Target Business Case Report --- SQL

1. **8 Dataset is imported to big-query for analysis of project.**
   a. Data type of all columns in the "customers" table.

```
select
    column_name,
    data_type
    FROM
    Scaler_Sql_Project.INFORMATION_SCHEMA.COLUMNS
    where
    table_name ='customers'
```

Query results

JOB INFORMATION | RESULTS | CHART

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

Insights: In this customer table we found that the data types are string and integer data type

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

```
SELECT
    MIN(order_purchase_timestamp) AS First_order,
    MAX(order_purchase_timestamp) AS Last_order
FROM
    Scaler_Sql_Project.orders
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXE

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

Insights: The time range which the customers make orders is between 2016 to 2018

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

```
SELECT
    COUNT(DISTINCT c.Customer_city) AS Num_Cities,
    COUNT(DISTINCT c.Customer_state) AS Num_States
FROM
    Scaler_Sql_Project.customers c
JOIN
    Scaler_Sql_Project.orders o ON c.Customer_id = o.customer_id;
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXEC

Row	Num_Cities	Num_States
1	4119	27

Insights: There are 4119 cities and 27 states of customers who ordered the products.

2. **In-depth Exploration:**

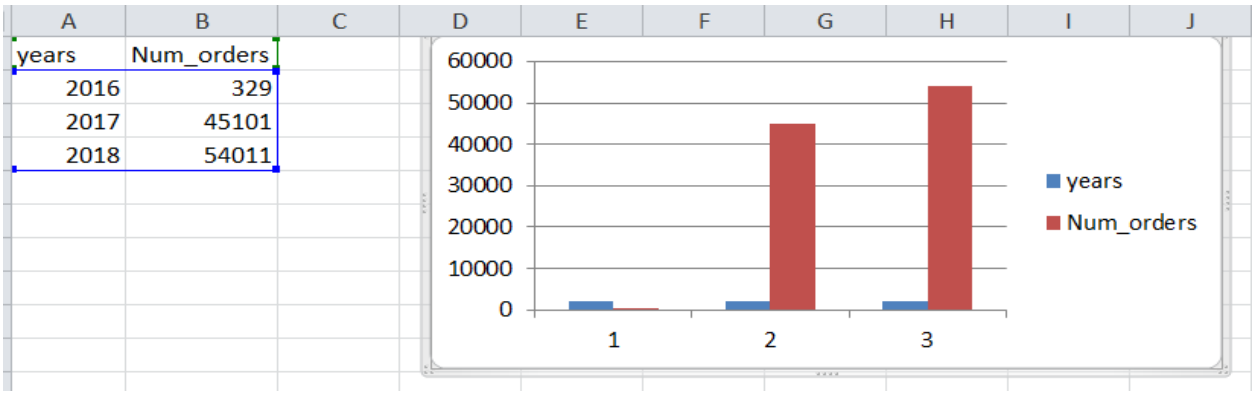
   a) **Is there a growing trend in the no. of orders placed over the past years?**

```
SELECT
    EXTRACT(YEAR FROM order_purchase_timestamp) AS years,
    COUNT(order_id) AS Num_orders
FROM
    Scaler_Sql_Project.orders
GROUP BY
    years
Order By 1
```

Query results

JOB INFORMATION | RESULTS | CHART

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



Insights: There is a exponential growth on number of orders over year on year

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

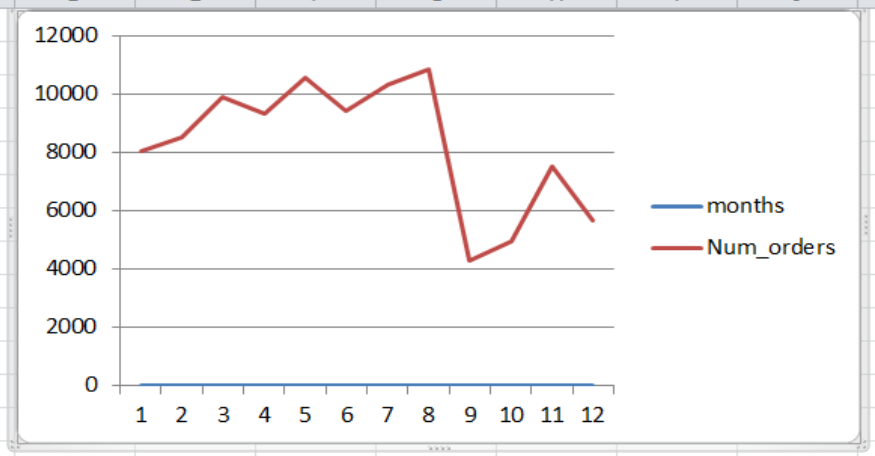
```
SELECT
    EXTRACT(MONTH FROM order_purchase_timestamp) AS months,
    COUNT(order_id) AS Num_orders
FROM
    Scaler_Sql_Project.orders
GROUP BY
    months
ORDER BY
    months
```

Query results

JOB INFORMATION | RESULTS | CHART

Row	months	Num_orders
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

months	Num_orders
1	8069
2	8508
3	9893
4	9343
5	10573
6	9412
7	10318
8	10843
9	4305
10	4959
11	7544
12	5674



Insights: From the line chart we can easily identify that there is an up and down in number of orders on monthly basis but these is high sales order in August month.

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

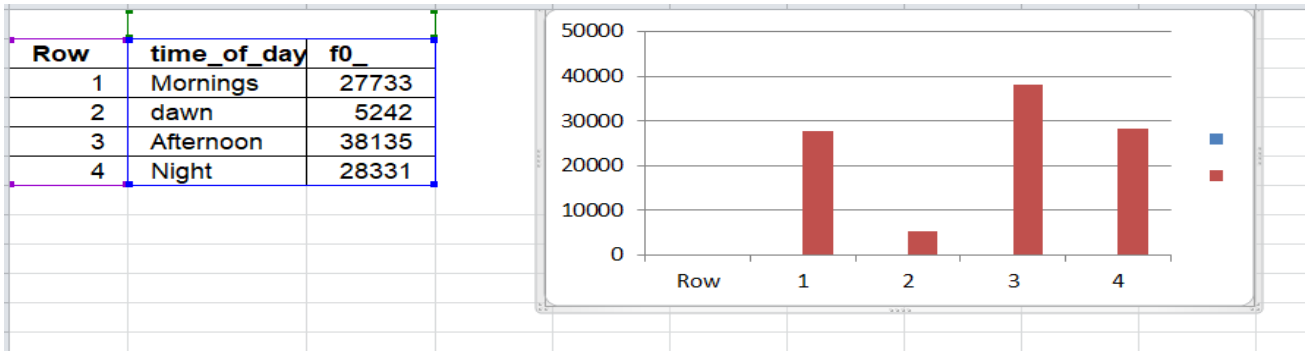
a. 0-6 hrs : Dawn
b. 7-12 hrs : Mornings
c. 13-18 hrs : Afternoon
d. 19-23 hrs : Night

```
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'
End As time_of_day,
count(order_id)
from Scaler_Sql_Project.orders
group by
    time_of_day
```

Query results

JOB INFORMATION | RESULTS | CHART | J

Row	time_of_day	f0_
1	Mornings	27733
2	dawn	5242
3	Afternoon	38135
4	Night	28331



Insights: From this graph we can easily identify that during afternoon time Brazilian customers mostly place their order while dawn records the fewest order followed by Night and mornings.

c) **Evolution of E-commerce orders in the Brazil region:**

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

```
select
      Extract(Month from O.order_purchase_timestamp),
c.customer_state as State,
      count(order_id) as Num_Orders
from Scaler_Sql_Project.orders o
inner join Scaler_Sql_Project.customers c on o.customer_id =
c.customer_id
group by Extract(Month from O.order_purchase_timestamp),
c.customer_state
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS

Row	f0_	State	Num_Orders
1	11	RJ	1048
2	12	RS	283
3	12	SP	2357
4	2	DF	196
5	11	PR	378
6	4	MT	92

Insights: From the result we can find the State (SP) has highest number of sales (4982) and the states (RR & AP) has lowest sales order of 2 only.

b) How are the customers distributed across all the states?

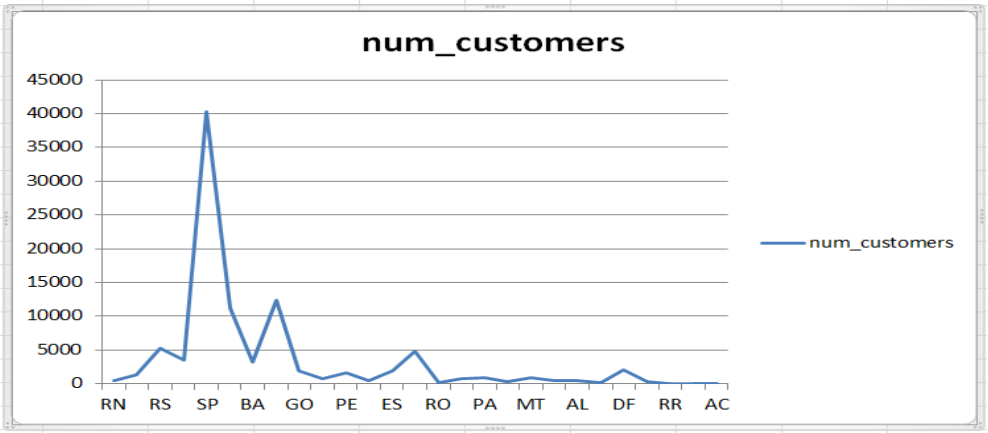
```
SELECT
    c.customer_state as State,
    COUNT(Distinct c.customer_unique_id) AS num_customers
FROM
    `Scaler_Sql_Project.customers`as c
GROUP BY
    c.customer_state
```

Query results

JOB INFORMATION | RESULTS | CHART

Row	State	num_customers
1	RN	474
2	CE	1313
3	RS	5277
4	SC	3534
5	SP	40302
6	MG	11259
7	BA	3277

State	num_customers
RN	474
CE	1313
RS	5277
SC	3534
SP	40302
MG	11259
BA	3277
RJ	12384
GO	1952
MA	726
PE	1609
PB	519
ES	1964
PR	4882
RO	240
MS	694
PA	949
TO	273
MT	876
PI	482
AL	401
AM	143
DF	2075
SE	342
RR	45
AP	67
AC	77



Insights: From this graph we can find the state (SP) has high distribution of sales order as compared to other states.

4. 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). You can use the "payment_value" column in the payments table to get the cost of orders.

```sql
WITH final AS (
    SELECT *
    FROM Scaler_Sql_Project.payments p
    JOIN Scaler_Sql_Project.orders o ON p.order_id = o.order_id
    WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) BETWEEN 2017 AND 2018
    AND EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
),
final_Cal AS (
    SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS yr,
           SUM(payment_value) AS cost
    FROM final
    GROUP BY 1
    ORDER BY 1
)

SELECT *,
       (100 * (LEAD(cost, 1) OVER (ORDER BY yr) - cost) / cost) AS perc_yoy
FROM final_Cal
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS

Row	yr	cost	perc_yoy
1	2018	8694733.839999...	null
2	2017	3669022.120000...	136.9768716466...

Insights: The query computes the total payment value for each year, focusing on orders placed within the first eight months and there is approximately 136% high in order from 2017 to 2018

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

```sql
SELECT
   c.customer_state AS State,
   SUM(oi.price) AS Total_order_price,
   AVG(oi.price) AS Avg_order_price
FROM Scaler_Sql_Project.order_items oi
JOIN Scaler_Sql_Project.orders o ON oi.order_id = o.order_id
JOIN Scaler_Sql_Project.customers c ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY State
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS

Row	State	Total_order_price	Avg_order_price
1	AC	15982.94999999...	173.7277173913...
2	AL	80314.80999999...	180.8892117117...
3	AM	22356.84000000...	135.4959999999...
4	AP	13474.29999999...	164.3207317073...
5	BA	511349.9900000...	134.6012082126...
6	CE	227254.7099999...	153.7582611637...
7	DF	302603.9399999...	125.7705486284...

Insights: This query provides a breakdown of the total number of unique customers per state. São Paulo (SP) leads with the highest number of unique customers, followed by Rio de Janeiro (RJ) and Minas Gerais (MG).

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

```
SELECT
   c.customer_state AS State,
   SUM(oi.freight_value) AS Total_order_freight,
   AVG(oi.freight_value) AS Avg_order_freight
FROM Scaler_Sql_Project.order_items oi
JOIN Scaler_Sql_Project.orders o ON oi.order_id = o.order_id
JOIN Scaler_Sql_Project.customers c ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY State
```

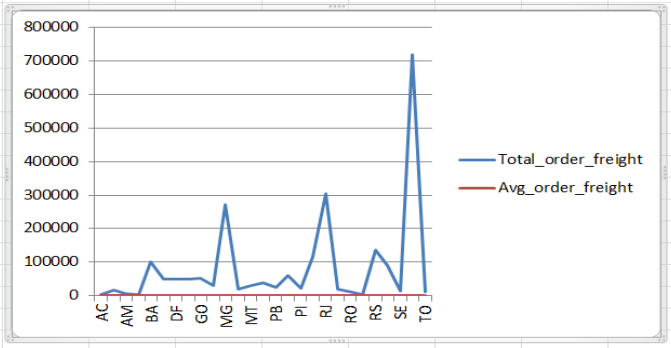
Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS

Row	State	Total_order_freight	Avg_order_freight
1	AC	3686.750000000...	40.07336956521...
2	AL	15914.58999999...	35.84367117117...
3	AM	5478.890000000...	33.20539393939...
4	AP	2788.500000000...	34.00609756097...
5	BA	100156.6799999...	26.36395893656...
6	CE	48351.58999999...	32.71420162381...
7	DF	50625.49999999...	21.04135494596...

Chart 1

	A	B	C
1	State	Total_order_freight	Avg_order_freight
2	AC	3686.75	40.07336957
3	AL	15914.59	35.84367117
4	AM	5478.89	33.20539394
5	AP	2788.5	34.00609756
6	BA	100156.68	26.36395894
7	CE	48351.59	32.71420162
8	DF	50625.5	21.04135495
9	ES	49764.6	22.0587766
10	GO	53114.98	22.76681526
11	MA	31523.77	38.25700243
12	MG	270853.46	20.63016681
13	MS	19144.03	23.374884
14	MT	29715.43	28.16628436
15	PA	38699.3	35.83268519
16	PB	25719.73	42.72380399
17	PE	59449.66	32.91786268
18	PI	21218.2	39.14797048
19	PR	117851.68	20.53165157
20	RJ	305589.31	20.96092393
21	RN	18860.1	35.65236295
22	RO	11417.38	41.06971223
23	RR	2235.19	42.98442308
24	RS	135522.74	21.73580433
25	SC	89660.26	21.47036877
26	SE	14111.47	36.65316883
27	SP	718723.07	15.14727539
28	TO	11732.68	37.24660317



Insights: Form below graph we found that state (SP) having highest value of freight and state (MS) having lowest freight value.

5. 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.
   Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
   Do this in a single query.

   You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:
   - **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
   - **diff_estimated_delivery** = order_delivered_customer_date - order_estimated_delivery_date

```
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_esimated_delivery
from Scaler_Sql_Project.orders
where
    order_delivered_customer_date is Not Null
AND order_estimated_delivery_date is Not Null
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS

Row	order_id	time_to_deliver	diff_esimated_delive
1	1950d777989f6a877539f5379...	30	12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	-28
3	65d1e226dfaeb8cdc42f66542...	35	-16
4	635c894d068ac37e6e03dc54e...	30	-1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	-1
7	276e9ec344d3bf029ff83a161c...	43	4
8	54e1a3c2b97fb0809da548a59...	40	4
9	fd04fa4105ee8045f6a0139ca5...	37	1
10	302bb8109d097a9fc6e9cefc5...	33	5

Insights: The query calculates the time taken to deliver each order and the difference between the actual delivery date and the estimated delivery date. The output indicates that the maximum time taken to deliver an order is 209 days.

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

   For Lowest freight value

```sql
select customer_state, avg(oi.freight_value) as lowest_freight_value
from Scaler_Sql_Project.customers c
Join Scaler_Sql_Project.orders o
ON o.customer_id = c.customer_id
Join Scaler_Sql_Project.order_items oi
ON o.order_id = oi.order_id
Group by 1
Order by 2
limit 5
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON

Row	customer_state	lowest_freight_value
1	SP	15.14727539041...
2	PR	20.53165156794...
3	MG	20.63016680630...
4	RJ	20.96092393168...
5	DF	21.04135494596...

Insights: The top 5 states with the lowest average freight costs. São Paulo (SP) stands out with the lowest average freight value at $15.0, followed by Paraná (PR), Minas Gerais (MG), Rio de Janeiro (RJ), and Distrito Federal (DF), all sharing an average freight value of $21.0.

For Highest freight Value

```sql
select customer_state, avg(oi.freight_value) as highest_freight_value
from Scaler_Sql_Project.customers c
Join Scaler_Sql_Project.orders o
ON o.customer_id = c.customer_id
Join Scaler_Sql_Project.order_items oi
ON o.order_id = oi.order_id
Group by 1
Order by 2 Desc
limit 5
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON

Row	customer_state	highest_freight_value
1	RR	42.98442307692...
2	PB	42.72380398671...
3	RO	41.06971223021...
4	AC	40.07336956521...
5	PI	39.14797047970...

Insights: The query identifies the top 5 states with the highest average freight costs. Roraima (RR) and Paraíba (PB) lead with the highest freight value of $42.98 and $42.72, followed closely by Rondônia (RO) at $41.06. Acre (AC) ranks fourth with $40.07, while Piauí (PI) completes the list with $39.14.

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

For lowest average delivery time

```
SELECT
    c.customer_state,
    sum(timestamp_diff(order_delivered_customer_date,
order_purchase_timestamp, day)) / Count(order_id) AS avg_delivery_time
FROM
    Scaler_Sql_Project.orders o
JOIN
    Scaler_Sql_Project.customers c ON o.customer_id = c.customer_id
WHERE
    o.order_status = 'delivered'
GROUP BY
    c.customer_state
ORDER BY
    avg_delivery_time
LIMIT 5
```

Query results

JOB INFORMATION | RESULTS | CHART

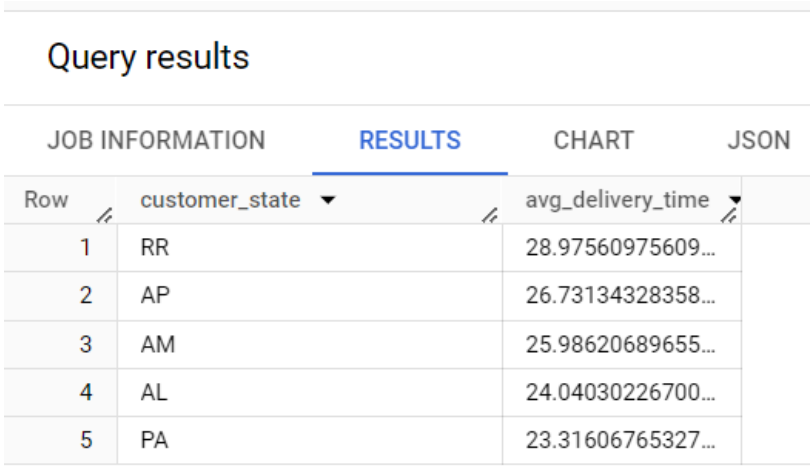
Row	customer_state	avg_delivery_time
1	SP	8.296659341744...
2	PR	11.52671135486...
3	MG	11.54218777523...
4	DF	12.50913461538...
5	SC	14.47518330513...

Insights: The query identifies the top 5 states with the lowest average delivery times. São Paulo leads with approximately 8.29 days, followed closely by Paraná and Minas Gerais, both around 11.52 days. Distrito Federal and Santa Catarina complete the list with average delivery times of roughly 12.50 and 14.47 days, respectively.

For highest average delivery time

```
SELECT
    c.customer_state,
    sum(timestamp_diff(order_delivered_customer_date,
order_purchase_timestamp, day)) / Count(order_id) AS avg_delivery_time
FROM
    Scaler_Sql_Project.orders o
JOIN
    Scaler_Sql_Project.customers c ON o.customer_id = c.customer_id
```

```
WHERE
    o.order_status = 'delivered'
GROUP BY
    c.customer_state
ORDER BY
    avg_delivery_time desc
LIMIT 5
```



Query results

JOB INFORMATION | RESULTS | CHART | JSON

Row	customer_state	avg_delivery_time
1	RR	28.97560975609...
2	AP	26.73134328358...
3	AM	25.98620689655...
4	AL	24.04030226700...
5	PA	23.31606765327...

Insights: The query identifies the top 5 states with the highest delivery times. The state Roraima has high delivery time compared to other states

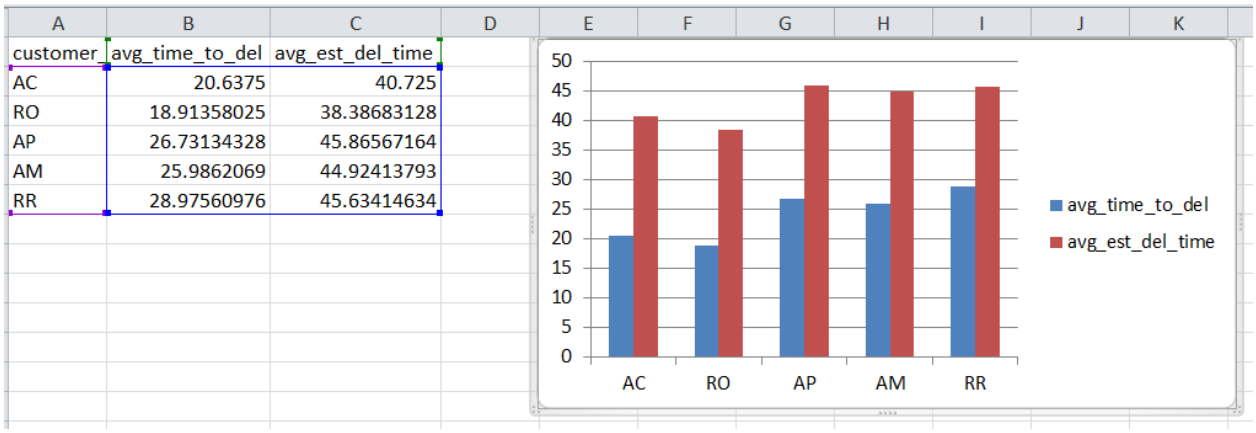
4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

```
SELECT
    c.customer_state,
    sum(timestamp_diff(order_delivered_customer_date,
                order_purchase_timestamp, day)) / Count(order_id) AS
avg_time_to_del,
    sum(timestamp_diff(order_estimated_delivery_date,
                order_purchase_timestamp, day)) / Count(order_id) AS
avg_est_del_time
FROM
    Scaler_Sql_Project.orders o
JOIN
    Scaler_Sql_Project.customers c ON o.customer_id = c.customer_id
WHERE
    o.order_status = 'delivered'
GROUP BY
    c.customer_state
ORDER BY (avg_time_to_del - avg_est_del_time)
LIMIT 5
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS

Row	customer_state	avg_time_to_del	avg_est_del_time
1	AC	20.6375	40.725
2	RO	18.91358024691...	38.38683127572...
3	AP	26.73134328358...	45.86567164179...
4	AM	25.98620689655...	44.92413793103...
5	RR	28.97560975609...	45.63414634146...



customer	avg_time_to_del	avg_est_del_time
AC	20.6375	40.725
RO	18.91358025	38.38683128
AP	26.73134328	45.86567164
AM	25.9862069	44.92413793
RR	28.97560976	45.63414634

Insights: Based on analysis we can find that the average time taken to deliver orders and the average estimated delivery time for each customer state. Ceará and Espírito Santo demonstrate slightly longer delivery times but still maintain a relatively small disparity between actual and estimated durations

6. **Analysis based on the payments:**

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

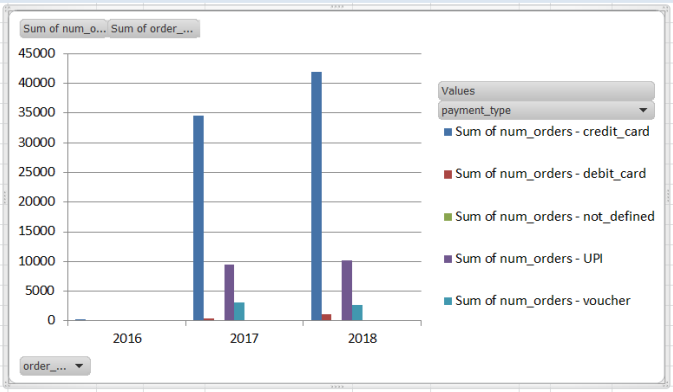
```
SELECT
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
    EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
    p.payment_type,
    COUNT(o.order_id) AS num_orders
FROM
    Scaler_Sql_Project.orders o
JOIN
    Scaler_Sql_Project.payments p ON o.order_id = p.order_id
GROUP BY
    order_year,
    order_month,
    p.payment_type
ORDER BY
    order_year,
    order_month,
    num_orders DESC
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | E

Row	order_year	order_month	payment_type	num_orders
1	2016	9	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	1	credit_card	583

	Column Labels											
	Sum of num_orders					Sum of order_month					Total Sum of num_orders	Total Sum of order_month
Row Labels	credit_card	debit_card	not_defined	UPI	voucher	credit_card	debit_card	not_defined	UPI	voucher		
2016	258	2		63	23	31	10		10	10	346	61
2017	34568	422		9508	3027	78	78		78	78	47525	312
2018	41969	1105	3	10213	2725	36	36	17	36	55	56015	180
Grand Total	**76795**	**1529**	**3**	**19784**	**5775**	**145**	**124**	**17**	**124**	**143**	**103886**	**553**



Insights: From the graph we easily found that the payment through credit card is made compared to other payment methods.

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

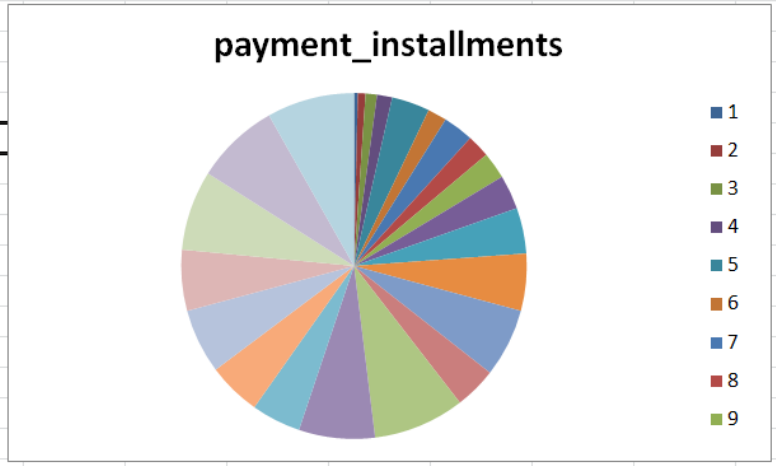
```
select payment_installments,
       count(order_id) as num_orders
from Scaler_Sql_Project.payments
where payment_installments >=1
group by 1
order by 2 desc
```

Query results

JOB INFORMATION | RESULTS | CHART

Row	payment_installment	num_orders
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268

payment_install	num_orders
1	52546
2	12413
3	10461
4	7098
10	5328
5	5239
8	4268
6	3920
7	1626
9	644
12	133
15	74
18	27
11	23
24	18
20	17
13	16
14	15
17	8
16	5
21	3
22	1
23	1



Insights: The data highlights a preference for single payment installments among customers, with over 52,000 orders falling into this category.