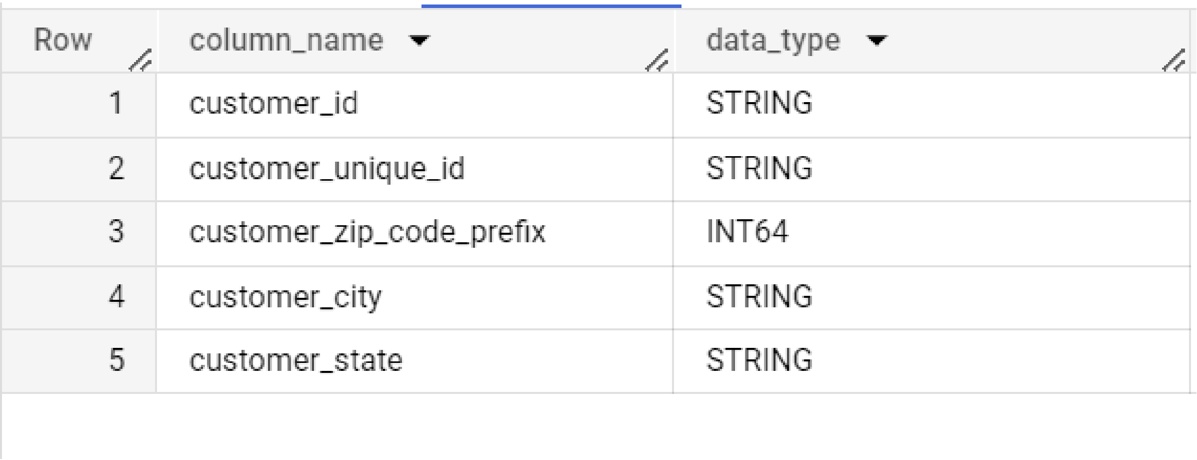
1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**
    1. Data type of all columns in the "customers" table.
    2. Get the time range between which the orders were placed.
    3. Count the Cities & States of customers who ordered during the given period.

ANS_1.1: Data type of all columns in the "customers" table.

```
SELECT column_name, data_type
FROM  `target-project-tami.target_project_tami`.INFORMATION_SCHEMA.COLUMNS
WHERE table_name ='customer'
```

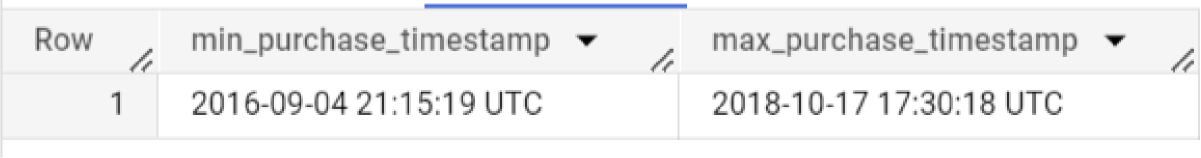


Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

INFERENCE: The data type schema in BIG QUERY Console can be either bought by this query or by simply clicking on the data set table will give the data types.

ANS_1.2: Get the time range between which the orders were placed.

```
SELECT
min(order_purchase_timestamp) AS min_purchase_timestamp,
max(order_purchase_timestamp) AS max_purchase_timestamp
FROM `target-project-tami.target_project_tami.orders`
```



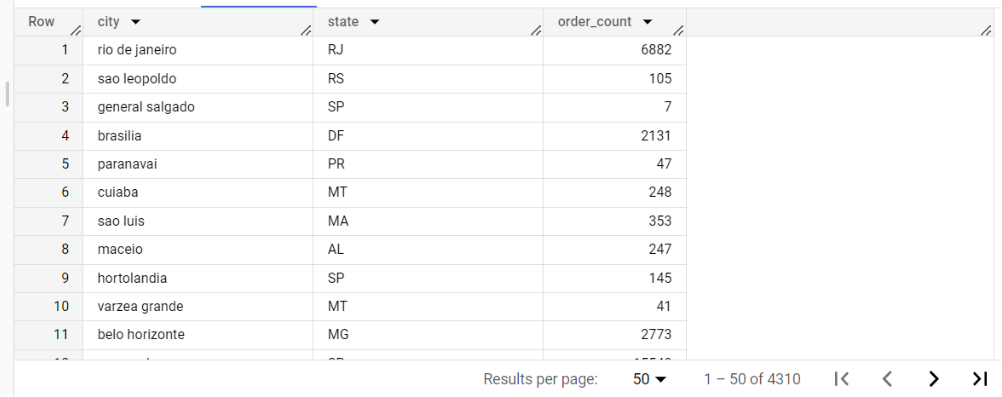
Row	min_purchase_timestamp	max_purchase_timestamp
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

INFERENCE: The aggregation function of min and max is done on the order table has information of 100k orders from 2016 to 2018 made at Target in Brazil.

ANS_1.3: Count the Cities & States of customers who ordered during the given period.

```
SELECT c.customer_city AS city,
       c.customer_state AS state,
       COUNT (o.customer_id) order_count
```

```
FROM `target-project-tami.target_project_tami.customer` c
JOIN `target_project_tami.orders` o
ON c.customer_id = o.customer_id
WHERE o.order_purchase_timestamp
BETWEEN '2016-09-04 21:15:19 UTC'
AND '2018-10-17 17:30:18 UTC'
GROUP BY c.customer_city, c.customer_state
```



Row	city	state	order_count
1	rio de janeiro	RJ	6882
2	sao leopoldo	RS	105
3	general salgado	SP	7
4	brasilia	DF	2131
5	paranavai	PR	47
6	cuiaba	MT	248
7	sao luis	MA	353
8	maceio	AL	247
9	hortolandia	SP	145
10	varzea grande	MT	41
11	belo horizonte	MG	2773

INFERENCE: The aggregation function on order table has information of city and states orders count from 2016 to 2018 made at Target in Brazil. It can be observed that there is a growing trend in e-commerce in Brazil.

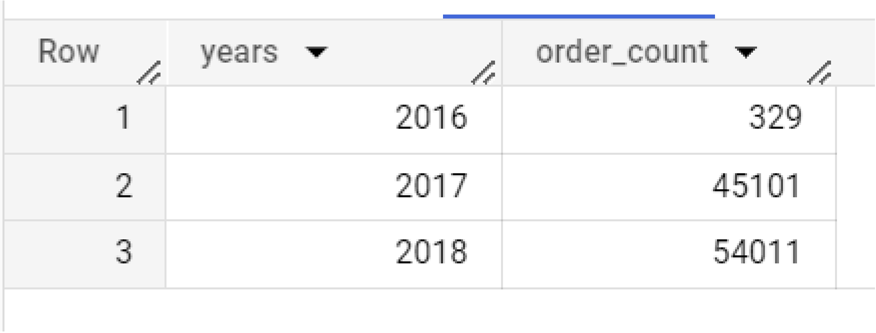
---

2. In-depth Exploration:

1. Is there a growing trend in the no. of orders placed over the past years?
2. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?
3. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
    1. 0-6 hrs : Dawn
    2. 7-12 hrs : Mornings
    3. 13-18 hrs : Afternoon
    4. 19-23 hrs : Night

ANS_2.1: Is there a growing trend in the no. of orders placed over the past years?

```
SELECT
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS years,
COUNT(DISTINCT o.order_id) AS order_count
FROM `target-project-tami.target_project_tami.orders` o
JOIN `target-project-tami.target_project_tami.customer` c
ON o.customer_id = c.customer_id
GROUP BY years
ORDER BY years
```

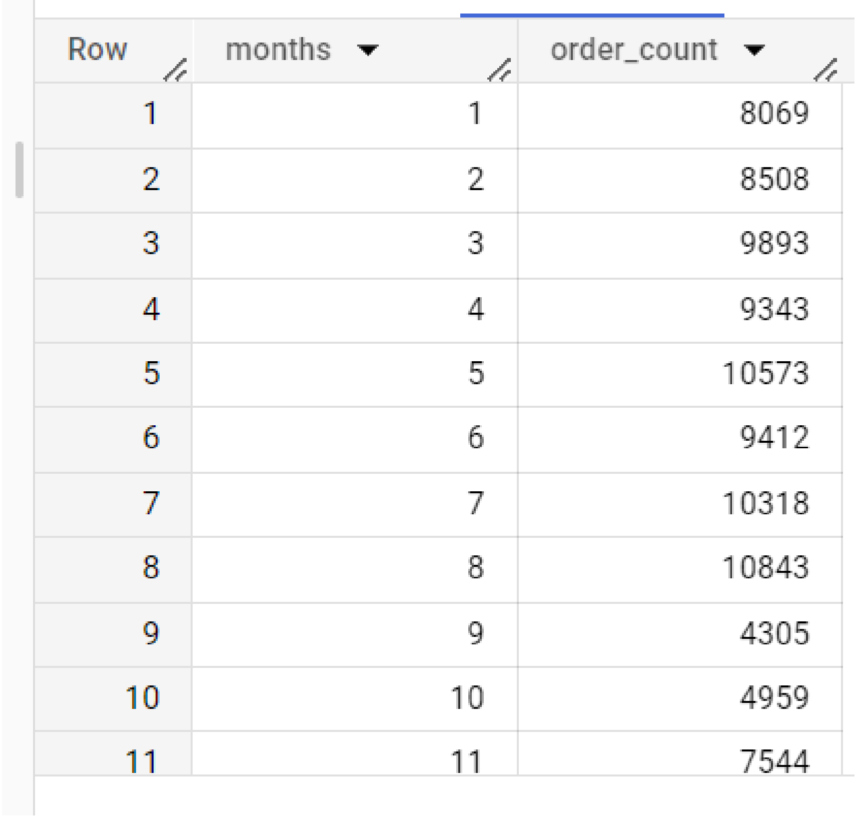


Row	years	order_count
1	2016	329
2	2017	45101
3	2018	54011

INFERENCE: The aggregation function on order table has information of yearly orders count from 2016 to 2018 made at Target in Brazil.

ANS_2.2: Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

```
SELECT
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS months,
COUNT(DISTINCT o.order_id) AS order_count
FROM `target-project-tami.target_project_tami.orders` o
JOIN `target-project-tami.target_project_tami.customer` c
ON o.customer_id = c.customer_id
GROUP BY months
ORDER BY months
```



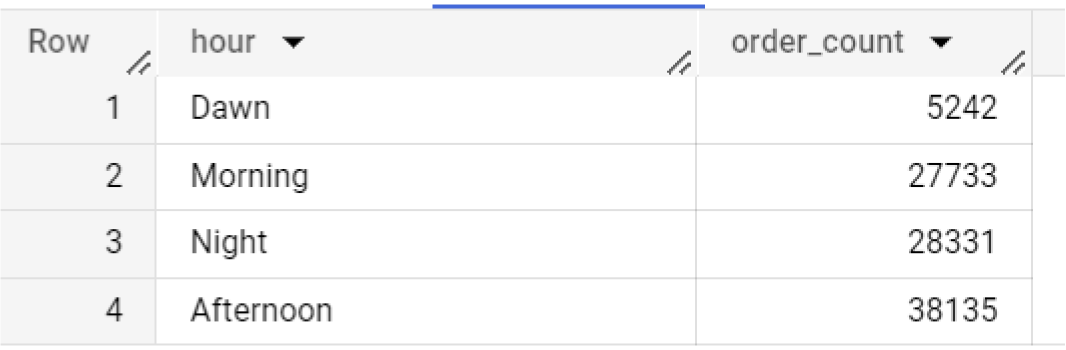
Row	months	order_count
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959
11	11	7544

INFERENCE: The aggregation function on order table has information of monthly orders count from 2016 to 2018 made at Target in Brazil. It is important to note that further analysis with a larger dataset would be required to validate these seasonality trends.

ANS_2.3: During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

1. **0-6 hrs : Dawn**
2. **7-12 hrs : Mornings**
3. **13-18 hrs : Afternoon**
4. **19-23 hrs : Night**

```
SELECT CASE
  WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 0 AND 6 THEN 'Dawn'
  WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 7 AND 12 THEN
'Morning'
  WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 13 AND 18 THEN
'Afternoon'
  WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 19 AND 23 THEN 'Night'
  END AS hour,
COUNT(o.order_id) AS order_count
FROM `target_project_tami.orders` o
JOIN `target_project_tami.customer` c
ON o.customer_id = c.customer_id
GROUP BY HOUR
ORDER BY order_count
```



Row	hour	order_count
1	Dawn	5242
2	Morning	27733
3	Night	28331
4	Afternoon	38135

INFERENCE: The aggregation function on order table has information of most favourable customers order time from 2016 to 2018 made at Target in Brazil. Based on the analysis, we found that Brazilian customers tend to place most orders during the daytime, specifically in the afternoon and night. This indicates that customers prefer to shop online when they have leisure time or after completing their daily activities.

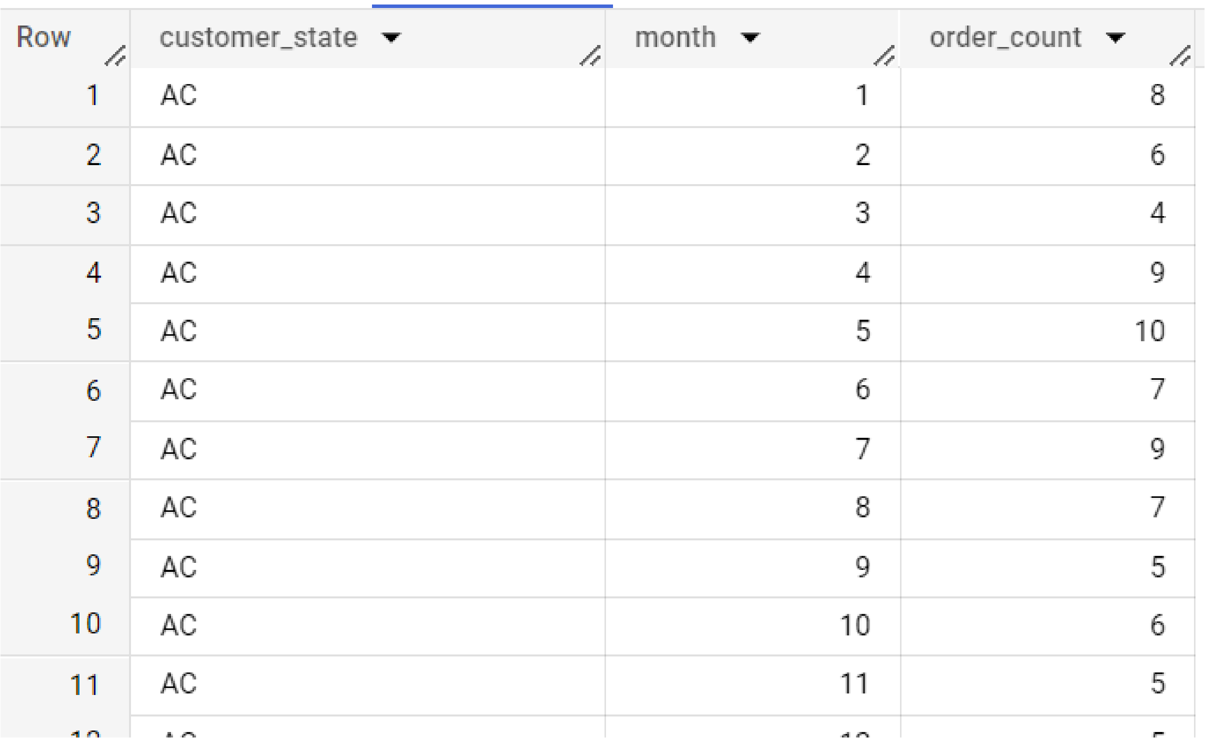
----

3. **Evolution of E-commerce orders in the Brazil region:**
    1. Get the month-on-month no. of orders placed in each state.
    2. How are the customers distributed across all the states?

ANS_3.1: Get the month-on-month no. of orders placed in each state.

```
SELECT
  c.customer_state,
  EXTRACT(month FROM o.order_purchase_timestamp) AS month,
  COUNT(o.order_purchase_timestamp) AS order_count
FROM
  `target-project-tami.target_project_tami.orders` o
```

```
JOIN
  `target-project-tami.target_project_tami.customer` c
ON
  o.customer_id = c.customer_id
GROUP BY
  c.customer_state, month
ORDER BY
  c.customer_state, month;
```

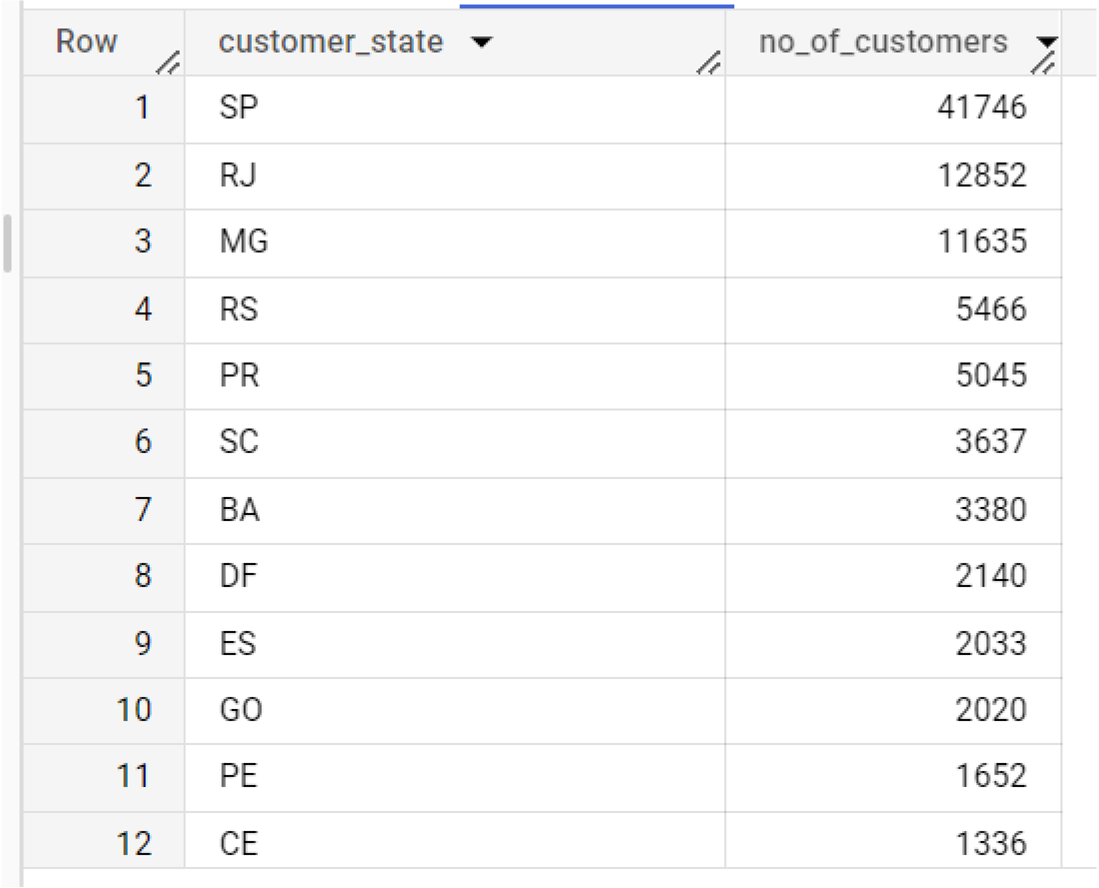


Row	customer_state	month	order_count
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6
11	AC	11	5

INFERENCE: Brazil, providing valuable insights into the customer purchase trends on a state-by-state basis.

ANS_3.2: How are the customers distributed across all the states?

```
SELECT
  c.customer_state,
  COUNT(c.customer_id) AS no_of_customers
FROM
  `target-project-tami.target_project_tami.customer` c
GROUP BY
  c.customer_state
ORDER BY
  no_of_customers DESC
```



Row	customer_state	no_of_customers
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
12	CE	1336

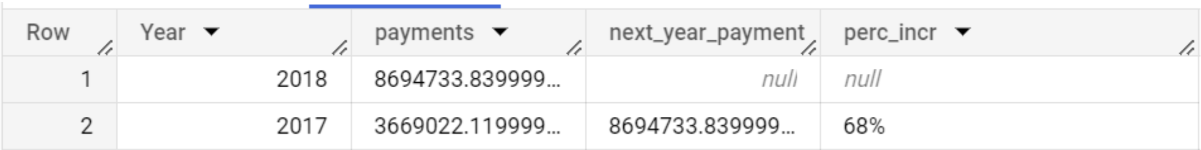
INFERENCE: It is evident that (SP) consistently has the highest number of customers in any given states, followed by (RJ) and (MG).

----

4. **Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.**
   1. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).
      You can use the "payment_value" column in the payments table to get the cost of orders.
   2. Calculate the Total & Average value of order price for each state.
   3. Calculate the Total & Average value of order freight for each state.

ANS_4.1: Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

```
with base as(
SELECT
      EXTRACT(year FROM order_purchase_timestamp) AS Year,
      EXTRACT(month FROM order_purchase_timestamp) AS Month,
      SUM (p.payment_value) AS payments
      FROM `target-project-tami.target_project_tami.orders` o
      INNER JOIN `target-project-tami.target_project_tami.payments` p
      ON o.order_id = p.order_id
      WHERE EXTRACT(month FROM order_purchase_timestamp) BETWEEN 1 AND 8
      GROUP By 1,2
      ORDER BY 1,2
),
base2 as (
          SELECT year,
          SUM(payments) AS payments FROM base
          GROUP BY 1
),
base3 as(
          SELECT *,
          LEAD(payments, 1) OVER (ORDER BY year) AS next_year_payment FROM base2
)
SELECT *,
CONCAT (ROUND ((next_year_payment- payments)/payments *100/2), '%')  AS perc_incr
FROM base3
```

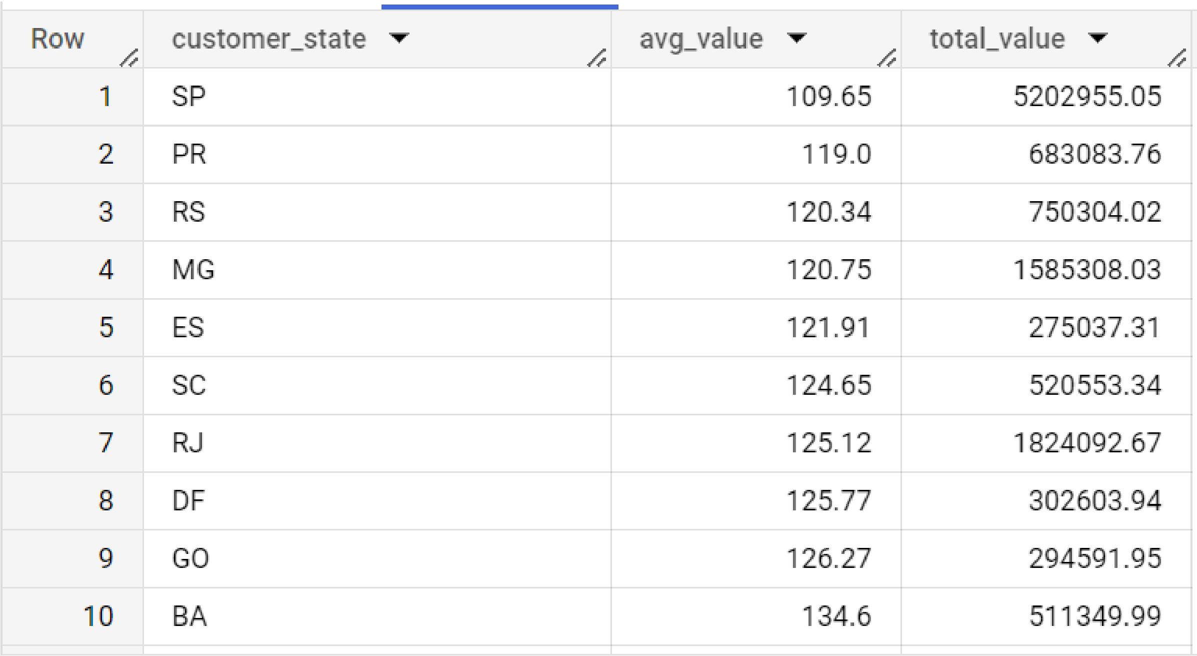


Row	Year	payments	next_year_payment	perc_incr
1	2018	8694733.839999...	null	null
2	2017	3669022.119999...	8694733.839999...	68%

INFERENCE: The overall percentage increase in the cost of orders from 2017 to 2018, including only the months from January to August, is 68%

ANS_4.2: Calculate the Total & Average value of order price for each state.

```
SELECT c.customer_state,
       ROUND(AVG(oi.price), 2) AS avg_value,
       ROUND(SUM(oi.price), 2) AS total_value,
FROM `target-project-tami.target_project_tami.orders` o
JOIN `target_project_tami.order_items` oi ON o.order_id = oi.order_id
JOIN `target_project_tami.customer` c ON o.customer_id = c.customer_id
GROUP BY 1
ORDER BY 2,3
```



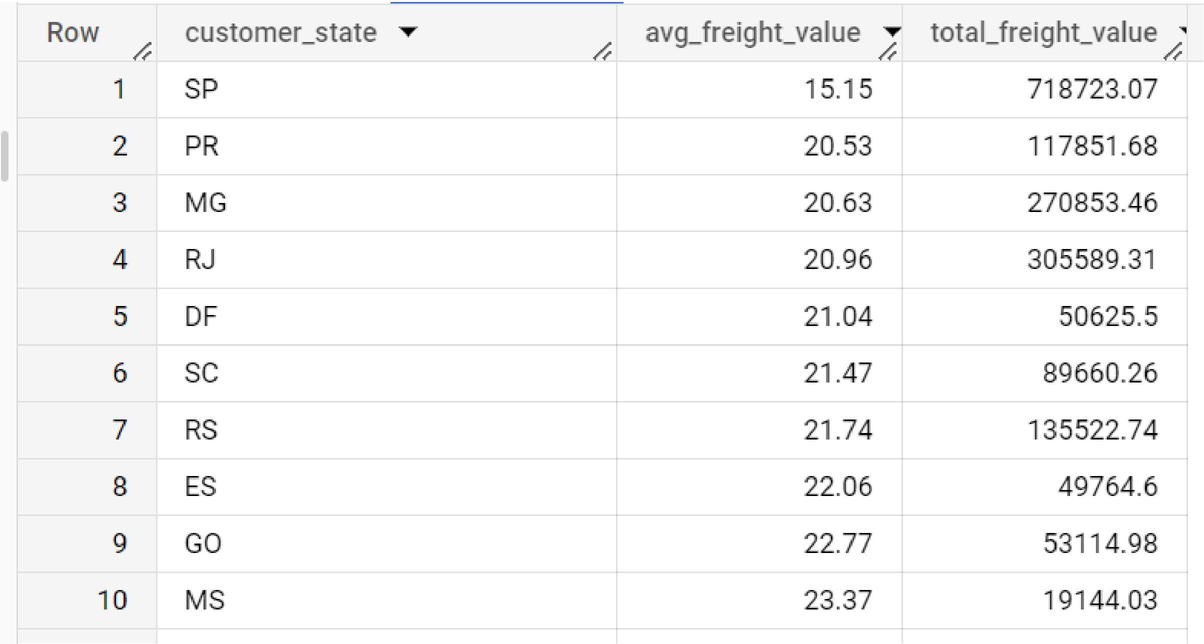
Row	customer_state	avg_value	total_value
1	SP	109.65	5202955.05
2	PR	119.0	683083.76
3	RS	120.34	750304.02
4	MG	120.75	1585308.03
5	ES	121.91	275037.31
6	SC	124.65	520553.34
7	RJ	125.12	1824092.67
8	DF	125.77	302603.94
9	GO	126.27	294591.95
10	BA	134.6	511349.99

INFERENCE: The state SP of Brazil has the highest value of orders

---

ANS_4.3: **Calculate the Total & Average value of order freight for each state.**

```
SELECT c.customer_state,
       ROUND(AVG(oi.freight_value), 2) AS avg_freight_value,
       ROUND(SUM(oi.freight_value), 2) AS total_freight_value,
FROM `target-project-tami.target_project_tami.orders` o
JOIN `target_project_tami.order_items` oi ON o.order_id = oi.order_id
JOIN `target_project_tami.customer` c ON o.customer_id = c.customer_id
GROUP BY 1
ORDER BY 2,3
```



Row	customer_state	avg_freight_value	total_freight_value
1	SP	15.15	718723.07
2	PR	20.53	117851.68
3	MG	20.63	270853.46
4	RJ	20.96	305589.31
5	DF	21.04	50625.5
6	SC	21.47	89660.26
7	RS	21.74	135522.74
8	ES	22.06	49764.6
9	GO	22.77	53114.98
10	MS	23.37	19144.03

INFERENCE: The state SP of Brazil has the lowest average freight value.

---

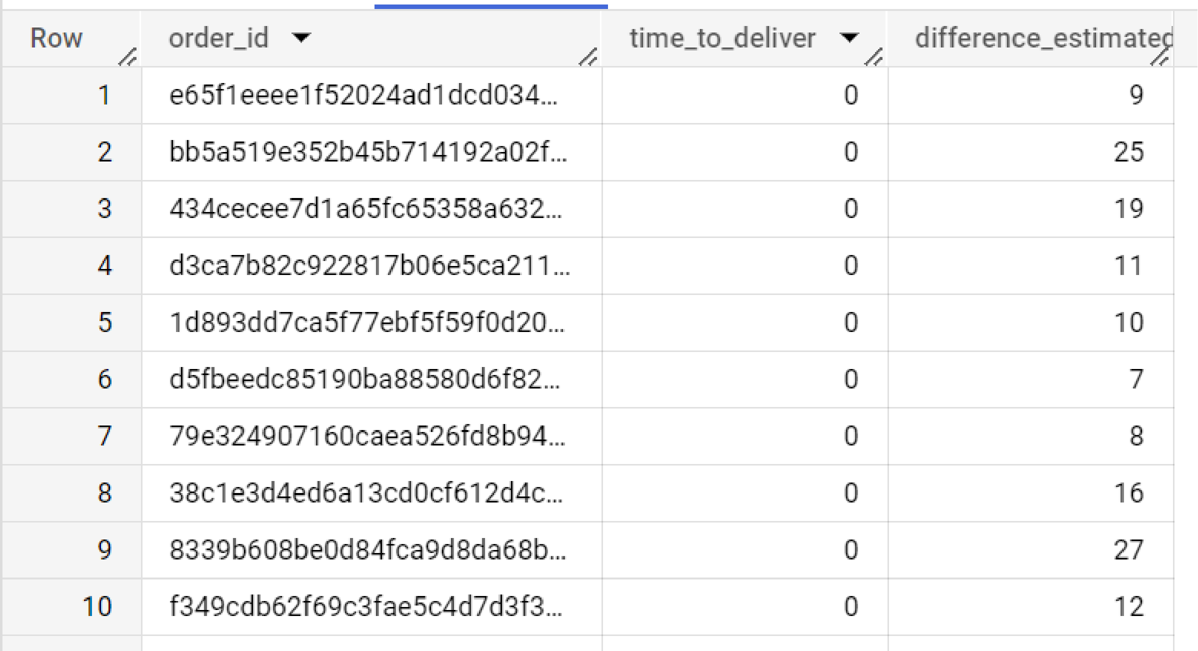
5. **Analysis based on sales, freight and delivery time.**
    1. Find the no. of days taken to deliver each order from the order's purchase date as delivery time.
       Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
       Do this in a single query.

       You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:
        1. **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
        2. **diff_estimated_delivery** = order_estimated_delivery_date - order_delivered_customer_date
    2. Find out the top 5 states with the highest & lowest average freight value.
    3. Find out the top 5 states with the highest & lowest average delivery time.
    4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

ANS_5.1: Find the no. of days taken to deliver each order from the order's purchase date as delivery time.
Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
Do this in a single query.

```
SELECT
  order_id,
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)
  AS time_to_deliver,
  DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY)
  AS difference_estimated_delivery
FROM
  `target-project-tami.target_project_tami.orders`
WHERE
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) IS NOT
NULL
ORDER BY
  time_to_deliver
```



Row	order_id	time_to_deliver	difference_estimated
1	e65f1eeee1f52024ad1dcd034...	0	9
2	bb5a519e352b45b714192a02f...	0	25
3	434cecee7d1a65fc65358a632...	0	19
4	d3ca7b82c922817b06e5ca211...	0	11
5	1d893dd7ca5f77ebf5f59f0d20...	0	10
6	d5fbeedc85190ba88580d6f82...	0	7
7	79e324907160caea526fd8b94...	0	8
8	38c1e3d4ed6a13cd0cf612d4c...	0	16
9	8339b608be0d84fca9d8da68b...	0	27
10	f349cdb62f69c3fae5c4d7d3f3...	0	12

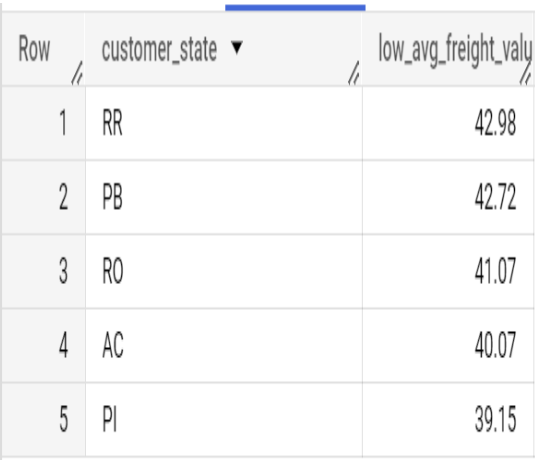
INFERENCE: The time to deliver is showing zero because we have placed the condition day wise.

ANS_5.2: Find out the top 5 states with the highest & lowest average freight value.

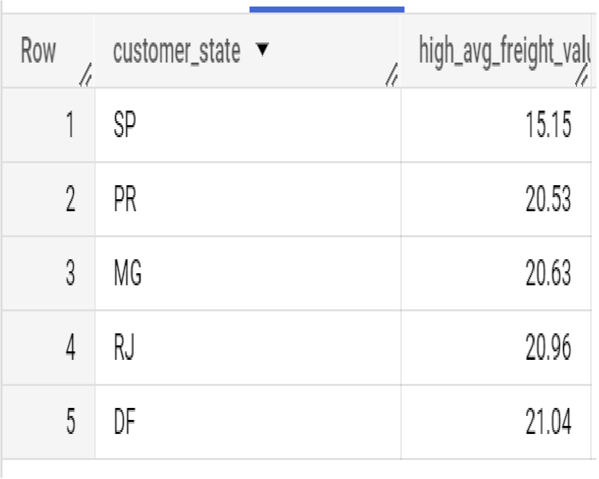
```
SELECT
  c.customer_state,
  ROUND(AVG(oi.freight_value), 2) AS high_avg_freight_value,
FROM
  `target-project-tami.target_project_tami.orders` o
JOIN
```

```
  `target-project-tami.target_project_tami.order_items` oi ON o.order_id =
oi.order_id
JOIN
  `target-project-tami.target_project_tami.customer` c ON o.customer_id =
c.customer_id
GROUP BY
 c.customer_state
ORDER BY high_avg_freight_value ASC
LIMIT 5
```

```
SELECT
  c.customer_state,
  ROUND(AVG(oi.freight_value), 2) AS low_avg_freight_value,
FROM
  `target-project-tami.target_project_tami.orders` o
JOIN
  `target-project-tami.target_project_tami.order_items` oi ON o.order_id =
oi.order_id
JOIN
  `target-project-tami.target_project_tami.customer` c ON o.customer_id =
c.customer_id
GROUP BY
 c.customer_state
ORDER BY low_avg_freight_value DESC
LIMIT 5
```



Row	customer_state ▼	low_avg_freight_valu
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15



Row	customer_state ▼	high_avg_freight_valu
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04

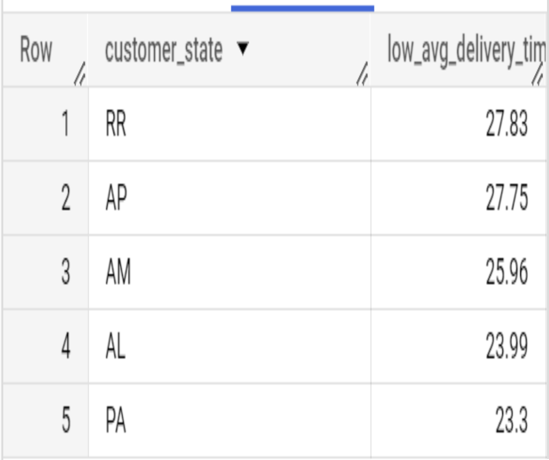
INFERENCE: The highest fright value is SP state and lowest fright value SP state.

ANS_5.3: Find out the top 5 states with the highest & lowest average delivery time.

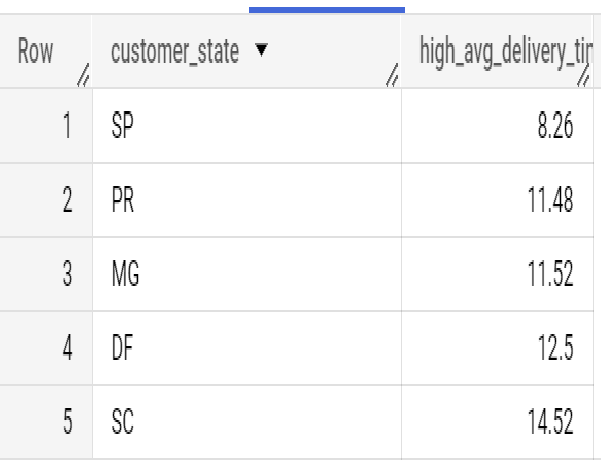
```
SELECT
  c.customer_state,
  ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
DAY)), 2)
  AS high_avg_delivery_time
FROM
  `target-project-tami.target_project_tami.orders` o
```

```
JOIN
  `target-project-tami.target_project_tami.order_items` oi ON o.order_id =
oi.order_id
JOIN
  `target-project-tami.target_project_tami.customer` c ON o.customer_id =
c.customer_id
GROUP BY
 c.customer_state
ORDER BY high_avg_delivery_time
LIMIT 5
```

```
SELECT
  c.customer_state,
  ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
DAY)), 2)
  AS low_avg_delivery_time
FROM
  `target-project-tami.target_project_tami.orders` o
JOIN
  `target-project-tami.target_project_tami.order_items` oi ON o.order_id =
oi.order_id
JOIN
  `target-project-tami.target_project_tami.customer` c ON o.customer_id =
c.customer_id
GROUP BY
 c.customer_state
ORDER BY low_avg_delivery_time DESC
LIMIT 5
```



Row	customer_state ▼	low_avg_delivery_tim
1	RR	27.83
2	AP	27.75
3	AM	25.96
4	AL	23.99
5	PA	23.3



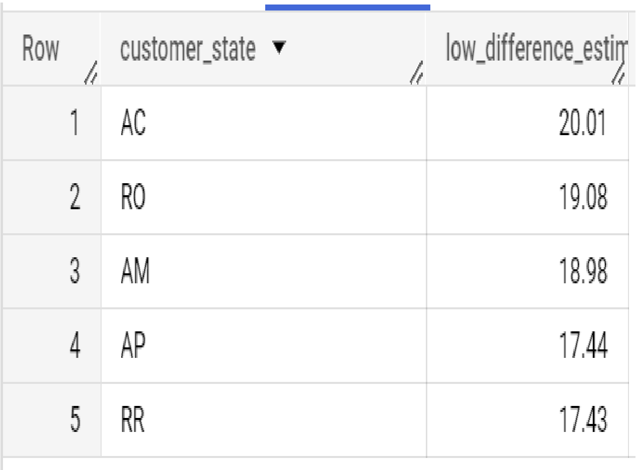
Row	customer_state ▼	high_avg_delivery_tir
1	SP	8.26
2	PR	11.48
3	MG	11.52
4	DF	12.5
5	SC	14.52

INFERENCE: The highest fright value SP state has highest average delivery time and lowest fright value RR state has lowest average delivery time. By this we can say that SP state is the largest state and has placed many orders compared to other states.

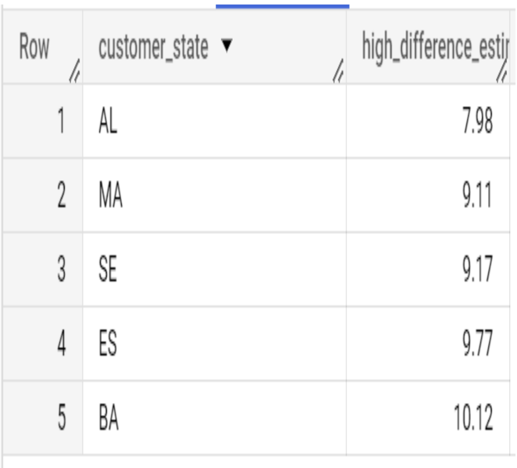
ANS_5.4: Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

```
SELECT
  c.customer_state,
  ROUND(AVG(DATE_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date, DAY)), 2)
  AS high_difference_estimated_delivery
FROM
  `target-project-tami.target_project_tami.orders` o
JOIN
  `target-project-tami.target_project_tami.order_items` oi ON o.order_id =
oi.order_id
JOIN
  `target-project-tami.target_project_tami.customer` c ON o.customer_id =
c.customer_id
GROUP BY
 c.customer_state
ORDER BY high_difference_estimated_delivery
LIMIT 5
```

```
SELECT
  c.customer_state,
  ROUND(AVG(DATE_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date, DAY)), 2)
  AS low_difference_estimated_delivery
FROM
  `target-project-tami.target_project_tami.orders` o
JOIN
  `target-project-tami.target_project_tami.order_items` oi ON o.order_id =
oi.order_id
JOIN
  `target-project-tami.target_project_tami.customer` c ON o.customer_id =
c.customer_id
GROUP BY
 c.customer_state
ORDER BY low_difference_estimated_delivery DESC
LIMIT 5
```



Row	customer_state ▼	low_difference_estir
1	AC	20.01
2	RO	19.08
3	AM	18.98
4	AP	17.44
5	RR	17.43



Row	customer_state ▼	high_difference_esti
1	AL	7.98
2	MA	9.11
3	SE	9.17
4	ES	9.77
5	BA	10.12

INFERENCE: The analysis reveals a weak positive correlation between avg freight value and time to delivery.

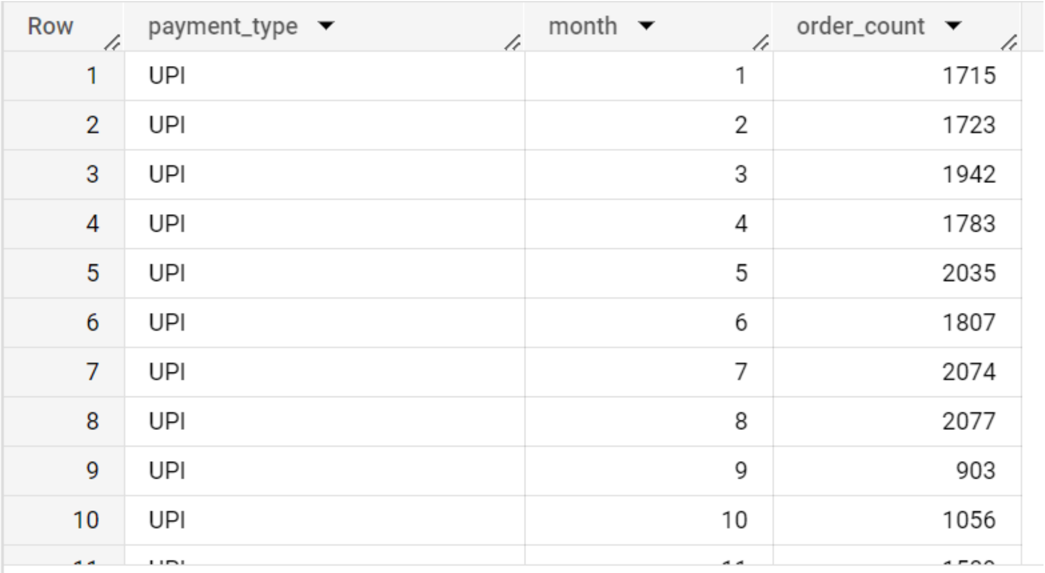
----

6. **Analysis based on the payments:**

1. Find the month on month no. of orders placed using different payment types.
2. Find the no. of orders placed on the basis of the payment installments that have been paid.

ANS_6.1: Find the month on month no. of orders placed using different payment types.

```
SELECT
      p.payment_type,
      EXTRACT (MONTH FROM o.order_purchase_timestamp) AS month,
      COUNT(DISTINCT o.order_id) AS order_count
FROM `target_project_tami.orders` o
JOIN `target_project_tami.payments` p
ON o.order_id = p.order_id
GROUP BY 1, 2
ORDER BY 1,2
```

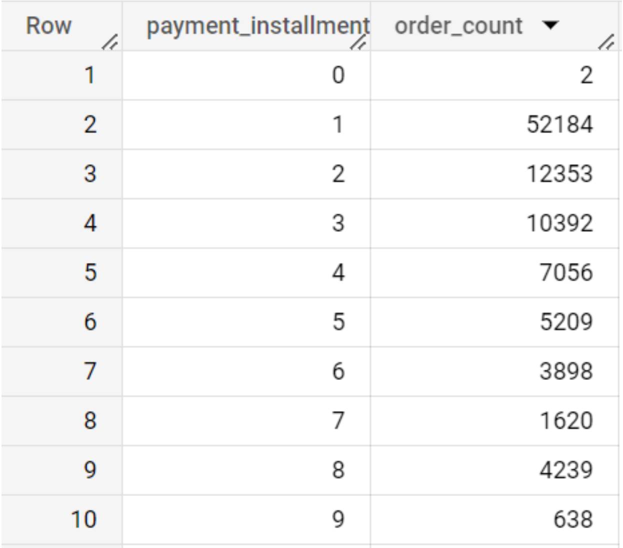


Row	payment_type	month	order_count
1	UPI	1	1715
2	UPI	2	1723
3	UPI	3	1942
4	UPI	4	1783
5	UPI	5	2035
6	UPI	6	1807
7	UPI	7	2074
8	UPI	8	2077
9	UPI	9	903
10	UPI	10	1056
11	UPI	11	[illegible]

INFERENCE: The analysis is based in the payment.

ANS_6.2: Find the no. of orders placed on the basis of the payment installments that have been paid.

```
SELECT
      p.payment_installments,
      COUNT(o.order_id) AS order_count
FROM `target_project_tami.orders` o
JOIN `target_project_tami.payments` p
ON o.order_id = p.order_id
WHERE o.order_status != 'canceled'
GROUP BY 1
ORDER BY 1,2
```



Row	payment_installment	order_count ▼
1	0	2
2	1	52184
3	2	12353
4	3	10392
5	4	7056
6	5	5209
7	6	3898
8	7	1620
9	8	4239
10	9	638

INFERENCE: The number of orders that has been made by that means which has been not cancelled and the payment method chosen has been installments.
