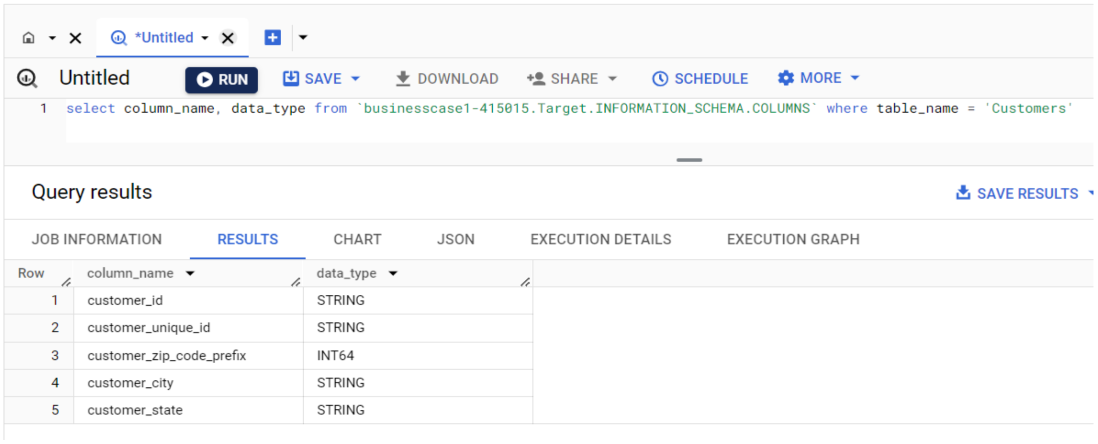
Target SQL Business Case

Q1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

1. **DataType of all columns in customer table**

Query : `select column_name, data_type from `businesscase1-415015.Target.INFORMATION_SCHEMA.COLUMNS` where table_name = 'Customers'`

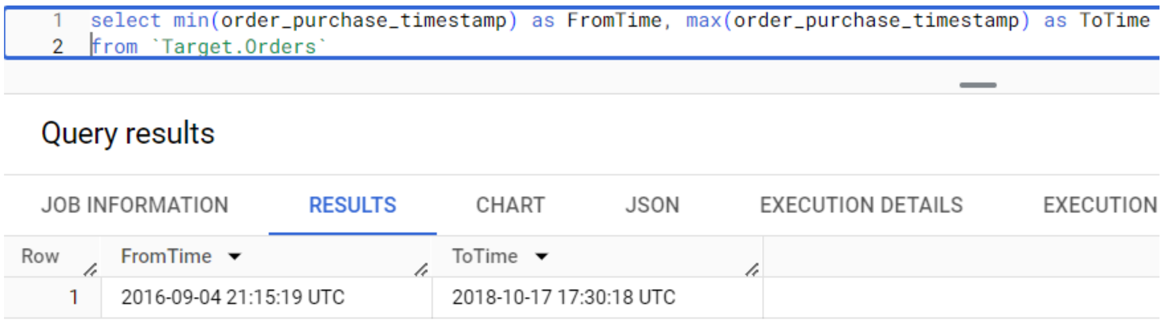


```
select column_name, data_type from `businesscase1-415015.Target.INFORMATION_SCHEMA.COLUMNS` where table_name = 'Customers'
```

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

2. **Get the time range between which the orders were placed.**

Query: `select min(order_purchase_timestamp) as FromTime, max(order_purchase_timestamp) as ToTime from `Target.Orders``

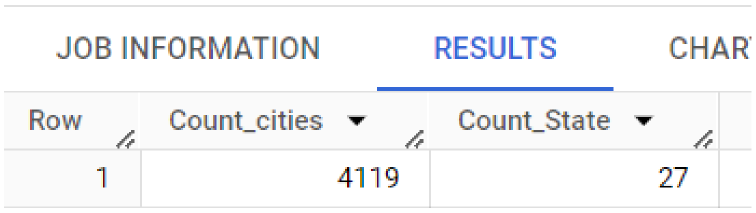


```
select min(order_purchase_timestamp) as FromTime, max(order_purchase_timestamp) as ToTime
from `Target.Orders`
```

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

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

Query:
```
select count(Distinct c.customer_city) as Count_cities, count(Distinct
c.customer_state) as Count_State  from `Target.Customers` c join `Target.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'
```



JOB INFORMATION | RESULTS | CHAR

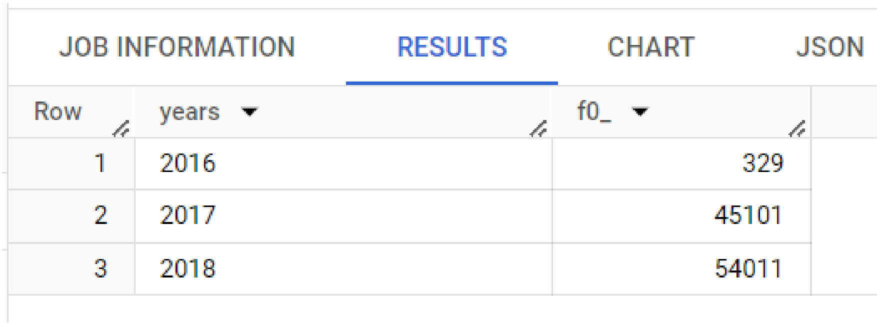
Row	Count_cities	Count_State
1	4119	27

Q2. In-depth Exploration:

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

Query:
```
select years, count(order_id) as Num_of_Orders from

(select Distinct(Format_Date('%Y',order_purchase_timestamp)) as Years, order_id
 from `Target.Orders` order by 1) a
group by a.years
order by a.Years
```



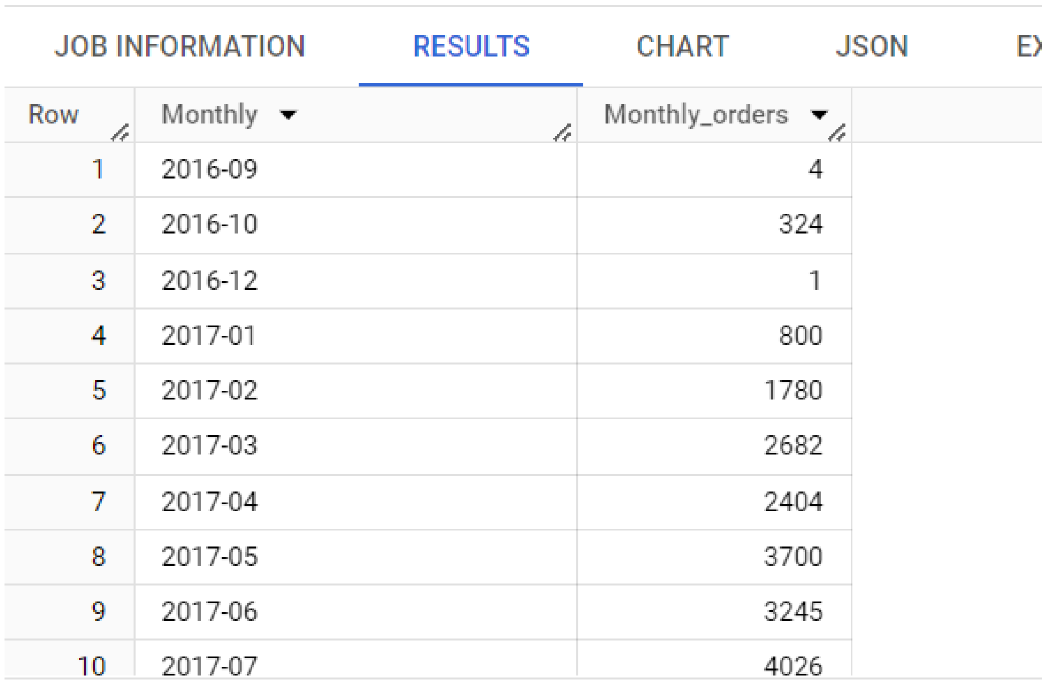
JOB INFORMATION | RESULTS | CHART | JSON

Row	years	f0_
1	2016	329
2	2017	45101
3	2018	54011

Insights: We see growing trend in the number of orders placed every year.
Huge growth in the sales between 2016 and 2017
Recommendations : To keep up the growing trend. Increase the speed of delivery and offers

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

Query:
```
select Monthly, count(order_id) as Monthly_orders from
(select Distinct(Format_Date('%Y-%m',order_purchase_timestamp)) as Monthly, order_id
 from `Target.Orders` order by 1) a
group by a.Monthly
order by a.Monthly
```



JOB INFORMATION | RESULTS | CHART | JSON | EX

Row	Monthly	Monthly_orders
1	2016-09	4
2	2016-10	324
3	2016-12	1
4	2017-01	800
5	2017-02	1780
6	2017-03	2682
7	2017-04	2404
8	2017-05	3700
9	2017-06	3245
10	2017-07	4026

Insights: Order Sales are less during September to December in 2016 and 2018
No sales in the month of November 2016

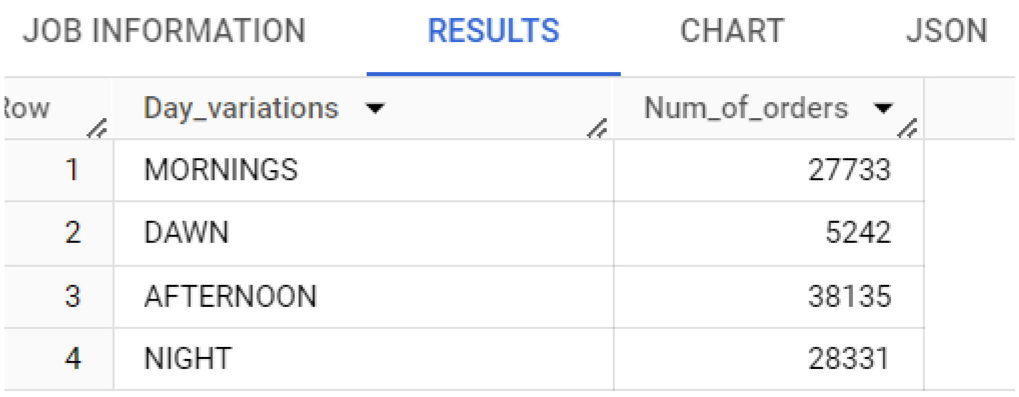
Recommendations : Give discounts or combo offers during the low or No sale period to increase the sales

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

Query:

```
with final as
(select parse_numeric(Format_Date('%H',order_purchase_timestamp)) as Hours from
`Target.Orders` order by order_purchase_timestamp),
dawn_mrng_aft_nht as
(select case when final.Hours between 0 and 6 then 'DAWN'
when final.Hours between 7 and 12 then 'MORNINGS'
when final.Hours between 13 and 18 then 'AFTERNOON'
when final.Hours between 19 and 24 then 'NIGHT' end as Day_variations
from final)

select *, count(Day_variations) as Num_of_orders from dawn_mrng_aft_nht
group by 1
```



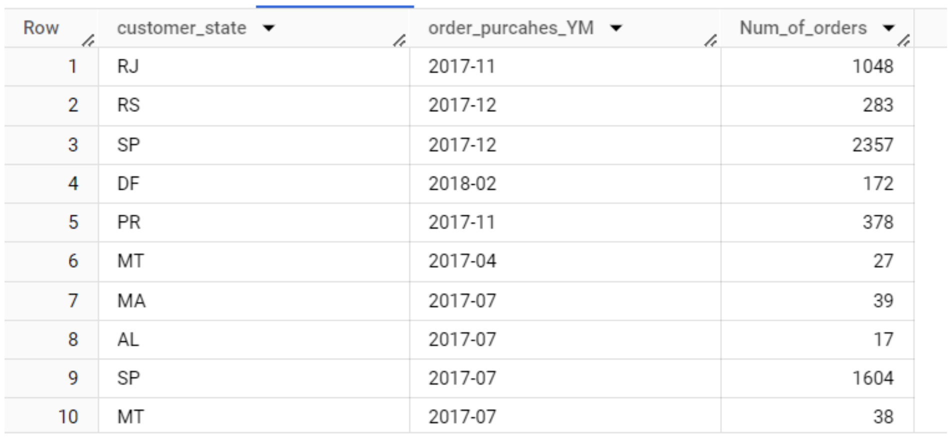
Row	Day_variations	Num_of_orders
1	MORNINGS	27733
2	DAWN	5242
3	AFTERNOON	38135
4	NIGHT	28331

Insights: Brazil customers mostly place more orders in Afternoon
Recommendation : Can have some exciting offers in the morning to increase the sales.

Q3. Evolution of E-commerce orders in the Brazil region:

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

Query:
```
select c.customer_state, Format_date('%Y-%m', o.order_purchase_timestamp) as
order_purcahes_YM , count(o.order_id) as Num_of_orders from `Target.Customers` c
join `Target.Orders` o on c.customer_id = o.customer_id
    group by c.customer_state, order_purcahes_YM
```

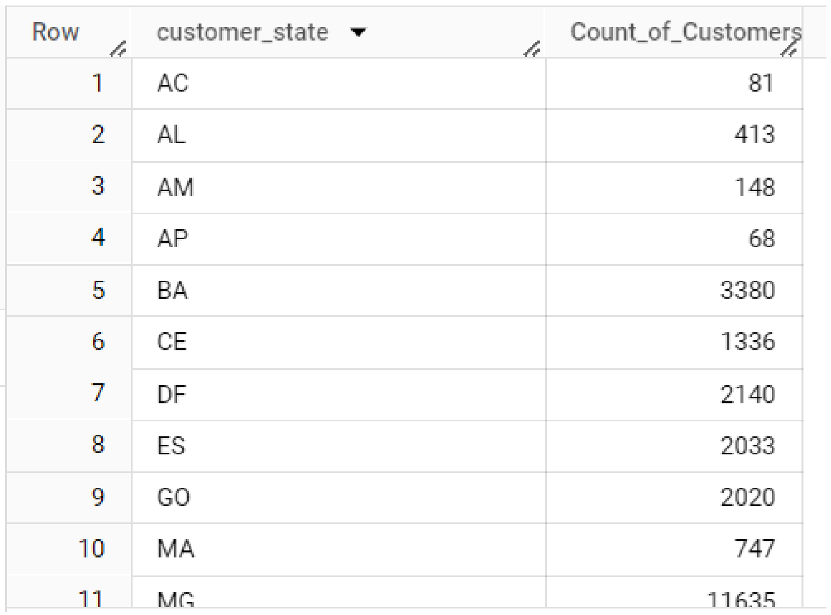


Row	customer_state	order_purcahes_YM	Num_of_orders
1	RJ	2017-11	1048
2	RS	2017-12	283
3	SP	2017-12	2357
4	DF	2018-02	172
5	PR	2017-11	378
6	MT	2017-04	27
7	MA	2017-07	39
8	AL	2017-07	17
9	SP	2017-07	1604
10	MT	2017-07	38

Insights: Few states like AL and MT have very less number of orders placed in the year 2017
Recommendations : Give more discounts or offers to increase the sales during the low sale period

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

Query:
```
select customer_state, count(customer_id) as Count_of_Customers from
`Target.Customers`
group by customer_state
order by 1
```



Row	customer_state	Count_of_Customers
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747
11	MG	11635

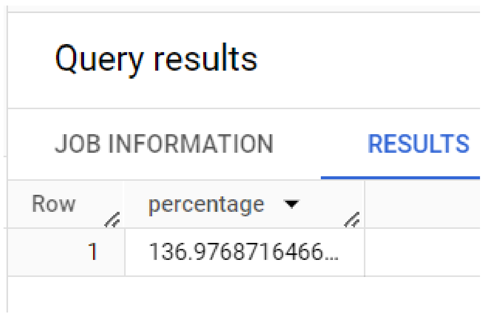
Insights: Customers are not evenly distributed across the state
SP, RJ and MG are top 3 states where there are more customers
Recommendations: RR AP and AC states have very low customers < 100. Increase the customer base by giving regional level offers and also free delivery

Q4. 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).

```
with pay_years as
(select extract(year from o.order_purchase_timestamp) as order_year,
extract(month from o.order_purchase_timestamp) as order_month,
p.payment_value
FROM `Target.Orders` o join `Target.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) between 1 and 8)

select (sum(case when order_year = 2018 then payment_value else 0 end) -
sum(case when order_year = 2017 then payment_value else 0 end))/sum(case when
order_year = 2017 then payment_value else 0 end) *100 as percentage from pay_years
```


Query results

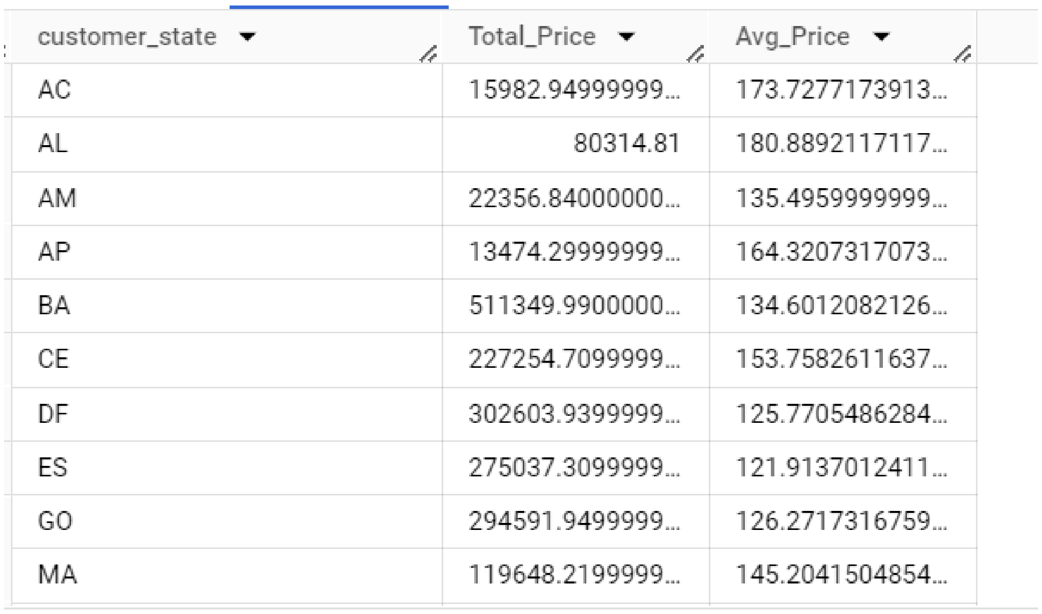
JOB INFORMATION RESULTS

Row	percentage
1	136.9768716466...

Insights: There is 136 percent increase in cost of orders from 2017 to 2018
Recommendations: Percentage increase can still be increased by increasing the number of orders

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

Query:
```
select c.customer_state , sum(item.price) as Total_Price, AVG (item.price) as
Avg_Price from `Target.Customers` c join `Target.Orders` o on c.customer_id =
o.customer_id
join `Target.Order_Items` item on o.order_id = item.order_id
group by c.customer_state
order by c.customer_state
```

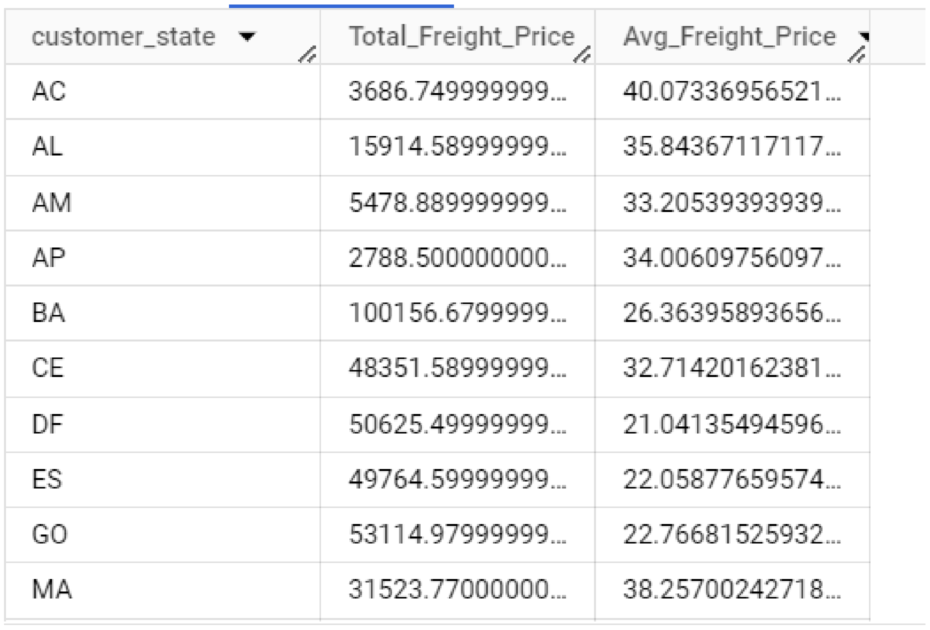


customer_state	Total_Price	Avg_Price
AC	15982.94999999...	173.7277173913...
AL	80314.81	180.8892117117...
AM	22356.84000000...	135.4959999999...
AP	13474.29999999...	164.3207317073...
BA	511349.9900000...	134.6012082126...
CE	227254.7099999...	153.7582611637...
DF	302603.9399999...	125.7705486284...
ES	275037.3099999...	121.9137012411...
GO	294591.9499999...	126.2717316759...
MA	119648.2199999...	145.2041504854...

Insights: Total Price is greater that average price

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

Query:
```
select c.customer_state , sum(item.freight_value) as Total_Freight_Price, AVG
(item.freight_value) as Avg_Freight_Price from `Target.Customers` c join
`Target.Orders` o on c.customer_id = o.customer_id
join `Target.Order_Items` item on o.order_id = item.order_id
group by c.customer_state
order by c.customer_state
```



customer_state	Total_Freight_Price	Avg_Freight_Price
AC	3686.749999999...	40.07336956521...
AL	15914.58999999...	35.84367117117...
AM	5478.889999999...	33.20539393939...
AP	2788.500000000...	34.00609756097...
BA	100156.6799999...	26.36395893656...
CE	48351.58999999...	32.71420162381...
DF	50625.49999999...	21.04135494596...
ES	49764.59999999...	22.05877659574...
GO	53114.97999999...	22.76681525932...
MA	31523.77000000...	38.25700242718...

Insights: Average Freight price is more for the states RR, PB, RO, AC > 40.

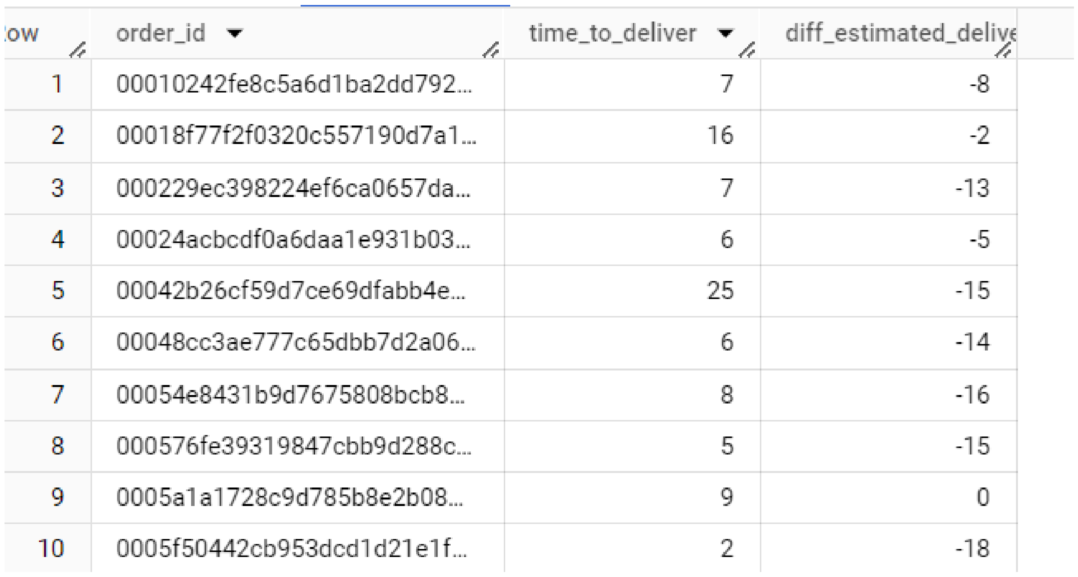
Recommendations: Decrease the Avg freight price for the states like RR, PB, RO, AC to increase the profit

Q5. 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.

Query:

Query:
```
select order_id,
 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` where order_delivered_customer_date is not null
order by order_id
```



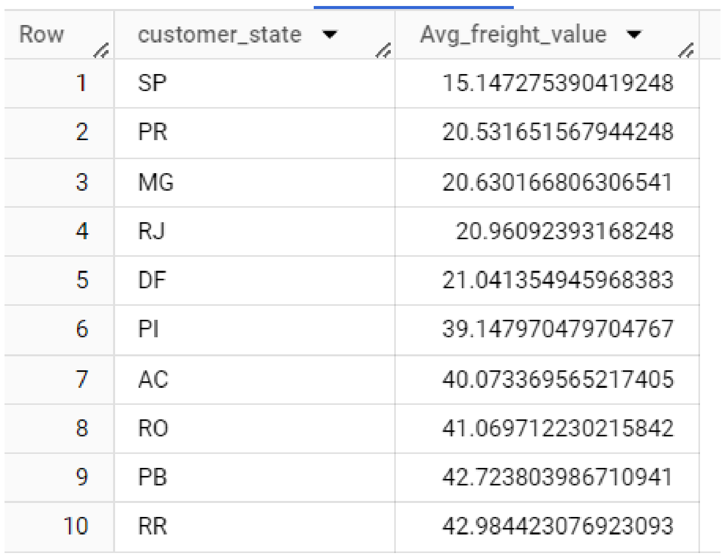
ow	order_id	time_to_deliver	diff_estimated_delive
1	00010242fe8c5a6d1ba2dd792...	7	-8
2	00018f77f2f0320c557190d7a1...	16	-2
3	000229ec398224ef6ca0657da...	7	-13
4	00024acbcdf0a6daa1e931b03...	6	-5
5	00042b26cf59d7ce69dfabb4e...	25	-15
6	00048cc3ae777c65dbb7d2a06...	6	-14
7	00054e8431b9d7675808bcb8...	8	-16
8	000576fe39319847cbb9d288c...	5	-15
9	0005a1a1728c9d785b8e2b08...	9	0
10	0005f50442cb953dcd1d21e1f...	2	-18

Insights: 2 orders took more than 200 days to deliver and more number of orders took more than 100 days to deliver. We also have orders which were delivered on same day or next day

Recommendations: Speed up the delivery time by hiring more delivery partners or by stocking up the inventory to reduce the delivery time.

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

Query:
```
with cte_Avg_freight_value as
(select c.customer_state,Avg(oitem.freight_value) as Avg_freight_value from
`Target.Customers` c join `Target.Orders` o on c.customer_id = o.customer_id
 join `Target.Order_Items` oitem on o.order_id = oitem.order_id
 group by c.customer_state
)
(select * from cte_Avg_freight_value order by Avg_freight_value desc limit 5)
union all
(select * from cte_Avg_freight_value order by Avg_freight_value limit 5)
order by Avg_freight_value
```



Row	customer_state	Avg_freight_value
1	SP	15.147275390419248
2	PR	20.531651567944248
3	MG	20.630166806306541
4	RJ	20.96092393168248
5	DF	21.041354945968383
6	PI	39.147970479704767
7	AC	40.073369565217405
8	RO	41.069712230215842
9	PB	42.723803986710941
10	RR	42.984423076923093

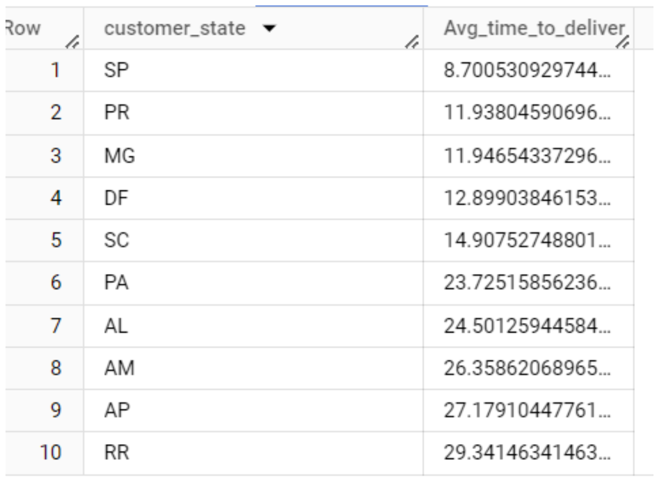
Insights: States like SP, PR, MG, RJ and DF have lowest average freight value and PI, AC, RO, PB and RR have highest freight value

Recommendations: Decrease the Freight value of top 5 states with highest avd freight time to increase the profitability.

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

Query:
```
with CTE_Avg_delivery_time as
(select c.customer_state, Avg(Date_diff(Extract(date from
order_delivered_customer_date), Extract(date from order_purchase_timestamp), day)) as
Avg_time_to_deliver   from `Target.Customers` c join `Target.Orders` o on
c.customer_id = o.customer_id
 where order_delivered_customer_date is not null
group by c.customer_state)

(select * from CTE_Avg_delivery_time order by Avg_time_to_deliver desc limit 5)
union all
(select * from CTE_Avg_delivery_time order by Avg_time_to_deliver limit 5)
order by Avg_time_to_deliver
```



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

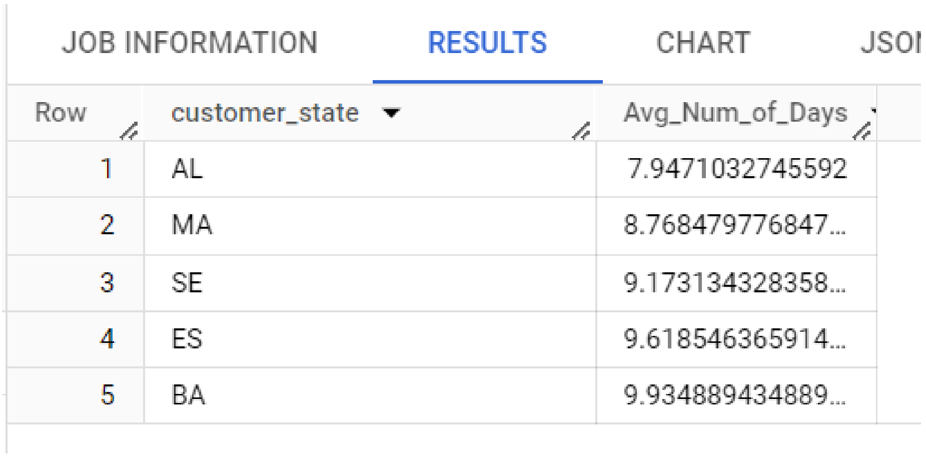
Insights: SP, PR, MG, DF and SC states took less time to deliver.

PA, AL, AM, AP and RR states took more time to deliver

Recommendations: Increase the delivery time for the states with highest average delivery time by hiring more delivery partners.

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

Query:
```
select c.customer_state,
AVG(DATE_DIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date, day))
Avg_Num_of_Days from `Target.Customers` c join `Target.Orders` o on c.customer_id =
o.customer_id where order_delivered_customer_date is not null
group by c.customer_state
order by Avg_Num_of_Days
limit 5
```



Row	customer_state	Avg_Num_of_Days
1	AL	7.9471032745592
2	MA	8.768479776847...
3	SE	9.173134328358...
4	ES	9.618546365914...
5	BA	9.934889434889...

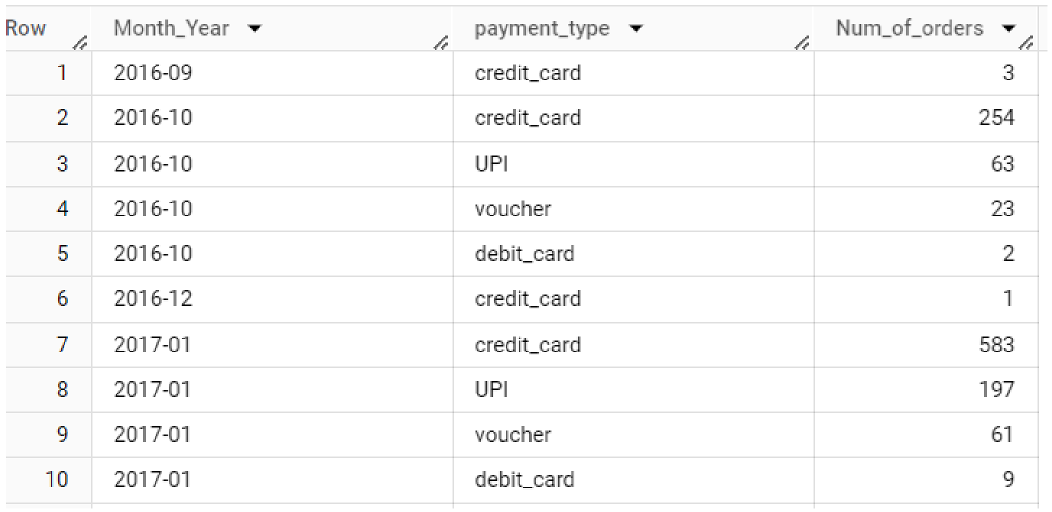
Insights : AL, MA, SE, ES and BA states made the fastest delivery compared to estimated dates

Recommendations: Increase the delivery speed in other states by hiring more delivery partners and stock up the inventory.

6.Analysis based on the payments:

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

Query:
```
select FORMAT_DATE('%Y-%m' , order_purchase_timestamp) as Month_Year,
p.payment_type, count(o.order_id) as Num_of_orders from `Target.Orders` o join
`Target.Payments` p on o.order_id = p.order_id group by Month_Year, p.payment_type
order by Month_Year
```



Row	Month_Year	payment_type	Num_of_orders
1	2016-09	credit_card	3
2	2016-10	credit_card	254
3	2016-10	UPI	63
4	2016-10	voucher	23
5	2016-10	debit_card	2
6	2016-12	credit_card	1
7	2017-01	credit_card	583
8	2017-01	UPI	197
9	2017-01	voucher	61
10	2017-01	debit_card	9

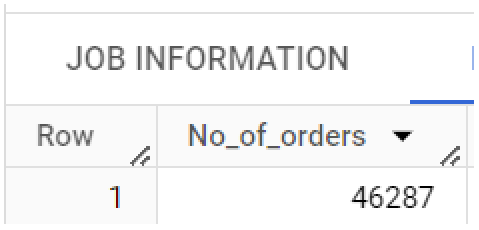
Insights: Voucher payment type is less in most of the months

Recommendations: Give some offers on UPI or debit card payment to increase the payment type to UPI or debit card

---

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

Query:
```
select count(*) as No_of_orders from
(select o.order_id,  p.payment_installments , max(p.payment_sequential) as Seq_max
from `Target.Orders` o join `Target.Payments` p on o.order_id = p.order_id group by
o.order_id, p.payment_installments) t
where t.payment_installments = t.Seq_max
```



JOB INFORMATION

Row	No_of_orders
1	46287

Insights: 46287 number of orders made the full payment installments

Recommendations : Decrease the payment installation (EMI) to get the complete payments at the earliest.