TARGET

This business case has information of 100k orders from 2016 to 2018 made at Target in Brazil.

Dataset has following tables to analyse and give recommendations

1. customers
2. geolocation
3. order_items
4. payments
5. reviews
6. orders
7. products
8. sellers

QUESTION SET

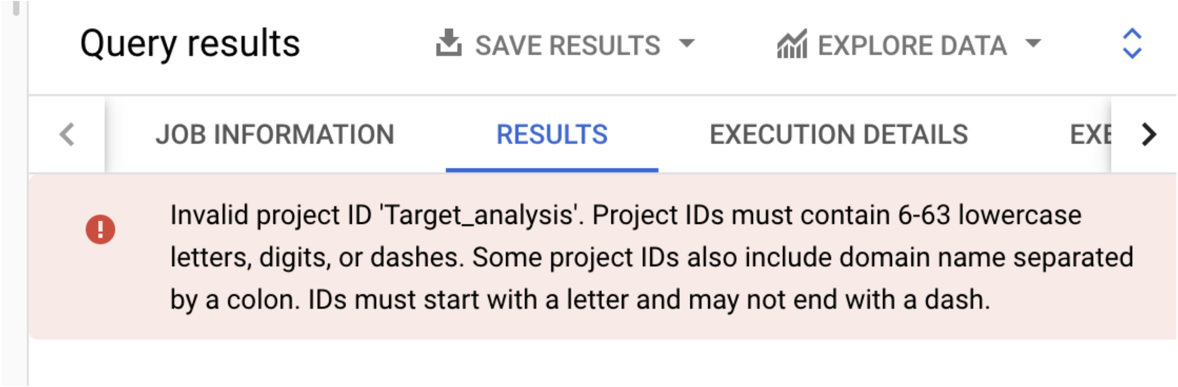
Q1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.

1. **Data type of columns in a table**

QUERY :

```
SELECT
  column_name,
  data_type
FROM
  Target_analysis.INFORMATION_SCHEMA.columns;
```

SCREENSHOT :



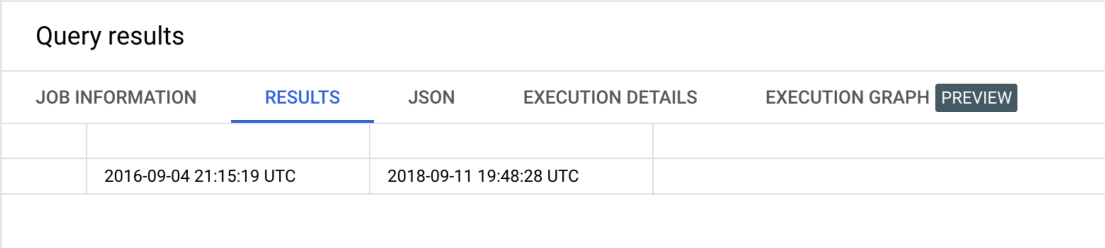
Invalid project ID 'Target_analysis'. Project IDs must contain 6-63 lowercase letters, digits, or dashes. Some project IDs also include domain name separated by a colon. IDs must start with a letter and may not end with a dash.

2. Time period for which the data is given

QUERY :

```
SELECT
  MIN(order_purchase_timestamp) AS first_day,
  MAX(order_delivered_carrier_date) AS last_day
FROM
  `Target_analysis.orders`;
```

SCREENSHOT :



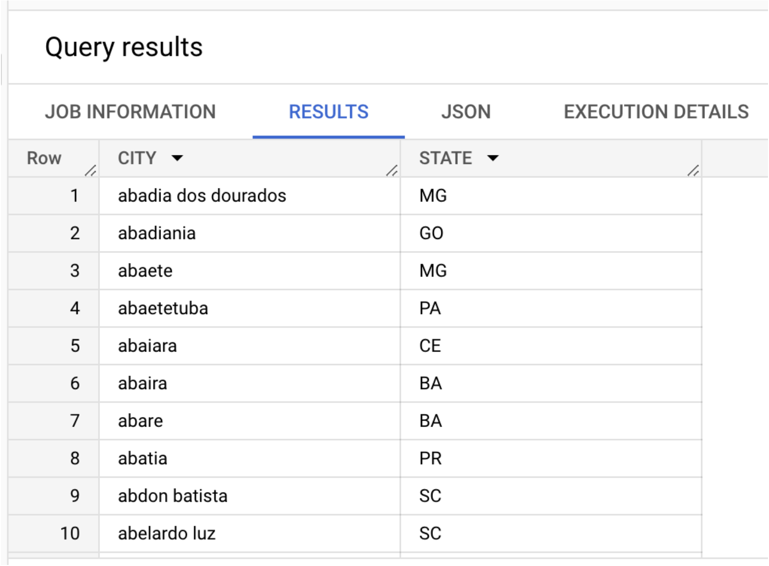
	2016-09-04 21:15:19 UTC	2018-09-11 19:48:28 UTC

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

QUERY:

```
SELECT
  DISTINCT c.customer_city AS CITY,
  c.customer_state AS STATE
FROM
  `Target_analysis.customers` c
INNER JOIN
  `Target_analysis.orders` o
ON
  c.customer_id = o.customer_id
WHERE
  o.order_purchase_timestamp BETWEEN (
  SELECT
    MIN(order_purchase_timestamp) AS first_day
  FROM
    `Target_analysis.orders` )
  AND (
  SELECT
    MAX(order_delivered_carrier_date) AS last_day
  FROM
    `Target_analysis.orders` )
ORDER BY
  city;
```

SCREENSHOT:



Query results

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

INSIGHTS:

we have analyzed the database schema to gain insights into its structure and contents. The schema reveals the number of tables and columns, providing an understanding of the data organization.

Examining the data of the company "Target" , we found that the data is available for a time period of nearly 2 years from April 2016 to September 2018. This allows us to uncover trends, patterns, and changes within that period.

Moreover, the database schema may contain more detailed data, such as information at the state or city level. This granularity enables us to perform regional analysis, identifying variations and trends specific to different locations.

ACTIONABLE ITEMS:

Targeted Marketing Campaigns: Utilize the granular data available, such as data at the state or city level, to tailor marketing campaigns based on regional preferences and customer behavior. This can help improve marketing effectiveness and drive customer engagement.

Seasonal Planning: Analyze the data to identify seasonal trends and patterns. This information can assist in optimizing inventory management, ensuring sufficient stock availability during peak seasons and minimizing excess inventory during slower periods.

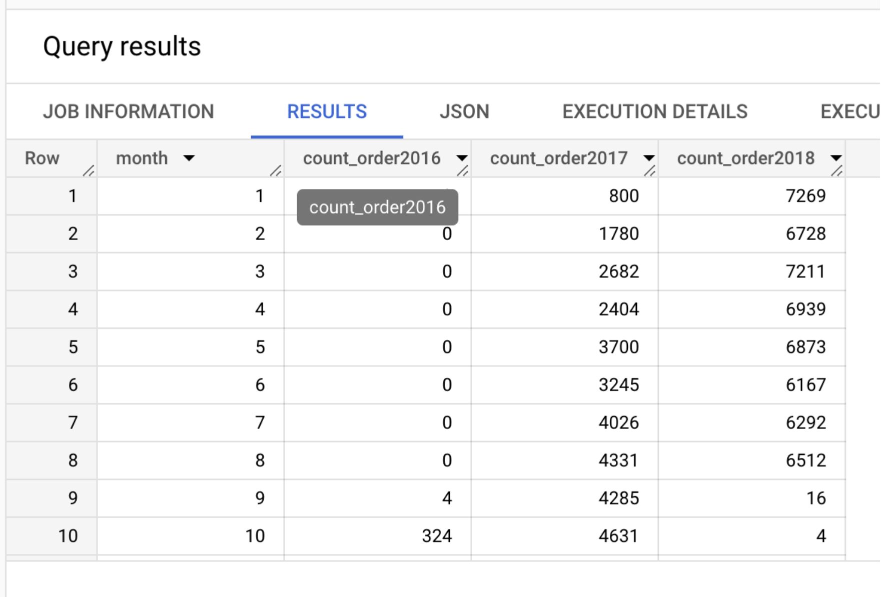
Q2: In-depth Exploration:

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?**

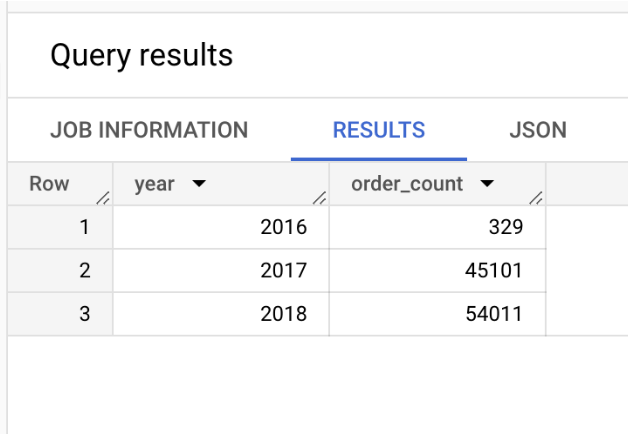
QUERY:

```
SELECT

  EXTRACT(MONTH
  FROM
    order_purchase_timestamp) AS month,
  COUNT(CASE
      WHEN EXTRACT(YEAR FROM order_purchase_timestamp) = 2016 THEN 1
  END
    ) AS count_order2016,
  COUNT(CASE
      WHEN EXTRACT(YEAR FROM order_purchase_timestamp) = 2017 THEN 1
  END
    ) AS count_order2017,
  COUNT(CASE
      WHEN EXTRACT(YEAR FROM order_purchase_timestamp) = 2018 THEN 1
  END
    ) AS count_order2018,

FROM
  `Target_analysis.orders`
GROUP BY
  month
ORDER BY
  month;

and also we can do this :

SELECT
  EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
  COUNT(*) AS order_count
FROM
  `Target_analysis.orders`
GROUP BY
  year
ORDER BY
  year;
```

SCREENSHOT:



Row	month	count_order2016	count_order2017	count_order2018
1	1		800	7269
2	2	0	1780	6728
3	3	0	2682	7211
4	4	0	2404	6939
5	5	0	3700	6873
6	6	0	3245	6167
7	7	0	4026	6292
8	8	0	4331	6512
9	9	4	4285	16
10	10	324	4631	4



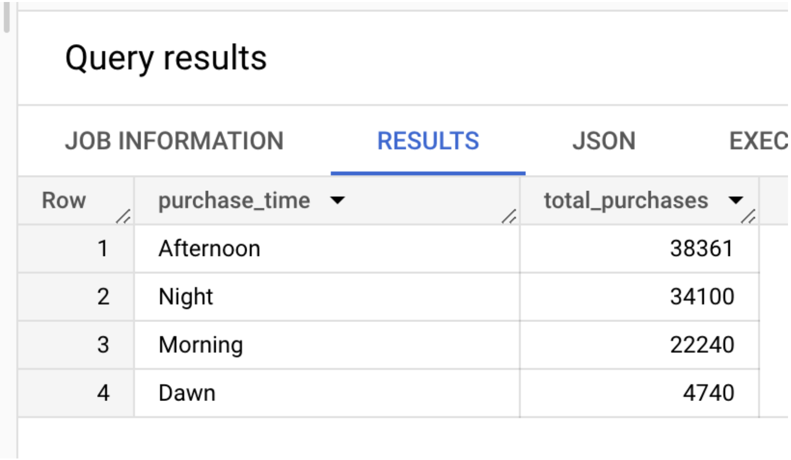
Row	year	order_count
1	2016	329
2	2017	45101
3	2018	54011

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

QUERY:

```
SELECT
   CASE
       WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 0 AND
EXTRACT(HOUR FROM order_purchase_timestamp) < 6 THEN 'Dawn'
       WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 6 AND
EXTRACT(HOUR FROM order_purchase_timestamp) < 12 THEN 'Morning'
       WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 12 AND
EXTRACT(HOUR FROM order_purchase_timestamp) < 18 THEN 'Afternoon'
       ELSE 'Night'
   END AS purchase_time,
   COUNT(*) AS total_purchases
FROM
   `Target_analysis.orders`
GROUP BY
   purchase_time
ORDER BY
   total_purchases desc;
```

SCREENSHOT :



Query results

Row	purchase_time	total_purchases
1	Afternoon	38361
2	Night	34100
3	Morning	22240
4	Dawn	4740

INSIGHTS:

The Brazilian e-commerce market shows substantial year-by-year growth, indicating increasing consumer adoption of online shopping. Month-by-month analysis reveals a consistent upward trend in order volumes, indicating sustained business growth and customer engagement throughout the year. Although 2018 shows stable activity without significant seasonal fluctuations, customers predominantly make purchases in the afternoon and night. Mornings also contribute a significant portion of daily orders, accounting for approximately one-fifth of the total.

ACTIONABLE:

Enhance Morning Experience: Offer morning-specific deals, personalized recommendations, or loyalty incentives to further increase customer activity during this time.

Optimize Marketing: Allocate resources and design targeted campaigns during peak hours in the afternoon and night to maximize customer engagement and sales.

Monitor Outliers: Keep an eye on unexpected peaks or fluctuations and adjust inventory management, marketing strategies, or customer support accordingly.

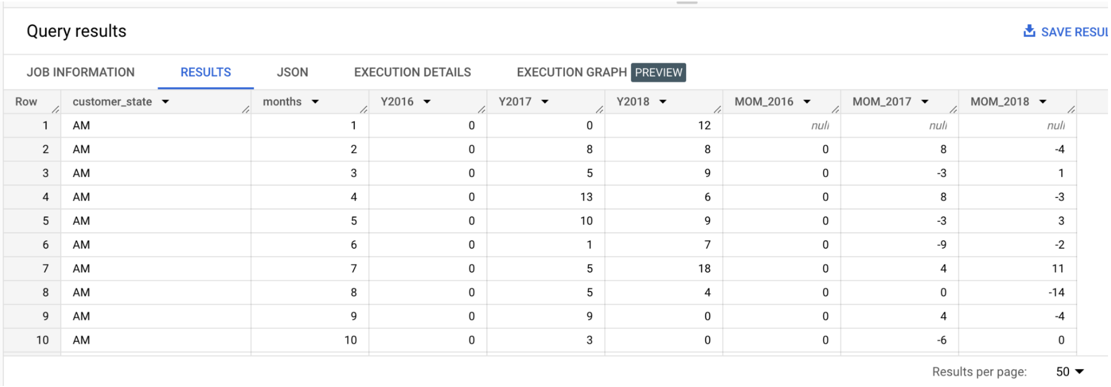
Q3 Evolution of E-commerce orders in the Brazil region:

Q3.1 Get month on month orders by states

QUERY:

```
With MOM as (

SELECT
  customer_state,
  EXTRACT(month FROM order_purchase_timestamp) AS months,
    COUNT(CASE when extract(year from order_purchase_timestamp) = 2016 then
1 end) as Y2016,
    COUNT(CASE when extract(year from order_purchase_timestamp) = 2017 then
1 end) as Y2017,
    COUNT(CASE when extract(year from order_purchase_timestamp) = 2018 then
1 end) as Y2018,
  COUNT(order_id) AS total_orders
FROM
  `Target_analysis.customers` c
INNER JOIN
  `Target_analysis.orders` o
ON
  c.customer_id = o.customer_id
GROUP BY
  customer_state,
  months
ORDER BY
  customer_state,
  months
)

select customer_state, months,Y2016,Y2017,Y2018,
(Y2016 - LAG(Y2016,1,NULL) over(partition by customer_state order by
customer_state , months)) as MOM_2016,
(Y2017 - LAG(Y2017,1,NULL) over(partition by customer_state order by
customer_state , months)) as MOM_2017,
(Y2018 - LAG(Y2018,1,NULL) over(partition by customer_state order by
customer_state , months)) as MOM_2018
 from MOM ;
```

SCREENSHOT:



Row	customer_state	months	Y2016	Y2017	Y2018	MOM_2016	MOM_2017	MOM_2018
1	AM	1	0	0	12	null	null	null
2	AM	2	0	8	8	0	8	-4
3	AM	3	0	5	9	0	-3	1
4	AM	4	0	13	6	0	8	-3
5	AM	5	0	10	9	0	-3	3
6	AM	6	0	1	7	0	-9	-2
7	AM	7	0	5	18	0	4	11
8	AM	8	0	5	4	0	0	-14
9	AM	9	0	9	0	0	4	-4
10	AM	10	0	3	0	0	-6	0

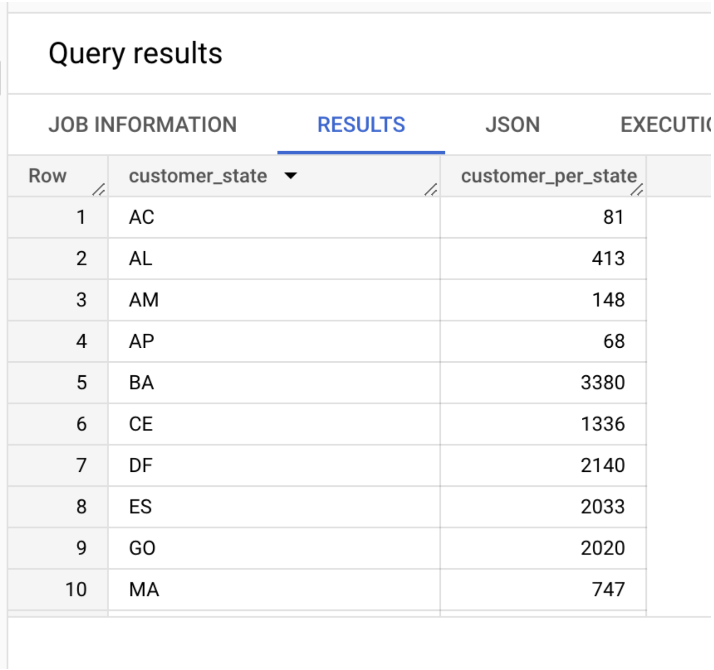
Results per page: 50

Q3.2 Distribution of customers across the states in Brazil

QUERY:

```
SELECT customer_state,
 count(customer_id) as customer_per_state
FROM
  `Target_analysis.customers`
GROUP BY
  customer_state
  order by
  customer_state;
```

SCREENSHOT :



Row	customer_state	customer_per_state
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747

INSIGHT:

The analysis unveils the month-on-month and year-on-year trends of orders within a specific state. Moreover, it highlights the notable variability in customer distribution, implying substantial variations in customer preferences, behaviors, and purchasing patterns across different regions within the state.

ACTIONABLE ITEM: Recognizing and understanding this variability becomes imperative to facilitate targeted marketing efforts and implement customized strategies that effectively engage and serve customers in diverse areas of the state.

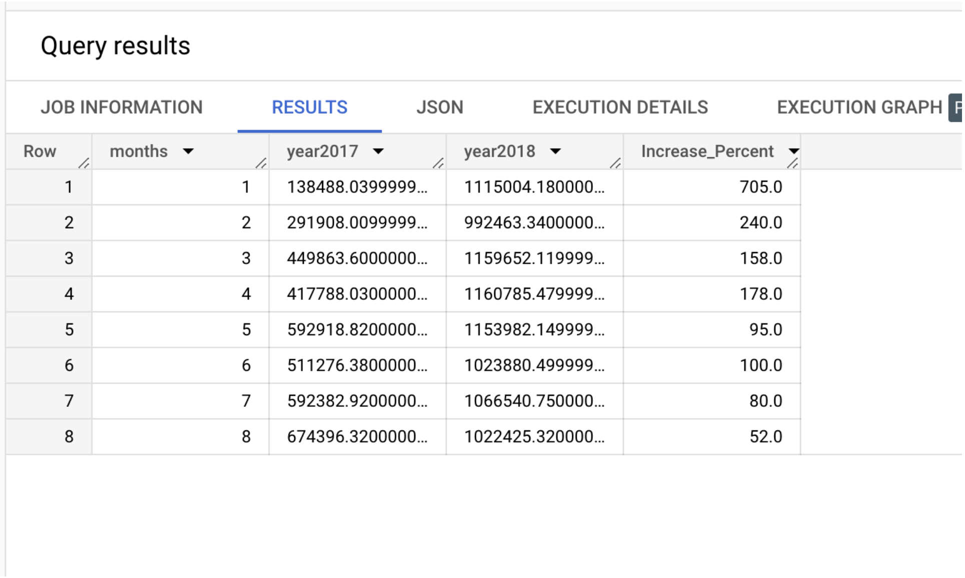
Q4 : Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

Q4.1 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 sales_table AS (select
extract(MONTH FROM o.order_purchase_timestamp) as Months,
extract(YEAR FROM o.order_purchase_timestamp) as Years,
sum(payment_value)as Payments,

 from
`Target_analysis.orders` o inner join `Target_analysis.payments`p
on
o.order_id = p.order_id
where extract(MONTH FROM o.order_purchase_timestamp)  between 1 and 8
 group by Months, Years
 order by Months, Years)

select
sales_table.months,
MAX(CASE WHEN sales_table.Years = 2017 THEN sales_table.Payments END) as
year2017,
MAX(CASE WHEN sales_table.Years = 2018 THEN sales_table.Payments END) as
year2018,

round(((MAX(CASE WHEN sales_table.Years = 2018 THEN sales_table.Payments
END)-MAX(CASE WHEN sales_table.Years = 2017 THEN sales_table.Payments
END))/MAX(CASE WHEN sales_table.Years = 2017 THEN sales_table.Payments END))
* 100) as Increase_Percent
FROM sales_table
GROUP BY sales_table.months
ORDER BY sales_table.months;
```

SCREENSHOT:



Query results

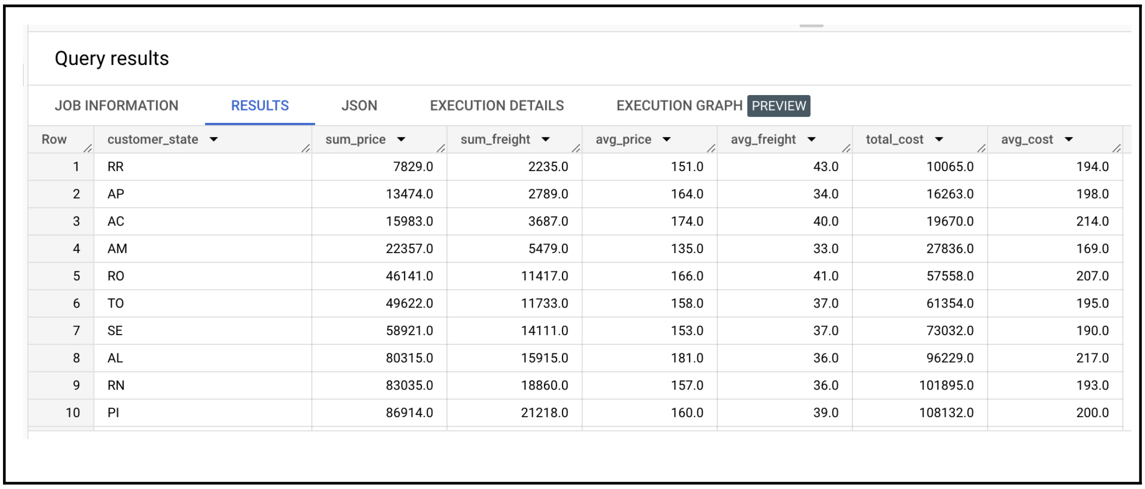
Row	months	year2017	year2018	Increase_Percent
1	1	138488.0399999...	1115004.180000...	705.0
2	2	291908.0099999...	992463.3400000...	240.0
3	3	449863.6000000...	1159652.119999...	158.0
4	4	417788.0300000...	1160785.479999...	178.0
5	5	592918.8200000...	1153982.149999...	95.0
6	6	511276.3800000...	1023880.499999...	100.0
7	7	592382.9200000...	1066540.750000...	80.0
8	8	674396.3200000...	1022425.320000...	52.0

Q4.2 Mean & Sum of price and freight value by customer state :

QUERY:

```
select c.customer_state, round(sum(oi.price)) as sum_price ,
round(sum(oi.freight_value)) as sum_freight, round(avg(oi.price)) as
avg_price, round(avg(oi.freight_value)) as avg_freight,
round(sum(oi.price)+sum(oi.freight_value)) as total_cost,
round(avg(oi.price)+avg(oi.freight_value)) as avg_cost
from
`Target_analysis.customers` c inner join `Target_analysis.orders` o
on c.customer_id = o.customer_id
inner join `Target_analysis.order_items` oi
on  o.order_id = oi.order_id
group by c.customer_state
order by total_cost;
```

SCREENSHOT:



Row	customer_state	sum_price	sum_freight	avg_price	avg_freight	total_cost	avg_cost
1	RR	7829.0	2235.0	151.0	43.0	10065.0	194.0
2	AP	13474.0	2789.0	164.0	34.0	16263.0	198.0
3	AC	15983.0	3687.0	174.0	40.0	19670.0	214.0
4	AM	22357.0	5479.0	135.0	33.0	27836.0	169.0
5	RO	46141.0	11417.0	166.0	41.0	57558.0	207.0
6	TO	49622.0	11733.0	158.0	37.0	61354.0	195.0
7	SE	58921.0	14111.0	153.0	37.0	73032.0	190.0
8	AL	80315.0	15915.0	181.0	36.0	96229.0	217.0
9	RN	83035.0	18860.0	157.0	36.0	101895.0	193.0
10	PI	86914.0	21218.0	160.0	39.0	108132.0	200.0

INSIGHT:

The cost of orders from 2017 to 2018 (Jan to Aug) witnessed a significant increase, with a percentage ranging from 52% to 705% with highest delta of increase in January(705%) and lowest delta in Dec(52%). This indicates a substantial growth in the monetary value of orders during this period.

When examining the mean and sum of price and freight value by customer state, it is evident that there are variations across different states. The state-wise analysis reveals differences in average order prices and freight values, indicating potential disparities in customer spending patterns and logistics costs.

ACTIONABLE:

Investigate the state-wise variations in average order prices and freight values. Identify potential reasons for these differences, such as regional preferences, market dynamics, or logistical challenges. Utilize this information to tailor marketing strategies and optimize logistics operations in different states.

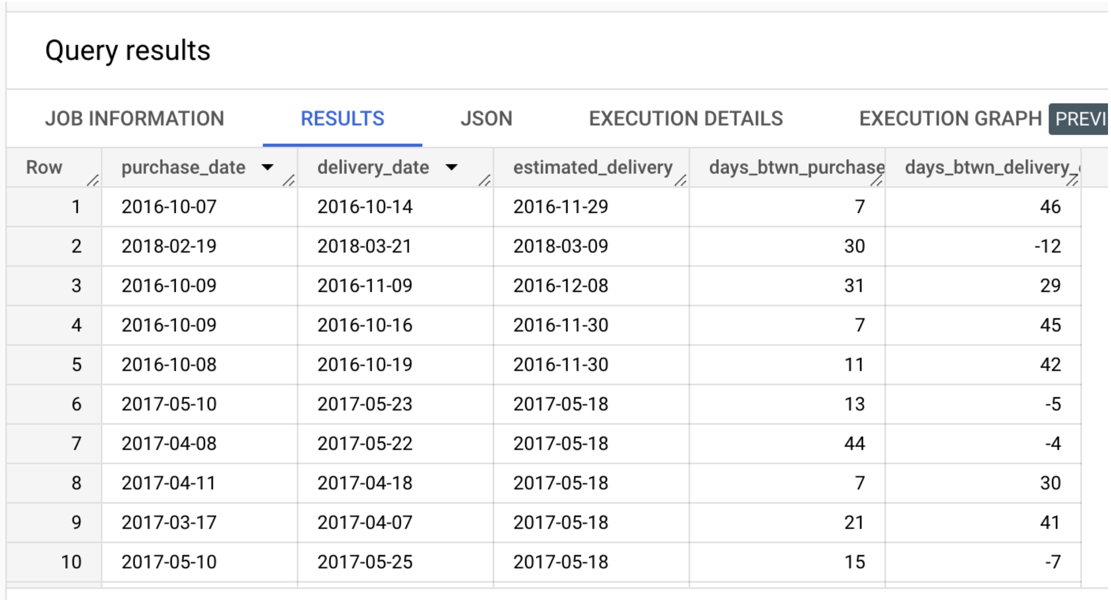
Q5 Analysis on sales, freight and delivery time

Q5.1 Calculate days between purchasing, delivering and estimated delivery

QUERY:

```
SELECT

  DATE(order_purchase_timestamp) AS purchase_date,
  DATE(order_delivered_customer_date) AS delivery_date,
  DATE(order_estimated_delivery_date) AS estimated_delivery,

DATE_DIFF(DATE(order_delivered_customer_date),DATE(order_purchase_timestamp)
,day)AS days_btwn_purchase_delivery,

DATE_DIFF(DATE(order_estimated_delivery_date),DATE(order_delivered_customer_
date),day)AS days_btwn_delivery_estimated_delivery
FROM
  `Target_analysis.orders`;
```

SCREENSHOT:



Row	purchase_date	delivery_date	estimated_delivery	days_btwn_purchase	days_btwn_delivery_
1	2016-10-07	2016-10-14	2016-11-29	7	46
2	2018-02-19	2018-03-21	2018-03-09	30	-12
3	2016-10-09	2016-11-09	2016-12-08	31	29
4	2016-10-09	2016-10-16	2016-11-30	7	45
5	2016-10-08	2016-10-19	2016-11-30	11	42
6	2017-05-10	2017-05-23	2017-05-18	13	-5
7	2017-04-08	2017-05-22	2017-05-18	44	-4
8	2017-04-11	2017-04-18	2017-05-18	7	30
9	2017-03-17	2017-04-07	2017-05-18	21	41
10	2017-05-10	2017-05-25	2017-05-18	15	-7

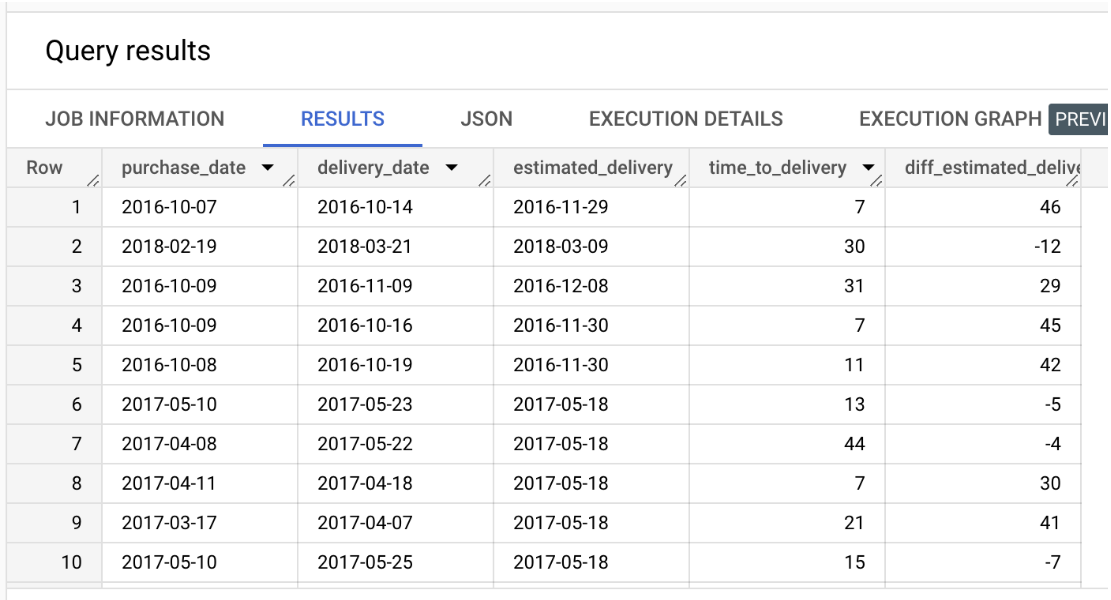
Q5.2 Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

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

QUERY:

```
SELECT

  DATE(order_purchase_timestamp) AS purchase_date,
  DATE(order_delivered_customer_date) AS delivery_date,
  DATE(order_estimated_delivery_date) AS estimated_delivery,

DATE_DIFF(DATE(order_delivered_customer_date),DATE(order_purchase_timestamp)
,day)AS time_to_delivery,

DATE_DIFF(DATE(order_estimated_delivery_date),DATE(order_delivered_customer_
date),day)AS diff_estimated_delivery
FROM
  `Target_analysis.orders`;
```

SCREENSHOT:



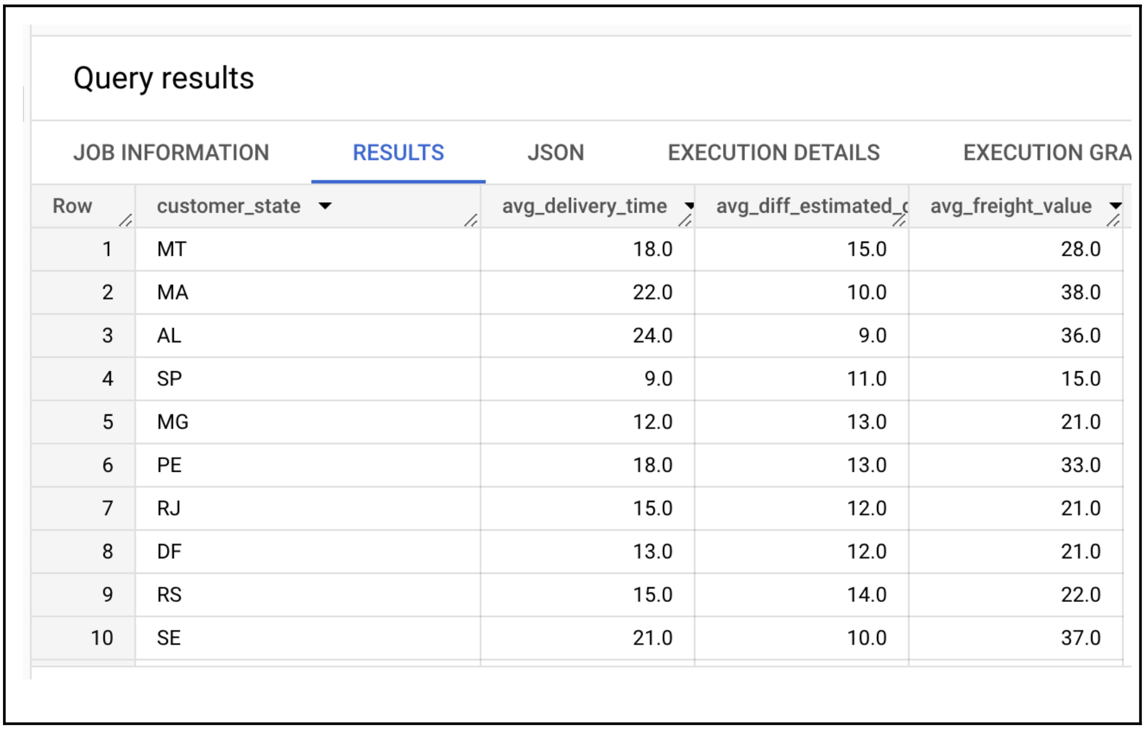
Row	purchase_date	delivery_date	estimated_delivery	time_to_delivery	diff_estimated_delive
1	2016-10-07	2016-10-14	2016-11-29	7	46
2	2018-02-19	2018-03-21	2018-03-09	30	-12
3	2016-10-09	2016-11-09	2016-12-08	31	29
4	2016-10-09	2016-10-16	2016-11-30	7	45
5	2016-10-08	2016-10-19	2016-11-30	11	42
6	2017-05-10	2017-05-23	2017-05-18	13	-5
7	2017-04-08	2017-05-22	2017-05-18	44	-4
8	2017-04-11	2017-04-18	2017-05-18	7	30
9	2017-03-17	2017-04-07	2017-05-18	21	41
10	2017-05-10	2017-05-25	2017-05-18	15	-7

Q5.3 Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

QUERY:

```
WITH

  sales AS(
  SELECT
    customer_state,
    freight_value,
    DATE(order_purchase_timestamp) AS purchase_date,
    DATE(order_delivered_customer_date) AS delivery_date,
    DATE(order_estimated_delivery_date) AS estimated_delivery,

DATE_DIFF(DATE(order_delivered_customer_date),DATE(order_purchase_timestamp)
,day)AS time_to_delivery,

DATE_DIFF(DATE(order_estimated_delivery_date),DATE(order_delivered_customer_
date),day)AS diff_estimated_delivery,
  FROM
    `Target_analysis.orders` o
  INNER JOIN
    `Target_analysis.customers`c
  ON
    c.customer_id = o.customer_id
  INNER JOIN
    `Target_analysis.order_items`oi
  ON
    oi.order_id = o.order_id )

SELECT
  customer_state,
  ROUND(AVG(time_to_delivery))AS avg_delivery_time,
  ROUND(AVG(diff_estimated_delivery)) AS avg_diff_estimated_delivery,
  ROUND(AVG(freight_value)) AS avg_freight_value
FROM
  sales
GROUP BY
  customer_state;
```

SCREENSHOT:



Query results

Row	customer_state	avg_delivery_time	avg_diff_estimated_	avg_freight_value
1	MT	18.0	15.0	28.0
2	MA	22.0	10.0	38.0
3	AL	24.0	9.0	36.0
4	SP	9.0	11.0	15.0
5	MG	12.0	13.0	21.0
6	PE	18.0	13.0	33.0
7	RJ	15.0	12.0	21.0
8	DF	13.0	12.0	21.0
9	RS	15.0	14.0	22.0
10	SE	21.0	10.0	37.0

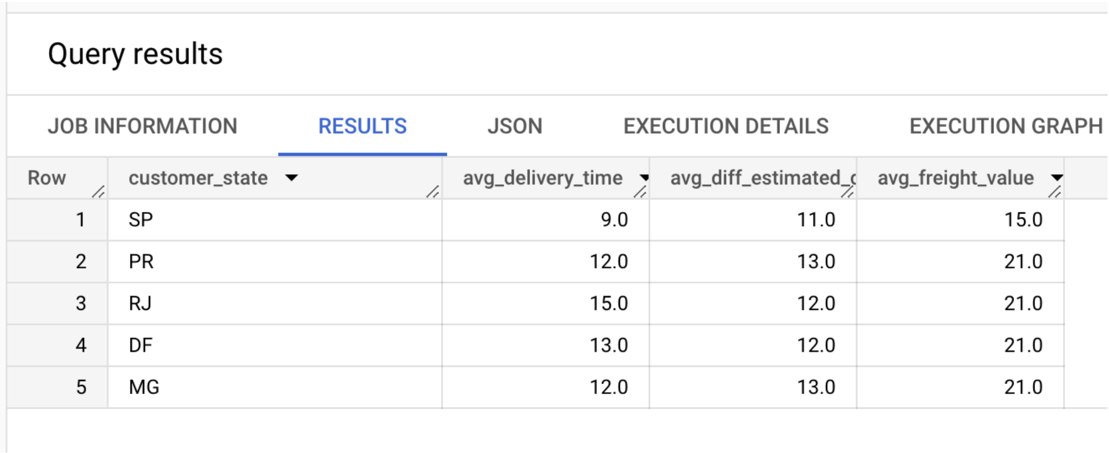
Q5.4 Sort the data to get the following:

Q5.5 Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

QUERY:

```
WITH

  sales AS(
  SELECT
    customer_state,
    freight_value,
    DATE(order_purchase_timestamp) AS purchase_date,
    DATE(order_delivered_customer_date) AS delivery_date,
    DATE(order_estimated_delivery_date) AS estimated_delivery,

DATE_DIFF(DATE(order_delivered_customer_date),DATE(order_purchase_timestamp)
,day)AS time_to_delivery,

DATE_DIFF(DATE(order_estimated_delivery_date),DATE(order_delivered_customer_
date),day)AS diff_estimated_delivery,
  FROM
    `Target_analysis.orders` o
  INNER JOIN
    `Target_analysis.customers`c
  ON
    c.customer_id = o.customer_id
  INNER JOIN
    `Target_analysis.order_items`oi
  ON
    oi.order_id = o.order_id )

SELECT
  customer_state,
  ROUND(AVG(time_to_delivery))AS avg_delivery_time,
  ROUND(AVG(diff_estimated_delivery)) AS avg_diff_estimated_delivery,
  ROUND(AVG(freight_value)) AS avg_freight_value
FROM
  sales
GROUP BY
  customer_state
  order by avg_freight_value
  limit 5;
```

SCREENSHOT



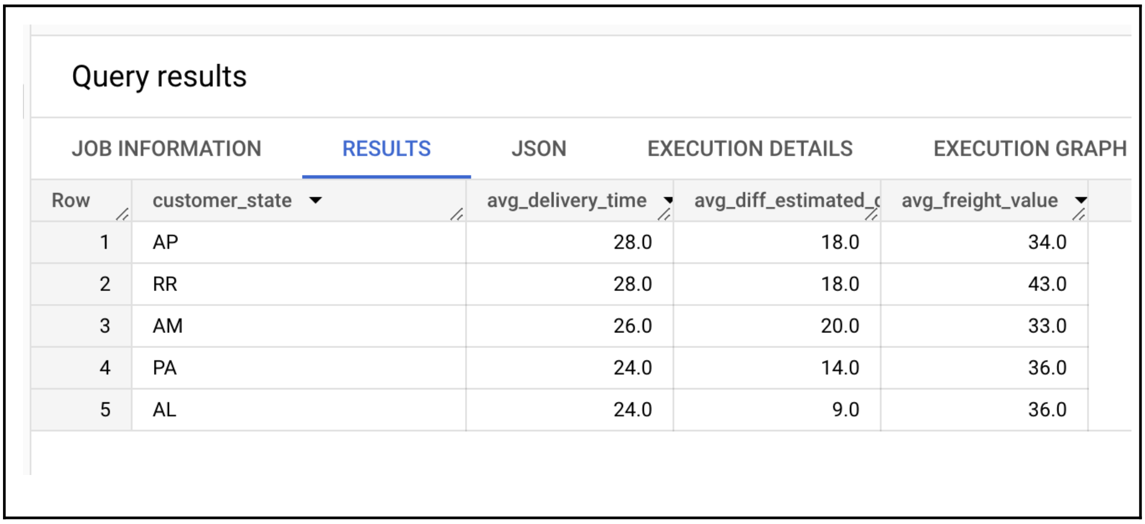
Row	customer_state	avg_delivery_time	avg_diff_estimated_	avg_freight_value
1	SP	9.0	11.0	15.0
2	PR	12.0	13.0	21.0
3	RJ	15.0	12.0	21.0
4	DF	13.0	12.0	21.0
5	MG	12.0	13.0	21.0

Q5.6 Top 5 states with highest/lowest average time to delivery

QUERY:

```
WITH
  sales AS(
  SELECT
    customer_state,
    freight_value,
    DATE(order_purchase_timestamp) AS purchase_date,
    DATE(order_delivered_customer_date) AS delivery_date,
    DATE(order_estimated_delivery_date) AS estimated_delivery,

DATE_DIFF(DATE(order_delivered_customer_date),DATE(order_purchase_timestamp)
,day)AS time_to_delivery,

DATE_DIFF(DATE(order_estimated_delivery_date),DATE(order_delivered_customer_
date),day)AS diff_estimated_delivery,
  FROM
    `Target_analysis.orders` o
  INNER JOIN
    `Target_analysis.customers`c
  ON
    c.customer_id = o.customer_id
  INNER JOIN
    `Target_analysis.order_items`oi
  ON
    oi.order_id = o.order_id )
SELECT
  customer_state,
  ROUND(AVG(time_to_delivery))AS avg_delivery_time,
  ROUND(AVG(diff_estimated_delivery)) AS avg_diff_estimated_delivery,
  ROUND(AVG(freight_value)) AS avg_freight_value
FROM
  sales
GROUP BY
  customer_state
ORDER BY
  avg_delivery_time DESC
LIMIT
  5;
```

SCREENSHOT:



Row	customer_state	avg_delivery_time	avg_diff_estimated_c	avg_freight_value
1	AP	28.0	18.0	34.0
2	RR	28.0	18.0	43.0
3	AM	26.0	20.0	33.0
4	PA	24.0	14.0	36.0
5	AL	24.0	9.0	36.0

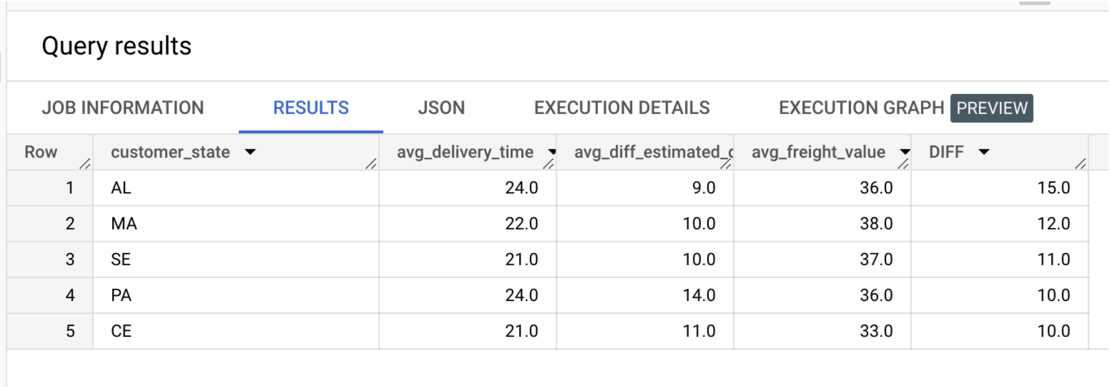
Q5.7 Top 5 states where delivery is really fast/ not so fast compared to estimated date

QUERY:

```
WITH
  sales AS(
  SELECT
    customer_state,
    freight_value,
    DATE(order_purchase_timestamp) AS purchase_date,
    DATE(order_delivered_customer_date) AS delivery_date,
    DATE(order_estimated_delivery_date) AS estimated_delivery,

DATE_DIFF(DATE(order_delivered_customer_date),DATE(order_purchase_timestamp)
,day)AS time_to_delivery,

DATE_DIFF(DATE(order_estimated_delivery_date),DATE(order_delivered_customer_
date),day)AS diff_estimated_delivery,
  FROM
    `Target_analysis.orders` o
  INNER JOIN
    `Target_analysis.customers`c
  ON
    c.customer_id = o.customer_id
  INNER JOIN
    `Target_analysis.order_items`oi
  ON
    oi.order_id = o.order_id )
SELECT
```

```
  customer_state,
  ROUND(AVG(time_to_delivery))AS avg_delivery_time,
  ROUND(AVG(diff_estimated_delivery)) AS avg_diff_estimated_delivery,
  ROUND(AVG(freight_value)) AS avg_freight_value,
 (ROUND(AVG(time_to_delivery))- ROUND(AVG(diff_estimated_delivery))) AS DIFF
 ,
FROM
  sales
GROUP BY
  customer_state
  ORDER BY DIFF desc
  LIMIT 5;
```

SCREENSHOT:



Query results

Row	customer_state	avg_delivery_time	avg_diff_estimated_	avg_freight_value	DIFF
1	AL	24.0	9.0	36.0	15.0
2	MA	22.0	10.0	38.0	12.0
3	SE	21.0	10.0	37.0	11.0
4	PA	24.0	14.0	36.0	10.0
5	CE	21.0	11.0	33.0	10.0

INSIGHTS:

The analysis of sales, freight, and delivery time reveals that there is a high variation between the days between purchasing and delivery across orders.

The days between delivery and estimated delivery variation suggests instances where orders were delivered earlier or later than the estimated date.

ACTIONABLE:

Identify the factors contributing to longer delivery times and instances of delayed deliveries. Analyze potential bottlenecks in the supply chain, logistics operations, or order processing that may cause delays.For states with higher average freight values, explore options to optimize shipping costs without compromising service quality. Negotiate contracts with shipping partners, explore alternative shipping methods, or implement pricing strategies to manage freight expenses effectively.

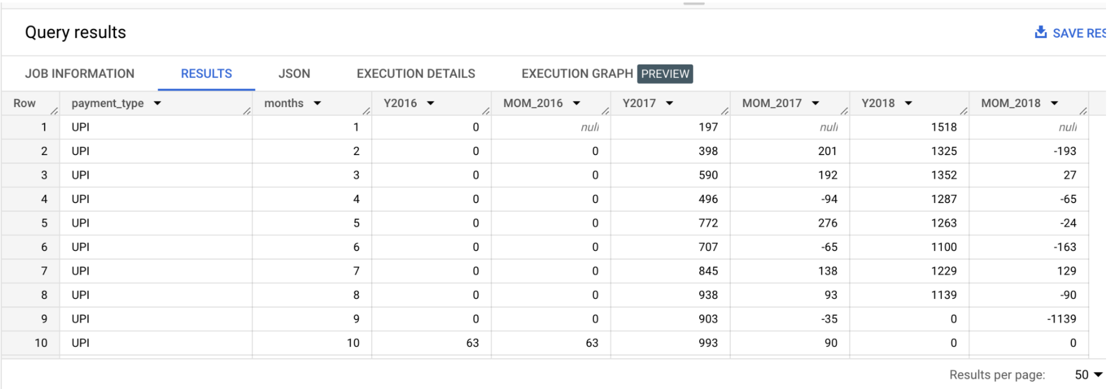
Q6. Payment type analysis:

Q6.1 Month over Month count of orders for different payment types

QUERY:

```
WITH
  MOM AS(
  SELECT
    payment_type,
    EXTRACT(month
    FROM
      o.order_purchase_timestamp) AS months,
    COUNT(CASE
        WHEN EXTRACT(year FROM order_purchase_timestamp) = 2016 THEN 1
    END
      ) AS Y2016,
    COUNT(CASE
        WHEN EXTRACT(year FROM order_purchase_timestamp) = 2017 THEN 1
    END
      ) AS Y2017,
    COUNT(CASE
        WHEN EXTRACT(year FROM order_purchase_timestamp) = 2018 THEN 1
    END
      ) AS Y2018,
    COUNT(p.order_id) AS total_order_count
  FROM
    `Target_analysis.payments`p
  INNER JOIN
    `Target_analysis.orders` o
  ON
    p.order_id = o.order_id
  GROUP BY
    payment_type,
    months
  ORDER BY
    payment_type,
    months)
SELECT
  payment_type, months, Y2016, (Y2016 - LAG(Y2016,1,NULL) OVER(PARTITION BY
payment_type ORDER BY payment_type, months)) AS MOM_2016, Y2017, (Y2017 -
LAG(Y2017,1,NULL) OVER(PARTITION BY payment_type ORDER BY payment_type,
months)) AS MOM_2017, Y2018, (Y2018 - LAG(Y2018,1,NULL) OVER(PARTITION BY
payment_type ORDER BY payment_type, months)) AS MOM_2018
FROM
  MOM
ORDER BY
  payment_type, months ;
```

SCREENSHOT:



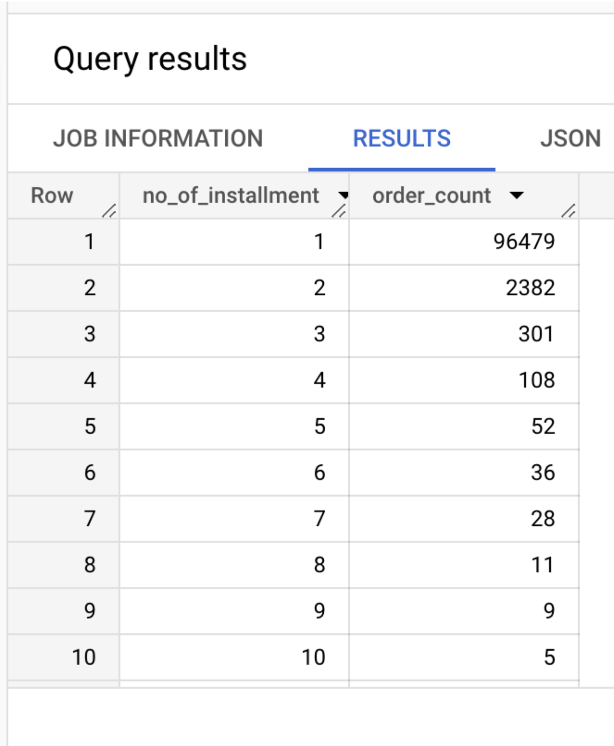
Row	payment_type	months	Y2016	MOM_2016	Y2017	MOM_2017	Y2018	MOM_2018
1	UPI	1	0	null	197	null	1518	null
2	UPI	2	0	0	398	201	1325	-193
3	UPI	3	0	0	590	192	1352	27
4	UPI	4	0	0	496	-94	1287	-65
5	UPI	5	0	0	772	276	1263	-24
6	UPI	6	0	0	707	-65	1100	-163
7	UPI	7	0	0	845	138	1229	129
8	UPI	8	0	0	938	93	1139	-90
9	UPI	9	0	0	903	-35	0	-1139
10	UPI	10	63	63	993	90	0	0

Q6.2 Count of orders based on the no. of payment installments

QUERY:

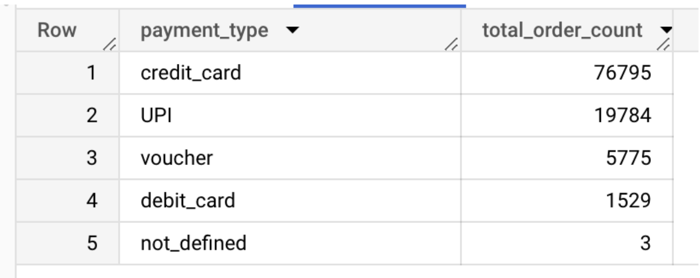
```
WITH

  PO AS (
  SELECT
    COUNT(payment_installments) as pi,
    order_id
  FROM
    `Target_analysis.payments`
  GROUP BY
    order_id)

SELECT pi as no_of_installment, COUNT(order_id) as order_count
from PO
group by pi
order by pi;
```

SCREENSHOT :



Row	no_of_installment	order_count
1	1	96479
2	2	2382
3	3	301
4	4	108
5	5	52
6	6	36
7	7	28
8	8	11
9	9	9
10	10	5

INSIGHTS:



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

The majority of orders are made using credit cards, with a total order count of 76,795 and second by UPI with 19,784 orders which is like 1/4th of the former.

Most orders (96,479) are made with a single payment installment. As the number of installments increases, the count of orders decreases. This suggests that customers prefer to pay for their orders in fewer installments. There is a steep decline in order counts after 2 installments.

ACTIONABLE ITEMS:
Evaluate the popularity of different payment types and assess the associated transaction fees or costs. Consider negotiating contracts or partnerships with payment processors to optimize transaction costs and provide customers with convenient and cost-effective payment options.

Continuously monitor and analyze payment trends, keeping an eye on emerging payment technologies or methods that may gain popularity among customers. Stay updated with industry trends and be prepared to adapt payment options accordingly to meet evolving customer preferences.

Recommendations

- Utilize granular data for targeted marketing campaigns based on regional preferences and customer behavior.
- Optimize seasonal planning to ensure sufficient stock availability and minimize excess inventory.
- Enhance the morning experience by offering morning-specific deals and personalized recommendations.
- Allocate resources for targeted marketing campaigns during peak hours in the afternoon and night.
- Monitor outliers and adjust strategies to capitalize on opportunities or mitigate risks.
- Investigate state-wise variations in order prices and freight values to optimize marketing and logistics.

Continuously analyze payment trends and adapt to evolving customer preferences. Implementing these recommendations will improve marketing effectiveness, drive customer engagement, optimize inventory management, and provide cost-effective payment options. Stay updated with customer behavior and market dynamics to stay ahead in the Brazilian e-commerce market.