Target SQL Business Case

Q-1 A. Data type of all columns in the “customers” table.

Ans -

```
SELECT
    column_name,
    data_type
FROM
    `target-shop1.target.INFORMATION_SCHEMA.COLUMNS`
WHERE
    table_name = 'customers';
```

Screenshot -
Query Result

Insights - All columns have appropriate data types (likely VARCHAR for IDs and city/state, INT for zip codes)

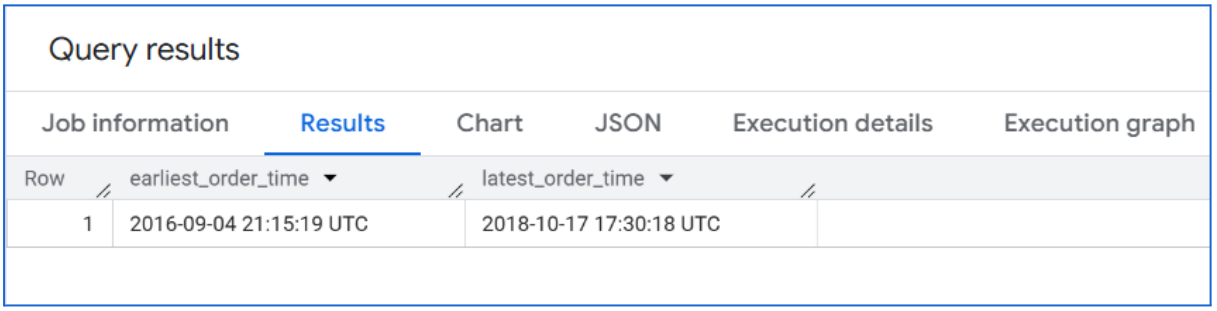
Recommendation - N/A

Q-1 B. Get the time range between which the orders were placed.

Ans -

```
SELECT
    MIN(order_purchase_timestamp) AS earliest_order_time,
    MAX(order_purchase_timestamp) AS latest_order_time
FROM
   `target-shop1.target.orders`;
```

Screenshot -
Query Result



Insights - Orders were placed between Sept 2016 and Oct 2018 with timezone UTC

Recommendation - N/A

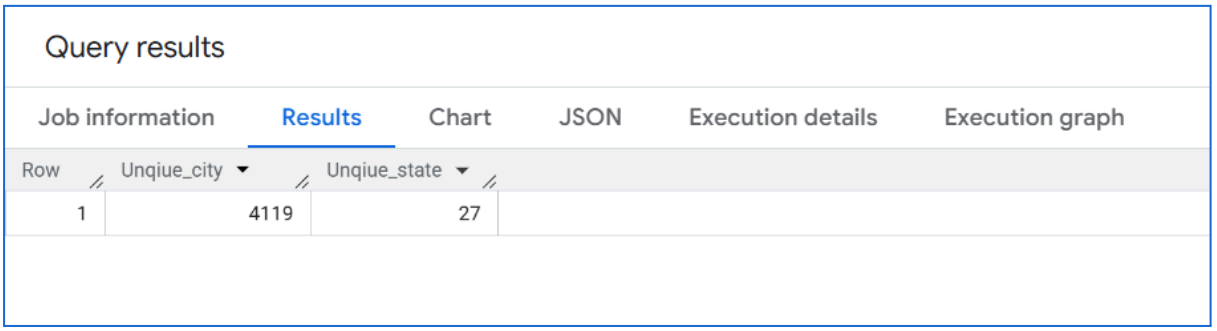
Q-1 C. Count the Cities & States of customers who ordered during the given period.

Ans -

```
select count(distinct c.customer_city ) as unique_city,
count( distinct c.customer_state) as unique_state
From `target.customers` c
JOIN `target.orders` o ON c.customer_id = o.customer_id
WHERE
order_purchase_timestamp Between '2016-09-04' AND '2018-10-17'
```

Screenshot -

Query Result



Insights - Unique City is 4119 so it has wide delivery in every city Long network and strong Presence in 27 States.

Recommendation - Brazil have approx 5500 (in Real) city we also Need To Focus On Remaining City and Continue focusing on nationwide delivery so Our 4119 city customers don't Lose.

II. In-depth Exploration:

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

Ans -
```
select Extract( year from order_purchase_timestamp ) as order_year,
FORMAT_TIMESTAMP ( '%B' , order_purchase_timestamp ) as order_month,
count(order_id) as total_orders_id
 From `target-shop1.target.orders`
Group By 1,2
Order by 2, 1 desc;
```

Screenshot -
Query Result

Insights - There is a consistent increase in monthly orders over the years 2018, 2017 is Very Good Year For us. We get a high Number of Order in Both Year (By Arranging Total_order DESC) increasing order trend suggests strong growth.

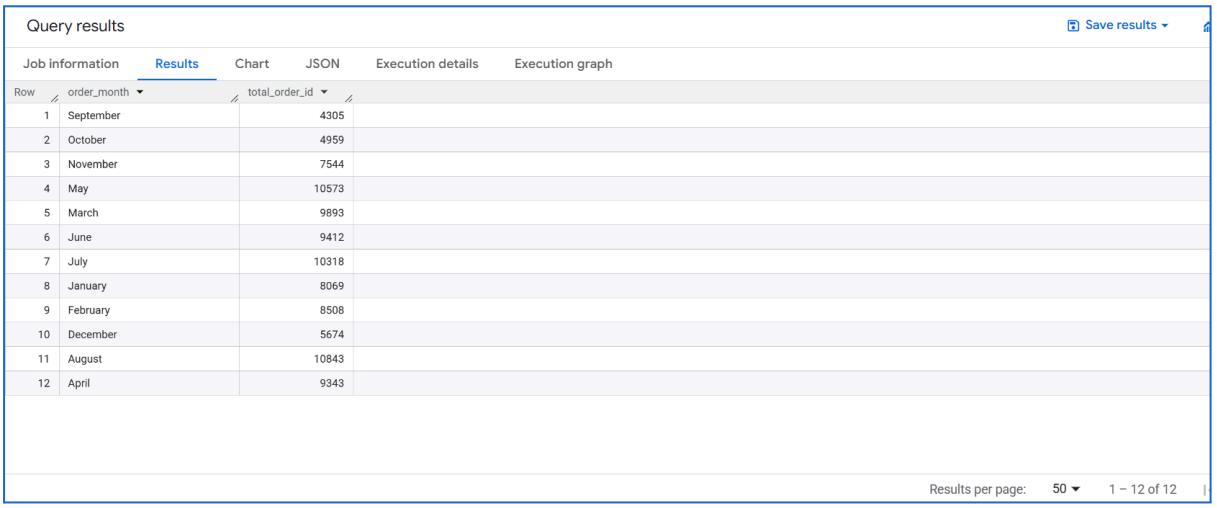
Recommendation - Invest Some In marketing so we can Continue this Growth and launch some Program To retain our Loyal customers

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

Ans -

```
select
FORMAT_TIMESTAMP ( '%B' , order_purchase_timestamp ) as Order_month,
count(order_id) as Total_orders
 From `target-shop1.target.orders`
 Group By Order_month
 Order By Order_month desc;
```

Screenshot -
Query Result



Row	order_month	total_order_id
1	September	4305
2	October	4959
3	November	7544
4	May	10573
5	March	9893
6	June	9412
7	July	10318
8	January	8069
9	February	8508
10	December	5674
11	August	10843
12	April	9343

Insights - in Monsoon Season Order are increased and less order in Autumn Season peak order was (july-aug) and less order was (oct - dec)

Recommendation - launch some Mid year promotion to boost Our sales and In Winter do Christmas sales and new year sales. For Continued sales in peak Months use Discounts by SMS or mail Reminders.

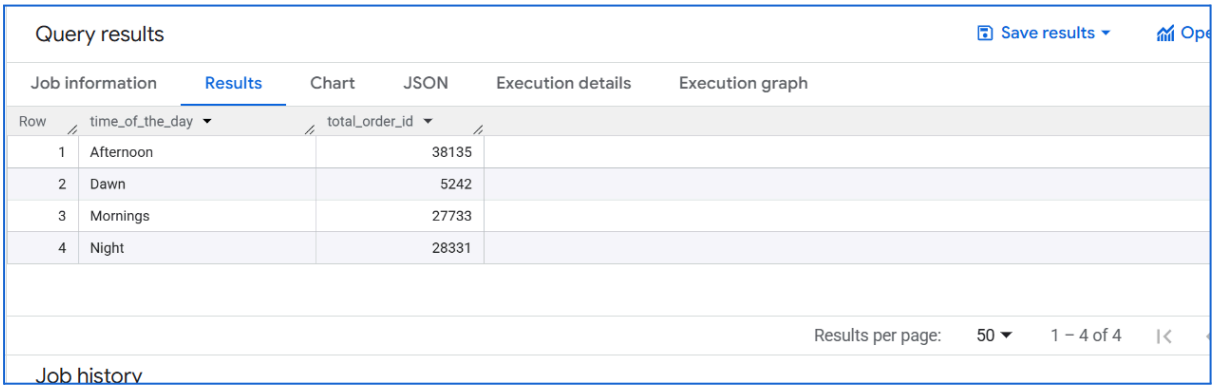
C. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

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

Ans -

```
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'
else 'Night'
end as time_of_the_day,
Count(order_id) as total_order_id
 from target-shop1.target.orders
group by 1
order by 1;
```

Screenshot -
Query Result



Insights - Most orders are placed during the Afternoon and Fewest orders are placed during Dawn hours.

Recommendation - Send email offers or push notifications in the Dawn. and Run Offers in morning and dawn

III. Evolution of E-commerce orders in the Brazil region:
A. Get the month on month no. of orders placed in each state.

Ans -

```
select Extract( year from o.order_purchase_timestamp ) as Order_Year,
FORMAT_TIMESTAMP ( '%B' , o.order_purchase_timestamp ) as Order_month,
c.customer_state,
count(o.order_id) as Total_orders
 From `target-shop1.target.orders` o
 JOIN `target.customers` c ON o.customer_id = c.customer_id
Group by Order_Year,Order_month,c.customer_state
order by Order_Year,Order_month,c.customer_state
```

Screenshot -

Query Result

Query results — Save results

Job information | Results | Chart | JSON | Execution details | Execution graph

Row	Order_Year	Order_month	customer_state	Total_orders
1	2016	December	PR	1
2	2016	October	AL	2
3	2016	October	BA	4
4	2016	October	CE	8
5	2016	October	DF	6
6	2016	October	ES	4
7	2016	October	GO	9
8	2016	October	MA	4

Results per page: 50 — 1 – 50 of 565

Insights - São Paulo (SP) , Minas Gerais (MG) ,Rio de Janeiro (RJ) have large orders because these cities have a large population of Brazil and some states have consistently low order volumes.

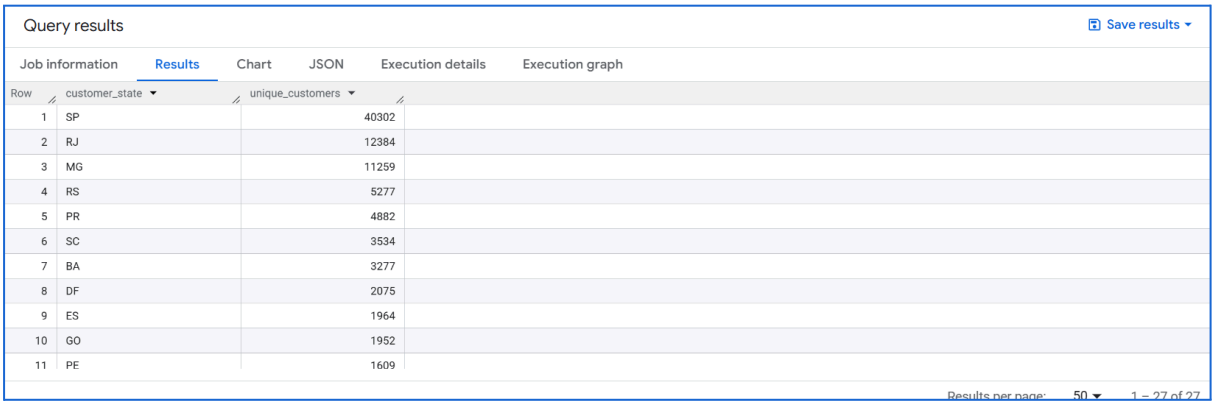
Recommendation - Make speed delivery in Top 5 City Like 10 Minutes delivery and Open WareHouse . Use ads in low order city or Investigate there any Delivery issues in these states

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

Ans-

```
SELECT
    customer_state,
    COUNT(DISTINCT customer_unique_id) AS unique_customers
FROM
    `target.customers`
GROUP BY
    customer_state
ORDER BY
    unique_customers DESC;
```

Screenshot -
Query Result



Query results — Save results

Job information | Results | Chart | JSON | Execution details | Execution graph

Row	customer_state	unique_customers
1	SP	40302
2	RJ	12384
3	MG	11259
4	RS	5277
5	PR	4882
6	SC	3534
7	BA	3277
8	DF	2075
9	ES	1964
10	GO	1952
11	PE	1609

Results per page: 50 — 1 – 27 of 27

Insights - SP, RJ,MG These have large population in Brazil Increase scalability in these city make large promotion in These state because of population Focus on faster shipping

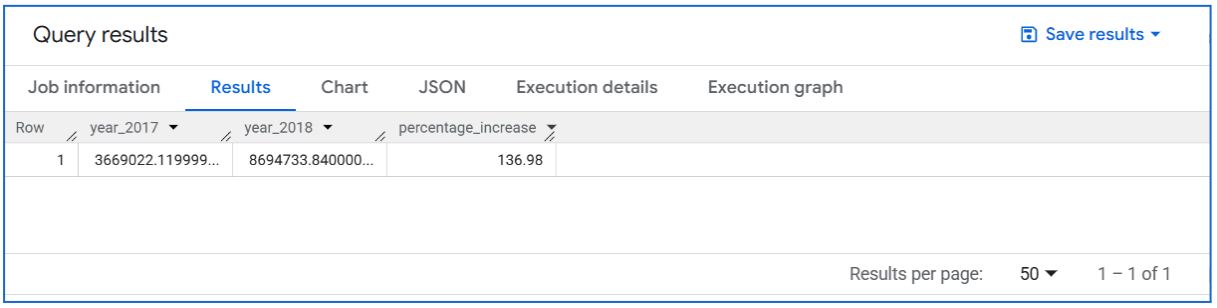
Recommendation - Run Discounts Coupons and Promotions in Low order Cities and Ground Promotions in low Orders

IV. 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).

Ans -
```
with year_total as
(
select  extract (year from o.order_purchase_timestamp) as order_year,
sum(p.payment_value) as  total_payment

From target-shop1.target.orders o

JOIN target.payments 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 order_year
)

 select sum(case when order_year = 2017 then total_payment else 0 END )  as
year_2017,
   sum(case when order_year = 2018 then total_payment else 0 END )  as year_2018,
 ROUND(
        ((SUM(CASE WHEN order_year = 2018 THEN total_payment ELSE 0 END) -
          SUM(CASE WHEN order_year = 2017 THEN total_payment ELSE 0 END)) /
          NULLIF(SUM(CASE WHEN order_year = 2017 THEN total_payment ELSE 0 END), 0)
* 100),
        2
    ) as percentage_increase
from
year_total;
```

Screenshot -

Query Result



Query results — Save results

Job information | Results | Chart | JSON | Execution details | Execution graph

Row	year_2017	year_2018	percentage_increase
1	3669022.119999...	8694733.840000...	136.98

Results per page: 50 — 1 – 1 of 1

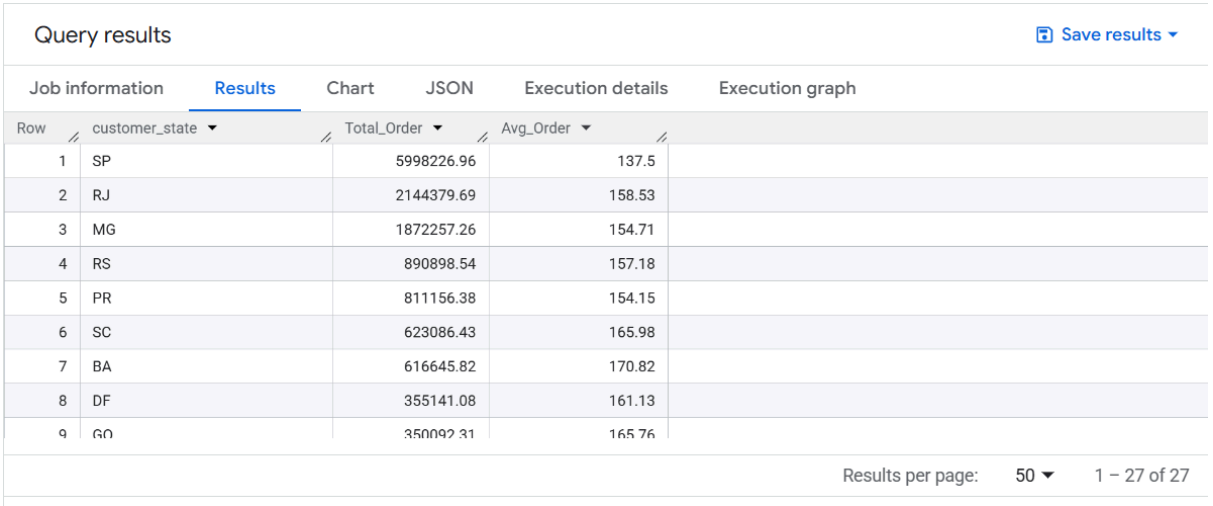
Insights - The total order value increased by 130% from 2018 to 2017.
Recommendation - Explore partnerships with payment gateways and logistics providers to optimize transaction and delivery costs.

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

Ans -

```
select c.customer_state ,
ROUND (sum(p.payment_value),2) as Total_Order,
ROUND (Avg(p.payment_value),2) as Avg_Order
 from `target.customers` c
 JOIN `target.orders` o ON c.customer_id = o.customer_id
 JOIN  `target.payments` p  ON o.order_id = p.order_id
 Group by c.customer_state
 order by Total_Order desc ;
```

Screenshot -
Query Result



Query results — Save results

Job information | Results | Chart | JSON | Execution details | Execution graph

Row	customer_state	Total_Order	Avg_Order
1	SP	5998226.96	137.5
2	RJ	2144379.69	158.53
3	MG	1872257.26	154.71
4	RS	890898.54	157.18
5	PR	811156.38	154.15
6	SC	623086.43	165.98
7	BA	616645.82	170.82
8	DF	355141.08	161.13
9	GO	350092.31	165.76

Results per page: 50 — 1 – 27 of 27

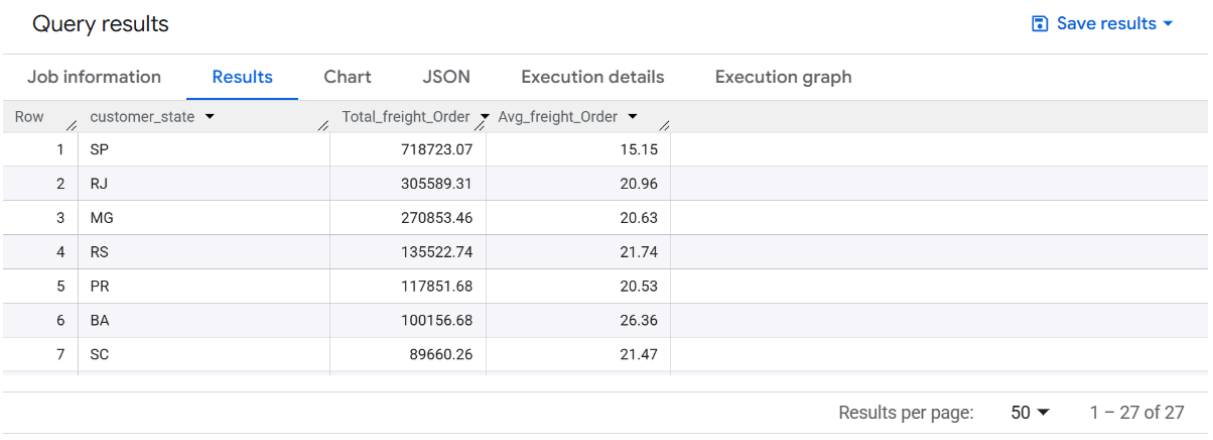
Insights - Average order value is slightly higher in SP compared to other states

Recommendation - Offer premium products and bundles in these states to maximize revenue.

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

Ans -
```
select c.customer_state ,
ROUND (sum(p.freight_value),2) as Total_freight_Order,
ROUND (Avg(p.freight_value),2) as Avg_freight_Order
 from target.customers c
 JOIN target.orders o ON c.customer_id = o.customer_id
 JOIN  target.order_items p  ON o.order_id = p.order_id
 Group by c.customer_state
 order by Total_freight_Order desc
```

Screenshot -
Query Result



Row	customer_state	Total_freight_Order	Avg_freight_Order
1	SP	718723.07	15.15
2	RJ	305589.31	20.96
3	MG	270853.46	20.63
4	RS	135522.74	21.74
5	PR	117851.68	20.53
6	BA	100156.68	26.36
7	SC	89660.26	21.47

Results per page: 50 1 – 27 of 27

Insights - States like São Paulo (SP) and Rio de Janeiro (RJ) have the highest total freight costs and some states have lower average freight per order**.**

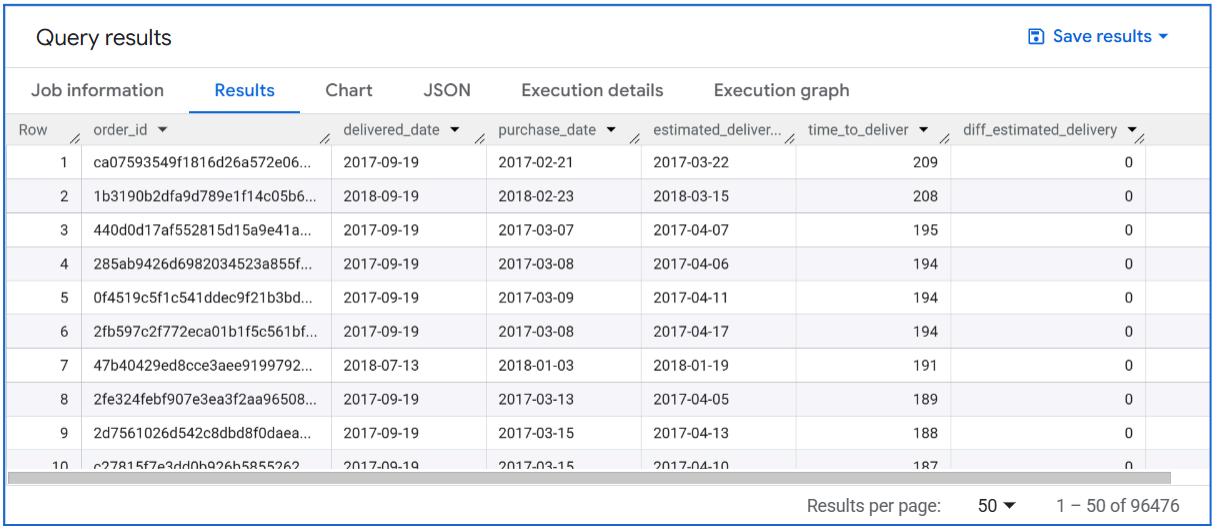
Recommendation - partner with local carries in SP and RJ so Ship with large speed and make high Stock in warehouse So no out of stock Product.

V. 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.
Do this in a single query.

Ans -
```
select order_id,
```

```
date(order_delivered_customer_date) as delivered_date,
date(order_purchase_timestamp) as purchase_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_estimated_delivery_date,order_estimated_delivery_date , day) as
diff_estimated_delivery

from target-shop1.target.orders
where order_delivered_customer_date IS NOT NULL
order by time_to_deliver desc , diff_estimated_delivery desc ;
```

Screenshot -
Query Result



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

Insights - In 2017 and few starting months in 2018 The order took a very long time (almost 7 months) to be delivered.

Recommendation - Review and Optimize Supply Chain Use faster shipping methods if possible and open small ware with Popular products in Large population in city and Collect feedback specifically about delivery times

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

Ans -
```
with Freight_avg as
(select c.customer_state ,
ROUND (Avg(p.freight_value),2) as Avg_freight_Order
 from target.customers c
 JOIN target.orders o ON c.customer_id = o.customer_id
 JOIN  target.order_items p  ON o.order_id = p.order_id
```

```
 Group by c.customer_state
 )

(select * From Freight_avg
order by Avg_freight_Order desc
limit 5 )

union all

(select * From Freight_avg
order by Avg_freight_Order
limit 5);
```

Screenshot -
Query Result

Query results

Job information | Results | Chart | JSON | Execution details | Execution graph

Row	customer_state	Avg_freight_Order
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15
6	SP	15.15
7	PR	20.53
8	MG	20.63
9	RJ	20.96
10	DF	21.04

Insights - States like RJ and DF have the lowest average freight costs. States like RR and PB have the highest freight costs the Gap Between Top To Bottom is (10-20)

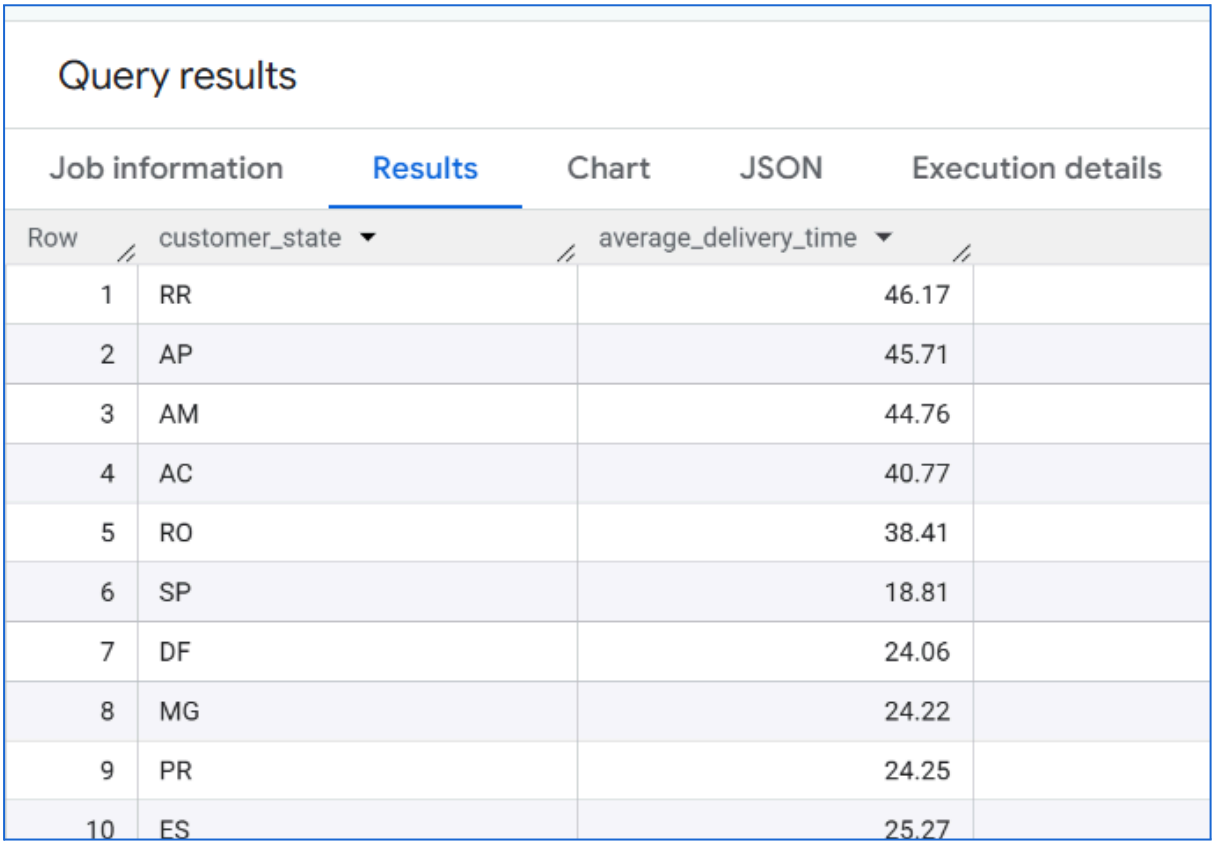
Recommendation - Consider these areas as low-cost shipping zones for potential discounts. We need to fix this Because RR have High Order and Large population also focus on delivery routes

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

Ans -
```
with Avg_delivery AS
(
  select  c.customer_state ,
```

```
ROund(avg(date_diff(o.order_estimated_delivery_date,o.order_purchase_timestamp ,
day)),2) as average_delivery_time
From `target-shop1.target.orders` o
JOIN `target-shop1.target.customers` c ON o.customer_id = c.customer_id
where o.order_estimated_delivery_date is not null
Group by c.customer_state
)
 (select *  From Avg_delivery
 order by average_delivery_time desc
 limit 5 )
 UNION ALL
 (select *  From Avg_delivery
 order by average_delivery_time
 limit 5 );
```

Screenshot -
Query Result



Query results

Job information | Results | Chart | JSON | Execution details

Row	customer_state	average_delivery_time
1	RR	46.17
2	AP	45.71
3	AM	44.76
4	AC	40.77
5	RO	38.41
6	SP	18.81
7	DF	24.06
8	MG	24.22
9	PR	24.25
10	ES	25.27

Insights - (AP) and (RR) have the fastest deliveries and (PR) and (ES) Slowest deliveries

Recommendation - Make a Base Hub or ware House in slowest deliveries and make deliveries routes arrange in the order For Delivery boys or hire local deliveries partners

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

Ans -

```
with delivery_gap as (

SELECT c.customer_state,
ROUND(max(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date
,day )),2) as avg_delivery_gap
From `target.orders` o
JOIN `target.customers` c ON o.customer_id = c.customer_id
where o.order_delivered_customer_date is not null
group by c.customer_state )

select * From delivery_gap
order by avg_delivery_gap desc
limit 5 ;
```

Screenshot -
Query Result

Row	customer_state	avg_delivery_gap
1	SP	146.0
2	MA	139.0
3	RS	134.0
4	RJ	108.0
5	ES	77.0

Insights - States like SP and MA consistently deliver much earlier than estimated. ES delivery gap drop up to (80)

Recommendation - Promote faster than expected deliveries and use SP or MA model regions to improve performance elsewhere.

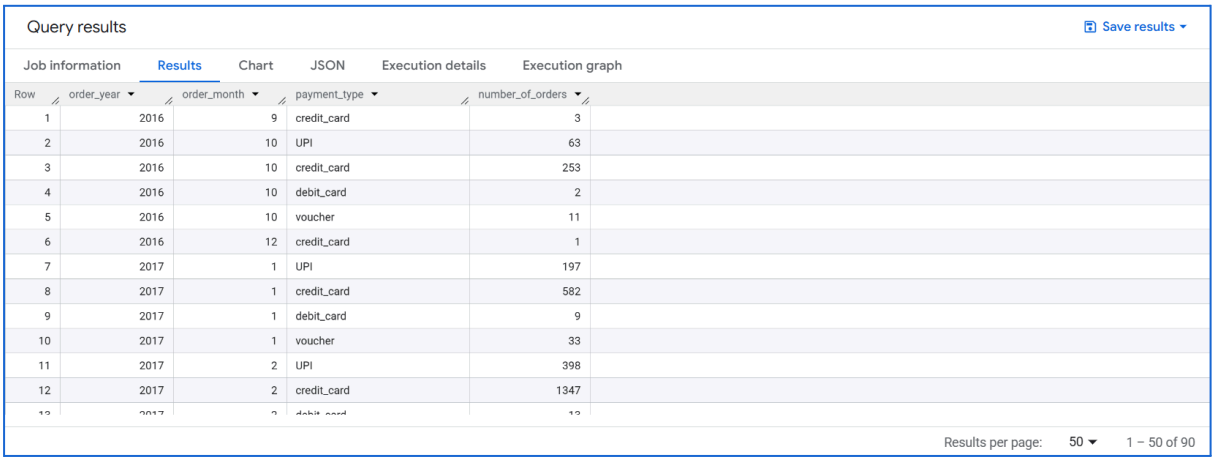
VI. Analysis based on the payments:

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

Ans -

```
select extract( year from o.order_purchase_timestamp) as order_year,
extract( month from o.order_purchase_timestamp) as order_month,
p.payment_type,
count(distinct o.order_id) as number_of_orders
from `target.orders` o
JOIN `target.payments` p ON o.order_id = p.order_id
GROUP BY  order_year,order_month,payment_type
ORDER BY
    order_year, order_month, p.payment_type;
```

Screenshot -
Query Result



Query results — Save results

Job information | Results | Chart | JSON | Execution details | Execution graph

Row	order_year	order_month	payment_type	number_of_orders
1	2016	9	credit_card	3
2	2016	10	UPI	63
3	2016	10	credit_card	253
4	2016	10	debit_card	2
5	2016	10	voucher	11
6	2016	12	credit_card	1
7	2017	1	UPI	197
8	2017	1	credit_card	582
9	2017	1	debit_card	9
10	2017	1	voucher	33
11	2017	2	UPI	398
12	2017	2	credit_card	1347

Results per page: 50 — 1 – 50 of 90

Insights - Credit cards dominate most months and 2 is UPI and 3 is Debit card. And Seasonal payment are shift in monsoon orders are high

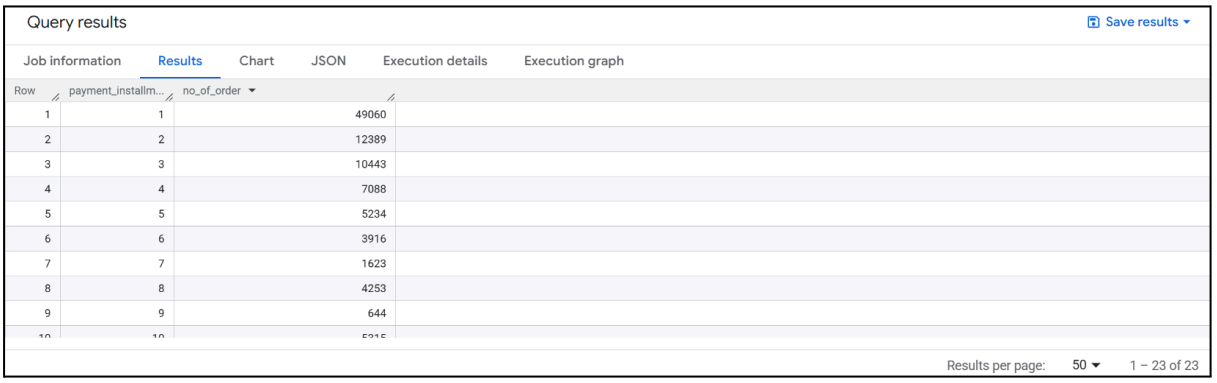
Recommendation - offer a discount on credit cards to maintain these. Do cashback offers and different Coupon codes On UPI and Debit Card for increased

B. Find the no. of orders placed on the basis of the payment installments that have been paid.

Ans -

```
select payment_installments ,
count ( distinct order_id ) as  no_of_order
from `target.payments`
where payment_installments > 0
Group by payment_installments
order by payment_installments ;
```

Screenshot -
Query Result



Insights - Most orders are placed with 1 installment. And other installment is Less Compared to 1

Recommendation - Consider promoting longer installment plans to encourage higher-value purchases and Offer incentives like discounts for one-time payments to maintain this trend.