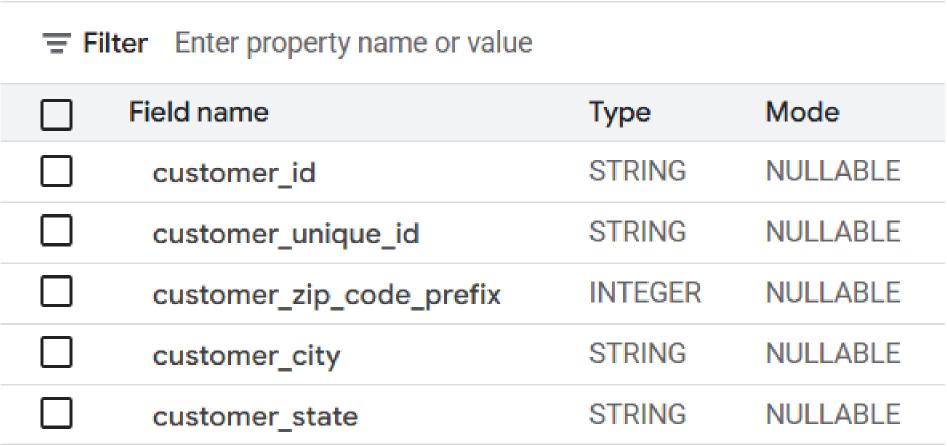
Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

a. Data type of all columns in the "customers" table.



Filter Enter property name or value

□	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

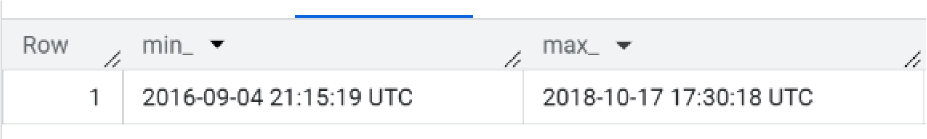
b. Get the time range between which the orders were placed.

- Query:

```
SELECT min(order_purchase_timestamp) as min_,

    max(order_purchase_timestamp) as max_
FROM `target-case-study-project.target.orders`
```

- Results:



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

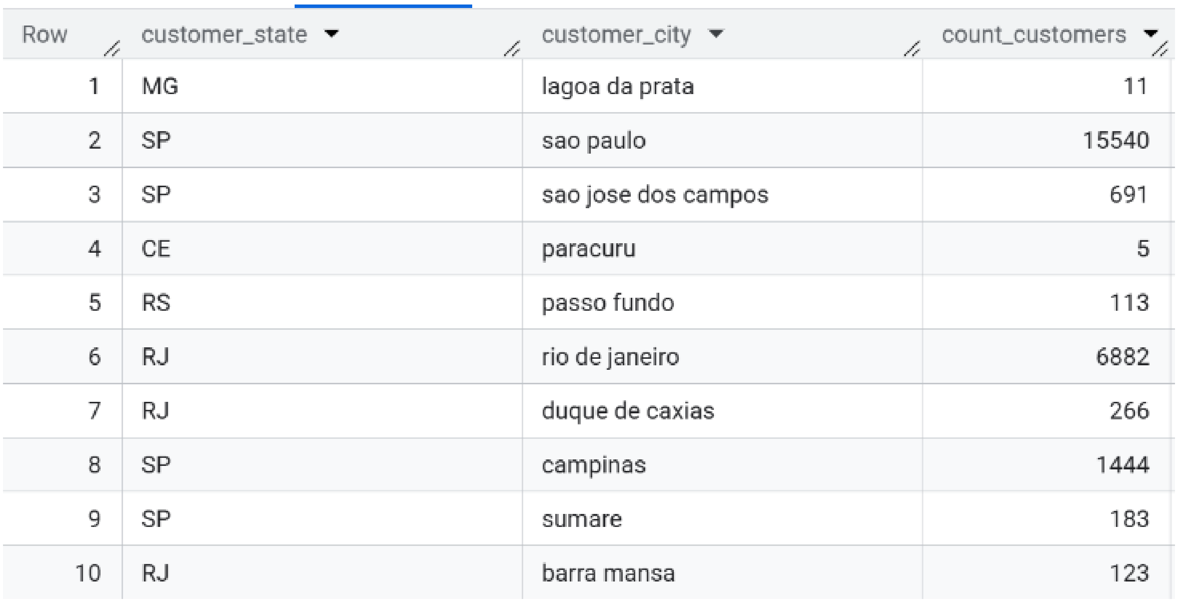
- Insights:
  The data available with us is for the time period of the placed orders is between September 2016-October 2018.

c. Count the Cities & States of customers who ordered during the given period.

- Query:

```
SELECT c.customer_state, c.customer_city, COUNT(DISTINCT c.customer_id) as
count_customers
FROM `target-case-study-project.target.orders` o
INNER JOIN `target-case-study-project.target.customers` c
ON o.customer_id=c.customer_id
GROUP BY 1,2
```

- Results:



Row	customer_state	customer_city	count_customers
1	MG	lagoa da prata	11
2	SP	sao paulo	15540
3	SP	sao jose dos campos	691
4	CE	paracuru	5
5	RS	passo fundo	113
6	RJ	rio de janeiro	6882
7	RJ	duque de caxias	266
8	SP	campinas	1444
9	SP	sumare	183
10	RJ	barra mansa	123

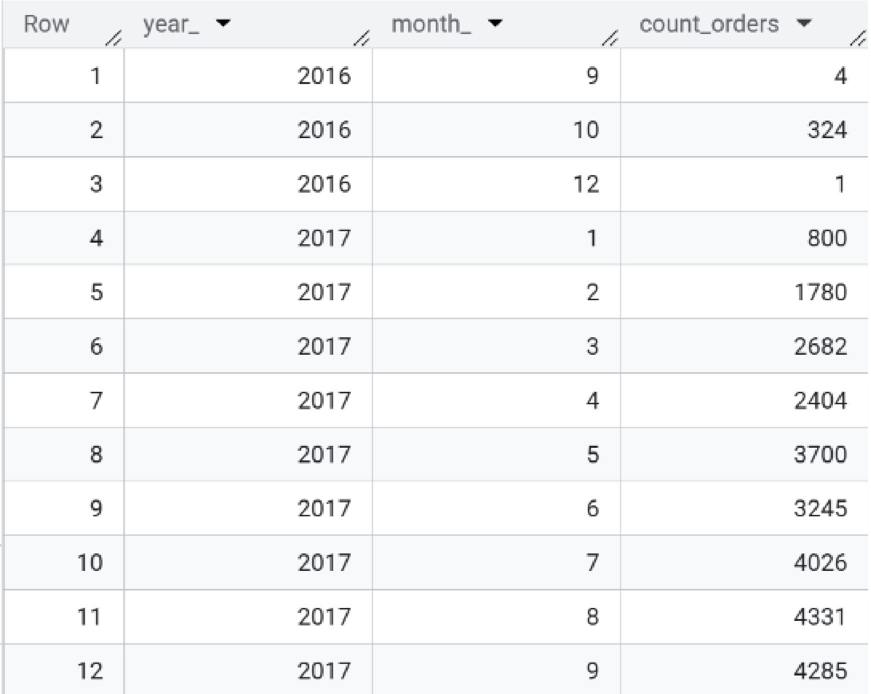
In-depth Exploration:

a. Is there a growing trend in the no. of orders placed over the past years?
   Trend of orders placed month-by-month:

   - Query:

```
SELECT
EXTRACT(year FROM order_purchase_timestamp) as year_,
EXTRACT(month FROM order_purchase_timestamp) as month_,
COUNT(order_id) as count_orders
FROM `target-case-study-project.target.orders`
GROUP BY EXTRACT(year FROM order_purchase_timestamp), EXTRACT(month FROM
order_purchase_timestamp)
ORDER BY year_, month_
```

   - Results:



Row	year_	month_	count_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
11	2017	8	4331
12	2017	9	4285

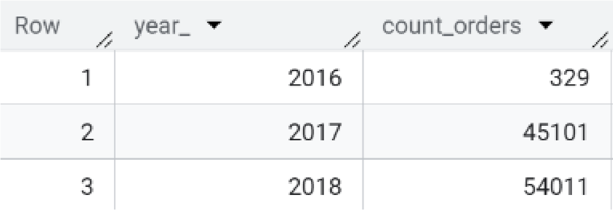
- Insights:
  The month-on month number of orders increase monotonically from Jan 2017 till Jan 2018 after which it somewhat remains constant.

Trend of orders placed year-by-year:

- Query:

```
SELECT
EXTRACT(year FROM order_purchase_timestamp) as year_,
COUNT(order_id) as count_orders
 FROM `target-case-study-project.target.orders`
GROUP BY EXTRACT(year FROM order_purchase_timestamp)
ORDER BY year_
```

- Results:



Row	year_	count_orders
1	2016	329
2	2017	45101
3	2018	54011

- Insights:
  The number of orders placed increase monotonically over years.

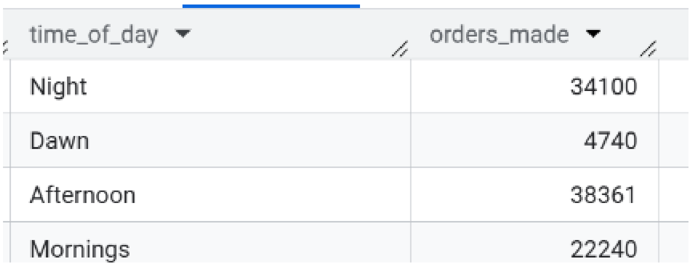
b. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?
   No

c. During what time of the day, do the Brazilian customers mostly place their orders?

- Query:

```
SELECT time_of_day, Count(order_id) as orders_made
FROM
(
SELECT *,
    CASE
      WHEN hour_ BETWEEN 0 AND 5 THEN 'Dawn'
      WHEN hour_ BETWEEN 6 AND 11 THEN 'Mornings'
      WHEN hour_ BETWEEN 12 AND 17 THEN 'Afternoon'
      ELSE 'Night'
    END AS time_of_day
FROM (
   SELECT order_id,
         order_purchase_timestamp,
         EXTRACT(HOUR FROM order_purchase_timestamp) AS hour_
   FROM `target-case-study-project.target.orders`
) a
) b
GROUP BY time_of_day
```

- Result:



time_of_day	orders_made
Night	34100
Dawn	4740
Afternoon	38361
Mornings	22240

- Insights:
  Most orders in the Brazil are placed in the Afternoon time (38361 orders total) followed by Night, Morning. The least number of orders get placed at Dawn (Only 4740 total orders).

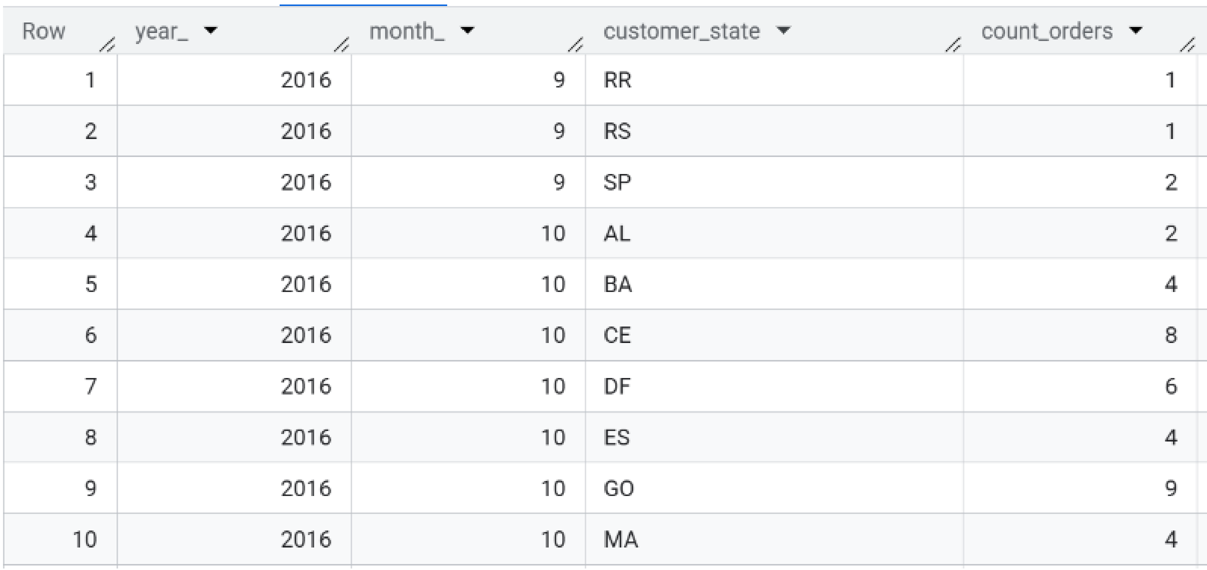
Evolution of E-commerce orders in the Brazil region:

a. Get the month on month no. of orders placed in each state.

- Query:

```
SELECT year_, month_, customer_state, Count(order_id) as count_orders
FROM (
  SELECT order_id,
       customer_state,
        EXTRACT(year FROM order_purchase_timestamp) AS year_,
        EXTRACT(month FROM order_purchase_timestamp) AS month_
  FROM `target-case-study-project.target.orders` oi
  INNER JOIN `target-case-study-project.target.customers` c
  ON oi.customer_id = c.customer_id
) a
GROUP BY year_, month_, customer_state
Order BY year_, month_, customer_state
```

- Results:



Row	year_	month_	customer_state	count_orders
1	2016	9	RR	1
2	2016	9	RS	1
3	2016	9	SP	2
4	2016	10	AL	2
5	2016	10	BA	4
6	2016	10	CE	8
7	2016	10	DF	6
8	2016	10	ES	4
9	2016	10	GO	9
10	2016	10	MA	4

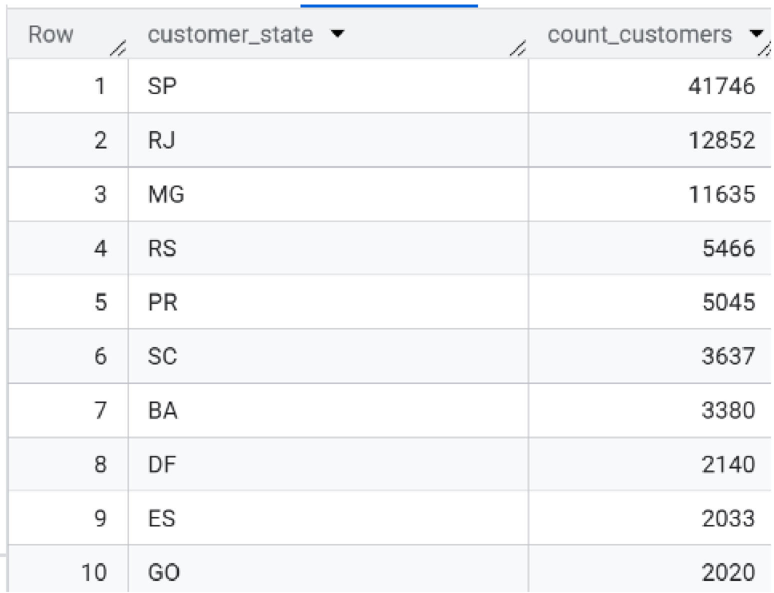
- Insights:
  The maximum number of orders were placed by customers from SP state in the month of Jan-2017 with total orders placed were 299.

b. How are the customers distributed across all the states?

- Query:

```
SELECT customer_state,
 COUNT(customer_id) as count_customers
 FROM `target-case-study-project.target.customers`
 GROUP BY customer_state
 Order BY count_customers DESC
```

- Results:



Row	customer_state	count_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

- Insights:
  The maximum number of customers are from the Sao Paulo followed by Rio De Janeiro. There are only 68 customers from the state Amapa and 46 from Roraima which is the lowest.

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

a. 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.

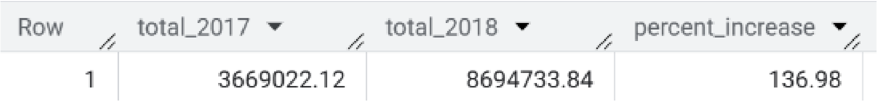
- Query:

```
WITH monthly_costs AS (
  SELECT
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year_,
    ROUND(SUM(p.payment_value), 2) AS total_cost
  FROM `target-case-study-project.target.orders` o
  INNER JOIN (
    SELECT
      order_id,
      ROUND(SUM(payment_value), 2) AS payment_value
    FROM `target-case-study-project.target.payments`
    GROUP BY order_id
  ) p ON o.order_id = p.order_id
  WHERE EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
```

```
     AND EXTRACT(YEAR FROM o.order_purchase_timestamp) IN (2017, 2018)
  GROUP BY year_
)

SELECT
  MAX(CASE WHEN year_ = 2017 THEN total_cost END) AS total_2017,
  MAX(CASE WHEN year_ = 2018 THEN total_cost END) AS total_2018,
  ROUND(
    (MAX(CASE WHEN year_ = 2018 THEN total_cost END) -
     MAX(CASE WHEN year_ = 2017 THEN total_cost END))
    / MAX(CASE WHEN year_ = 2017 THEN total_cost END) * 100,
    2
  ) AS percent_increase
FROM monthly_costs;
```

- Results:



Row	total_2017	total_2018	percent_increase
1	3669022.12	8694733.84	136.98

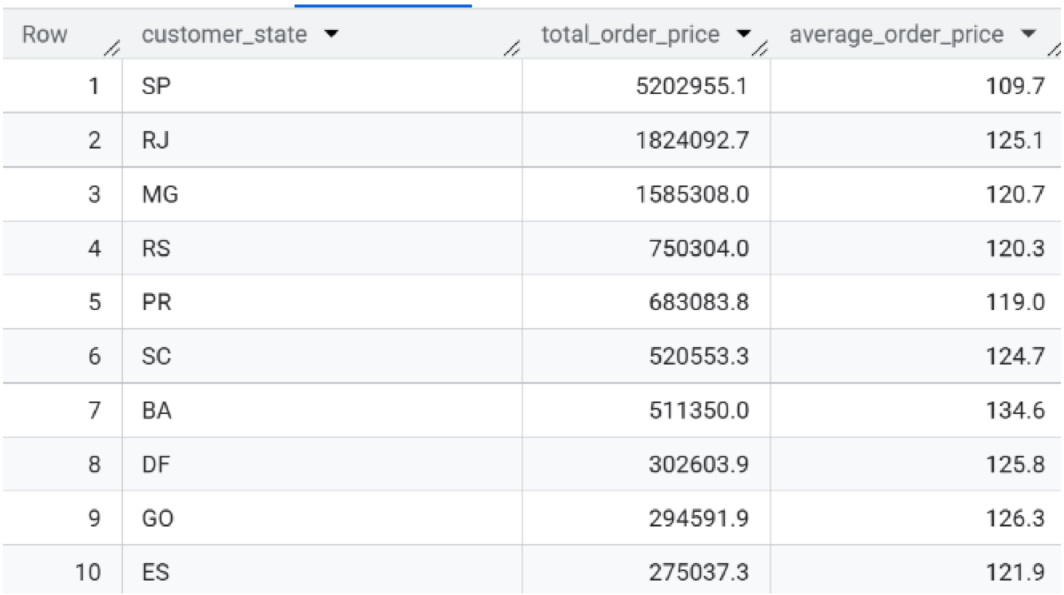
- Insights:
  There is approximately 130% increase in the cost of orders from year 2017 to 2018

b. Calculate the Total & Average value of order price for each state.

- Query:

```
SELECT
 customer_state,
 ROUND(SUM(price), 1) AS total_order_price,
 ROUND(AVG(price), 1) AS average_order_price
FROM `target-case-study-project.target.order_items` oi
INNER JOIN `target-case-study-project.target.orders` o
 ON oi.order_id = o.order_id
INNER JOIN `target-case-study-project.target.customers` c
 ON o.customer_id = c.customer_id
GROUP BY customer_state
ORDER BY total_order_price DESC;
```

- Results:



Row	customer_state	total_order_price	average_order_price
1	SP	5202955.1	109.7
2	RJ	1824092.7	125.1
3	MG	1585308.0	120.7
4	RS	750304.0	120.3
5	PR	683083.8	119.0
6	SC	520553.3	124.7
7	BA	511350.0	134.6
8	DF	302603.9	125.8
9	GO	294591.9	126.3
10	ES	275037.3	121.9

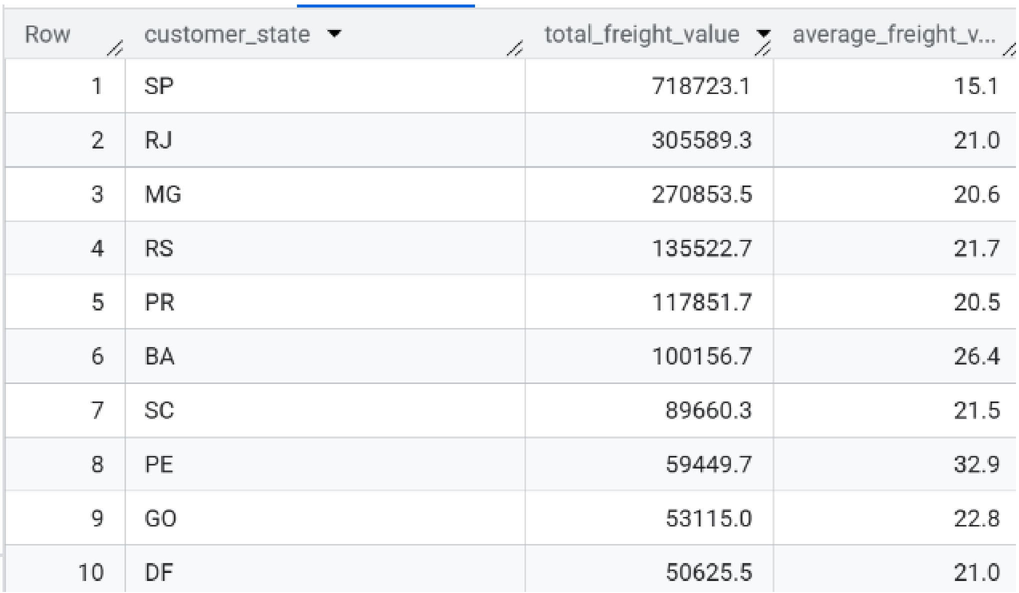
- Insights:
  The total value of order price is maximum for Sao Paulo and minimum for Roraima. Whereas the average value of order price is maximum for Paraiba and minimum for Sao Paulo. Meaning the per customer order price for SP is the minimum but the total order price is maximum.

c. Calculate the Total & Average value of order freight for each state.

- Query:

```
SELECT
  customer_state,
  ROUND(SUM(freight_value), 1) AS total_freight_value,
  ROUND(AVG(freight_value), 1) AS average_freight_value
FROM `target-case-study-project.target.order_items` oi
INNER JOIN `target-case-study-project.target.orders` o
  ON oi.order_id = o.order_id
INNER JOIN `target-case-study-project.target.customers` c
  ON o.customer_id = c.customer_id
GROUP BY customer_state
ORDER BY total_freight_value DESC;
```

- Results:



Row	customer_state	total_freight_value	average_freight_v...
1	SP	718723.1	15.1
2	RJ	305589.3	21.0
3	MG	270853.5	20.6
4	RS	135522.7	21.7
5	PR	117851.7	20.5
6	BA	100156.7	26.4
7	SC	89660.3	21.5
8	PE	59449.7	32.9
9	GO	53115.0	22.8
10	DF	50625.5	21.0

- Insights:
  The total value of order freight is maximum for Sao Paulo and minimum for Roraima. Whereas the average value of order freight is maximum for Roraima and minimum for Sao Paulo. Meaning the per customer order freight for SP is the minimum but the total order freight is maximum. Also, the per customer order freight for RR is the maximum but the total order freight is minimum.

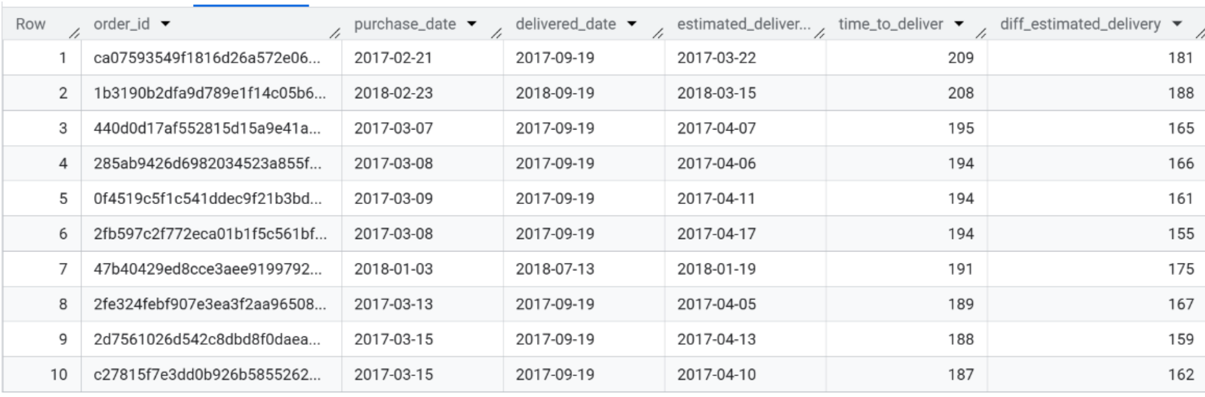
Analysis based on sales, freight and delivery time.

a. 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.

- Query:

```
SELECT
  order_id,
  DATE(order_purchase_timestamp) AS purchase_date,
  DATE(order_delivered_customer_date) AS delivered_date,
  DATE(order_estimated_delivery_date) AS estimated_delivery_date,
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
time_to_deliver,
  DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date, DAY) AS
diff_estimated_delivery
FROM `target-case-study-project.target.orders`
WHERE order_delivered_customer_date IS NOT NULL
  AND order_estimated_delivery_date IS NOT NULL
ORDER BY time_to_deliver DESC;
```

- Results:



Row	order_id	purchase_date	delivered_date	estimated_deliver...	time_to_deliver	diff_estimated_delivery
1	ca07593549f1816d26a572e06...	2017-02-21	2017-09-19	2017-03-22	209	181
2	1b3190b2dfa9d789e1f14c05b6...	2018-02-23	2018-09-19	2018-03-15	208	188
3	440d0d17af552815d15a9e41a...	2017-03-07	2017-09-19	2017-04-07	195	165
4	285ab9426d6982034523a855f...	2017-03-08	2017-09-19	2017-04-06	194	166
5	0f4519c5f1c541ddec9f21b3bd...	2017-03-09	2017-09-19	2017-04-11	194	161
6	2fb597c2f772eca01b1f5c561bf...	2017-03-08	2017-09-19	2017-04-17	194	155
7	47b40429ed8cce3aee9199792...	2018-01-03	2018-07-13	2018-01-19	191	175
8	2fe324febf907e3ea3f2aa96508...	2017-03-13	2017-09-19	2017-04-05	189	167
9	2d7561026d542c8dbd8f0daea...	2017-03-15	2017-09-19	2017-04-13	188	159
10	c27815f7e3dd0b926b5855262...	2017-03-15	2017-09-19	2017-04-10	187	162

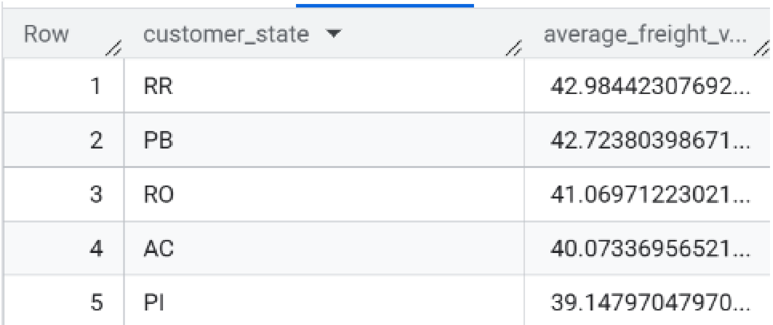
- Insights:
  The maximum delay between the estimated and actual delivery of any product has been 188 days.

b. Find out the top 5 states with the highest & lowest average freight value.

- Query:

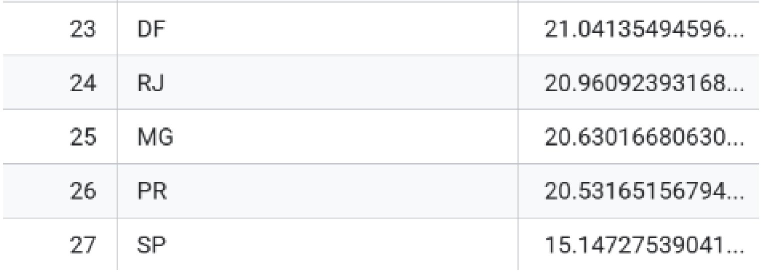
```
SELECT customer_state, AVG(freight_value) as average_freight_value
FROM `target-case-study-project.target.order_items` oi
INNER JOIN `target-case-study-project.target.orders` o
  ON oi.order_id = o.order_id
INNER JOIN `target-case-study-project.target.customers` c
  ON o.customer_id = c.customer_id
GROUP BY customer_state
ORDER BY average_freight_value DESC;
```

- Results:
  Top 5 highest average freight value states:



Row	customer_state	average_freight_v...
1	RR	42.98442307692...
2	PB	42.72380398671...
3	RO	41.06971223021...
4	AC	40.07336956521...
5	PI	39.14797047970...

Top 5 lowest average freight value states:



23	DF	21.04135494596...
24	RJ	20.96092393168...
25	MG	20.63016680630...
26	PR	20.53165156794...
27	SP	15.14727539041...

- Insights:
  The top 5 states with highest average freight value are:
  RR, PB, RO, AC, PI.
  The top 5 states with lowest average freight value are:

DF, RJ, MG, PR, SP.

c. Find out the top 5 states with the highest & lowest average delivery time.

- Query:

  Top 5 states with the highest average delivery time:

```
-- CTE to calculate delivery time per order
WITH delivery_time_per_order AS (
  SELECT
    o.order_id,
    c.customer_state,
    DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
time_to_deliver
  FROM `target-case-study-project.target.orders` o
  JOIN `target-case-study-project.target.customers` c
    ON o.customer_id = c.customer_id
  WHERE order_delivered_customer_date IS NOT NULL
)

-- Average delivery time by state
SELECT customer_state,
       ROUND(AVG(time_to_deliver), 2) AS avg_delivery_days
FROM delivery_time_per_order
GROUP BY customer_state
ORDER BY avg_delivery_days DESC
LIMIT 5;
```

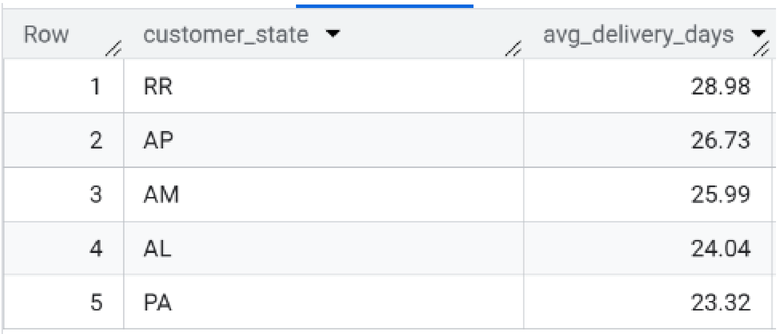
  Top 5 states with the highest average delivery time:

```
-- CTE to calculate delivery time per order
WITH delivery_time_per_order AS (
  SELECT
    o.order_id,
    c.customer_state,
    DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
time_to_deliver
  FROM `target-case-study-project.target.orders` o
  JOIN `target-case-study-project.target.customers` c
    ON o.customer_id = c.customer_id
  WHERE order_delivered_customer_date IS NOT NULL
)

-- Average delivery time by state
SELECT customer_state,
       ROUND(AVG(time_to_deliver), 2) AS avg_delivery_days
FROM delivery_time_per_order
GROUP BY customer_state
ORDER BY avg_delivery_days Asc
LIMIT 5;
```

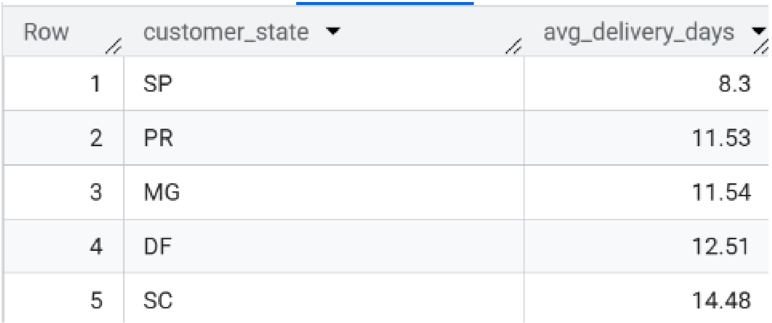
- Results and Insights:

  Top 5 states with the highest average delivery time:



Row	customer_state	avg_delivery_days
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32

Top 5 states with the lowest average delivery time:



Row	customer_state	avg_delivery_days
1	SP	8.3
2	PR	11.53
3	MG	11.54
4	DF	12.51
5	SC	14.48

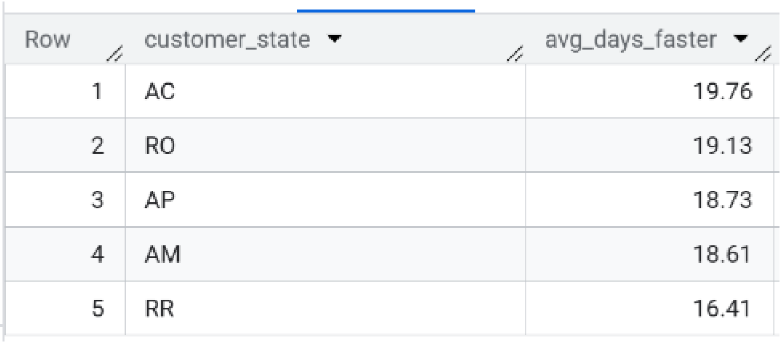
d. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

- Query:

```
WITH delivery_speed AS (
  SELECT
    c.customer_state,
    DATE_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY) AS days_faster
  FROM `target-case-study-project.target.orders` o
  JOIN `target-case-study-project.target.customers` c
    ON o.customer_id = c.customer_id
  WHERE o.order_delivered_customer_date IS NOT NULL
    AND o.order_estimated_delivery_date IS NOT NULL
)

SELECT
  customer_state,
  ROUND(AVG(days_faster), 2) AS avg_days_faster
FROM delivery_speed
GROUP BY customer_state
HAVING avg_days_faster > 0
ORDER BY avg_days_faster DESC
LIMIT 5;
```

- Results & Insights:



Row	customer_state	avg_days_faster
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

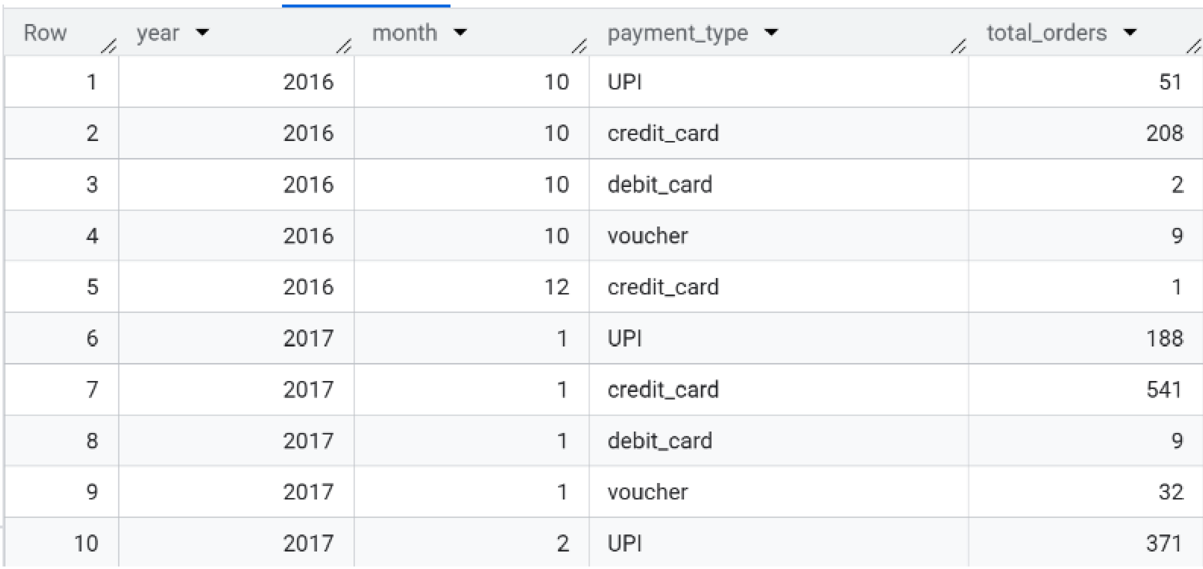
Analysis based on the payments:

a. Find the month on month no. of orders placed using different payment types.

- Query:

```
SELECT
 EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
 EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
 p.payment_type,
 COUNT(DISTINCT o.order_id) AS total_orders
FROM `target-case-study-project.target.orders` o
JOIN `target-case-study-project.target.payments` p
 ON o.order_id = p.order_id
WHERE o.order_status = 'delivered' -- optional: only count completed orders
GROUP BY year, month, p.payment_type
ORDER BY year, month, p.payment_type;
```

- Results:



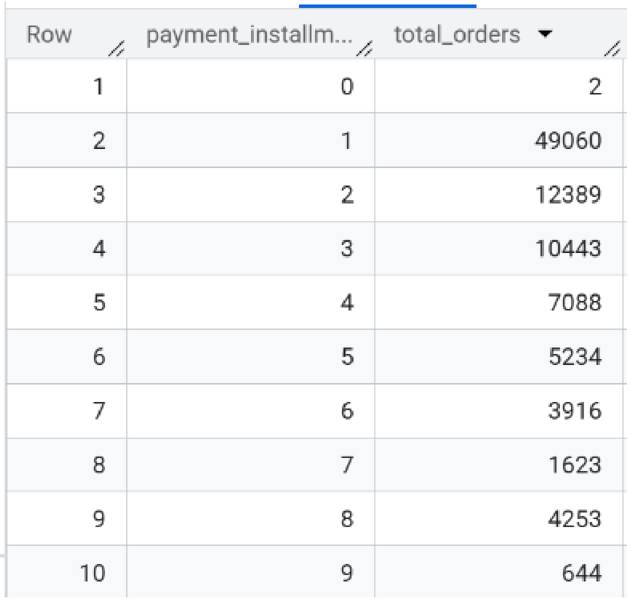
Row	year	month	payment_type	total_orders
1	2016	10	UPI	51
2	2016	10	credit_card	208
3	2016	10	debit_card	2
4	2016	10	voucher	9
5	2016	12	credit_card	1
6	2017	1	UPI	188
7	2017	1	credit_card	541
8	2017	1	debit_card	9
9	2017	1	voucher	32
10	2017	2	UPI	371

b. Find the no. of orders placed on the basis of the payment instalments that have been paid.

- Query:

```
SELECT
  payment_installments,
  COUNT(DISTINCT order_id) AS total_orders
FROM `target-case-study-project.target.payments`
GROUP BY payment_installments
ORDER BY payment_installments;
```

- Results:



Row	payment_installm...	total_orders
1	0	2
2	1	49060
3	2	12389
4	3	10443
5	4	7088
6	5	5234
7	6	3916
8	7	1623
9	8	4253
10	9	644