

INSIGHTS from dataset of operations of TARGET in Brazil between 2016 and 2018

Question 1

Part 1: Data type of all columns in the "customers" table

```
Select column_name, data_type
from `Target_Scalar_proj1.INFORMATION_SCHEMA.COLUMNS`
WHERE TABLE_NAME = 'Customer';
```

Query results			SAVE RESULTS	EXPLORE DATA	
JOB INFORMATION			RESULTS	CHART	JSON
EXECUTION DETAILS			EXECUTION GRAPH		
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			

Part 2 : Time range between which the orders were placed

```
select min(order_purchase_timestamp) as `first_order_time`,
max(order_purchase_timestamp) as `last_order_time`
from `Target_Scalar_proj1.orders`;
```

Query results			SAVE RESULTS	EXPLORE DATA	
JOB INFORMATION			RESULTS	CHART	JSON
EXECUTION DETAILS			EXECUTION GRAPH		
Row	first_order_time	last_order_time			
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC			

Part 3: The Cities & States of customers who ordered

```
select count(distinct customer_city) as `City_Count`, count(distinct customer_state) as
`State_Count`
from `Target_Scalar_proj1.Customer`
```

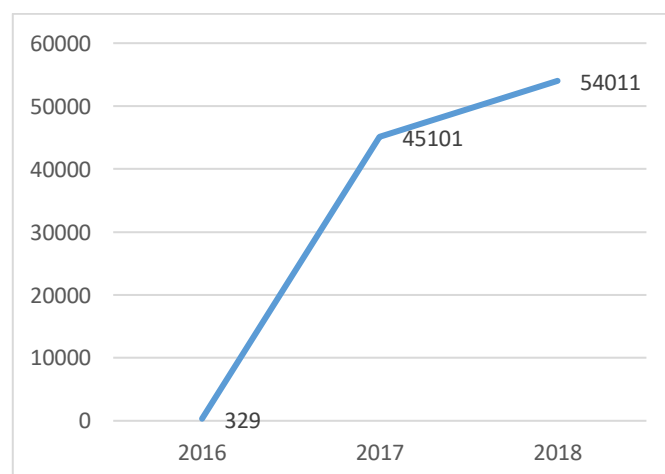
Query results				SAVE RESULTS	EXPLORE DATA	
JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	City_Count	State_Count				
1	4119	27				

Question2

Part 1: 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 `COUNT ORDERS`
from `Target_Scalar_proj1.orders`
GROUP BY EXTRACT(YEAR FROM order_purchase_timestamp)
ORDER BY YEAR;
```

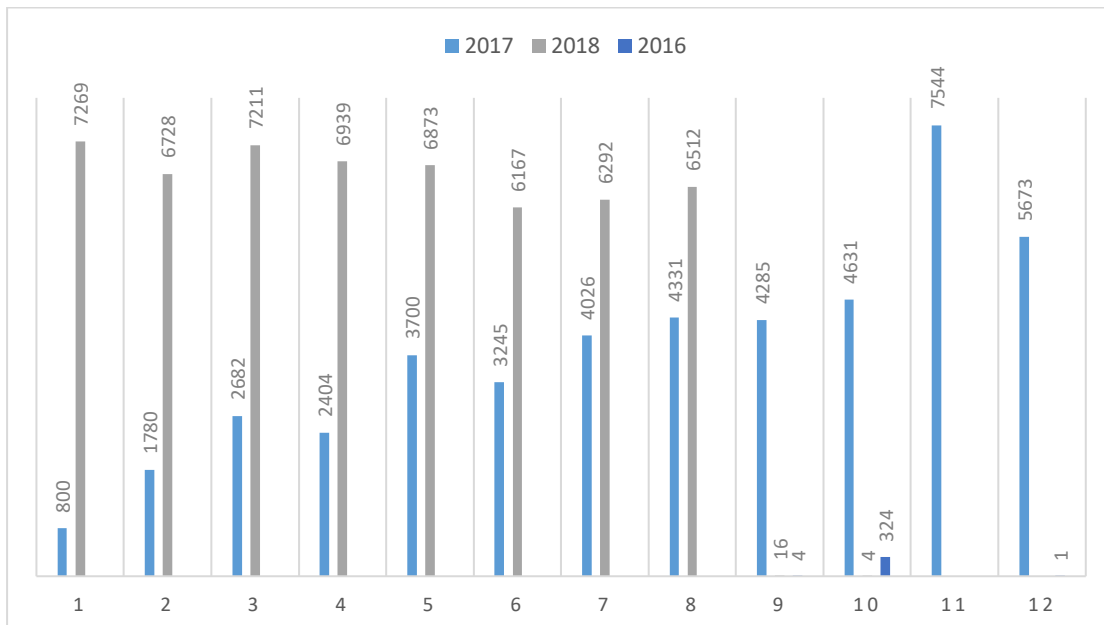
Query results				SAVE RESULTS	EXPLORE DATA	
JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	YEAR	COUNT ORDERS				
1	2016	329				
2	2017	45101				
3	2018	54011				



Part 2 : 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 `COUNT ORDERS`
from `Target_Scalar_proj1.orders`
GROUP BY EXTRACT(YEAR FROM order_purchase_timestamp),EXTRACT(MONTH FROM
order_purchase_timestamp)
ORDER BY YEAR,MONTH;
```

Query results					SAVE RESULTS	EXPLORE DATA	
JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS		EXECUTION GRAPH
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				





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


```
select
  (case when extract(hour from order_purchase_timestamp ) >=0 and extract(hour from
order_purchase_timestamp ) <=6 then "Dawn"
when extract(hour from order_purchase_timestamp ) >=7 and extract(hour from
order_purchase_timestamp )<=12 then "Mornings"
when extract(hour from order_purchase_timestamp ) >=13 and extract(hour from
order_purchase_timestamp )<=18 then "Afternoon"
when extract(hour from order_purchase_timestamp ) >=19 and extract(hour from
order_purchase_timestamp )<=23 then "Night"
end) as `time_of_day` , count(*)
from `Target_Scalar_proj1.orders`

group by 1
```

Query results

 SAVE RESULTS

 EXPLORE DATA



JOB INFORMATION

RESULTS

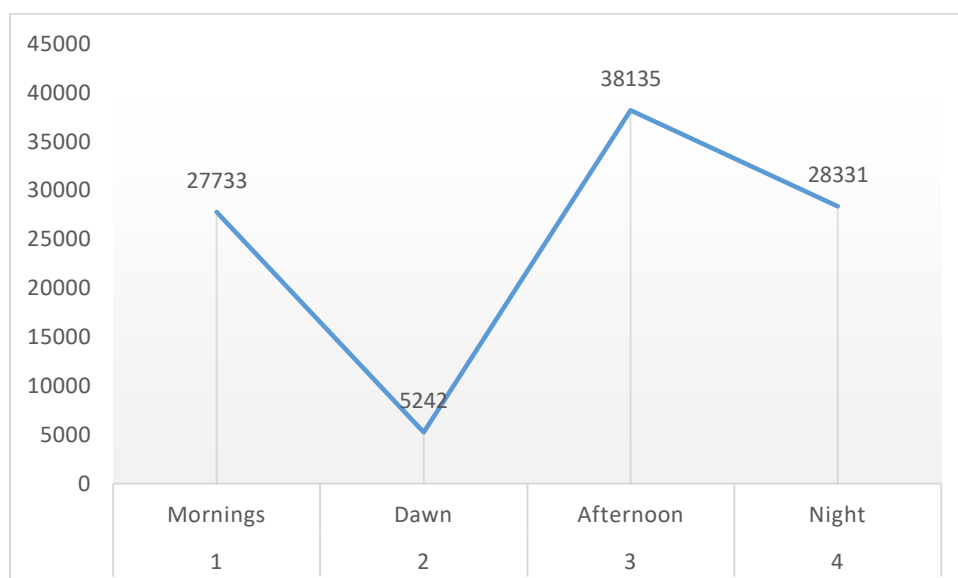
CHART

JSON

EXECUTION DETAILS

EXECUTION GRAPH

Row	time_of_day	no_of_orders
1	Mornings	27733
2	Dawn	5242
3	Afternoon	38135
4	Night	28331





Question 3


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

```
select customer_state, extract(month from order_purchase_timestamp) as `month`, count(*) as `no_of_orders`  
from `Target_Scalar_proj1.orders` join `Target_Scalar_proj1.Customer`  
using (customer_id)  
where order_status not in ('canceled','unavailable')  
group by customer_state, extract(month from order_purchase_timestamp)  
order by customer_state, month;
```

Query results

 SAVE RESULTS

 EXPLORE DATA



JOB INFORMATION

RESULTS

CHART

JSON

EXECUTION DETAILS





EXECUTION GRAPH

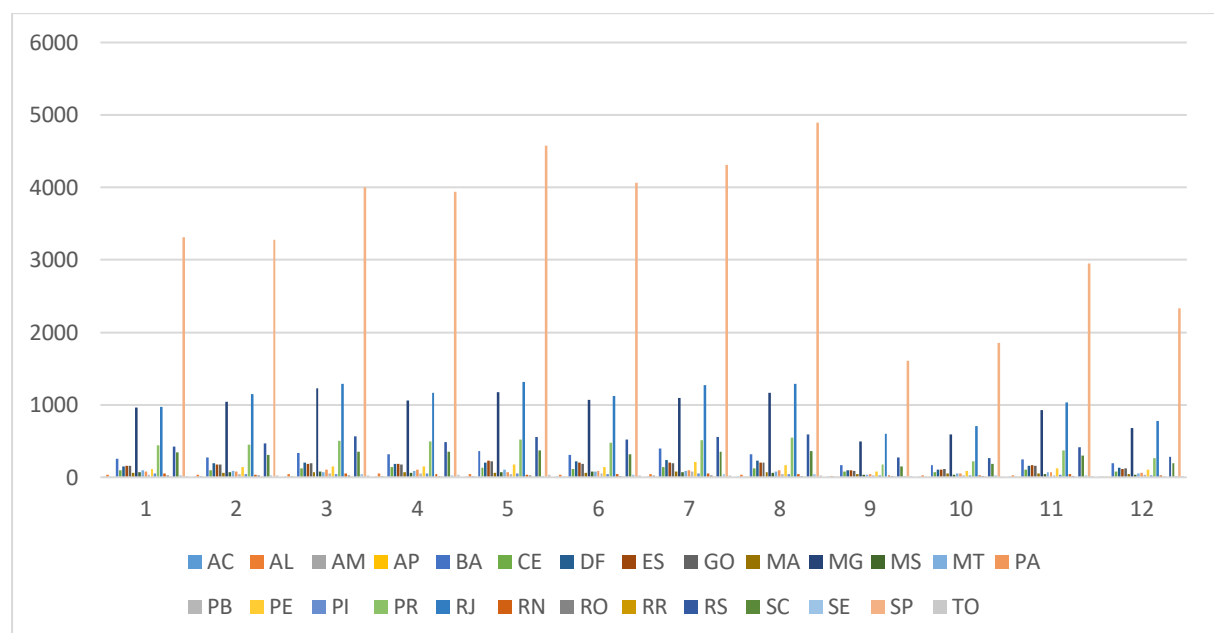
Row	customer_state	month	no_of_orders	
1	AC	1	8	
2	AC	2	6	
3	AC	3	4	
4	AC	4	9	
5	AC	5	10	
6	AC	6	7	
7	AC	7	9	
8	AC	8	7	
9	AC	9	5	
10	AC	10	6	

Results per page:

50

1 – 50 of 322

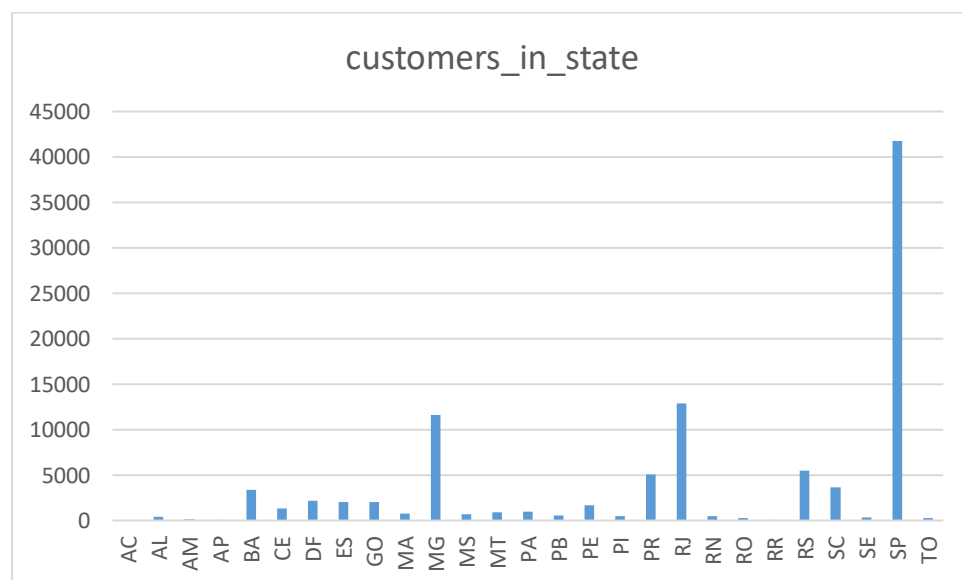




Part 2 : How are the customers distributed across all the states?

```
select customer_state, count(distinct customer_id) as `customer_in_state`
from `Target_Scalar_proj1.Customer`
group by customer_state
order by customer_state;
```

Query results		SAVE RESULTS	EXPLORE DATA	
JOB INFORMATION		RESULTS	CHART	JSON
EXECUTION DETAILS		EXECUTION GRAPH		
Row	customer_state	customer_in_state		
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		



Question 4

Part 1 : Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

```

with a as(select *,extract(MONTH from order_purchase_timestamp) as month, extract(year from
order_purchase_timestamp) as year
from `Target_Scalar_proj1.orders`),

b as (select sum(payment_value) as `2017_sum`
from `Target_Scalar_proj1.payments` join a using(order_id)
where year = 2017 and month between 1 and 8),

c as ( select sum(payment_value) as `2018_sum`
from `Target_Scalar_proj1.payments` join a using(order_id)
where year = 2018 and month between 1 and 8)

select (`2018_sum`-`2017_sum`)/`2017_sum` *100 as `percentage_increase`
from b,c;

```

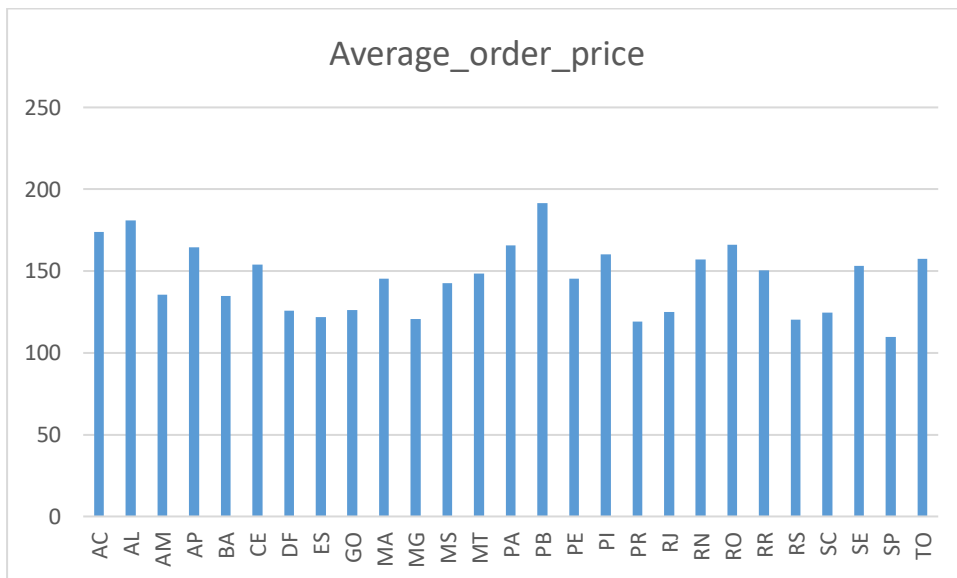
Query results		SAVE RESULTS	EXPLORE DATA	
JOB INFORMATION		RESULTS	CHART	JSON
		EXECUTION DETAILS	EXECUTION GRAPH	
Row	percentage_increase			
1	136.9768716466...			

Part 2 : The Total & Average value of order price for each state


```


select customer_state, round(sum(price),2) as `Total_order_price`, round(avg(price),2) as
`Average_order_price`
from `Target_Scalar_proj1.Customer` join `Target_Scalar_proj1.orders` using (customer_id)
join `Target_Scalar_proj1.order_items` using (order_id)
group by customer_state
order by customer_state;


```



Query results

 SAVE RESULTS ▾

 EXPLORE DATA ▾



JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	customer_state ▾	Total_order_price ▾	Average_order_price ▾			
1	AC	15982.95	173.73			
2	AL	80314.81	180.89			
3	AM	22356.84	135.5			
4	AP	13474.3	164.32			
5	BA	511349.99	134.6			
6	CE	227254.71	153.76			
7	DF	302603.94	125.77			
8	ES	275037.31	121.91			
9	GO	294591.95	126.27			
10	MA	119648.22	145.2			

Results per page: 50 ▾ 1 – 27 of 27

⏪

⏴

⏵

⏩


Part 3 : The Total & Average value of order freight for each state


```


select customer_state, round(sum(freight_value),2) as `Total_order_freight`,
round(avg(freight_value),2) as `Average_order_freight`
from `Target_Scalar_proj1.Customer` join `Target_Scalar_proj1.orders` using (customer_id)
join `Target_Scalar_proj1.order_items` using (order_id)
group by customer_state
order by customer_state;

```

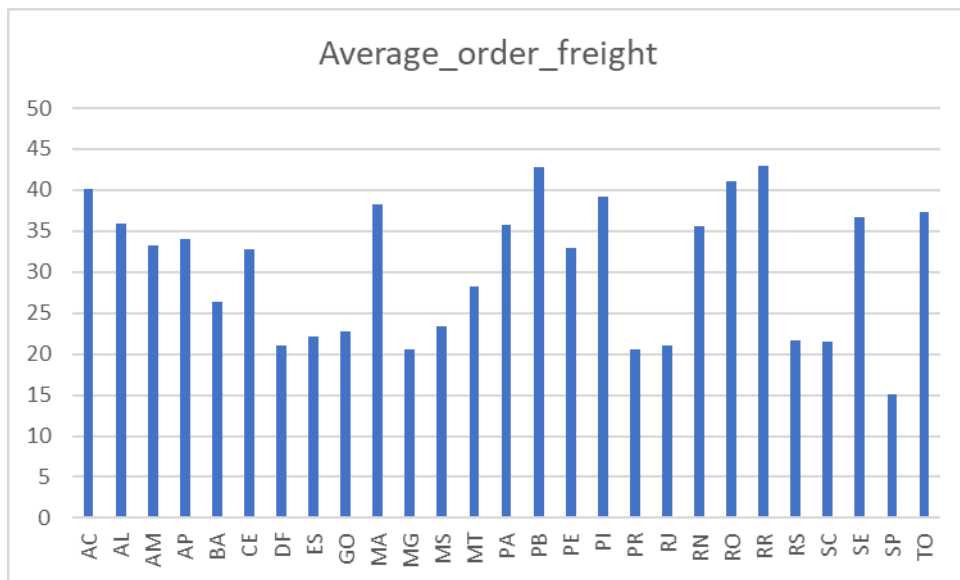
Query results

 SAVE RESULTS ▾

 EXPLORE DATA ▾



JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	customer_state ▾	Total_order_freight	Average_order_freight			
1	AC	3686.75	40.07			
2	AL	15914.59	35.84			
3	AM	5478.89	33.21			
4	AP	2788.5	34.01			
5	BA	100156.68	26.36			
6	CE	48351.59	32.71			
7	DF	50625.5	21.04			
8	ES	49764.6	22.06			
9	GO	53114.98	22.77			
10	MA	31523.77	38.26			



Question 5

Part 1: The no. of days taken to deliver each order from the order's purchase date as delivery time and the difference (in days) between the estimated & actual delivery date of an order

```
select order_id,date_diff(order_delivered_customer_date, order_purchase_timestamp,day ) as
`time_to_deliver_in_Days`,
date_diff(order_delivered_customer_date,order_estimated_delivery_date,day ) as
`diff_estimated_delivery`
from `Target_Scalar_proj1.orders`
where order_status = 'delivered';
```

Query results				SAVE RESULTS	EXPLORE DATA	
JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	order_id	time_to_deliver_in_D	diff_estimated_delive			
1	635c894d068ac37e6e03dc54e...	30	-1			
2	3b97562c3aee8bdedcb5c2e45...	32	0			
3	68f47f50f04c4cb6774570cfde...	29	-1			
4	276e9ec344d3bf029ff83a161c...	43	4			
5	54e1a3c2b97fb0809da548a59...	40	4			
6	fd04fa4105ee8045f6a0139ca5...	37	1			
7	302bb8109d097a9fc6e9cefc5...	33	5			
8	66057d37308e787052a32828...	38	6			
9	19135c945c554eebfd7576c73...	36	2			
10	4493e45e7ca1084efcd38ddeb...	34	0			
				Results per page:	50	1 – 50 of 96478

Part 2: Top 5 states with the highest & lowest average freight value

```

select customer_state, average_order_freight, r
from
(select customer_state, round(avg(freight_value),2) as `Average_order_freight`,
dense_rank() over(order by round(avg(freight_value),2) desc) as r
from `Target_Scalar_proj1.Customer`
join `Target_Scalar_proj1.orders` using (customer_id)
join `Target_Scalar_proj1.order_items` using (order_id)
group by customer_state
order by Average_order_freight desc) as t
where r<=5 or r>=22

```

Query results				SAVE RESULTS	EXPLORE DATA	
JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	customer_state					
1	SP					
2	PR					
3	MG					
4	RJ					
5	DF					
6	PI					
7	AC					
8	RO					
9	PB					
10	RR					

Part 3: Top 5 states with the highest & lowest average delivery time

```
select customer_state from
(
select customer_state,round(avg(time_to_deliver_in_Days),2) as `avg_time`,dense_rank()
over(order by avg(time_to_deliver_in_Days) desc) as `r`
from (select order_id,customer_id,date_diff(order_delivered_customer_date,
order_purchase_timestamp,day ) as `time_to_deliver_in_Days`
from `Target_Scalar_proj1.orders`
where order_status = 'delivered') t join `Target_Scalar_proj1.Customer` using(customer_id)
group by customer_state
order by r
) as t
where r<6 or r >22
order by avg_time desc
```


Query results		SAVE RESULTS	EXPLORE DATA	
JOB INFORMATION		RESULTS	CHART	JSON
EXECUTION DETAILS		EXECUTION GRAPH		
Row	customer_state			
1	RR			
2	AP			
3	AM			
4	AL			
5	PA			
6	SC			
7	DF			
8	MG			
9	PR			
10	SP			


Part 4 : Top 5 states where the order delivery is really fast as compared to the estimated date of delivery


```
select customer_state
from
(select customer_state, avg(diff_estimated_delivery) as `early_delivery`
from
(select order_id, customer_id,
date_diff(order_delivered_customer_date,order_estimated_delivery_date,day )
as`diff_estimated_delivery`
From `Target_Scalar_proj1.orders`
where order_status = 'delivered'and order_delivered_customer_date is not
null
order by diff_estimated_delivery) as t
join `Target_Scalar_proj1.Customer` using (customer_id)
group by customer_state)
order by early_delivery
```

limit 5

Query results

 SAVE RESULTS

 EXPLORE DATA



JOB INFORMATION

RESULTS

CHART

JSON

EXECUTION DETAILS

EXECUTION GRAPH


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


Question 6


Part 1 : Month on month no. of orders placed using different payment types

```
select payment_type,extract(month from order_purchase_timestamp) as
`Month`,count(payment_type) as `No_of_orders`
from `Target_Scalar_proj1.payments` join `Target_Scalar_proj1.orders`
using(order_id)
group by payment_type, extract(month from order_purchase_timestamp)
order by month, payment_type;
```

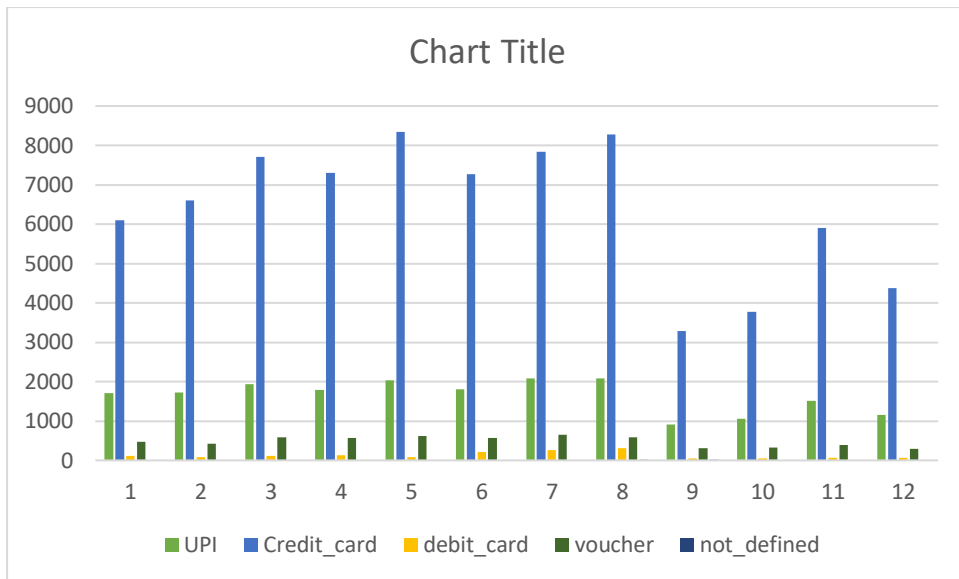
Query results

 SAVE RESULTS ▾

 EXPLORE DATA ▾



JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	payment_type ▾	Month ▾	No_of_orders ▾			
1	UPI	1	1715			
2	credit_card	1	6103			
3	debit_card	1	118			
4	voucher	1	477			
5	UPI	2	1723			
6	credit_card	2	6609			
7	debit_card	2	82			
8	voucher	2	424			
9	UPI	3	1942			



Part 2: No. of orders placed on the basis of the payment instalments that have been paid

```
select payment_installments,count(payment_installments) as `No_of_orders`
from `Target_Scalar_proj1.payments`
group by payment_installments;
```

Query results

SAVE RESULTS

EXPLORE DATA

JOB INFORMATION

RESULTS

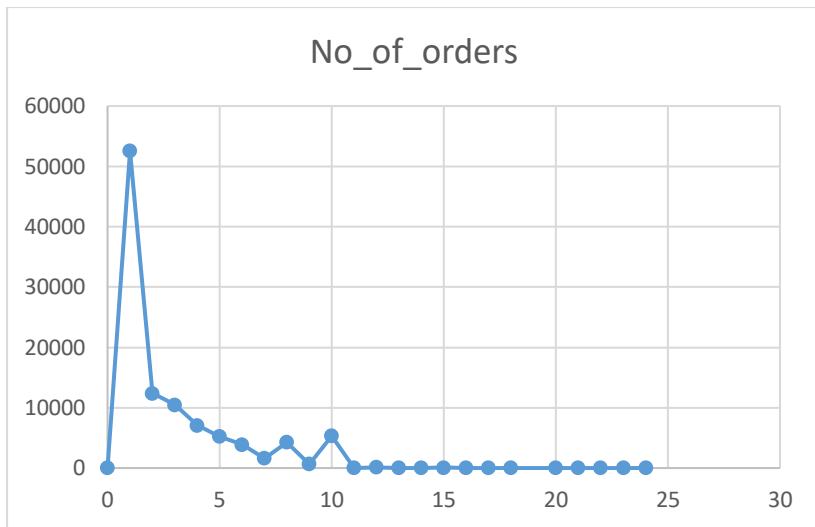
CHART

JSON

EXECUTION DETAILS

EXECUTION GRAPH

Row	payment_installment	No_of_orders
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268



INSIGHTS:

- The dataset covers 4119 cities in 27 Brazilian states from September 4, 2016, to October 17, 2018.
- There is a growing trend in the number of orders placed from 2016 to 2018
- There is no significant monthly seasonality in the orders placed during these year
- The majority of orders are placed in the afternoon (38,135), followed by night (28,331), and morning (27,733). Dawn sees the least number of orders (5,242)
- Top Customer States: SP, RJ, MG, PI, RS
States with Minimal Customers: AC, AM, AP, RR
- Freight price is the highest for stat SP followed by PR, MG,RJ and DF
- But for the same states we see that the delivery time for these states is very high.
- Credit cards are the preferred payment method throughout the year, with a significant portion of customers opting to pay in one instalment.