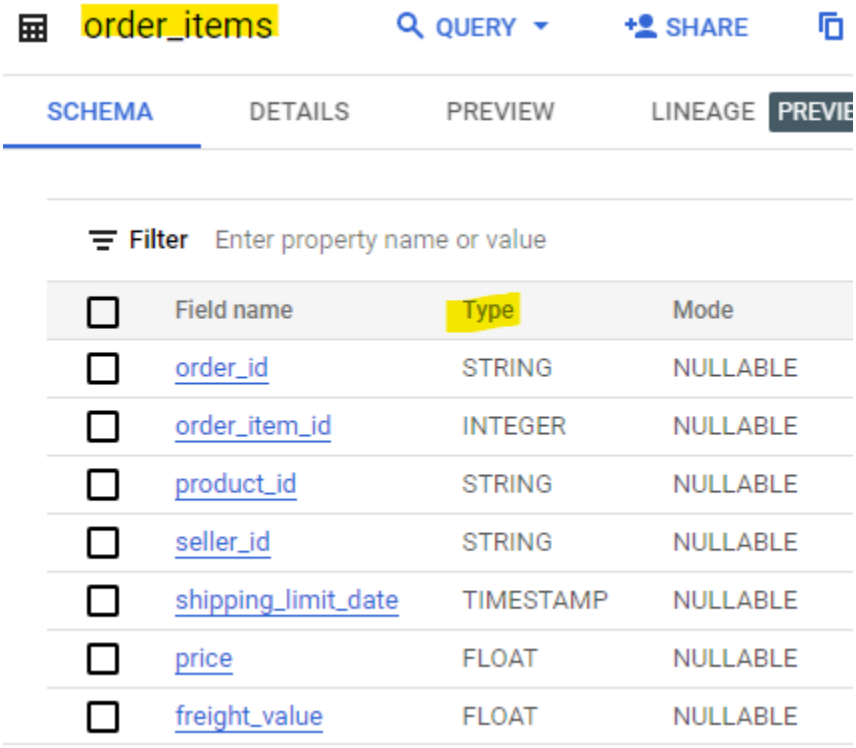
1) **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset**

   a) **Data type of columns in a table**



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

Actionable Insight:

Data type for each column can be observed in the "Type" attribute of each table, which helps us understand the type of data in the corresponding column of every table. Data types like String, Integer, Float, Timestamp can be observed in the attributes of the given data.

   b) **Time period for which the data is given**

CODE:

```
SELECT MIN(order_purchase_timestamp) AS start_date,
MAX(order_purchase_timestamp) AS end_date
FROM `Target_Business_Case.orders`
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION D

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

Actionable Insight:

The time period of the given data is 04-09-2016 to 17-10-2018.
Exactly '2 years 1 month 13 days' of data is presented in the business case.

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

CODE:

```
SELECT DISTINCT
customer_city,
customer_state
FROM `Target_Business_Case.customers`
ORDER BY customer_city, customer_state
```

Query results

JOB INFORMATION | RESULTS

Row	customer_city	customer_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

Actionable Insight:

The States and Cities from which the customers have purchased the data

can be seen using the above SQL code. This data lets us know the stretch till which Target has customers geographically.

---

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?

CODE:

```
SELECT
EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
count(order_id) AS Number_of_orders
FROM `Target_Business_Case.orders`
GROUP BY year, month
ORDER BY year, month
```

Query results

JOB INFORMATION | RESULTS | JSON | EXEC

Row	year	month	Number_of_orders
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

Actionable Insight:

There is definitely a growing trend in e-commerce in Brazil. Initially, the number of orders placed were extremely low. As the time passed, we could see a gradual upward inclination of orders placed.

To understand the Seasonality:

CODE:

```
SELECT
EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
count(order_id) AS Number_of_orders
FROM `Target_Business_Case.orders`
GROUP BY month
ORDER BY Number_of_orders DESC
```

Query results

JOB INFORMATION | RESULTS

Row	month	Number_of_orders
1	8	10843
2	5	10573
3	7	10318
4	3	9893
5	6	9412
6	4	9343
7	2	8508
8	1	8069
9	11	7544
10	12	5674
11	10	4959
12	9	4305

Actionable Insight:

From the above code, during the 8th, 5th and 7th months, we can see the peak in orders placed, compared to other months.

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

CODE:

```
SELECT DISTINCT
time_of_the_day,
count(time_of_the_day) OVER(PARTITION BY
time_of_the_day) as count_time_of_day
```

```
FROM
(SELECT
CASE
  WHEN EXTRACT(hour FROM order_purchase_timestamp)
between 0 AND 5
  THEN "Dawn"
  WHEN EXTRACT(hour FROM order_purchase_timestamp)
between 6 AND 11
  THEN "Morning"
  WHEN EXTRACT(hour FROM order_purchase_timestamp)
between 12 AND 17
  THEN "Afternoon"
  WHEN EXTRACT(hour FROM order_purchase_timestamp)
between 18 AND 23
  THEN "Night"
END AS time_of_the_day
FROM `Target_Business_Case.orders`)
ORDER BY count_time_of_day DESC;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXEC

Row	time_of_the_day	count_time_of_day
1	Afternoon	38361
2	Night	34100
3	Morning	22240
4	Dawn	4740

Actionable Insight:

It can be clearly observed that during Afternoon Brazilian customers tend to buy more compared to any other time of the day.

---

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

a) Get month on month orders by states

CODE:

```
SELECT DISTINCT
cus.customer_state,
EXTRACT(MONTH FROM ord.order_purchase_timestamp) AS
month,
count(ord.order_id) AS Number_of_orders
FROM `Target_Business_Case.customers` AS cus
JOIN `Target_Business_Case.orders` AS ord
ON cus.customer_id=ord.customer_id
GROUP BY customer_state, month
ORDER BY customer_state, month
```

Query results

JOB INFORMATION | RESULTS | JSON

Row	customer_state	month	Number_of_orders
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6
11	AC	11	5
12	AC	12	5
13	AL	1	39
14	AL	2	39

Actionable Insight:

From the above code, we can ascertain the monthly sales for each state. It can be observed that the sales in the 5th month for the State 'AC' are the highest and in the 3rd month, the lowest.

b) Distribution of customers across the states in Brazil

CODE:

```
SELECT DISTINCT
customer_state,
count(customer_id) AS Number_of_people
FROM `Target_Business_Case.customers`
GROUP BY customer_state
ORDER BY Number_of_people DESC
```

Query results

JOB INFORMATION | RESULTS | JS

Row	customer_state	Number_of_people
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

Actionable Insight:

From the distribution of customers across the States of Brazil, we can find that the state 'SP' has the maximum number of customers compared to other regions. The state 'RR' has the least number of customers with just 46 of them.

---

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

CODE:

```
SELECT DISTINCT
Year,
ROUND(((SUM(pay.payment_value)) -
(LAG(SUM(pay.payment_value)) OVER(ORDER BY
year)))/(LAG(SUM(pay.payment_value)) OVER(ORDER BY
year))*100,2) AS percentage_cost_increase
FROM
(SELECT
order_id,
EXTRACT (YEAR FROM order_purchase_timestamp) AS year,
EXTRACT (MONTH FROM order_purchase_timestamp) AS month
FROM `Target_Business_Case.orders`) AS ord
JOIN `Target_Business_Case.payments` As pay
ON ord.order_id=pay.order_id
WHERE ord.year BETWEEN 2017 AND 2018 AND ord.month
BETWEEN 1 AND 8
GROUP BY ord.year
ORDER BY ord.year
```

Query results

JOB INFORMATION | RESULTS | JSON

Row	year	percentage_cost_increase
1	2017	*null*
2	2018	136.98

Actionable Insight:

The cost of orders has been increased from 2017 to 2018 (during the months from Jan to Aug) by 136.98%.

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

CODE:

```
SELECT DISTINCT
cus.customer_state,
ROUND(AVG (ordi.price) OVER(PARTITION BY
```

```
cus.customer_state),2) as mean_price,
ROUND(AVG (ordi.freight_value) OVER(PARTITION BY
cus.customer_state),2) as mean_freight_value,
ROUND(SUM (ordi.price) OVER(PARTITION BY
cus.customer_state),2) as price_sum,
ROUND(SUM (ordi.freight_value) OVER(PARTITION BY
cus.customer_state),2) as freight_value_sum
FROM `Target_Business_Case.customers` AS cus
JOIN `Target_Business_Case.orders` AS ord
ON cus.customer_id =ord.customer_id
JOIN `Target_Business_Case.order_items` AS ordi
ON ord.order_id = ordi.order_id
ORDER BY cus.customer_state
```

Query results

SAVE RESU

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPI

Row	customer_state	mean_price	mean_freight_value	price_sum	freight_value_sum
1	AC	173.73	40.07	15982.95	3686.75
2	AL	180.89	35.84	80314.81	15914.59
3	AM	135.5	33.21	22356.84	5478.89
4	AP	164.32	34.01	13474.3	2788.5
5	BA	134.6	26.36	511349.99	100156.68
6	CE	153.76	32.71	227254.71	48351.59
7	DF	125.77	21.04	302603.94	50625.5
8	ES	121.91	22.06	275037.31	49764.6
9	GO	126.27	22.77	294591.95	53114.98
10	MA	145.2	38.26	119648.22	31523.77
11	MG	120.75	20.63	1585308.03	270853.46
12	MS	142.63	23.37	116812.64	19144.03
13	MT	148.3	28.17	156453.53	29715.43
14	PA	165.69	35.83	178947.81	38699.3

Actionable Insight:

The Sum and Average of price and freight value is arranged state wise in the above table.

5) Analysis on sales, freight and delivery time

a) Calculate days between purchasing, delivering and estimated delivery

CODE:

```
SELECT DISTINCT
order_id,
TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY) AS purchased_delivered_days,
TIMESTAMP_DIFF(order_estimated_delivery_date,order_purchase_timestamp,DAY) AS purchased_estimated_days,
TIMESTAMP_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY) AS estimated_delivered_days
FROM `Target_Business_Case.orders`
ORDER BY order_id
```

Query results

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

Row	order_id	purchased_delivered_days	purchased_estimated_days	estimated_delivered_days
1	00010242fe8c5a6d1ba2dd792...	7	15	8
2	00018f77f2f0320c557190d7a1...	16	18	2
3	000229ec398224ef6ca0657da...	7	21	13
4	00024acbcdf0a6daa1e931b03...	6	11	5
5	00042b26cf59d7ce69dfabb4e...	25	40	15
6	00048cc3ae777c65dbb7d2a06...	6	21	14
7	00054e8431b9d7675808bcb8...	8	24	16
8	000576fe39319847cbb9d288c...	5	20	15
9	0005a1a1728c9d785b8e2b08...	9	9	0
10	0005f50442cb953dcd1d21e1f...	2	20	18

Actionable Insight:

The difference between purchase, delivery and estimated delivery days are displayed as shown above. This result helps us understand how quickly an order is delivered since the order is placed, and also it shows the gap between the estimated delivery and actual delivery.

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

i) **time_to_delivery = order_purchase_timestamp-order_delivered_customer_date**

ii) **diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date**

CODE:

```
SELECT DISTINCT
order_id,
TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase_timestamp,HOUR) AS time_to_delivery,
TIMESTAMP_DIFF(order_estimated_delivery_date,order_delivered_customer_date,HOUR) AS diff_estimated_delivery
FROM `Target_Business_Case.orders`
ORDER BY order_id
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	order_id	time_to_delivery	diff_estimated_delivery
1	00010242fe8c5a6d1ba2dd792...	182	192
2	00018f77f2f0320c557190d7a1...	389	55
3	000229ec398224ef6ca0657da...	190	322
4	00024acbcdf0a6daa1e931b03...	147	130
5	00042b26cf59d7ce69dfabb4e...	602	367
6	00048cc3ae777c65dbb7d2a06...	160	346
7	00054e8431b9d7675808bcb8...	202	385
8	000576fe39319847cbb9d288c...	121	369
9	0005a1a1728c9d785b8e2b08...	239	-18
10	0005f50442cb953dcd1d21e1f...	51	438

Note: time_to_delivery and diff_estimated_delivery has been represented in terms of hours.

Actionable Insight:

The time_to_delivery and diff_estimated_delivery for each order provides the time taken for actual delivery and the time estimated for delivery in terms of hours. The negative sign in diff_estimated_delivery attribute indicates that the order was delivered way before the estimated delivery time.

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

CODE:

```
SELECT DISTINCT
cus.customer_state,
ROUND(AVG (ordi.freight_value) OVER(PARTITION BY
cus.customer_state ORDER BY cus.customer_state),3) as
mean_freight_value,
ROUND(AVG(ord.time_to_delivery) OVER(PARTITION BY
cus.customer_state ORDER BY cus.customer_state),3) as
mean_time_to_delivery,
ROUND(AVG(ord.diff_estimated_delivery) OVER(PARTITION
BY cus.customer_state ORDER BY cus.customer_state),3)
as mean_diff_estimated_delivery
FROM `Target_Business_Case.customers` as cus
JOIN
(SELECT
customer_id,
order_id,
TIMESTAMP_DIFF(order_delivered_customer_date,order_pur
chase_timestamp,HOUR) AS time_to_delivery,
TIMESTAMP_DIFF(order_estimated_delivery_date,order_del
ivered_customer_date,HOUR) AS diff_estimated_delivery
FROM `Target_Business_Case.orders`) as ord
ON cus.customer_id = ord.customer_id
JOIN `Target_Business_Case.order_items` as ordi
ON ord.order_id = ordi.order_id
```

Query results

SAVE RES

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRA

Row	customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	AC	40.073	496.67	487.593
2	AL	35.844	587.227	193.133
3	AM	33.205	632.847	460.969
4	AP	34.006	676.457	426.012
5	BA	26.364	461.451	246.573
6	CE	32.714	503.205	249.532
7	DF	21.041	310.518	275.42
8	ES	22.059	375.08	238.415
9	GO	22.767	369.184	277.892
10	MA	38.257	519.059	221.118

Note: mean_time_to_delivery and mean_diff_estimated_delivery has been represented in terms of hours.

Actionable Insight:

The average freight value, average time taken for delivery and the average difference of estimated delivery time can be seen above, state wise.

d) Sort the data to get the following:

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

Highest average freight value:

CODE:

```
SELECT DISTINCT
cus.customer_state,
ROUND(AVG (ordi.freight_value) OVER(PARTITION BY
cus.customer_state),3) as mean_freight_value,
ROUND(AVG(ord.time_to_delivery) OVER(PARTITION BY
cus.customer_state),3) as mean_time_to_delivery,
ROUND(AVG(ord.diff_estimated_delivery) OVER(PARTITION BY
cus.customer_state),3) as mean_diff_estimated_delivery
```

```
FROM `Target_Business_Case.customers` as cus
JOIN
(SELECT
customer_id,
order_id,
TIMESTAMP_DIFF(order_delivered_customer_date,order_purch
ase_timestamp,HOUR) AS time_to_delivery,
TIMESTAMP_DIFF(order_estimated_delivery_date,order_deliv
ered_customer_date,HOUR) AS diff_estimated_delivery
FROM `Target_Business_Case.orders`) as ord
ON cus.customer_id = ord.customer_id
JOIN `Target_Business_Case.order_items` as ordi
ON ord.order_id = ordi.order_id
ORDER BY mean_freight_value DESC
LIMIT 5
```

Query results

SAVE RESUI

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPI

Row	customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	RR	42.984	676.978	422.435
2	PB	42.724	493.669	296.527
3	RO	41.07	473.271	463.769
4	AC	40.073	496.67	487.593
5	PI	39.148	464.751	260.117

Note: mean_time_to_delivery and mean_diff_estimated_delivery has been represented in terms of hours.

Actionable Insight:

States with codes RR, PB, RO, AC, PI are the top 5 States that have the highest mean freight values.

Lowest average freight value:

CODE:

```
SELECT DISTINCT
cus.customer_state,
ROUND(AVG (ordi.freight_value) OVER(PARTITION BY
cus.customer_state),3) as mean_freight_value,
```

```
ROUND(AVG(ord.time_to_delivery) OVER(PARTITION BY
cus.customer_state),3) as mean_time_to_delivery,
ROUND(AVG(ord.diff_estimated_delivery) OVER(PARTITION BY
cus.customer_state),3) as mean_diff_estimated_delivery
FROM `Target_Business_Case.customers` as cus
JOIN
(SELECT
customer_id,
order_id,
TIMESTAMP_DIFF(order_delivered_customer_date,order_purch
ase_timestamp,HOUR) AS time_to_delivery,
TIMESTAMP_DIFF(order_estimated_delivery_date,order_deliv
ered_customer_date,HOUR) AS diff_estimated_delivery
FROM `Target_Business_Case.orders`) as ord
ON cus.customer_id = ord.customer_id
JOIN `Target_Business_Case.order_items` as ordi
ON ord.order_id = ordi.order_id
ORDER BY mean_freight_value
LIMIT 5
```

Query results

SAVE RESUL

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

Row	customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	SP	15.147	208.869	251.894
2	PR	20.532	286.24	306.574
3	MG	20.63	287.112	302.913
4	RJ	20.961	363.063	271.044
5	DF	21.041	310.518	275.42

Note: mean_time_to_delivery and mean_diff_estimated_delivery has been represented in terms of hours.

Actionable Insight:

States with codes SP, PR, MG,RJ,DF are the top 5 States that have the lowest mean freight values.

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

Highest average time to delivery:

CODE:

```
SELECT DISTINCT
cus.customer_state,
ROUND(AVG (ordi.freight_value) OVER(PARTITION BY
cus.customer_state),3) as mean_freight_value,
ROUND(AVG(ord.time_to_delivery) OVER(PARTITION BY
cus.customer_state),3) as mean_time_to_delivery,
ROUND(AVG(ord.diff_estimated_delivery) OVER(PARTITION BY
cus.customer_state),3) as mean_diff_estimated_delivery
FROM `Target_Business_Case.customers` as cus
JOIN
(SELECT
customer_id,
order_id,
TIMESTAMP_DIFF(order_delivered_customer_date,order_purch
ase_timestamp,HOUR) AS time_to_delivery,
TIMESTAMP_DIFF(order_estimated_delivery_date,order_deliv
ered_customer_date,HOUR) AS diff_estimated_delivery
FROM `Target_Business_Case.orders`) as ord
ON cus.customer_id = ord.customer_id
JOIN `Target_Business_Case.order_items` as ordi
ON ord.order_id = ordi.order_id
ORDER BY mean_time_to_delivery DESC
LIMIT 5
```

Query results

SAVE RE

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GR

Row	customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	RR	42.984	676.978	422.435
2	AP	34.006	676.457	426.012
3	AM	33.205	632.847	460.969
4	AL	35.844	587.227	193.133
5	PA	35.833	569.601	325.289

Note: mean_time_to_delivery and mean_diff_estimated_delivery has been represented in terms of hours.

Actionable Insight:

States with codes RR, AP, AM, AL, PA are the top 5 States that have the highest mean time to delivery.

Lowest average time to delivery:

CODE:

```
SELECT DISTINCT
cus.customer_state,
ROUND(AVG (ordi.freight_value) OVER(PARTITION BY
cus.customer_state),3) as mean_freight_value,
ROUND(AVG(ord.time_to_delivery) OVER(PARTITION BY
cus.customer_state),3) as mean_time_to_delivery,
ROUND(AVG(ord.diff_estimated_delivery) OVER(PARTITION BY
cus.customer_state),3) as mean_diff_estimated_delivery
FROM `Target_Business_Case.customers` as cus
JOIN
(SELECT
customer_id,
order_id,
TIMESTAMP_DIFF(order_delivered_customer_date,order_purch
ase_timestamp,HOUR) AS time_to_delivery,
TIMESTAMP_DIFF(order_estimated_delivery_date,order_deliv
ered_customer_date,HOUR) AS diff_estimated_delivery
FROM `Target_Business_Case.orders`) as ord
ON cus.customer_id = ord.customer_id
JOIN `Target_Business_Case.order_items` as ordi
ON ord.order_id = ordi.order_id
ORDER BY mean_time_to_delivery
LIMIT 5
```

Query results

SAVE RESULT

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

Row	customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	SP	15.147	208.869	251.894
2	PR	20.532	286.24	306.574
3	MG	20.63	287.112	302.913
4	DF	21.041	310.518	275.42
5	SC	21.47	359.526	260.55

Note: mean_time_to_delivery and mean_diff_estimated_delivery has been

represented in terms of hours.

Actionable Insight:
States with codes SP, PR, MG,DF, SC are the top 5 States that have the lowest mean time to delivery.

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

Fastest delivery:

CODE:

```
SELECT DISTINCT
cus.customer_state,
ROUND(AVG (ordi.freight_value) OVER(PARTITION BY
cus.customer_state),3) as mean_freight_value,
ROUND(AVG(ord.time_to_delivery) OVER(PARTITION BY
cus.customer_state),3) as mean_time_to_delivery,
ROUND(AVG(ord.diff_estimated_delivery) OVER(PARTITION BY
cus.customer_state),3) as mean_diff_estimated_delivery
FROM `Target_Business_Case.customers` as cus
JOIN
(SELECT
customer_id,
order_id,
TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase
_timestamp,HOUR) AS time_to_delivery,
TIMESTAMP_DIFF(order_estimated_delivery_date,order_delivere
d_customer_date,HOUR) AS diff_estimated_delivery
FROM `Target_Business_Case.orders`) as ord
ON cus.customer_id = ord.customer_id
JOIN `Target_Business_Case.order_items` as ordi
ON ord.order_id = ordi.order_id
ORDER BY mean_diff_estimated_delivery DESC
LIMIT 5
```

Query results

SAVE F

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION G

Row	customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	AC	40.073	496.67	487.593
2	RO	41.07	473.271	463.769
3	AM	33.205	632.847	460.969
4	AP	34.006	676.457	426.012
5	RR	42.984	676.978	422.435

Note: mean_time_to_delivery and mean_diff_estimated_delivery has been represented in terms of hours.

Actionable Insight:

States with codes AC, RO, AM, AP, RR are the top 5 States that have the fastest delivery.

Slowest delivery:

CODE:

```
SELECT DISTINCT
cus.customer_state,
ROUND(AVG (ordi.freight_value) OVER(PARTITION BY
cus.customer_state),3) as mean_freight_value,
ROUND(AVG(ord.time_to_delivery) OVER(PARTITION BY
cus.customer_state),3) as mean_time_to_delivery,
ROUND(AVG(ord.diff_estimated_delivery) OVER(PARTITION BY
cus.customer_state),3) as mean_diff_estimated_delivery
FROM `Target_Business_Case.customers` as cus
JOIN
(SELECT
customer_id,
order_id,
TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase
_timestamp,HOUR) AS time_to_delivery,
TIMESTAMP_DIFF(order_estimated_delivery_date,order_delivere
d_customer_date,HOUR) AS diff_estimated_delivery
FROM `Target_Business_Case.orders`) as ord
ON cus.customer_id = ord.customer_id
JOIN `Target_Business_Case.order_items` as ordi
ON ord.order_id = ordi.order_id
```

```
ORDER BY mean_diff_estimated_delivery
LIMIT 5
```

Query results

SAVE

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION

Row	customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	AL	35.844	587.227	193.133
2	MA	38.257	519.059	221.118
3	SE	36.653	514.723	223.464
4	ES	22.059	375.08	238.415
5	BA	26.364	461.451	246.573

Note: mean_time_to_delivery and mean_diff_estimated_delivery has been represented in terms of hours.

Actionable Insight:

States with codes AL, MA, SE, ES, BA are the top 5 States that have the slowest delivery.

---

6) Payment type analysis:

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

CODE:

```
SELECT DISTINCT
pay.payment_type,
EXTRACT (MONTH FROM ord.order_purchase_timestamp) AS
month,
count(ord.order_id) AS order_count
FROM `Target_Business_Case.orders` as ord
JOIN `Target_Business_Case.payments` as pay
ON ord.order_id = pay.order_id
GROUP BY month, pay.payment_type
ORDER BY pay.payment_type, month
```

Query results

JOB INFORMATION | RESULTS | JSON | E

Row	payment_type	month	order_count
1	UPI	1	1715
2	UPI	2	1723
3	UPI	3	1942
4	UPI	4	1783
5	UPI	5	2035
6	UPI	6	1807
7	UPI	7	2074
8	UPI	8	2077
9	UPI	9	903
10	UPI	10	1056
11	UPI	11	1509
12	UPI	12	1160
13	credit_card	1	6103
14	credit_card	2	6609

Actionable Insight:

Month wise count of orders for various payment types can be seen above. For the UPI mode of payment, in the 9th month, the number of orders was the lowest with 903 orders. And in the 8th month, it was the highest with 2077 orders.

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

CODE:

```
SELECT DISTINCT
pay.payment_installments,
count(ord.order_id) AS order_count
FROM `Target_Business_Case.orders` as ord
JOIN `Target_Business_Case.payments` as pay
ON ord.order_id = pay.order_id
GROUP BY pay.payment_installments
ORDER BY pay.payment_installments
```

Query results

JOB INFORMATION | RESULTS | JS

Row	payment_installments	order_count
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
11	10	5328
12	11	23
13	12	133
14	13	16

Actionable Insight:

Number of orders based on the Payment Installments can be observed above. Installment 1 has the highest number of orders (52546) placed and the lowest number of order (1) is placed with installments 22 and 23.

Summary of Insights, and Recommendation:

From the above findings, we can come to the conclusion that:

1. The data is provided for the time period 04-09-2016 to 17-10-2018, with the attributes having data types String, Integer, Float, Timestamp.

2. Various cities and States from which the customers have placed orders could also be determined.

3. It is observed that e-commerce in Brazil, has a forward growing trend with peak sales during the months of May, July and August. Initially, the sales were minimal, and gradually there was a massive increase in the number of orders placed. Also it is evident from the buying pattern of Brazilian Customers that high amount of orders are placed during the Afternoon

4. The rise and fall of sales for each State month wise can be observed.

5. Similarly, the distribution of customers across different states of Brazil can be identified.

6. A massive increase by 136.98% in the cost of orders can be observed when comparing from 2017 to 2018.

7. Also, State wise Sum and Average of price and freight value tabulated is seen.

8. The fastness of delivery and the gap between estimated delivery and actual delivery time can be observed from the table showing the difference between purchase, delivery and estimated delivery days.

9. From the data collected, it can be ascertained that in some cases the products were delivered sooner than the estimated delivery date.

10. With respect to different States the average of freight value, delivery time, and the difference between estimated and actual delivery is analyzed.

11. Top five States with the highest and lowest mean freight value, and mean delivery time are evaluated. Similarly top five States with the fastest and slowest delivery are also evaluated.

12. Significant changes in the usage of various methods of payment per month can be observed.

13. Number of orders based on the Payment Installments can be observed in the table generated.

Certain Recommendations are:

1. As it is ascertained that in certain months the sales are higher, we need to make sure that proper sales and marketing techniques must be adopted so as to ensure a constant stream of sales in all the months, thereby increasing the profit.

2. There is a massive difference in distribution of customers when it comes to different states. Quality advertisements and customer satisfaction packages must be introduced in the states with low amount of customers. By creating customer oriented management, one can attract huge amount of customers.

3. Significant increase in the cost of orders can be used as an advantage by the companies to invest in areas that require more managerial attention.

4. The organization must ensure quick delivery of products and must also develop a quality interface so that there is not much difference between the estimated and actual delivery time. Doing this will ensure customer loyalty as well as helps in attracting new customers.

5. The states with the highest mean freight value and delivery time must be given careful consideration to review and redesign organizational policies so as to increase efficiency. By doing this, the states that are lacking can catch up and the states that are performing great can improve.

6. With the advent of technology, an organization must always be ready and forward to accommodate new technological improvements. As we can see there are various kinds of payment methods used. Making internet banking more accessible and easy to use will help retain customers and also will increase sales by providing hassle free transaction options.

Adopting these recommendations and attentiveness towards consumers' needs and organizational capabilities will help maximize both customer satisfaction as well as profits.