

## **Business Case: Target SQL – Analysis Report**

**Q.1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**

1. Data type of all columns in the "customers" table.

**Solution 1.1:**

**Table(s): 'customers'**

**Query/Code:**

```
SELECT COLUMN_NAME, DATA_TYPE
FROM scaler-dsml-tushar-sql.BUSCASE1.INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME='customers';
```

**Results/Output:**

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
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		

**Observations/Remark:**

- a) The above query is return the data types of each table, this helps us to ensure the use of right logic for analysis and interpretation of the dataset.
- b) The further we can also get the various details for applying the functions and performing the suitable operation for data analysis such as table\_name; column\_name; ordinal\_position; is\_nullable; data\_type; is\_system\_defined, etc. from below link:

COLUMNS view/ INFORMATION\_SCHEMA.COLUMNS:

<https://cloud.google.com/bigquery/docs/information-schema-columns>

**Q.1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**

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

**Solution 1.2:**

**Table(s): 'orders'**

**Query/Code:**

```
SELECT min(order_purchase_timestamp) as Inidate,
       max(order_purchase_timestamp) as lastdate
FROM `BUSCASE1.orders`
```

## Results/Output:

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	Inidate		lastdate	
1	2016-09-04 21:15:19 UTC		2018-10-17 17:30:18 UTC	

## Observations/Remark:

- Order Period is from 4<sup>th</sup> September 2016 to 12<sup>th</sup> October 2018.
- This sets our analysis window to be for approx. 2 years, 1.5 month but only covers one whole Calander/ Financial Year.

## Q.1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

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

### Solution 1.3:

- Count of cities, State of customers are as under:**
  - Cities: 4119
  - States: 27
  - Customers: 99441
  - Orders: 99441

**Tables: 'orders' and 'customers'** as have requisite parameters/columns

### Joint ID: customer\_id

The orders.csv contain following features:		The customers.csv contain following features:	
Features	Description	Features	Description
order_id	A Unique ID of order made by the consumers	customer_id	ID of the consumer who made the purchase
customer_id	ID of the consumer who made the purchase	customer_unique_id	Unique ID of the consumer
order_status	Status of the order made i.e. delivered, shipped, etc.	customer_zip_code_prefix	Zip Code of consumer's location
order_purchase_timestamp	Timestamp of the purchase	customer_city	Name of the City from where order is made
order_delivered_carrier_date	Delivery date at which carrier made the delivery	customer_state	State Code from where order is made (Eg. são paulo)
order_delivered_customer_date	Date at which customer got the product		
order_estimated_delivery_date	Estimated delivery date of the products		

## Query/Code:

```
SELECT
COUNT(DISTINCT a.customer_city) AS City_Nos,
COUNT(DISTINCT a.customer_state) AS State_Nos,
COUNT(DISTINCT b.customer_id) as Cutomer_Nos,
COUNT(DISTINCT b.order_id) as Order_Nos
FROM `BUSCASE1.customers` a, `BUSCASE1.orders` b
WHERE a.customer_id = b.customer_id
```

## Results/Output:

### Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		CH/
Row	City_Nos	State_Nos	Customer_Nos	Order_Nos		
1	4119	27	99441	99441		

## Observations/Remark:

- The results represent the summary of the number of cities and states in Brazil who ordered during the period.
- From the results, it is observed that the customer and order numbers are equal that means only one item has been placed from one customer\_id. To see the customer wise distribution following code will be used:

## Query/Code:

```
SELECT b.customer_city, b.customer_state, COUNT(a.order_id) as orders
FROM `BUSCASE1.orders` a, `BUSCASE1.customers` b
WHERE a.customer_id = b.customer_id
GROUP BY 1, 2
ORDER BY 3 DESC
LIMIT 10;
(LIMIT 10 only to display the results as per the instruction)
```

## Results/Output:

Row	customer_city	customer_state	orders
1	sao paulo	SP	15540
2	rio de janeiro	RJ	6882
3	belo horizonte	MG	2773
4	brasilia	DF	2131
5	curitiba	PR	1521
6	campinas	SP	1444
7	porto alegre	RS	1379
8	salvador	BA	1245
9	guarulhos	SP	1189
10	sao bernardo do campo	SP	938

## Observations/Remark:

- Customer state does not have full names use only two digit code. However the same can be easily traced from the state code details<sup>1,2</sup> :
- About ~15% of the Total Orders(99441) are from only one city i.e. 'Sao Paulo' in 'Sao Paulo' (SP) State
- The Top 10 consist of 35.23% of Total Order.
- Under Top 10 Cities

<sup>1</sup> [https://en.wikipedia.org/wiki/Federative\\_units\\_of\\_Brazil](https://en.wikipedia.org/wiki/Federative_units_of_Brazil)

<sup>2</sup> [https://en.wikipedia.org/wiki/List\\_of\\_cities\\_in\\_Brazil\\_by\\_population](https://en.wikipedia.org/wiki/List_of_cities_in_Brazil_by_population)

- i) 'Sao Paulo' and 'Rio de Janerio' consist of 64% (Sao Paulo: ~44% + Rio de Janerio: ~20% ) contribution in top 10 ordered city.
- ii) 'Sao Paulo' has 2.25 times more orders than its nearest 2<sup>nd</sup> highest order city 'Rio de Janerio'
- iii) 55% of orders are from single state 'SP': Sao Paulo(State) . The order are from four cities 'sao paulo': 44%, campinas: 4%, guarulhos: 3%, sao bernardo do campo: 3%.
- e) From the public domain information, it is observed that the Sao Paulo state is the most populated state and also have the highest GDP in all the Brazilian States<sup>3</sup>.

## Q2. In-depth Exploration:

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

### Solution 2.1:

**Tables: 'orders'**

**Query/Code:**

```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS Years,
       EXTRACT(MONTH FROM order_purchase_timestamp) AS Months,
       COUNT(order_id) AS orders
FROM `BUSCASE1.orders`
GROUP BY 1,2
ORDER BY 1 desc,2 desc
LIMIT 10;
(LIMIT 10 only to display the results)
```

### Results/Output:

Query results				
JOB INFORMATION		RESULTS	JSON	EXECUTION I
Row	Years	Months	orders	
1	2018	10	4	
2	2018	9	16	
3	2018	8	6512	
4	2018	7	6292	
5	2018	6	6167	
6	2018	5	6873	
7	2018	4	6939	
8	2018	3	7211	
9	2018	2	6728	
10	2018	1	7269	

<sup>3</sup> [https://en.wikipedia.org/wiki/Federative\\_units\\_of\\_Brazil](https://en.wikipedia.org/wiki/Federative_units_of_Brazil)



### Observations/Remark:

Based on analysis of order count by using the Excel following are the observational

- There is an overall upward trend in e-commerce in Brazil in period 2016 to 2018.
- The count of purchases has shown an overall upward trend.
- From September 2016 to November 2017 this shows the interest of customers in the e-market platform.
- But if we look year wise trend during the ending duration the orders sustained in range of 6000-7000 during the Sept 2017 to August 2018 the with some fluctuations.
- The initial orders low because of the starting duration of the business. In ending the sudden decrease may be due to some technical issue.
- It's may be noted that the order count alone does not indicate the total growth of business.
- To gain a more accurate understanding, we should also consider revenue growth.

### Q2. In-depth Exploration:

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

#### Solution 2.2:

**Table(s): orders**

**Query/Code:**

```
SELECT EXTRACT (MONTH FROM order_purchase_timestamp) AS Months,
       COUNT (order_id) AS orders
FROM `BUSCASE1.orders`
GROUP BY 1
ORDER BY 1
```

## Results/Output:

### Query results

JOB INFORMATION		RESULTS		JSOI
row	Months	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
11		11		7544
12		12		5674

## Observations/Remark:

If we look at the summary of the total of Month-on-Month Orders (over total period) the most number of orders were placed during march to August. However, it does not give the yearly details to find the yearly details we can follow following approach.

## Query/Code:

```
WITH ONE AS
(SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS Years,
    EXTRACT(MONTH FROM order_purchase_timestamp) AS Months,
    COUNT(order_id) AS orders
FROM `BUSCASE1.orders`
GROUP BY 1,2
ORDER BY 1 desc,2 desc)

SELECT
    Months,
    SUM(IF(Years < 2017, orders,0)) as Y_2016,
    SUM(IF(Years = 2017, orders,0)) as Y_2017,
    SUM(IF(Years > 2017, orders,0)) as Y_2018,
    SUM(orders) as total_orders
FROM ONE
GROUP BY 1
ORDER BY 1;
```

## Results/Output:

Row	Months	Y_2016	Y_2017	Y_2018	totalOrders
1	1	0	800	7269	8069
2	2	0	1780	6728	8508
3	3	0	2682	7211	9893
4	4	0	2404	6939	9343
5	5	0	3700	6873	10573
6	6	0	3245	6167	9412
7	7	0	4026	6292	10318
8	8	0	4331	6512	10843
9	9	4	4285	16	4305
10	10	324	4631	4	4959
11	11	0	7544	0	7544
12	12	1	5673	0	5674

## Observations/Remark:

- The from Nov 2017 to August 2018, we can see a settled trend. The order frequency is on higher side from November 2017 to May 2018 during the other months.
- There is a growing trend from Jan 2017 to Jan 2018, with peak in Nov 2017 and in nearby months. However, the seasonal for the same can't be verified as the data for the same is not available.
- The Overall details seem to be insufficient for a comparative seasonal trend analysis. As initial 4 month from Sept 2016 to Feb 2017 has not sustained the Orders in the duration. Then, from March to July the orders have increased may be due to promotion of the platform.

## Q2. In-depth Exploration:

3. 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

**Solution 2.3:** Third Quarter of the day i.e. Afternoon :13-18 hrs

**Table(s):** orders

**Query/Code:**

```
SELECT
CASE
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 0 AND 5 THEN '1-Dawn'
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 6 AND 11 THEN '2-Morning'
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 12 AND 17 THEN '3-Afternoon'
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 18 AND 23 THEN '4-Night' END AS
hour,
COUNT(order_id) AS order_count
FROM
`BUSCASE1.orders`
GROUP BY 1
ORDER BY 2 DESC;
```

## Results/Output:

Row	hour	order_count
1	3-Afternoon	38361
2	4-Night	34100
3	2-Morning	22240
4	1-Dawn	4740

## Observations/Remark:

- The customers have placed most orders in afternoon to night period from 1:00 PM to 11:00 PM, specifically in the afternoon and night.
- This indicates that customers prefer to shop online when they are relaxed or after completing their daily activities.
- The analysis is based on the given timestamps data only.

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

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

### Solution 3.1:

**Table(s):** 'orders' and 'customers'

**Joint ID:** 'customer\_id'

### Query/Code:

```
SELECT a.customer_state,
       EXTRACT(MONTH FROM b.order_purchase_timestamp) AS Months,
       COUNT(order_id) AS order_count
FROM `BUSCASE1.customers` a, `BUSCASE1.orders` b
WHERE a.customer_id = b.customer_id
GROUP BY 1,2
ORDER BY 1,2
LIMIT 10;
(LIMIT 10 only to display the results as per the instruction)
```

## Results/Output:

Row	customer_state	Months	order_count
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



### Observations/Remark:

The Month on Month Analysis for each customer state has been prepared with Help of Excel is as under :

	Months	1	2	3	4	5	6	7	8	9	10	11	12	Grand Total
1	SP	3351	3357	4047	3967	4632	4104	4381	4982	1648	1908	3012	2357	41746
2	RJ	990	1176	1302	1172	1321	1128	1288	1307	612	725	1048	783	12852
3	MG	971	1063	1237	1061	1190	1080	1111	1177	511	600	943	691	11635
4	RS	427	473	569	488	559	526	565	599	279	276	422	283	5466
5	PR	443	460	504	500	524	478	523	556	183	225	378	271	5045
6	SC	345	316	362	351	379	321	356	365	157	189	303	193	3637
7	BA	264	273	340	318	368	307	405	323	170	170	250	192	3380
8	DF	151	196	207	183	208	220	243	232	97	104	168	131	2140
9	ES	159	186	182	188	228	204	206	200	93	104	170	113	2033
10	GO	164	176	199	177	226	184	192	213	88	117	157	127	2020
11	PE	113	146	153	154	174	140	210	170	76	87	126	103	1652
12	CE	99	101	126	143	136	121	140	130	77	74	108	81	1336
13	PA	82	83	109	107	75	92	96	104	41	58	70	58	975
14	MT	96	84	71	92	104	83	85	78	35	55	74	50	907
15	MA	66	67	77	73	65	59	79	70	42	52	56	41	747
16	MS	71	75	79	58	74	76	74	59	33	34	46	36	715
	Other States	277	276	329	311	310	289	364	278	163	181	213	164	3155
	Grand Total	8069	8508	9893	9343	10573	9412	10318	10843	4305	4959	7544	5674	99441

Observations on the data table:

- The above table insights into the customer purchase trends on a state-by-state basis.
- Top 10 states consist of ~90 % of total Orders placed.
- The São Paulo (SP) consistently has the highest number of orders in any given month and consist of ~42% of the total orders, followed by Rio de Janeiro (RJ) having ~13% and Minas Gerais (MG) with ~11% in total orders.
- These states consist on total share in order about ~66%
- Only 12 states are having the average order count above 100.

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

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

#### Solution 3.2:

**Table(s): 'customers'**

**Query/Code:**

```
SELECT customer_state,
COUNT(customer_id) AS customer_nos
FROM `BUSCASE1.customers`
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10;
(LIMIT 10 only to display the results as per the instruction)
```

## Results/Output:

Row	customer_state	customer_nos
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

## Observations/Remark:

As in solution to question number 1.3 is discussed that the orders and customer are same. And the same may also be verified through solution to above question number 3.1. Therefore, the table and analysis presented in respect of the solution 3.1 is also applicable to this question.

## 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).

You can use the "payment\_value" column in the payments table to get the cost of orders.

**Solution 4.1:** The increase is 136.98% in the cost of orders from 2017 to 2018 (include months between Jan to Aug only)

**Table(s):** 'orders' and 'payments'

**Joint ID:** 'order\_id'

**Query/Code:** The total change in % (sum of Jan to Aug of 2017 and 2018)

```
WITH ONE AS
(SELECT order_id,
    EXTRACT(YEAR FROM order_purchase_timestamp) AS Years,
    EXTRACT(MONTH FROM order_purchase_timestamp) AS Months
FROM `BUSCASE1.orders`
WHERE EXTRACT(YEAR FROM order_purchase_timestamp) IN (2017, 2018)
AND EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 AND 8
ORDER BY 3,2)

SELECT
ROUND(SUM(IF(a.Years = 2018 AND Months BETWEEN 1 AND 8, b.payment_value,null)),2) as Y_2018,
ROUND(SUM(IF(a.Years = 2017 AND Months BETWEEN 1 AND 8, b.payment_value,null)),2) as Y_2017,
ROUND((
    (SUM(IF(a.Years = 2018 AND a.Months BETWEEN 1 AND 8, b.payment_value,null))/
    SUM(IF(a.Years = 2017 AND a.Months BETWEEN 1 AND 8, b.payment_value,null))) - 1 ) *
100,2) AS Percentage_change
FROM ONE a
INNER JOIN `BUSCASE1.payments` b
ON a.order_id=b.order_id
```

**Results/Output:** The total change in % (sum of Jan to Aug of 2017 and 2018)

### Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION I
Row	Y_2018	Y_2017	Percentage_change	
1	8694733.84	3669022.12	136.98	

### Observations/Remark:

In Year 2018 the change in order placed is ~137% over the previous year i.e. Year 2017 for the given period. However, to get better insights we may go for the month-on-month change analysis. This can be obtained from following code.

**Query/Code:** The change in % (Jan to Aug of 2017 and 2018 month wise)

```
WITH ONE AS
(SELECT order_id,
       EXTRACT(YEAR FROM order_purchase_timestamp) AS Years,
       EXTRACT(MONTH FROM order_purchase_timestamp) AS Months
FROM `BUSCASE1.orders`
WHERE EXTRACT(YEAR FROM order_purchase_timestamp) IN (2017, 2018)
AND EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 AND 8
ORDER BY 3,2)

SELECT a.Months,
ROUND(SUM(IF(a.Years = 2018 AND Months BETWEEN 1 AND 8, b.payment_value,null)),2) as Y_2018,
ROUND(SUM(IF(a.Years = 2017 AND Months BETWEEN 1 AND 8, b.payment_value,null)),2) as Y_2017,
ROUND((
    (SUM(IF(a.Years = 2018 AND a.Months BETWEEN 1 AND 8, b.payment_value,null))/
    SUM(IF(a.Years = 2017 AND a.Months BETWEEN 1 AND 8, b.payment_value,null))) - 1 )*
100,2) AS Percentage_change
FROM ONE a
INNER JOIN `BUSCASE1.payments` b
ON a.order_id=b.order_id
GROUP BY 1
ORDER BY 1
```

**Results/Output:** The change in % (Jan to Aug of 2017 and 2018 month wise)

### Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CH.
Row	Months	Y_2018	Y_2017	Percentage_change	
1	1	1115004.18	138488.04	705.13	
2	2	992463.34	291908.01	239.99	
3	3	1159652.12	449863.6	157.78	
4	4	1160785.48	417788.03	177.84	
5	5	1153982.15	592918.82	94.63	
6	6	1023880.5	511276.38	100.26	
7	7	1066540.75	592382.92	80.04	
8	8	1022425.32	674396.32	51.61	

### Observations/Remark:

The observation from month-wise analysis is as under:

- During 5 months the order change is more than 100%
- January shows the highest percentage increase 705.13%, followed by February ~240 % and April ~177 %.

**Q4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.**

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

### Solution 4.2:

**Table(s):** 'customers', 'orders' and 'order\_item'

**Joint ID:** 'customer\_id' and 'order\_id'

### Query/Code:

```
SELECT a.customer_state,
       ROUND(AVG(c.price), 2) AS avg_value_of_price,
       ROUND(SUM(c.price), 2) AS total_value_of_price,
FROM `BUSCASE1.customers` a
JOIN `BUSCASE1.orders` b
ON a.customer_id = b.customer_id
JOIN `BUSCASE1.order_items` c
ON b.order_id = c.order_id
GROUP BY 1
ORDER BY 3 desc, 2 desc
LIMIT 10;
(LIMIT 10 only to display the results as per the instruction)
```

## Results/Output:

### Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	customer_state	avg_value_of_price	total_value_of_price	
1	SP	109.65	5202955.05	
2	RJ	125.12	1824092.67	
3	MG	120.75	1585308.03	
4	RS	120.34	750304.02	
5	PR	119.0	683083.76	
6	SC	124.65	520553.34	
7	BA	134.6	511349.99	
8	DF	125.77	302603.94	
9	GO	126.27	294591.95	
10	ES	121.91	275037.31	

## Observations/Remark:

- The lowest average value/ cost of orders is from São Paulo (109.65) followed by Paraná (119), Rio Grande do Sul (120.34).
- The highest average value/cost of orders is from Paraíba (191.48) followed by Alagoas (180.89), Acre (173.73).
- The highest price sum and orders are from the same is from São Paulo , followed by Rio de Janeiro, and Minas Gerais.

## Q4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

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

## Solution 4.3:

**Table(s):** 'customers', 'orders' and 'order\_item'

**Joint ID:** 'customer\_id' and 'order\_id'

## Query/Code:

```
SELECT a.customer_state,
       ROUND(AVG(c.freight_value), 2) AS avg_freight_value,
       ROUND(SUM(c.freight_value), 2) AS total_freight_value,
FROM `BUSCASE1.customers` a
JOIN `BUSCASE1.orders` b
ON a.customer_id = b.customer_id
JOIN `BUSCASE1.order_items` c
ON b.order_id = c.order_id
GROUP BY 1
ORDER BY 3 desc, 2 desc
LIMIT 10;
(LIMIT 10 only to display the results as per the instruction)
```

## Results/Output:

### Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	customer_state	avg_freight_value	total_freight_value	
1	SP	15.15	718723.07	
2	RJ	20.96	305589.31	
3	MG	20.63	270853.46	
4	RS	21.74	135522.74	
5	PR	20.53	117851.68	
6	BA	26.36	100156.68	
7	SC	21.47	89660.26	
8	PE	32.92	59449.66	
9	GO	22.77	53114.98	
10	DF	21.04	50625.5	

### Observations/Remark:

- The highest average freight value to deliver the orders in Roraima(42.98) followed by Paraíba (42.72) and Rondônia (41.07). the high cost is due to distance from populated places and terrain.
- The lowest average freight value to deliver the orders in São Paulo (15.15), followed by Paraná (20.53), and Minas Gerais(20.73). This shows that the order frequency has a significant impact on the avg freight value.

Note: Definition of freight rate is a price requested for the transport of cargo from one place to another.

### 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. 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:

$\text{time\_to\_deliver} = \text{order\_delivered\_customer\_date} - \text{order\_purchase\_timestamp}$

$\text{diff\_estimated\_delivery} = \text{order\_estimated\_delivery\_date} - \text{order\_delivered\_customer\_date}$

### Solution 5.1:

#### Table(s): orders

The initial query giving null values. To analyse it looked orders table (from priveview) more closely found that column states contains status delivered which is required for analysis.

order_status	order_purchase_timestamp	order_approved_at	order_delivered_carrier_date	order_delivered_customer_date	order_estimated_delivery_date
created	2017-11-25 11:10:33 UTC	null	null	null	2017-12-12 00:00:00 UTC
created	2017-12-05 01:07:58 UTC	null	null	null	2018-01-08 00:00:00 UTC
created	2017-12-05 01:07:52 UTC	null	null	null	2018-01-11 00:00:00 UTC
created	2018-02-09 17:21:04 UTC	null	null	null	2018-03-07 00:00:00 UTC
created	2017-11-06 13:12:34 UTC	null	null	null	2017-12-01 00:00:00 UTC
shipped	2017-04-20 12:45:34 UTC	2017-04-22 09:10:13 UTC	2017-04-24 11:31:17 UTC	null	2017-05-18 00:00:00 UTC
shipped	2017-07-13 11:03:05 UTC	2017-07-13 11:10:22 UTC	2017-07-18 18:17:30 UTC	null	2017-08-14 00:00:00 UTC
shipped	2017-07-11 13:36:30 UTC	2017-07-11 13:45:15 UTC	2017-07-13 17:55:46 UTC	null	2017-08-14 00:00:00 UTC
shipped	2017-07-29 18:05:07 UTC	2017-07-29 18:15:17 UTC	2017-07-31 16:41:59 UTC	null	2017-08-14 00:00:00 UTC
shipped	2017-07-13 10:02:47 UTC	2017-07-14 02:25:54 UTC	2017-07-20 20:02:58 UTC	null	2017-08-14 00:00:00 UTC
shipped	2017-07-19 12:44:59 UTC	2017-07-19 12:55:22 UTC	2017-07-20 14:38:54 UTC	null	2017-08-14 00:00:00 UTC
shipped	2018-05-11 18:24:01 UTC	2018-05-11 18:35:20 UTC	2018-05-14 15:49:00 UTC	null	2018-06-06 00:00:00 UTC
shipped	2018-05-20 18:58:04 UTC	2018-05-20 19:10:36 UTC	2018-05-24 06:53:00 UTC	null	2018-06-06 00:00:00 UTC
shipped	2018-05-03 21:09:38 UTC	2018-05-03 21:33:08 UTC	2018-05-04 14:34:00 UTC	null	2018-06-06 00:00:00 UTC

### Query/Code:

```
SELECT order_status, count(order_id)
FROM `BUSCASE1.orders`
GROUP BY 1
```

### Results/Output:

Query results			
JOB INFORMATION		RESULTS	JSON
Row	order_status	No_of_count	
1	created	5	
2	shipped	1107	
3	approved	2	
4	canceled	625	
5	invoiced	314	
6	delivered	96478	
7	processing	301	
8	unavailable	609	

So, to find the requisite details we need to use the delivered status. The following will give requisite result

### Query/Code:

```
SELECT order_id,
DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp, Day) AS
time_to_deliver,
DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, Day) AS
diff_estimated_delivery
FROM `BUSCASE1.orders`
WHERE order_status ='delivered'
ORDER BY 2 DESC,3 DESC
LIMIT 10
(LIMIT 10 only to display the results as per the instruction)
```

## Results/Output:

### Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	order_id	time_to_deliver	diff_estimated_deliver	
1	ca07593549f1816d26a572e06...	209	-181	
2	1b3190b2dfa9d789e1f14c05b...	208	-188	
3	440d0d17af552815d15a9e41a...	195	-165	
4	2fb597c2f772eca01b1f5c561b...	194	-155	
5	0f4519c5f1c541ddec9f21b3bd...	194	-161	
6	285ab9426d6982034523a855f...	194	-166	
7	47b40429ed8cce3aee9199792...	191	-175	
8	2fe324febf907e3ea3f2aa9650...	189	-167	
9	2d7561026d542c8dbd8f0daea...	188	-159	
10	437222e3fd1b07396f1d9ba8c...	187	-144	

## Observations/Remark:

The order wise average need to be analysed with customer and states, which is presented in next few solutions.

## Q5. Analysis based on sales, freight and delivery time.

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

### Solution 5.2:

**Table(s):** 'customers', 'orders' and 'order\_item'

**Joint ID:** 'customer\_id' and 'order\_id'

### Query/Code:

```
WITH ONE AS
(
  SELECT a.customer_state,
         ROUND(AVG(c.freight_value), 2) AS avg_freight_value
  FROM `BUSCASE1.customers` a
  INNER JOIN `BUSCASE1.orders` b
  ON a.customer_id = b.customer_id
  INNER JOIN `BUSCASE1.order_items` c
  ON b.order_id = c.order_id
  GROUP BY 1),
TWO AS
(
  SELECT MAX('Highest') OVER() AS TOP_5,
         DENSE_RANK() OVER(ORDER BY avg_freight_value DESC) as top_rank,
         customer_state, avg_freight_value
  FROM ONE
  ORDER BY top_rank
  LIMIT 5),
THREE AS
```



```

(SELECT MAX('Lowest')OVER() AS TOP_5,
 DENSE_RANK()OVER(ORDER BY avg_freight_value) as top_rank,
 customer_state, avg_freight_value
FROM ONE
ORDER BY top_rank
LIMIT 5)

SELECT * FROM TWO
UNION DISTINCT
SELECT * FROM THREE

```

## Results/Output:

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	TOP_5	top_rank	customer_state	avg_freight_value
1	Highest	1	RR	42.98
2	Highest	2	PB	42.72
3	Highest	3	RO	41.07
4	Highest	4	AC	40.07
5	Highest	5	PI	39.15
6	Lowest	1	SP	15.15
7	Lowest	2	PR	20.53
8	Lowest	3	MG	20.63
9	Lowest	4	RJ	20.96
10	Lowest	5	DF	21.04

## Observations/Remark:

- The 5 highest average freight value to deliver the orders in Roraima, Paraíba, Rondônia, Acre and Piauí. The high cost is due to distance from populated places and terrain.
- The 5 lowest average freight value to deliver the orders in São Paulo, Paraná, Minas Gerais, Rio de Janeiro and Distrito Federal. This shows that the order frequency has a significant impact on the avg freight value.

## Q5. Analysis based on sales, freight and delivery time.

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

### Solution 5.3:

**Table(s):** 'customers' and 'orders'

**Joint ID:** 'customer\_id'

### Query/Code:

```

WITH ONE AS
(SELECT a.customer_state,

```

```

ROUND(AVG(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp, Day)),2) AS
Avg_delivery_time
FROM `BUSCASE1.customers` a
INNER JOIN `BUSCASE1.orders` b
ON a.customer_id = b.customer_id
GROUP BY 1),
TWO AS
(SELECT MAX('Highest')OVER() AS TOP_5,
DENSE_RANK()OVER(ORDER BY Avg_delivery_time DESC) as top_rank,
customer_state, Avg_delivery_time
FROM ONE
ORDER BY top_rank
LIMIT 5),
THREE AS
(SELECT MAX('Lowest')OVER() AS TOP_5,
DENSE_RANK()OVER(ORDER BY Avg_delivery_time) as top_rank,
customer_state, Avg_delivery_time
FROM ONE
ORDER BY top_rank
LIMIT 5)

SELECT * FROM TWO
UNION DISTINCT
SELECT * FROM THREE

```

## Results/Output:

Query results						
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		CHART
row	TOP_5	top_rank	customer_state	Avg_delivery_time		
1	Highest	1	RR	28.98		
2	Highest	2	AP	26.73		
3	Highest	3	AM	25.99		
4	Highest	4	AL	24.04		
5	Highest	5	PA	23.32		
6	Lowest	1	SP	8.3		
7	Lowest	2	PR	11.53		
8	Lowest	3	MG	11.54		
9	Lowest	4	DF	12.51		
10	Lowest	5	SC	14.48		

### Observations/Remark:

- a) The 5 highest average deliver time states are Roraima, Amapá, Amazonas, Alagoas and Pará.
- b) The 5 lowest average deliver time states are São Paulo, Paraná, Minas Gerais, Distrito Federal and Santa Catarina.

### Q5. Analysis based on sales, freight and delivery time.

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.

#### Solution 5.4:

**Table(s):** 'customers' and 'orders'

**Joint ID:** 'customer\_id'

#### Query/Code:

```
SELECT a.customer_state,
       ROUND(AVG(DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date,
Day)),2) AS delivery_time_diff
FROM `BUSCASE1.customers` a
INNER JOIN `BUSCASE1.orders` b
ON a.customer_id = b.customer_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5
```

#### Results/Output:

##### Query results

JOB INFORMATION		RESULTS	JSON	EX
Row	customer_state	delivery_time_diff		
1	AC	19.76		
2	RO	19.13		
3	AP	18.73		
4	AM	18.61		
5	RR	16.41		

### Observations/Remark:

- a) The estimated delivery time and actual delivery time for top 5 States are Acre, Rondônia, Amapá, Amazonas and Roraima.

- b) However, it may be noted that the fastest delivery is only terms of how early the order is being delivered from its estimated delivery date at the time of Order.

#### Q6. Analysis based on the payments:

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

##### Solution 6.1:

**Table(s):** 'orders' and 'payments'

**Joint ID:** 'order\_id'

**Query/Code:**

```
SELECT EXTRACT(MONTH FROM a.order_purchase_timestamp) AS Months,
       b.payment_type,
       COUNT(a.order_id) AS Orders
FROM `BUSCASE1.orders` a
INNER JOIN `BUSCASE1.payments` b
ON a.order_id = b.order_id
GROUP BY 1, 2
ORDER BY 1, 2
LIMIT 10;
(LIMIT 10 only to display the results as per the instruction)
```

##### Results/Output:

###### Query results

JOB INFORMATION		RESULTS		JSON	EXECUTION DETAILS	
Row	Months		payment_type		Orders	
1	1	UPI	1715			
2	1	credit_card	6103			
3	1	debit_card	118			
4	1	voucher	477			
5	2	UPI	1723			
6	2	credit_card	6609			
7	2	debit_card	82			
8	2	voucher	424			
9	3	UPI	1942			
10	3	credit_card	7707			

### Observations/Remark:

Following data table is prepared by help of Excel for Orders placed.

	Payment Type					
Month	credit_card	debit_card	not_defined	UPI	voucher	Grand Total
1	6103	118		1715	477	8413
2	6609	82		1723	424	8838
3	7707	109		1942	591	10349
4	7301	124		1783	572	9780
5	8350	81		2035	613	11079
6	7276	209		1807	563	9855
7	7841	264		2074	645	10824
8	8269	311	2	2077	589	11248
9	3286	43	1	903	302	4535
10	3778	54		1056	318	5206
11	5897	70		1509	387	7863
12	4378	64		1160	294	5896
Grand Total	76795	1529	3	19784	5775	103886

- The analysis shows an overall uptrend from January to August and another uptrend from September to November.
- Credit card transactions are the most popular payment method about 74% , followed by UPI 19 % .
- Debit card transactions are the least preferred option 1%.
- 5% - 7% of the orders are placed with use of the vouchers.
- The credit card transactions are increasing compared to other payment methods, due to benefits like “buy now, pay later” options or cashback offers.

### Q6. Analysis based on the payments:

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

**Solution 6.2:** In aforementioned Solution 5.1, we have observed that the there are different status of the orders including status cancelled, so the cancelled orders may not have the payment and payment installments so here the condition cancelled not included in the approach.

**Table(s):** 'orders' and 'payments'

**Joint ID:** 'order\_id'

**Query/Code:**

```
SELECT a.payment_installments, COUNT(b.order_id) AS Orders
FROM `BUSCASE1.payments` a
INNER JOIN `BUSCASE1.orders` b
ON a.order_id = b.order_id
WHERE b.order_status != 'canceled'
GROUP BY 1
ORDER BY 1,2 DESC
LIMIT 10;
(LIMIT 10 only to display the results as per the instruction)
```

## Results/Output:

### Query results

JOB INFORMATION		RESULTS	JSON
Row	payment_installment	Orders	
1	0	2	
2	1	52184	
3	2	12353	
4	3	10392	
5	4	7056	
6	5	5209	
7	6	3898	
8	7	1620	
9	8	4239	
10	9	638	

### Observations/Remark:

- The majority of orders (maximum count) have only one payment installment about 51%.
- The 99% of the orders are having 10 or less installments
- The highest number of installments is 24 for 18 orders.
- However, this has to be looked with the order value instead of order numbers/ revenue for the deeper insight.

## Actionable Insights & Recommendations

The Observations are mentioned below each solution. Based on that the key actionable insights & Recommendations are as under:

- One customer has only placed one order. Need to find reasons for this trend on the platform.
- To boost sales, it is suggested to focus on customer retention strategies, such as marketing campaigns, promotional programs to show trustable customer service experiences.
- 10 states out of 27 consist of ~90 % of total Orders placed and São Paulo(SP) has highest Orders than the next five states combined. Indicates scope for improvement in the other states.
- The orders sustained in range of 6000-7000 during the Sept 2017 to August 2018 with some fluctuations. E-commerce businesses can plan their marketing and sales strategies accordingly. Offering discounts or promotions during off-peak seasons can encourage customers to make purchases during these times, resulting in increased sales.

- v) Imprudent in delivery times will have a positive impact on customer satisfaction and encourage repeat purchases. Streamlining logistics can be planned to achieve this. Improve logistics & shipping processes and optimise freight fee to shorten delivery times, competitiveness and enhance customer satisfaction.
- vi) As already mention that One customer (customer\_id) has only placed one order. This shows the platform needs to be improved for better experience. Investment in technology can enhance the e-commerce experience.
- vii) Collaboration & regular interaction with sellers is required to expand product offerings and to provide a diverse range of product to customers as per their needs and preferences.
- viii) The Credit card and UPI are the most popular payment systems and only 5% - 7% of the orders are placed with use of the vouchers, So credit card and UPI banks/ companies may be approach for no cost EMIs or better offer on purchasing.
- ix) Companies are suggested to provide quick Customer service & interaction experience by using effective and quick response chat bots and chat support services for customer inquiries.
- x) As the data is insufficient for better insights customer survey in high purchasing cities may be conducted in São Paulo and Rio de Jenario.