SQL PROJECT

TARGET: BUSINESS CASE

What is Target?

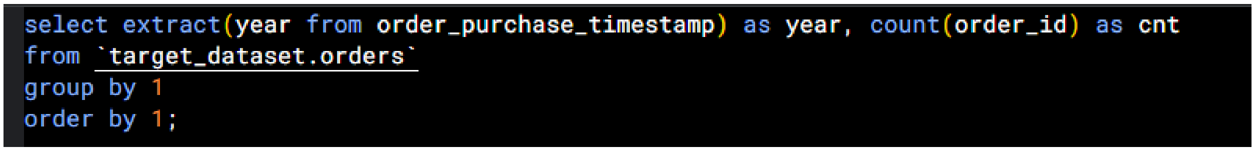
Target Corporation, or simply **Target**, is an American retail corporation that operates a chain of discount department stores and hypermarkets, headquartered in Minneapolis, Minnesota. It is the seventh-largest retailer in the United States, and a component of the S&P 500 Index. The company is one of the largest American-owned Private Employers in the United States.

This particular business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.

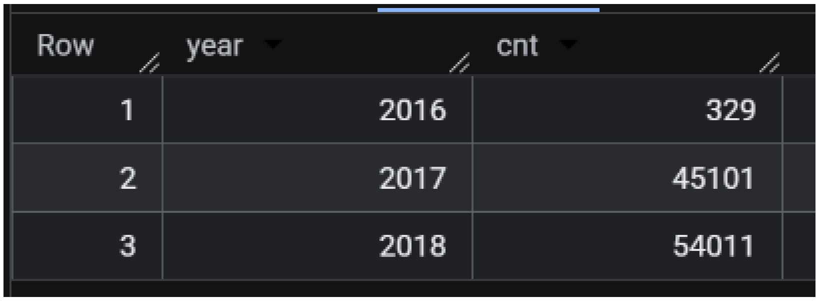
By analyzing this extensive dataset, it becomes possible to gain valuable insights into Target's operations in Brazil. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, etc.

Actionable Insights:

Q1. Is there a growing trend in the no. of orders placed over the past years?

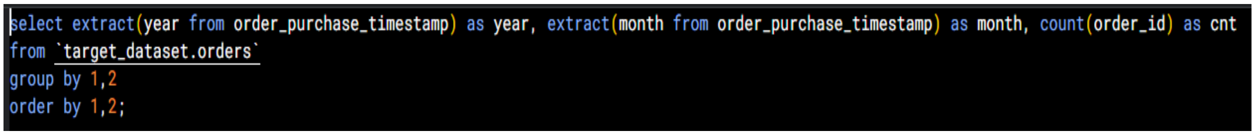


```
select extract(year from order_purchase_timestamp) as year, count(order_id) as cnt
from `target_dataset.orders`
group by 1
order by 1;
```

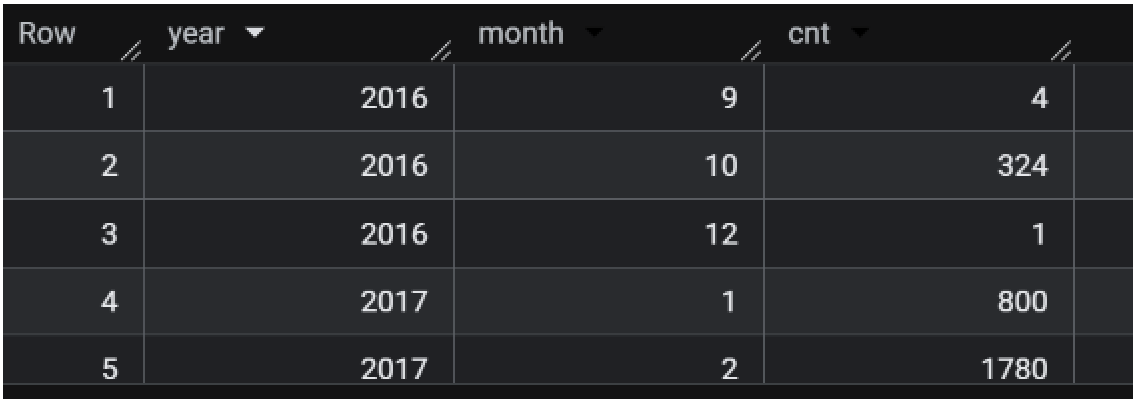


Row	year	cnt
1	2016	329
2	2017	45101
3	2018	54011

Q2. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

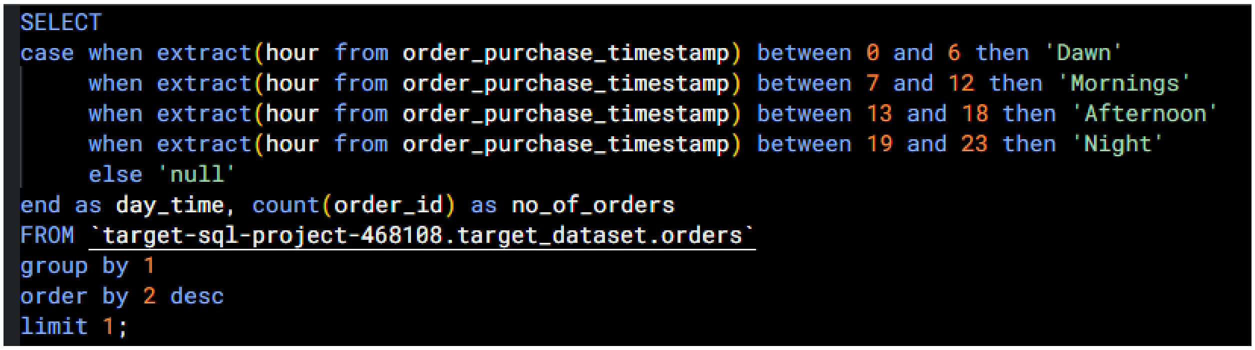


```
select extract(year from order_purchase_timestamp) as year, extract(month from order_purchase_timestamp) as month, count(order_id) as cnt
from `target_dataset.orders`
group by 1,2
order by 1,2;
```

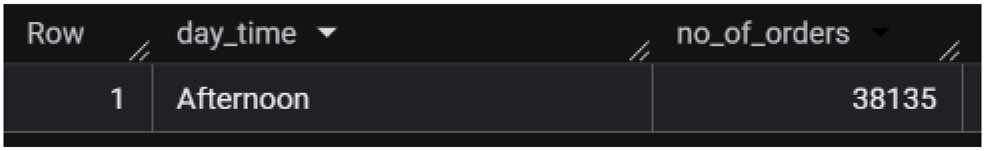


Row	year	month	cnt
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780

Q3. During what time of the day, do the Brazilian customers mostly place their orders?



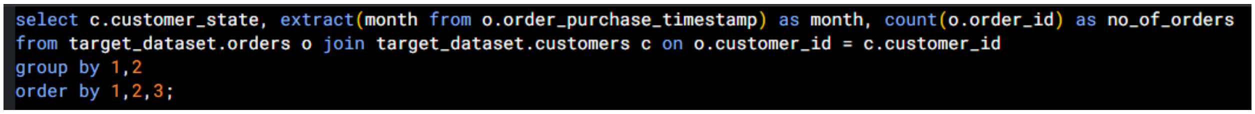
```
SELECT
case when extract(hour from order_purchase_timestamp) between 0 and 6 then 'Dawn'
     when extract(hour from order_purchase_timestamp) between 7 and 12 then 'Mornings'
     when extract(hour from order_purchase_timestamp) between 13 and 18 then 'Afternoon'
     when extract(hour from order_purchase_timestamp) between 19 and 23 then 'Night'
     else 'null'
end as day_time, count(order_id) as no_of_orders
FROM `target-sql-project-468108.target_dataset.orders`
group by 1
order by 2 desc
limit 1;
```



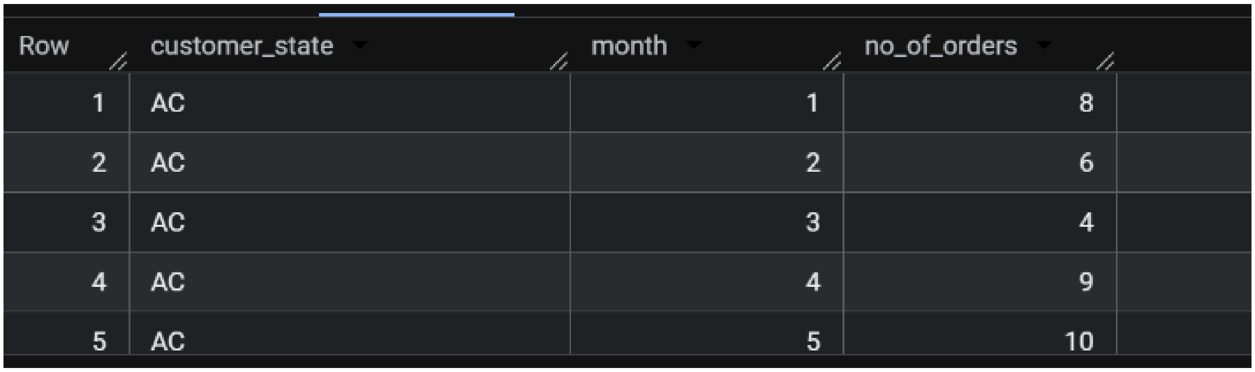
Row	day_time	no_of_orders
1	Afternoon	38135

<u>Insight</u>: As Brazilian customers order mostly during afternoon, they should get more delivery partners ready during afternoon time which will eventually reduce time to deliver the items and increase customer satisfaction.

Q4. Month on month no. of orders placed in each state.

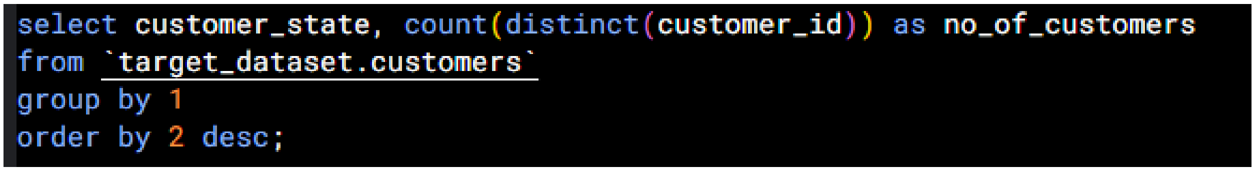


```
select c.customer_state, extract(month from o.order_purchase_timestamp) as month, count(o.order_id) as no_of_orders
from target_dataset.orders o join target_dataset.customers c on o.customer_id = c.customer_id
group by 1,2
order by 1,2,3;
```

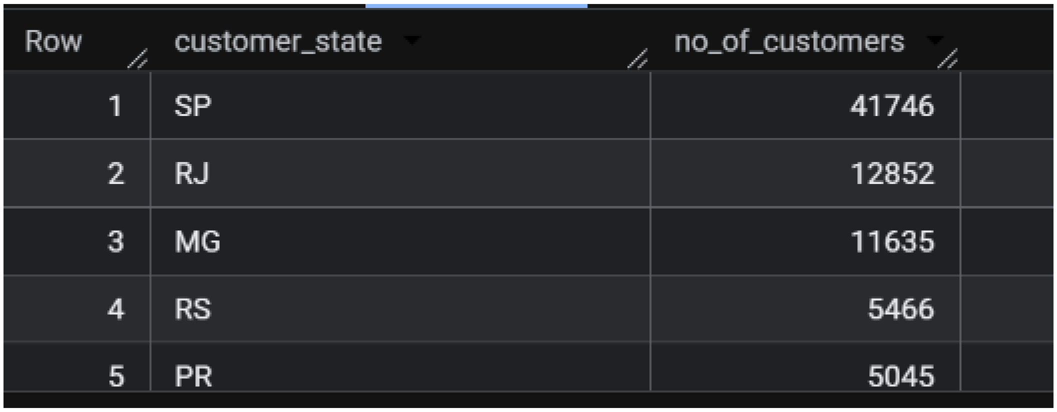


Row	customer_state	month	no_of_orders
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10

Q5. How are the customers distributed across all the states?

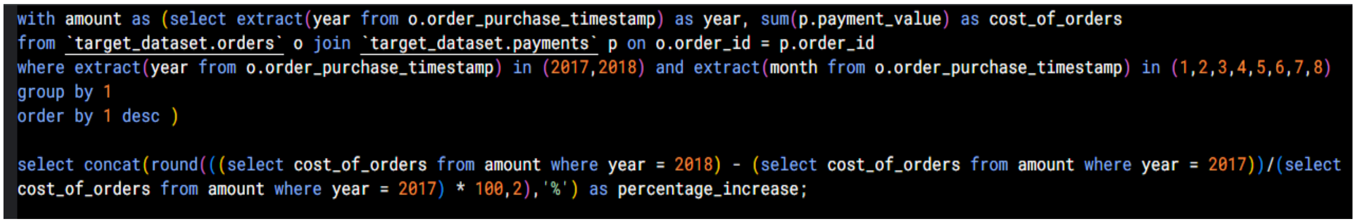


```
select customer_state, count(distinct(customer_id)) as no_of_customers
from `target_dataset.customers`
group by 1
order by 2 desc;
```

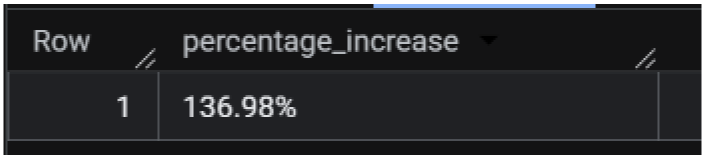


Row	customer_state	no_of_customers
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045

Q6. Get the % increase in the cost of orders from year 2017 to 2018 (months between Jan to Aug only).

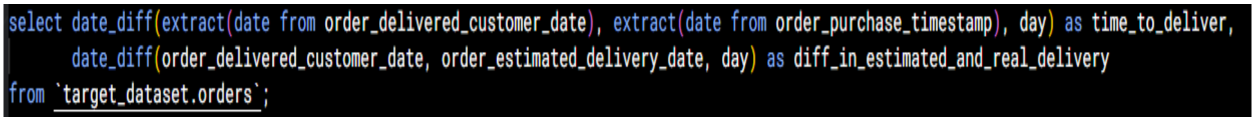


```
with amount as (select extract(year from o.order_purchase_timestamp) as year, sum(p.payment_value) as cost_of_orders
from `target_dataset.orders` o join `target_dataset.payments` p on o.order_id = p.order_id
where extract(year from o.order_purchase_timestamp) in (2017,2018) and extract(month from o.order_purchase_timestamp) in (1,2,3,4,5,6,7,8)
group by 1
order by 1 desc )

select concat(round(((select cost_of_orders from amount where year = 2018) - (select cost_of_orders from amount where year = 2017))/(select
cost_of_orders from amount where year = 2017) * 100,2),'%') as percentage_increase;
```

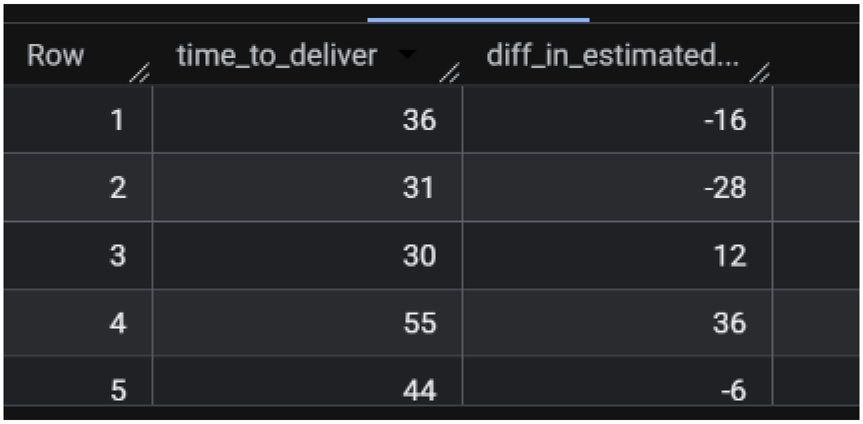


Row	percentage_increase
1	136.98%

Q7. 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.

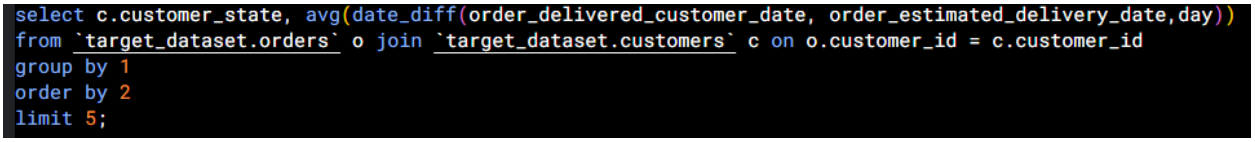


```
select date_diff(extract(date from order_delivered_customer_date), extract(date from order_purchase_timestamp), day) as time_to_deliver,
       date_diff(order_delivered_customer_date, order_estimated_delivery_date, day) as diff_in_estimated_and_real_delivery
from `target_dataset.orders`;
```

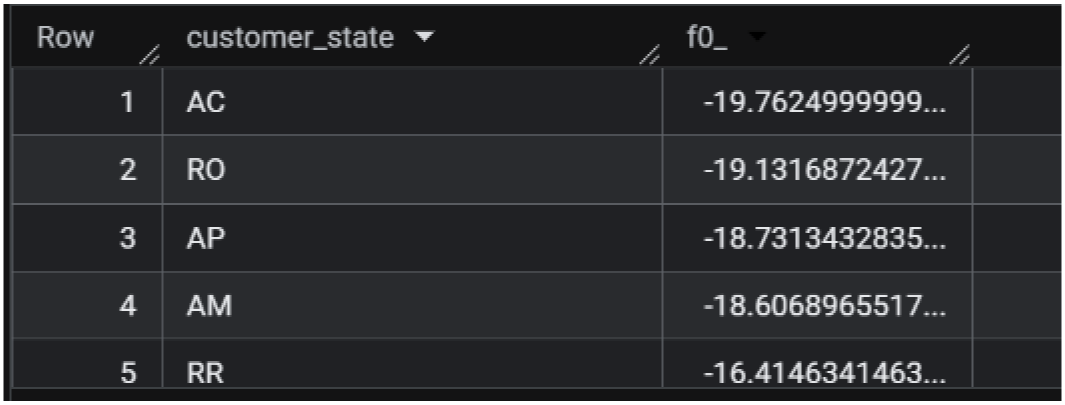


Row	time_to_deliver	diff_in_estimated...
1	36	-16
2	31	-28
3	30	12
4	55	36
5	44	-6

Q8. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

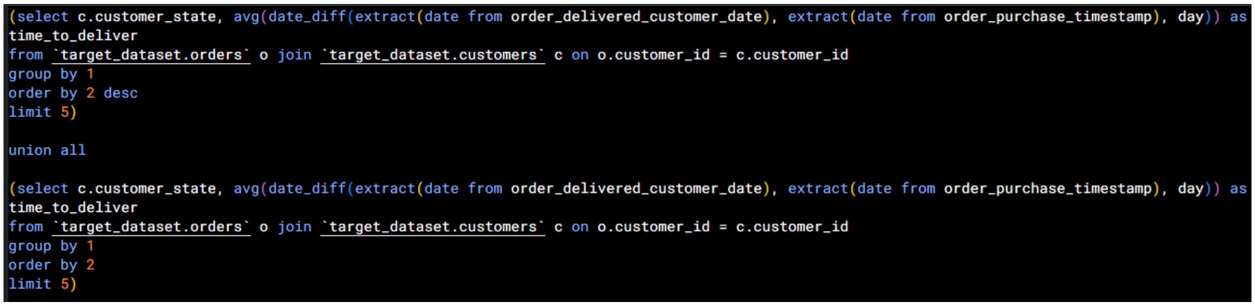


```
select c.customer_state, avg(date_diff(order_delivered_customer_date, order_estimated_delivery_date,day))
from `target_dataset.orders` o join `target_dataset.customers` c on o.customer_id = c.customer_id
group by 1
order by 2
limit 5;
```

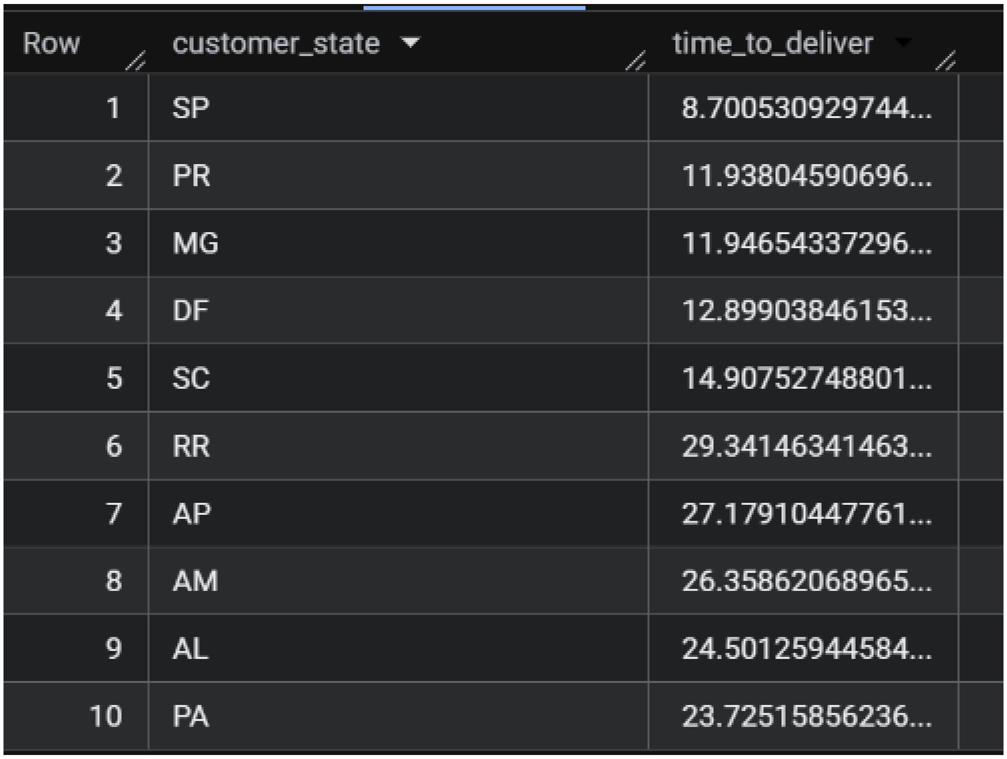


Row	customer_state	f0_
1	AC	-19.7624999999...
2	RO	-19.1316872427...
3	AP	-18.7313432835...
4	AM	-18.6068965517...
5	RR	-16.4146341463...

Q9. Find out the top 5 states with the highest & lowest average delivery time.

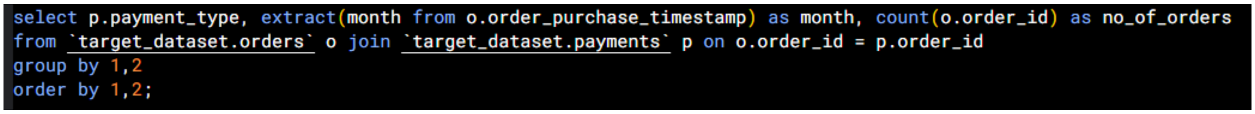


```
(select c.customer_state, avg(date_diff(extract(date from order_delivered_customer_date), extract(date from order_purchase_timestamp), day)) as
time_to_deliver
from `target_dataset.orders` o join `target_dataset.customers` c on o.customer_id = c.customer_id
group by 1
order by 2 desc
limit 5)

union all

(select c.customer_state, avg(date_diff(extract(date from order_delivered_customer_date), extract(date from order_purchase_timestamp), day)) as
time_to_deliver
from `target_dataset.orders` o join `target_dataset.customers` c on o.customer_id = c.customer_id
group by 1
order by 2
limit 5)
```

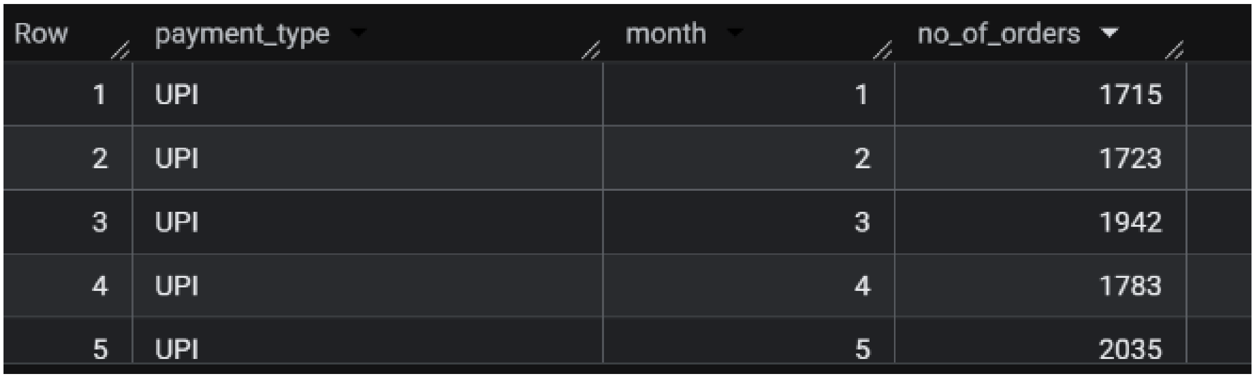


Row	customer_state	time_to_deliver
1	SP	8.700530929744...
2	PR	11.93804590696...
3	MG	11.94654337296...
4	DF	12.89903846153...
5	SC	14.90752748801...
6	RR	29.34146341463...
7	AP	27.17910447761...
8	AM	26.35862068965...
9	AL	24.50125944584...
10	PA	23.72515856236...

Q10. 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(o.order_id) as no_of_orders
from `target_dataset.orders` o join `target_dataset.payments` p on o.order_id = p.order_id
group by 1,2
order by 1,2;
```



Row	payment_type	month	no_of_orders
1	UPI	1	1715
2	UPI	2	1723
3	UPI	3	1942
4	UPI	4	1783
5	UPI	5	2035

Insight: There’s a gradual increase in payment type of UPI & Credit card till month 7-8 and then sudden decrease.