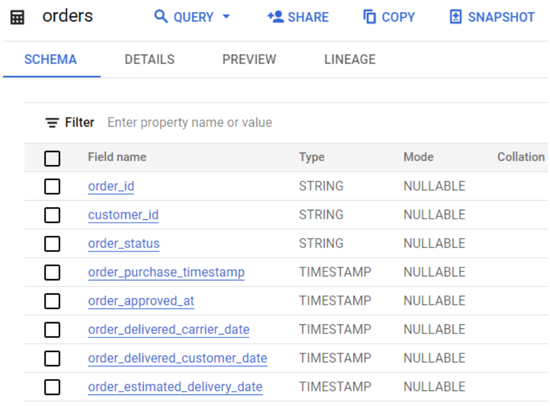
Target-SQL Business Case Study

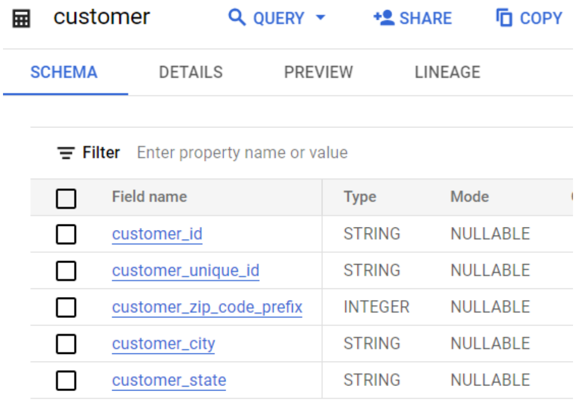
Question 1: Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset

Solution:

Sub question 1: Data type of columns in a table



□	Field name	Type	Mode	Collation
□	order_id	STRING	NULLABLE	
□	customer_id	STRING	NULLABLE	
□	order_status	STRING	NULLABLE	
□	order_purchase_timestamp	TIMESTAMP	NULLABLE	
□	order_approved_at	TIMESTAMP	NULLABLE	
□	order_delivered_carrier_date	TIMESTAMP	NULLABLE	
□	order_delivered_customer_date	TIMESTAMP	NULLABLE	
□	order_estimated_delivery_date	TIMESTAMP	NULLABLE	



□	Field name	Type	Mode
□	customer_id	STRING	NULLABLE
□	customer_unique_id	STRING	NULLABLE
□	customer_zip_code_prefix	INTEGER	NULLABLE
□	customer_city	STRING	NULLABLE
□	customer_state	STRING	NULLABLE

Data types of Orders table

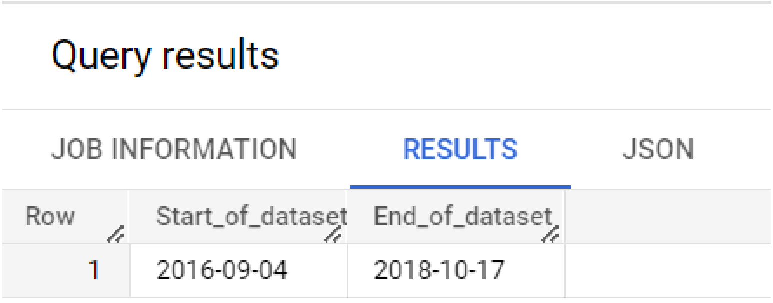
Data types of Customer table

Sub question 2 : Time period of which the data is given

Query:

```sql
SELECT DATE(MIN(order_purchase_timestamp)) AS Start_of_dataset ,
DATE(MAX(order_purchase_timestamp)) AS End_of_dataset
FROM `targetproject.orders`
```

Result:

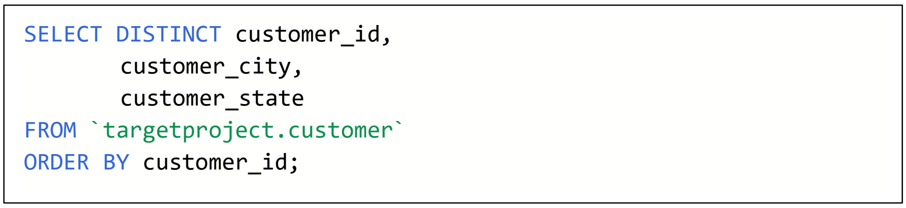


Row	Start_of_dataset	End_of_dataset
1	2016-09-04	2018-10-17

The orders table is used to find the first and last dates of orders placed, thereby giving us an idea that the data spans from September 2016 to October 2018.

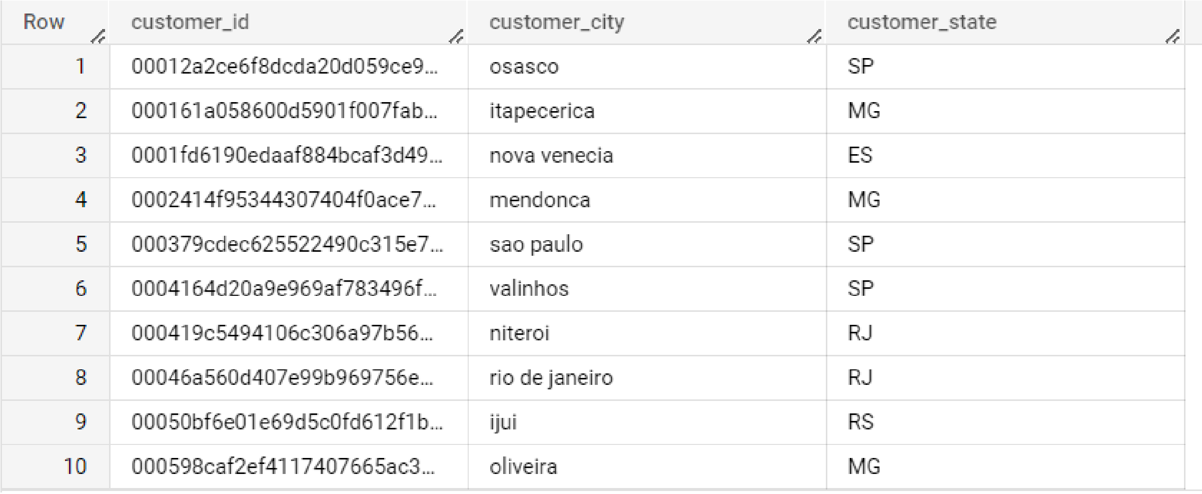
Sub question 3: Cities and States of customer ordered during the give period

Query:



```
SELECT DISTINCT customer_id,
        customer_city,
        customer_state
FROM `targetproject.customer`
ORDER BY customer_id;
```

Result:



Row	customer_id	customer_city	customer_state
1	00012a2ce6f8dcda20d059ce9...	osasco	SP
2	000161a058600d5901f007fab...	itapecerica	MG
3	0001fd6190edaaf884bcaf3d49...	nova venecia	ES
4	0002414f95344307404f0ace7...	mendonca	MG
5	000379cdec625522490c315e7...	sao paulo	SP
6	0004164d20a9e969af783496f...	valinhos	SP
7	000419c5494106c306a97b56...	niteroi	RJ
8	00046a560d407e99b969756e...	rio de janeiro	RJ
9	00050bf6e01e69d5c0fd612f1b...	ijui	RS
10	000598caf2ef4117407665ac3...	oliveira	MG

There are 99441 distinct customers in Brazil who have ordered from Target during the 2016-2018 period.

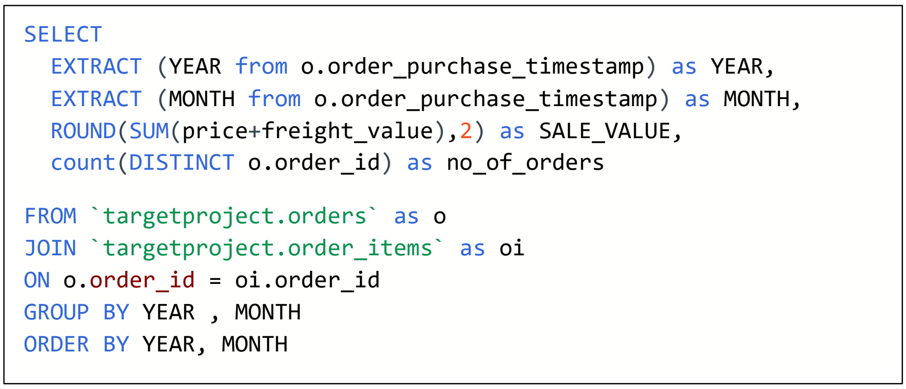
Question 2: In-depth Exploration:

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

Solution:

Sub question 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?

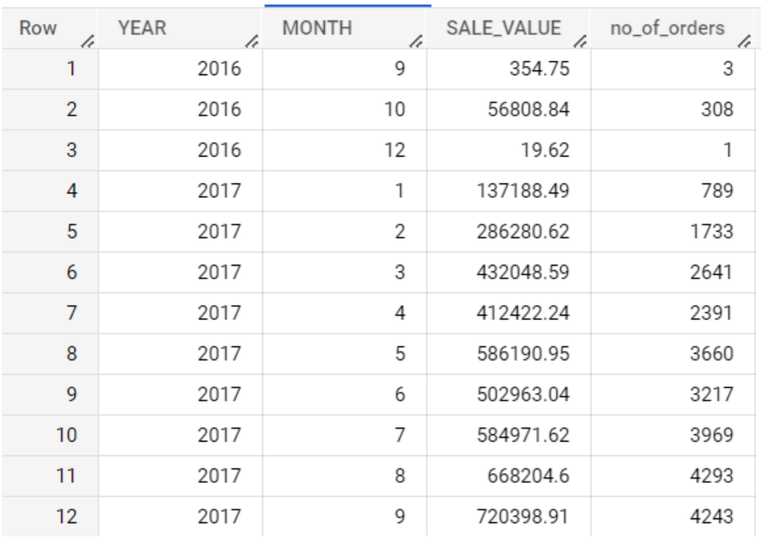
Query:



```
SELECT
  EXTRACT (YEAR from o.order_purchase_timestamp) as YEAR,
  EXTRACT (MONTH from o.order_purchase_timestamp) as MONTH,
  ROUND(SUM(price+freight_value),2) as SALE_VALUE,
  count(DISTINCT o.order_id) as no_of_orders

FROM `targetproject.orders` as o
JOIN `targetproject.order_items` as oi
ON o.order_id = oi.order_id
GROUP BY YEAR , MONTH
ORDER BY YEAR, MONTH
```

Result:



Row	YEAR	MONTH	SALE_VALUE	no_of_orders
1	2016	9	354.75	3
2	2016	10	56808.84	308
3	2016	12	19.62	1
4	2017	1	137188.49	789
5	2017	2	286280.62	1733
6	2017	3	432048.59	2641
7	2017	4	412422.24	2391
8	2017	5	586190.95	3660
9	2017	6	502963.04	3217
10	2017	7	584971.62	3969
11	2017	8	668204.6	4293
12	2017	9	720398.91	4243

Insights:

- It can be seen that there is a gradual increase in the sales value and orders month on month and also from 2017 to 2018. So there certainly is a growing market for the online retail industry in Brazil.
- We notice that there is a sudden increase in sales in the month of Nov, 2017 and then again in January,2018. However, there isn’t sufficient data to compare the numbers between the years and draw absolute conclusions.

- In the year 2018 from January to August, each month has a sale value over 10 lakhs, except for February. This could be due to the Rio Carnival/Festival that happens every year around February when the Brazilians are busy with the festival preparations and saving up for travel and such.

Recommendations:

- Brazil is a growing market for e-commerce and the company should continue operations and consider expanding to all parts of the country.

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

Time slots taken:

Dawn – 5AM to 7AM

Morning – 7AM to 12 PM

Afternoon – 12 PM to 4PM

Evening – 4PM to 8PM

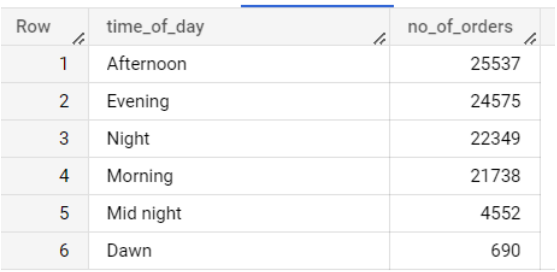
Night – 8PM to 12AM

Midnight – 12AM to 5AM

Query:

```
SELECT
  time_of_day,
  count(order_id) as no_of_orders
FROM
(SELECT
  order_id,
  time(order_purchase_timestamp) as order_time,
  CASE
    WHEN time(order_purchase_timestamp) BETWEEN "05:00:00" AND "07:00:00"
 THEN "Dawn"
    WHEN time(order_purchase_timestamp) BETWEEN "07:00:01" AND "12:00:00"
 THEN "Morning"
    WHEN time(order_purchase_timestamp) BETWEEN "12:00:01" AND "16:00:00"
 THEN "Afternoon"
    WHEN time(order_purchase_timestamp) BETWEEN "16:00:01" AND "20:00:00"
 THEN "Evening"
    WHEN time(order_purchase_timestamp) BETWEEN "20:00:01" AND "23:59:59"
 THEN "Night"
    ELSE "Mid night"
  END as time_of_day
FROM `targetproject.orders`)as xx
GROUP BY time_of_day
ORDER BY no_of_orders DESC;
```

Result:



Row	time_of_day	no_of_orders
1	Afternoon	25537
2	Evening	24575
3	Night	22349
4	Morning	21738
5	Mid night	4552
6	Dawn	690

Insights:

- As we can see from the result, Brazilians love to shop pretty much any time during the waking hours. However, maximum orders are in the afternoon to evening hours i.e between 12 noon and 8 in the evening.

Recommendations:

- The company could consider rolling out Flash Sales or Special Offers exclusive between these times in order to attract more customer and/or to increase the order value of current customers.

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

1. Get month on month orders by states
2. Distribution of customers across the states in Brazil

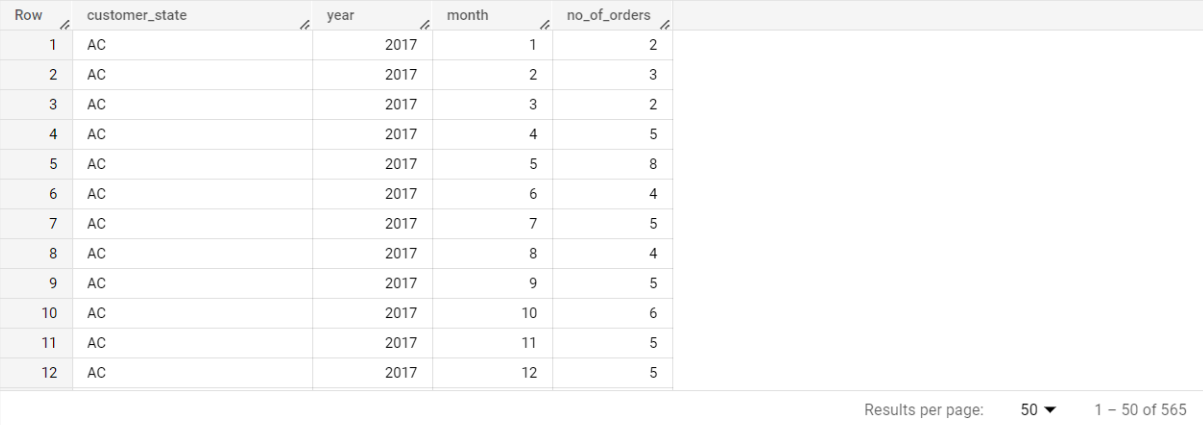
Solution:

Sub question 1: Get month on month orders by state

Query:

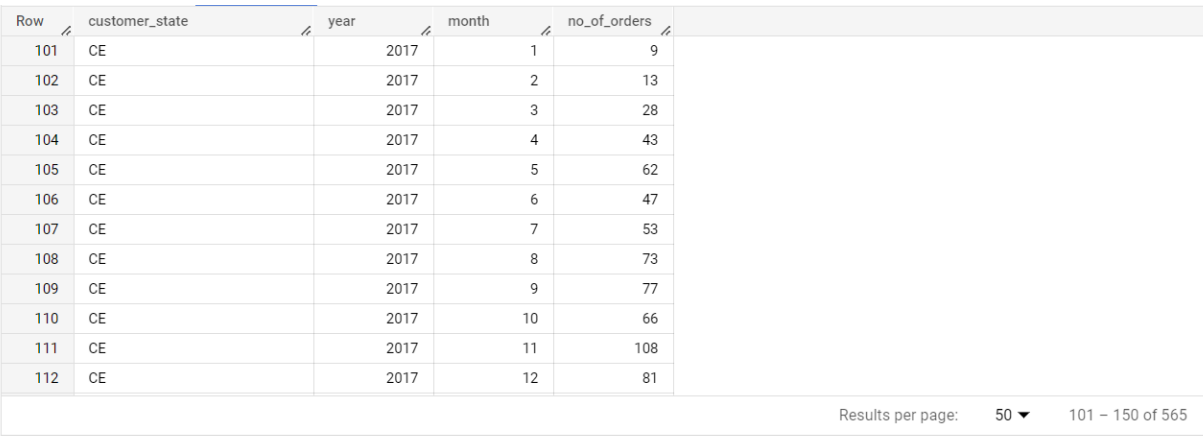
```
SELECT
  c.customer_state,
  EXTRACT(YEAR FROM o.order_purchase_timestamp) as year,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) as month,
  COUNT(o.order_id) as no_of_orders
FROM `targetproject.orders` as o
JOIN `targetproject.customer` as c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state, year, month
ORDER BY c.customer_state, year, month;
```

Result: Preview 1



Row	customer_state	year	month	no_of_orders
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
9	AC	2017	9	5
10	AC	2017	10	6
11	AC	2017	11	5
12	AC	2017	12	5

Preview 2



Row	customer_state	year	month	no_of_orders
101	CE	2017	1	9
102	CE	2017	2	13
103	CE	2017	3	28
104	CE	2017	4	43
105	CE	2017	5	62
106	CE	2017	6	47
107	CE	2017	7	53
108	CE	2017	8	73
109	CE	2017	9	77
110	CE	2017	10	66
111	CE	2017	11	108
112	CE	2017	12	81

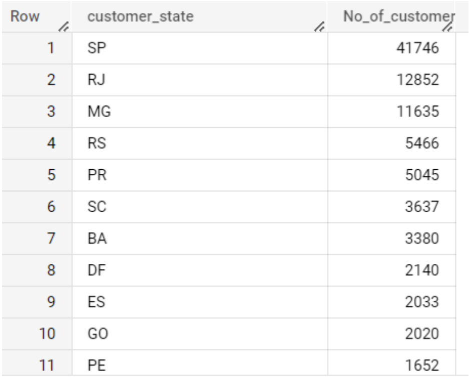
Insights: As we see in Preview 1, there isn't a proper trend in the number of orders in State AC however as we see in Preview 2, there is a growing trend in number of orders in State CE. But as we saw earlier, Brazil does show an overall growing trend in e-commerce market.

Sub question 2: Distribution of customers across the states in Brazil

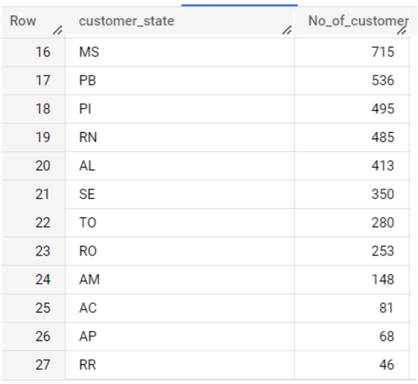
Query:

```
SELECT customer_state,
  count(customer_id) as No_of_customers
FROM `targetproject.customer`
GROUP BY customer_state
ORDER BY No_of_customers DESC
```

Result:



Row	customer_state	No_of_customer
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
11	PE	1652



Row	customer_state	No_of_custome
16	MS	715
17	PB	536
18	PI	495
19	RN	485
20	AL	413
21	SE	350
22	TO	280
23	RO	253
24	AM	148
25	AC	81
26	AP	68
27	RR	46

Question 4: Impact on Economy: Analyze the money movement 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) - You can use “payment_value” column in payments table
2. Mean & Sum of price and freight value by customer state

Solution:

Sub question 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

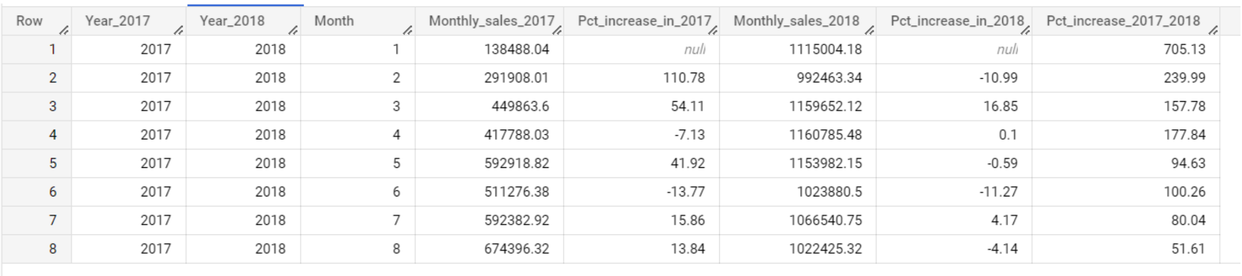
Query:

```
WITH data_2017 AS
(SELECT
  EXTRACT (YEAR from o.order_purchase_timestamp) AS Year,
  EXTRACT (MONTH from o.order_purchase_timestamp) AS Month,
  ROUND(SUM(p.payment_value),2) AS Monthly_sales_2017
FROM `targetproject.orders` AS o
JOIN `targetproject.payments` AS p
ON o.order_id = p.order_id
GROUP BY Year , Month
HAVING Year = 2017 AND Month IN (1,2,3,4,5,6,7,8)
ORDER BY Year , Month),

data_2018 AS
(SELECT
  EXTRACT (YEAR from o.order_purchase_timestamp) AS Year,
  EXTRACT (MONTH from o.order_purchase_timestamp) AS Month,
  ROUND(SUM(p.payment_value),2) AS Monthly_sales_2018
FROM `targetproject.orders` AS o
JOIN `targetproject.payments` AS p
ON o.order_id = p.order_id
GROUP BY Year , Month
HAVING Year = 2018 AND Month IN (1,2,3,4,5,6,7,8)
ORDER BY Year , Month)

SELECT data_2017.Year as Year_2017, data_2018.Year as Year_2018,
  data_2017.Month, Monthly_sales_2017,
  ROUND((Monthly_sales_2017 - LAG(Monthly_sales_2017,1,0) OVER(ORDER BY data_2017.Month))/
LAG(Monthly_sales_2017,1) OVER(ORDER BY data_2017.Month) * 100,2) as Pct_increase_in_2017,
  Monthly_sales_2018,
  ROUND((Monthly_sales_2018 - LAG(Monthly_sales_2018,1,0) OVER(ORDER BY data_2017.Month))/
LAG(Monthly_sales_2018,1) OVER(ORDER BY data_2017.Month) * 100,2) as Pct_increase_in_2018,
  ROUND((Monthly_sales_2018 - Monthly_sales_2017)/ Monthly_sales_2017 *100,2) AS Pct_increa
se_2017_2018
FROM data_2017
JOIN data_2018
ON data_2017.Month = data_2018.Month
ORDER BY Month
```

Result:



Row	Year_2017	Year_2018	Month	Monthly_sales_2017	Pct_increase_in_2017	Monthly_sales_2018	Pct_increase_in_2018	Pct_increase_2017_2018
1	2017	2018	1	138488.04	*null*	1115004.18	*null*	705.13
2	2017	2018	2	291908.01	110.78	992463.34	-10.99	239.99
3	2017	2018	3	449863.6	54.11	1159652.12	16.85	157.78
4	2017	2018	4	417788.03	-7.13	1160785.48	0.1	177.84
5	2017	2018	5	592918.82	41.92	1153982.15	-0.59	94.63
6	2017	2018	6	511276.38	-13.77	1023880.5	-11.27	100.26
7	2017	2018	7	592382.92	15.86	1066540.75	4.17	80.04
8	2017	2018	8	674396.32	13.84	1022425.32	-4.14	51.61

Insights:

- In 2017, we see that there is a huge jump in sales from January to February and February to March, this could be due to the initial surge to try out a newly introduced e-commerce segment in Brazil. However, we see that in the last 3 months that there is a change averaging around 14%
- In 2018, we see that a major fluctuation happens again in the months of January, February and March.
- The fluctuations in Quarter 1 in both the years could be due to the Rio Carnival.
- A year to year comparison shows that there is a significant increase in sales in 2018 when compared to 2017.

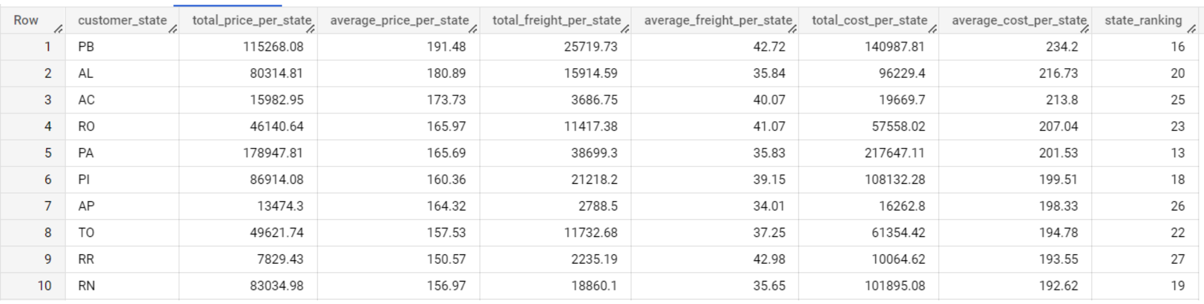
Recommendation:
Continue e-commerce operations in Brazil

Sub question 2: Mean and Sum of price and freight value by customer state

Query:

```
SELECT c.customer_state,
  ROUND(SUM(oi.price),2) AS total_price_per_state,
  ROUND(AVG(oi.price),2) AS average_price_per_state,
  ROUND(SUM(oi.freight_value),2) AS total_freight_per_state,
  ROUND(AVG(oi.freight_value),2) AS average_freight_per_state,
  ROUND(SUM(oi.price + oi.freight_value),2) AS total_cost_per_state,
  ROUND(AVG(oi.price + oi.freight_value),2) AS average_cost_per_state,
  RANK() OVER(ORDER BY ROUND(SUM(oi.price + oi.freight_value),2) DESC ) AS state_ranking
FROM `targetproject.order_items` AS oi
JOIN `targetproject.orders` AS o
ON oi.order_id = o.order_id
JOIN `targetproject.customer` AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY average_cost_per_state DESC
```

Result:



Row	customer_state	total_price_per_state	average_price_per_state	total_freight_per_state	average_freight_per_state	total_cost_per_state	average_cost_per_state	state_ranking
1	PB	115268.08	191.48	25719.73	42.72	140987.81	234.2	16
2	AL	80314.81	180.89	15914.59	35.84	96229.4	216.73	20
3	AC	15982.95	173.73	3686.75	40.07	19669.7	213.8	25
4	RO	46140.64	165.97	11417.38	41.07	57558.02	207.04	23
5	PA	178947.81	165.69	38699.3	35.83	217647.11	201.53	13
6	PI	86914.08	160.36	21218.2	39.15	108132.28	199.51	18
7	AP	13474.3	164.32	2788.5	34.01	16262.8	198.33	26
8	TO	49621.74	157.53	11732.68	37.25	61354.42	194.78	22
9	RR	7829.43	150.57	2235.19	42.98	10064.62	193.55	27
10	RN	83034.98	156.97	18860.1	35.65	101895.08	192.62	19

Insights:

- The average price of the above states are high and so is the average freight. However, when we see the state ranking based on total cost they fall between 13 and 27. This means that they are in bottom half of the states who contribute to total sales in Brazil.

Recommendations:

- The company can consider reduction in freight charges in order to attract more customers because customers tend avoid making purchases if the freight charges or shipping charges are high.
  - This could be achieved by setting up a warehouse around these areas so as to reduce shipping charges. However, the cost of setting up a warehouse and running it must not outweigh the potential increase in sales revenue.
  - An alternative is to provide free shipping for orders above a certain value.

Question 5: Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery
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
3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery
4. Sort the data to get the following:

   a) Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
   b) Top 5 states with highest/lowest average time to delivery
   c) Top 5 states where delivery is really fast/ not so fast compared to estimated date

Solution :

Query: Top 5 states with highest/lowest average freight value

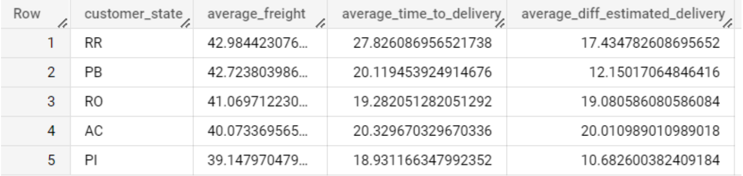
```
SELECT temp.customer_state,
  AVG(temp.freight_value) AS average_freight,
  AVG(temp.time_to_delivery) AS average_time_to_delivery,
  AVG(temp.diff_estimated_delivery) AS average_diff_estimated_delivery
FROM

(SELECT c.customer_state,
  freight_value,
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,DAY) AS time_to_delivery,
  DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS diff_estima
ted_delivery
FROM `targetproject.orders` AS o
JOIN `targetproject.order_items` AS oi
ON o.order_id = oi.order_id
JOIN `targetproject.customer` AS c
ON o.customer_id = c.customer_id) AS temp

GROUP BY temp.customer_state
ORDER BY average_freight DESC
LIMIT 5
```

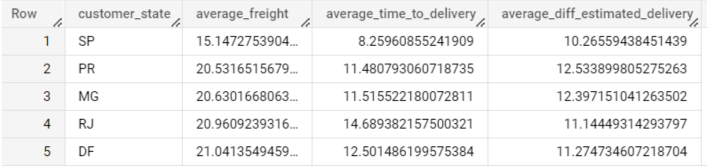
```
For lowest -
ORDER BY average_freight ASC
```

Result:



Row	customer_state	average_freight	average_time_to_delivery	average_diff_estimated_delivery
1	RR	42.984423076...	27.826086956521738	17.434782608695652
2	PB	42.723803986...	20.119453924914676	12.15017064846416
3	RO	41.069712230...	19.282051282051292	19.080586080586084
4	AC	40.073369565...	20.329670329670336	20.010989010989018
5	PI	39.147970479...	18.931166347992352	10.682600382409184

Top 5 highest average freight value



Row	customer_state	average_freight	average_time_to_delivery	average_diff_estimated_delivery
1	SP	15.1472753904...	8.25960855241909	10.26559438451439
2	PR	20.5316515679...	11.480793060718735	12.533899805275263
3	MG	20.6301668063...	11.515522180072811	12.397151041263502
4	RJ	20.9609239316...	14.689382157500321	11.14449314293797
5	DF	21.0413549459...	12.501486199575384	11.274734607218704

Top 5 lowest average freight value

Query : Top 5 states with highest/lowest average time to delivery

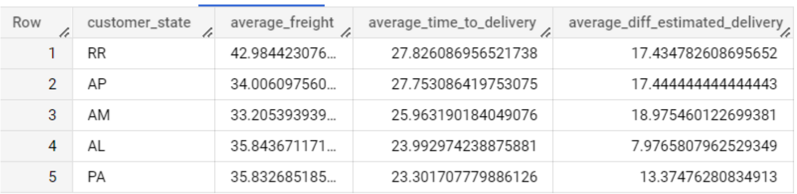
```
SELECT temp.customer_state,
  AVG(temp.freight_value) AS average_freight,
  AVG(temp.time_to_delivery) AS average_time_to_delivery,
  AVG(temp.diff_estimated_delivery) AS average_diff_estimated_delivery
FROM

(SELECT c.customer_state,
  freight_value,
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,DAY) AS time_to_delivery,
  DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS diff_estima
ted_delivery
FROM `targetproject.orders` AS o
JOIN `targetproject.order_items` AS oi
ON o.order_id = oi.order_id
JOIN `targetproject.customer` AS c
ON o.customer_id = c.customer_id) AS temp

GROUP BY temp.customer_state
ORDER BY average_time_to_delivery DESC
LIMIT 5
```

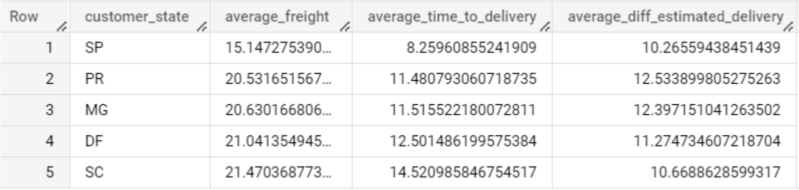
```
For lowest -
ORDER BY average_time_to_delivery ASC
```

Result:



Row	customer_state	average_freight	average_time_to_delivery	average_diff_estimated_delivery
1	RR	42.984423076...	27.826086956521738	17.434782608695652
2	AP	34.006097560...	27.753086419753075	17.444444444444443
3	AM	33.205393939...	25.963190184049076	18.975460122699381
4	AL	35.843671171...	23.992974238875881	7.9765807962529349
5	PA	35.832685185...	23.301707779886126	13.37476280834913

Top 5 highest average time to delivery



Row	customer_state	average_freight	average_time_to_delivery	average_diff_estimated_delivery
1	SP	15.147275390...	8.25960855241909	10.26559438451439
2	PR	20.531651567...	11.480793060718735	12.533899805275263
3	MG	20.630166806...	11.515522180072811	12.397151041263502
4	DF	21.041354945...	12.501486199575384	11.274734607218704
5	SC	21.470368773...	14.520985846754517	10.6688628599317

Top 5 average time to delivery

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

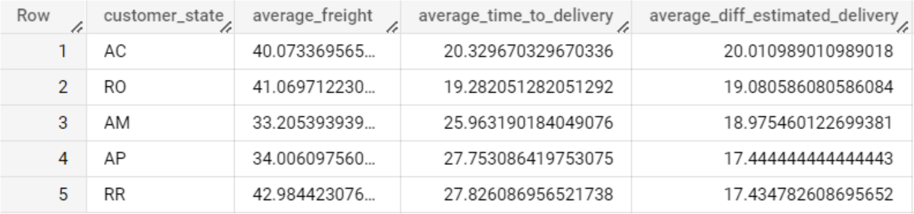
```
SELECT temp.customer_state,
  AVG(temp.freight_value) AS average_freight,
  AVG(temp.time_to_delivery) AS average_time_to_delivery,
  AVG(temp.diff_estimated_delivery) AS average_diff_estimated_delivery
FROM

(SELECT c.customer_state,
  freight_value,
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,DAY) AS time_to_delivery,
  DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS diff_estima
ted_delivery
FROM `targetproject.orders` AS o
JOIN `targetproject.order_items` AS oi
ON o.order_id = oi.order_id
JOIN `targetproject.customer` AS c
ON o.customer_id = c.customer_id) AS temp

GROUP BY temp.customer_state
ORDER BY average_diff_estimated_delivery DESC
LIMIT 5
```

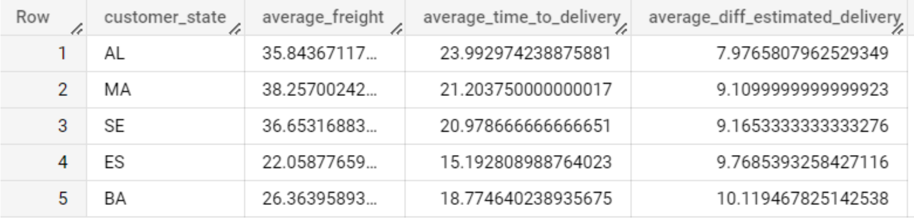
```
For fast -
ORDER BY average_diff_estimated_
_delivery ASC
```

Result:



Row	customer_state	average_freight	average_time_to_delivery	average_diff_estimated_delivery
1	AC	40.073369565...	20.329670329670336	20.010989010989018
2	RO	41.069712230...	19.282051282051292	19.080586080586084
3	AM	33.205393939...	25.963190184049076	18.975460122699381
4	AP	34.006097560...	27.753086419753075	17.444444444444443
5	RR	42.984423076...	27.826086956521738	17.434782608695652

Top 5 not so fast delivery compared to estimated time



Row	customer_state	average_freight	average_time_to_delivery	average_diff_estimated_delivery
1	AL	35.84367117...	23.992974238875881	7.9765807962529349
2	MA	38.25700242...	21.203750000000017	9.1099999999999923
3	SE	36.65316883...	20.978666666666651	9.1653333333333276
4	ES	22.05877659...	15.192808988764023	9.7685393258427116
5	BA	26.36395893...	18.774640238935675	10.119467825142538

Top 5 fast delivery compared to estimated time

Question 6: Payment type analysis:

1. Month over Month count of orders for different payment types
2. Count of orders based on the no. of payment instalments

Solution:

Sub-question 1: Month over month count of orders for different payment types

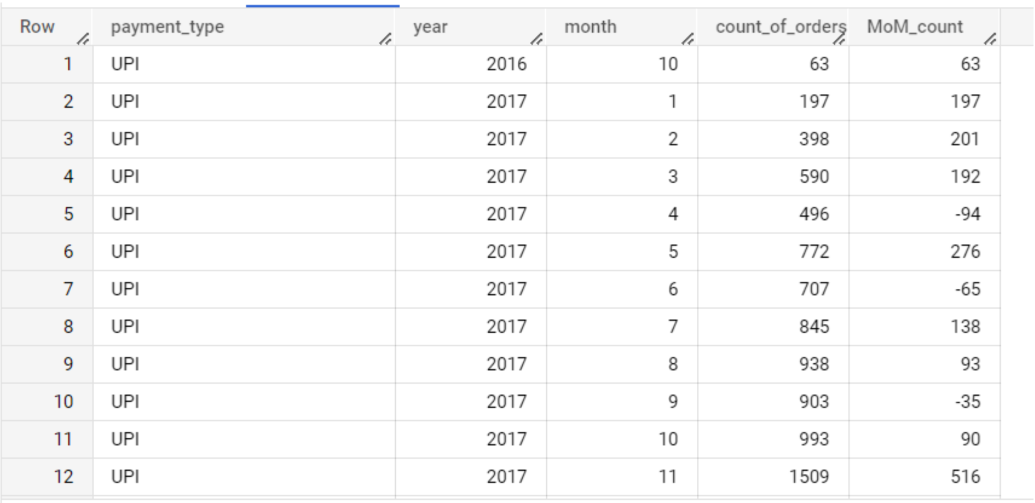
Query:

```
SELECT *,
  count_of_orders - LAG(count_of_orders,1,0) OVER(PARTITION BY payment_type, year
 ORDER BY month) as MoM_count

FROM
(SELECT
  payment_type,
  EXTRACT(YEAR FROM order_purchase_timestamp) as year,
  EXTRACT(MONTH FROM order_purchase_timestamp) as month,
  count(p.order_id) as count_of_orders
FROM `targetproject.payments` as p
JOIN `targetproject.orders` as o
ON p.order_id = o.order_id
GROUP BY payment_type, year, month) as temp

ORDER BY payment_type, year, month
```

Result:



Row	payment_type	year	month	count_of_orders	MoM_count
1	UPI	2016	10	63	63
2	UPI	2017	1	197	197
3	UPI	2017	2	398	201
4	UPI	2017	3	590	192
5	UPI	2017	4	496	-94
6	UPI	2017	5	772	276
7	UPI	2017	6	707	-65
8	UPI	2017	7	845	138
9	UPI	2017	8	938	93
10	UPI	2017	9	903	-35
11	UPI	2017	10	993	90
12	UPI	2017	11	1509	516

Insights:

- According to the results, majority number of orders were paid through credit card followed by UPI, vouchers and debit card.

Recommendations:

- The company could consider tying up with banks and UPI platforms to provide discounts and offers to customers.

Sub question 2: Count of orders based on the no. of payment instalments

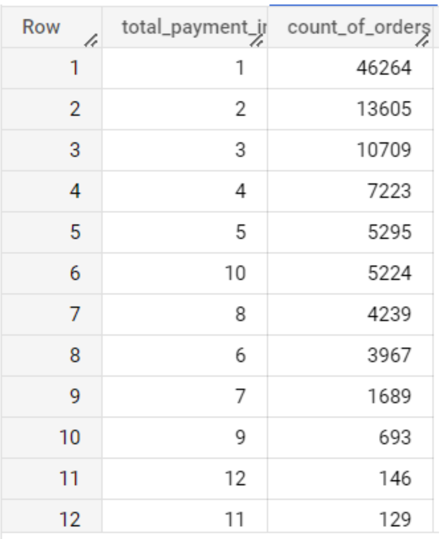
Query:

```
WITH installment_details AS

(SELECT
  *,
  sum(payment_installments) OVER(PARTITION BY order_id) as total_payment_installments,
FROM `targetproject.payments`)

SELECT
  total_payment_installments,
  count(DISTINCT order_id) as count_of_orders
FROM installment_details
GROUP BY total_payment_installments
ORDER BY count_of_orders DESC
```

Result:



Row	total_payment_i	count_of_orders
1	1	46264
2	2	13605
3	3	10709
4	4	7223
5	5	5295
6	10	5224
7	8	4239
8	6	3967
9	7	1689
10	9	693
11	12	146
12	11	129

Insights:

- Though the highest count goes to a one-time payment mode or single installment, majority customers still prefer to pay in installments.