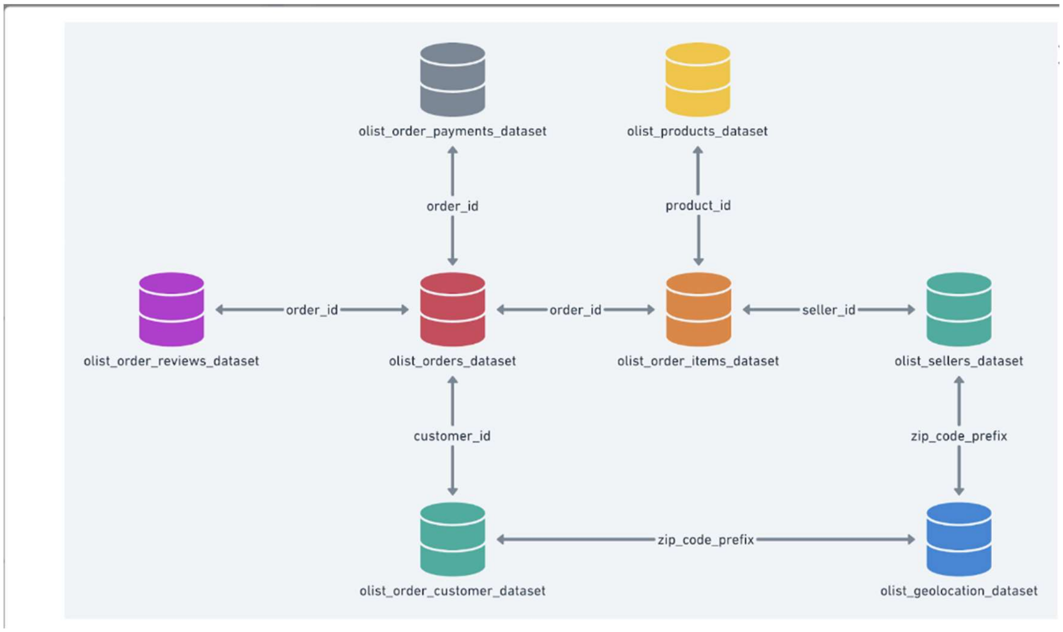
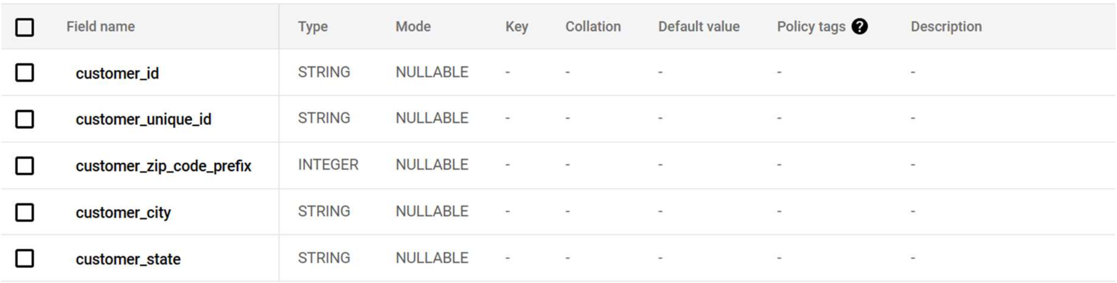
TARGET SQL CASE STUDY

[BigQuery – target – Google Cloud console]



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

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

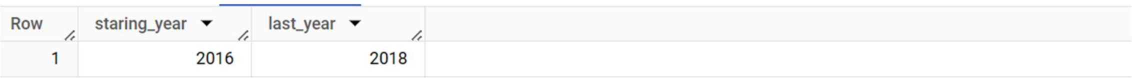


□	Field name	Type	Mode	Key	Collation	Default value	Policy tags	Description
□	customer_id	STRING	NULLABLE	-	-	-	-	-
□	customer_unique_id	STRING	NULLABLE	-	-	-	-	-
□	customer_zip_code_prefix	INTEGER	NULLABLE	-	-	-	-	-
□	customer_city	STRING	NULLABLE	-	-	-	-	-
□	customer_state	STRING	NULLABLE	-	-	-	-	-

Data type in customer table are of string, Integer type. Similarly for other tables datatypes are float, Date time etc.

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

```
SELECT
 EXTRACT (year
  FROM
    MIN (order_purchase_timestamp)) AS staring_year,
  EXTRACT (year
  FROM
    MAX (order_purchase_timestamp)) AS last_year
FROM target-391702.target.orders
```

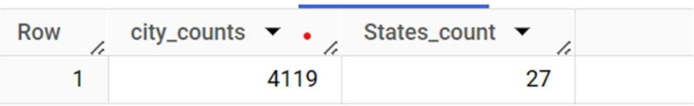


Row	staring_year	last_year
1	2016	2018

Range of date is between year 2016 and 2018.

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

```
SELECT
  count(distinct c.customer_city) as City_counts,
  count(distinct c.customer_state) as States_count
FROM
  target-391702.target.orders AS o
LEFT JOIN
  target-391702.target.customers AS c
ON
  o.customer_id = c.customer_id
```



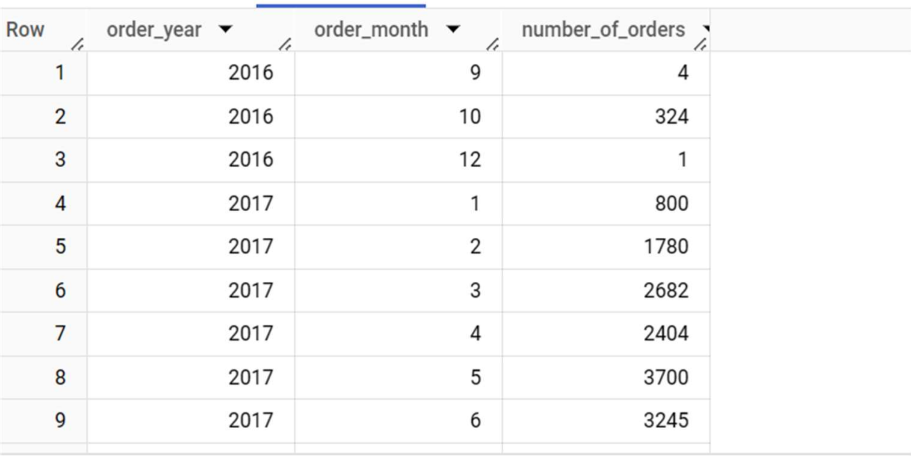
Row	city_counts	States_count
1	4119	27

There are 4119 cities and 27 states in the dataset.

2. In-depth Exploration:

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

```
SELECT
  order_year,
  order_month,
  COUNT(order_id) as number_of_orders
FROM (SELECT
    *, EXTRACT(year FROM order_purchase_timestamp) AS order_year,
      EXTRACT(month FROM order_purchase_timestamp) AS order_month
    FROM target.orders)  X
GROUP BY order_month, order_year
ORDER BY order_year, order_month
```



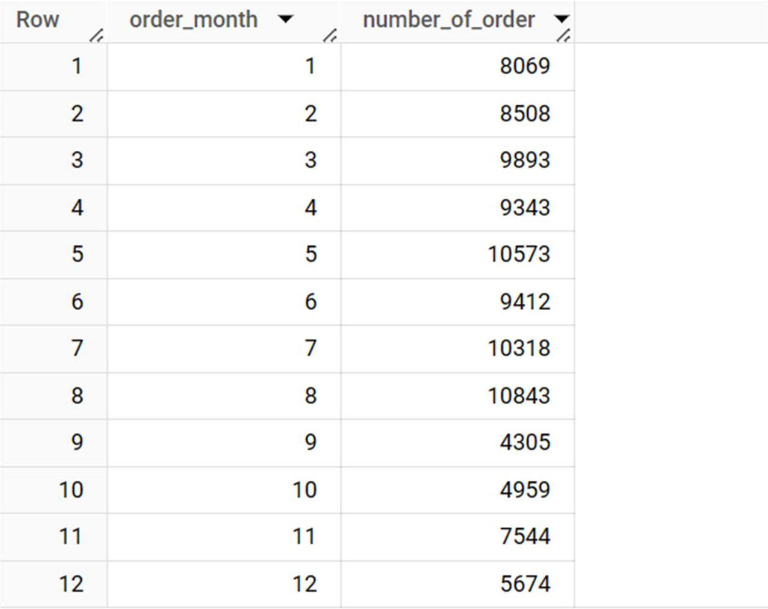
Row	order_year	order_month	number_of_orders
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245

There is increasing trend on number of order placed each year. Number of orders are increased gradually from 2016 to 2018, with its peak at November 2017

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

```
SELECT
 order_month,
 COUNT(order_id) as number_of_order
FROM (
 SELECT
  *,
  EXTRACT(year
  FROM
   order_purchase_timestamp) AS order_year,
  EXTRACT(month
  FROM
   order_purchase_timestamp) AS order_month
 FROM
  target.orders) X
GROUP BY
 order_month

 order by order_month
```



Row	order_month	number_of_order
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
12	12	5674

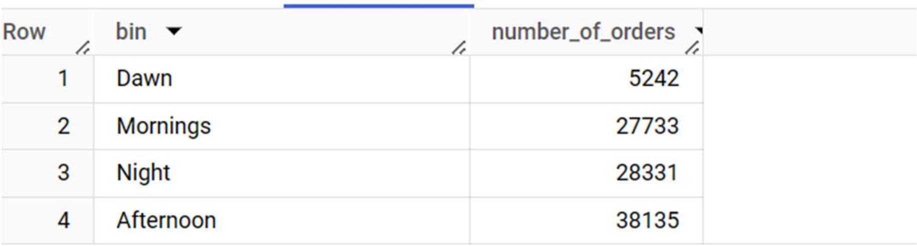
There is some kind of monthly seasonality no. of order places which peaks at May, Jun, July and August month.

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

- **0-6 hrs: Dawn**
- **7-12 hrs: Mornings**
- **13-18 hrs: Afternoon**
- **19-23 hrs: Night**

```
SELECT
 bin,
 COUNT(distinct order_id) AS number_of_orders

FROM (
 SELECT
  order_id,
  CASE
   WHEN EXTRACT(hour FROM order_purchase_timestamp) BETWEEN 0 AND 6 THEN "Dawn"
   WHEN EXTRACT(hour
  FROM
   order_purchase_timestamp) BETWEEN 7
  AND 12 THEN "Mornings"
   WHEN EXTRACT(hour FROM order_purchase_timestamp) BETWEEN 13 AND 18 THEN
"Afternoon"
  ELSE
  "Night"
 END
  AS bin
 FROM
  target.orders) AS X
GROUP BY
 bin
order by number_of_orders
```



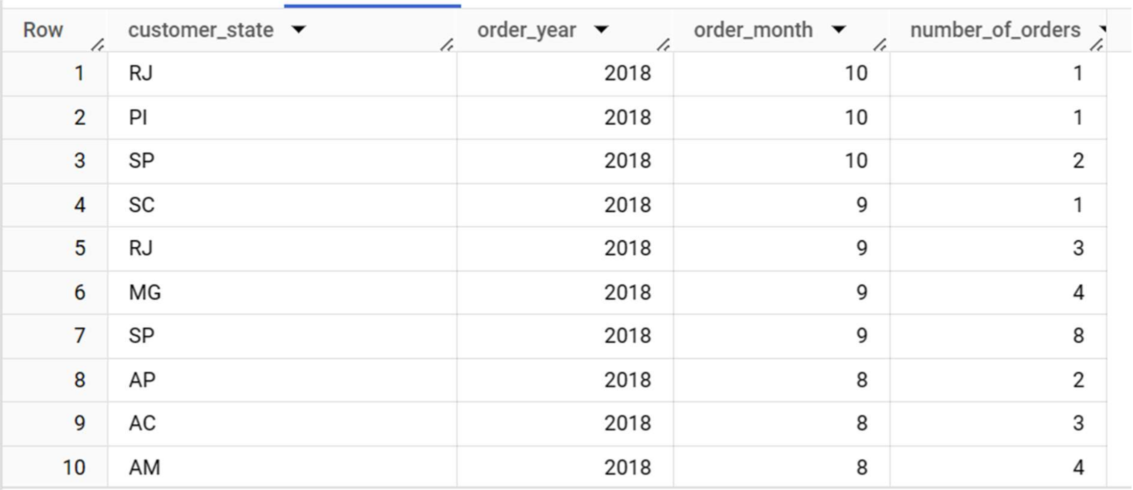
Row	bin	number_of_orders
1	Dawn	5242
2	Mornings	27733
3	Night	28331
4	Afternoon	38135

Most of the order were placed at Afternoon followed by Night and Morning, this could suggest the perfect time for flash sale shall be Dawn so the number of order can be increased at Dawn time.

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

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

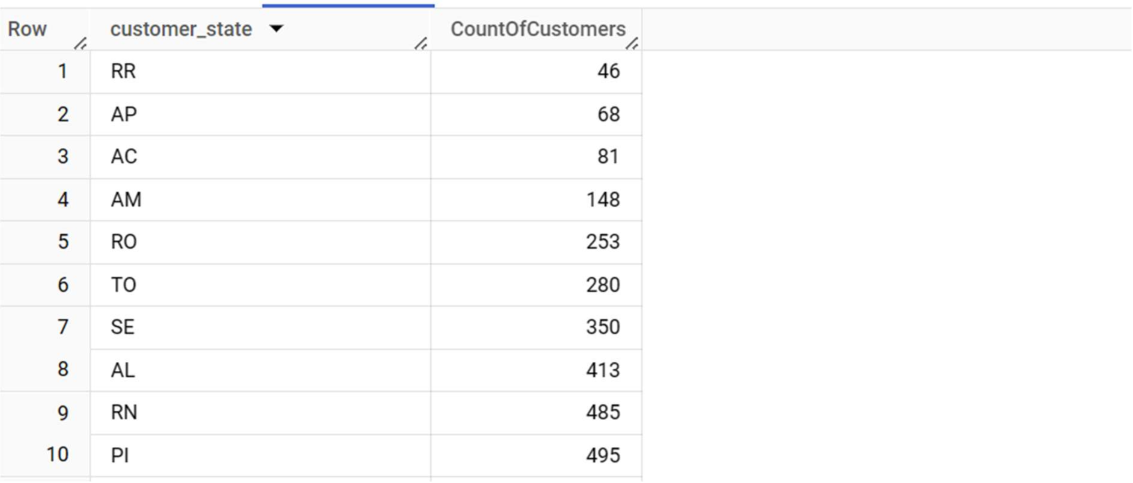
```
SELECT
 customer_state, order_year,
 order_month,
 COUNT(DISTINCT order_id) AS number_of_orders
FROM (
 SELECT
  order_id,
  EXTRACT(year
  FROM
   order_purchase_timestamp) AS order_year,
  EXTRACT(Month
  FROM
   order_purchase_timestamp) order_month,
  c.customer_state
 FROM
  target.orders AS o
 LEFT JOIN
  target.customers AS c
 ON
  c.customer_id = o.customer_id) X
GROUP BY
order_year, order_month, customer_state
ORDER BY
 order_year desc, order_month desc,
 COUNT(order_id)
```



Row	customer_state	order_year	order_month	number_of_orders
1	RJ	2018	10	1
2	PI	2018	10	1
3	SP	2018	10	2
4	SC	2018	9	1
5	RJ	2018	9	3
6	MG	2018	9	4
7	SP	2018	9	8
8	AP	2018	8	2
9	AC	2018	8	3
10	AM	2018	8	4

3.2 How are the customers distributed across all the states?

```
SELECT
  customer_state,
  COUNT(DISTINCT customer_id) AS CountOfCustomers
FROM
  `target.customers`
GROUP BY
  customer_state
ORDER BY
  CountOfCustomers
```

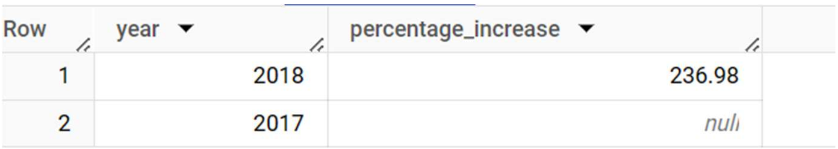


Row	customer_state	CountOfCustomers
1	RR	46
2	AP	68
3	AC	81
4	AM	148
5	RO	253
6	TO	280
7	SE	350
8	AL	413
9	RN	485
10	PI	495

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

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

```
WITH
 cte1 AS (
 SELECT
  EXTRACT(year
  FROM
   order_purchase_timestamp) AS year_jan_to_aug,
  payment_value
 FROM
  `target.payments` AS p
 LEFT JOIN
  target.orders AS o
 ON
  p.order_id= o.order_id
 WHERE
  EXTRACT(month
  FROM
   order_purchase_timestamp) BETWEEN 1
  AND 8),
 cte2 AS(
 SELECT
  year_jan_to_aug AS year,
  SUM(payment_value) AS payment_value
 FROM
  cte1
 GROUP BY
  year_jan_to_aug),
 cte3 AS (
 SELECT
  *,
  LAG(payment_value, 1) OVER (ORDER BY year) AS last_year_payment
 FROM
  cte2)
SELECT
 year,
 ROUND((payment_value*100/last_year_payment), 2) AS percentage_increase
FROM
 cte3
```

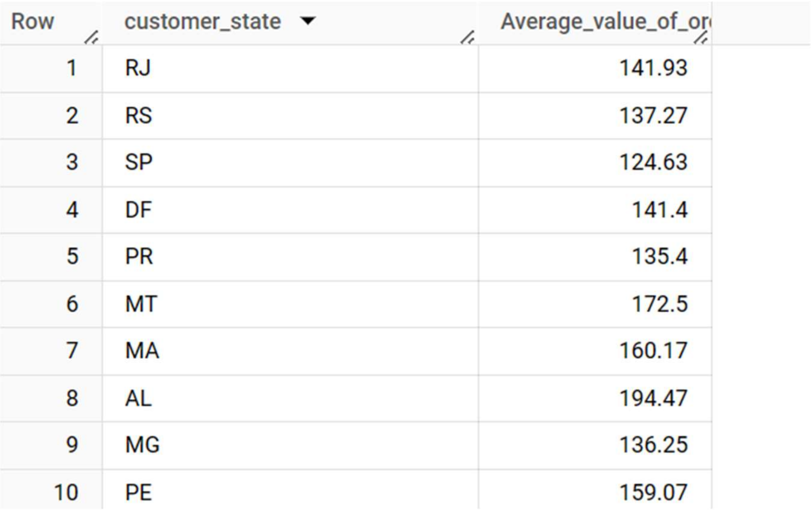


Row	year	percentage_increase
1	2018	236.98
2	2017	null

The year 2018 saw a 236% increase in cost of orders compared to 2017.

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

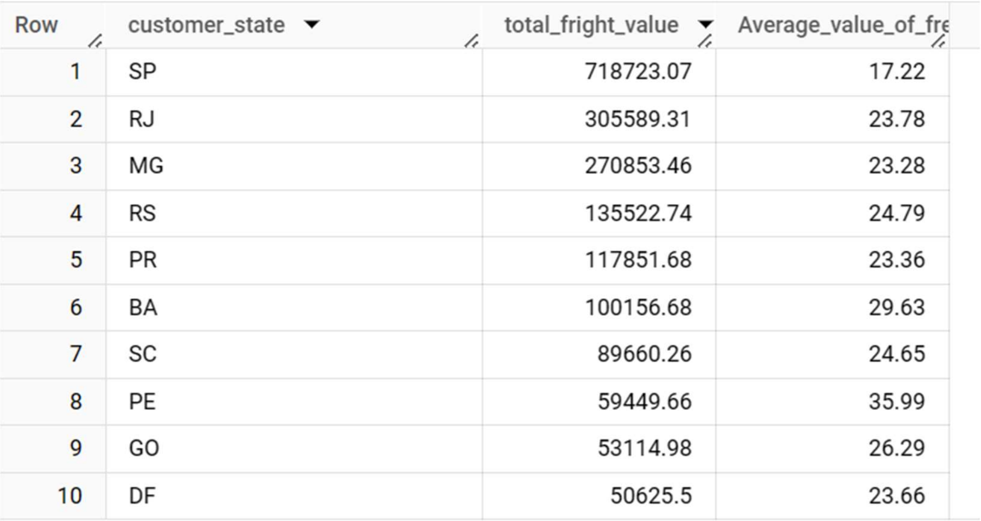
```
WITH
 cte1 AS(
 SELECT
  o.order_id,
  c.customer_state,
  price
 FROM
  target.orders o
 LEFT JOIN
  target.customers c
 ON
  c.customer_id = o.customer_id
 LEFT JOIN
  target.order_items oi
 ON
  oi.order_id = o.order_id)
SELECT
 customer_state,
 ROUND(SUM(price)/COUNT(DISTINCT order_id), 2) AS Average_value_of_order
FROM
 cte1
GROUP BY
 customer_state
```



Row	customer_state	Average_value_of_or
1	RJ	141.93
2	RS	137.27
3	SP	124.63
4	DF	141.4
5	PR	135.4
6	MT	172.5
7	MA	160.17
8	AL	194.47
9	MG	136.25
10	PE	159.07

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

```
WITH
 cte1 AS(
 SELECT
  o.order_id,
  c.customer_state,
  freight_value
 FROM
  target.orders o
 LEFT JOIN
  target.customers c
 ON
  c.customer_id = o.customer_id
 LEFT JOIN
  target.order_items oi
 ON
  oi.order_id = o.order_id)
SELECT
 customer_state,
 ROUND(SUM(freight_value), 2) AS total_fright_value,
 ROUND(SUM(freight_value)/COUNT(DISTINCT order_id), 2) AS Average_value_of_freight
FROM
 cte1
GROUP BY
 customer_state
ORDER BY
 ROUND(SUM(freight_value), 2) DESC,
 ROUND(SUM(freight_value)/COUNT(DISTINCT order_id), 2) DESC
LIMIT 10
```

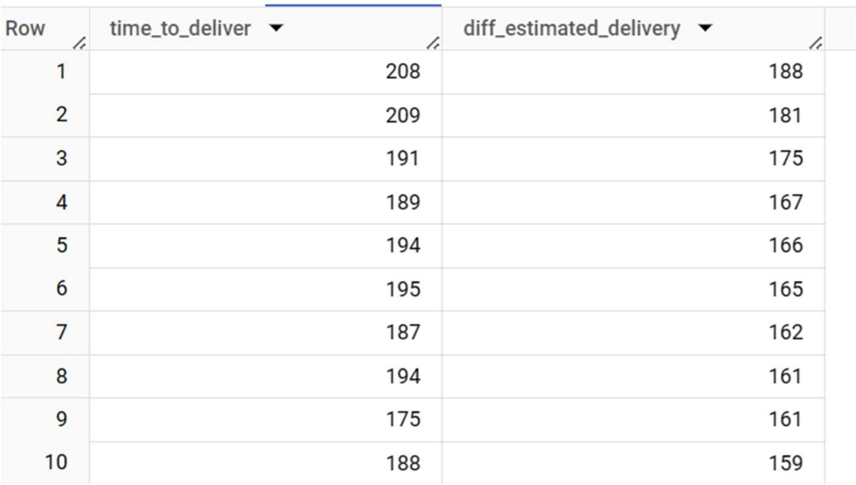


Row	customer_state	total_fright_value	Average_value_of_fre
1	SP	718723.07	17.22
2	RJ	305589.31	23.78
3	MG	270853.46	23.28
4	RS	135522.74	24.79
5	PR	117851.68	23.36
6	BA	100156.68	29.63
7	SC	89660.26	24.65
8	PE	59449.66	35.99
9	GO	53114.98	26.29
10	DF	50625.5	23.66

5. Analysis based on sales, freight and delivery time

5.1 Find the no. of days taken to deliver each order from the order’s purchase date as delivery time.

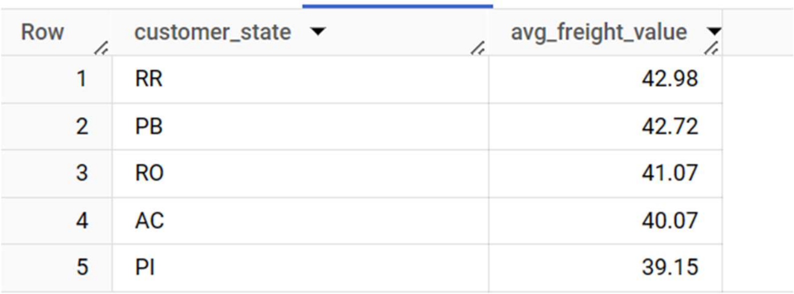
```
SELECT
 DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day) AS time_to_deliver,
 DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date, day) AS
diff_estimated_delivery
FROM
 `target.orders`
ORDER BY
 DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date, day) desc
```



Row	time_to_deliver	diff_estimated_delivery
1	208	188
2	209	181
3	191	175
4	189	167
5	194	166
6	195	165
7	187	162
8	194	161
9	175	161
10	188	159

5.2 Top 5 states with the highest average freight value and Top 5 with lowest.

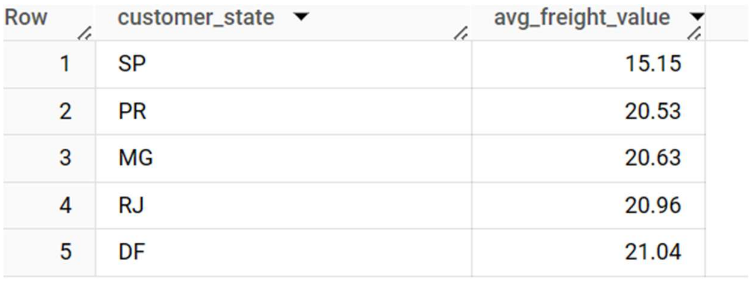
```
SELECT
  customer_state,
  ROUND(AVG(freight_value), 2) AS avg_freight_value
FROM
  target-391702.target.customers AS c
LEFT JOIN
  target-391702.target.orders AS o
ON c.customer_id = o.customer_id
LEFT JOIN target.order_items AS oi
ON o.order_id = oi.order_id
GROUP BY customer_state
ORDER BY AVG(freight_value) desc
LIMIT 5
```



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

Top 5 states with lowest average freight value.

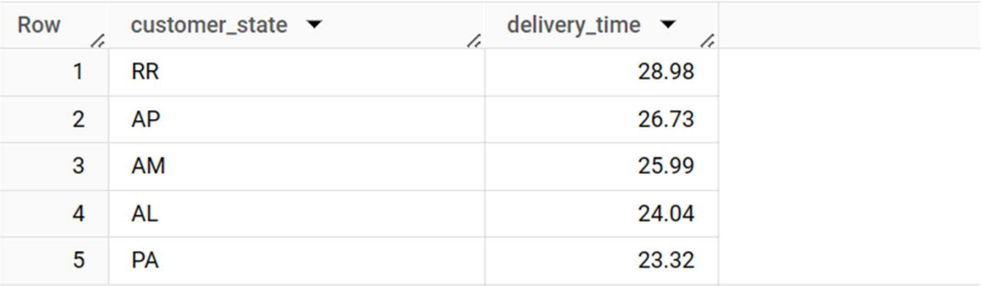
```
SELECT
 customer_state,
 ROUND(AVG(freight_value), 2) AS avg_freight_value
FROM
 target-391702.target.customers AS c
LEFT JOIN
 target-391702.target.orders AS o
ON
 c.customer_id = o.customer_id
LEFT JOIN
 target.order_items AS oi
ON
 o.order_id = oi.order_id
GROUP BY
 customer_state
ORDER BY
 AVG(freight_value)
LIMIT
 5
```



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

5.3 Find out the top 5 states with the highest average delivery time.

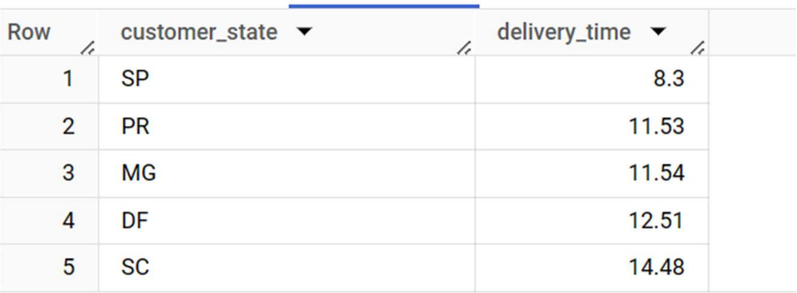
```
SELECT
  customer_state,
  ROUND(AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day)),2) AS
delivery_time,
FROM
  target.orders AS o
LEFT JOIN
  target.customers c
ON
  o.customer_id = c.customer_id
GROUP BY
  customer_state
ORDER BY
  AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day)) DESC
LIMIT 5
```



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

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

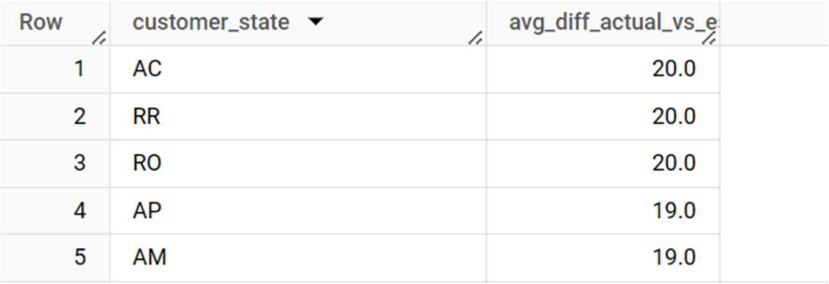
```
SELECT
 customer_state,
 ROUND(AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day)),2) AS
delivery_time,
FROM
 target.orders AS o
LEFT JOIN
 target.customers c
ON
 o.customer_id = c.customer_id
GROUP BY
 customer_state
ORDER BY
 AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day))
LIMIT 5
```



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

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

```
WITH
 cte1 AS(
 SELECT
   c.customer_state,
   SUM(DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp,
day))/COUNT(DISTINCT o.order_id) AS average_estimated_delivery,
   SUM(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,
day))/COUNT(DISTINCT o.order_id) AS average_delivery_days
 FROM
   target.orders AS o
 LEFT JOIN
   target.customers c
 ON
   o.customer_id = c.customer_id
 GROUP BY
   c.customer_state)
SELECT
 customer_state,
 ROUND(average_estimated_delivery - average_delivery_days, 0) AS avg_diff_actual_vs_estimated
FROM
 cte1
WHERE
 average_estimated_delivery > average_delivery_days
ORDER BY
 average_estimated_delivery - average_delivery_days desc
LIMIT 5
```

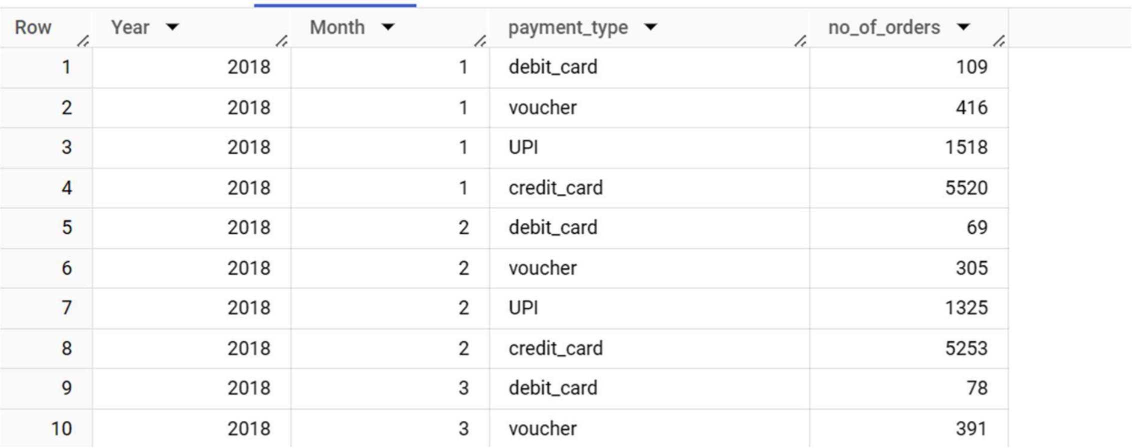


Row	customer_state	avg_diff_actual_vs_e
1	AC	20.0
2	RR	20.0
3	RO	20.0
4	AP	19.0
5	AM	19.0

6. Analysis based on the payments:

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

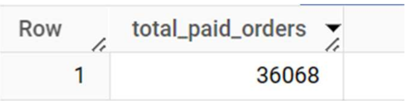
```
SELECT
 *
FROM (
 SELECT
  EXTRACT(year
  FROM
   order_purchase_timestamp) AS Year,
  EXTRACT(month FROM order_purchase_timestamp) AS Month,
  payment_type,
  COUNT(o.order_id) no_of_orders
 FROM
  target.orders o
 LEFT JOIN
  target.payments p
 ON
  o.order_id = p.order_id
 GROUP BY
  EXTRACT(month
  FROM
   order_purchase_timestamp),
  EXTRACT(year
  FROM
   order_purchase_timestamp),
  payment_type) X
ORDER BY  year DESC, Month, No_of_orders
```



Row	Year	Month	payment_type	no_of_orders
1	2018	1	debit_card	109
2	2018	1	voucher	416
3	2018	1	UPI	1518
4	2018	1	credit_card	5520
5	2018	2	debit_card	69
6	2018	2	voucher	305
7	2018	2	UPI	1325
8	2018	2	credit_card	5253
9	2018	3	debit_card	78
10	2018	3	voucher	391

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

```
WITH
 cte1 AS (
 SELECT
  o.order_id,
  payment_sequential,
  payment_installments,
  payment_value,
  order_status,
  (price + freight_value) AS total_to_be_paid
 FROM
  target.payments p
 LEFT JOIN
  target.orders o
 ON
  o.order_id = p.order_id
 LEFT JOIN
  target.order_items oi
 ON
  oi.order_id = p.order_id
 WHERE
  payment_installments > 1)
SELECT
 COUNT(DISTINCT order_id) AS total_paid_orders
FROM
 cte1
WHERE
 payment_value = cte1.total_to_be_paid
```



Row	total_paid_orders
1	36068

7. Actionable Insights & Recommendations:

Growing Trend in Orders Placed:
There's a consistent increase in the number of orders placed each year from 2016 to 2018, indicating growing market demand. To capitalize on this trend, Target should expand product offerings, improve inventory management, and enhance logistical capabilities to handle higher order volumes efficiently.

Monthly Seasonality in Orders:
Monthly order volumes show seasonal peaks, particularly in May, June, July, and August. To leverage these seasonal trends, Target should plan targeted marketing campaigns, promotions, and inventory stocking strategies aligned with seasonal demand patterns to maximize sales during peak months.

Time of Day for Orders:
Most orders are placed during the afternoon, followed by night and morning. To capitalize on peak order times, Target should implement flash sales or promotional activities during dawn to boost order numbers during this period.

Evolution of E-commerce Orders by State:
There's a variation in the number of orders placed across different states. To maximize market penetration and customer satisfaction, Target should tailor marketing strategies, product offerings, and logistics based on regional preferences and order patterns.

Impact on Economy - Money Movement Analysis:
The cost of orders increased significantly by 236% from 2017 to 2018. To maintain profitability while meeting customer expectations and market trends, Target should continuously monitor pricing strategies, freight costs, and payment options.

Total & Average Value of Orders and Freight by State:
There are variations in the average order value and freight costs across different states. Target should analyze high-value states to identify opportunities for upselling, cross-selling, and optimizing freight logistics for cost-effectiveness.

Analysis Based on Sales, Freight, and Delivery Time:
Delivery times vary across states, impacting customer satisfaction and operational efficiency. Target should focus on improving delivery logistics, reducing delivery times, and providing accurate estimated delivery dates to enhance customer experience and loyalty.

Payments and Instalments Analysis:
Customers opt for various payment types and installment plans, affecting order placements and cash flow. Target should offer flexible payment options, streamline installment processes, and monitor payment trends to facilitate smooth transactions and increase order conversions.