

TARGET BUSINESS CASE STUDY

1. Basic Analysis

1. Time period for which the data is given

Query:

```
SELECT
  MIN(order_purchase_timestamp) AS order_start_date,
  MAX(order_purchase_timestamp) AS order_end_date
FROM
  `scaler_project1.orders`
Result:
```

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

2. Cities and States of customers ordered during the given period

Query:

```
SELECT
  DISTINCT customer_city,
  customer_state
FROM
  `scaler_project1.customers` c
JOIN
  `scaler_project1.orders` o
ON
  o.customer_id=c.customer_id
```

Result:

Query results				SAVE RESULTS	EXPLORE DATA	
JOB INFORMATION				RESULTS	JSON	EXECUTION DETAILS
EXECUTION GRAPH				PREVIEW		
Row	customer_city	customer_state				
1	acu	RN				
2	ico	CE				
3	ipe	RS				
4	ipu	CE				
5	ita	SC				
6	itu	SP				

2. In-depth Exploration:

1. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

Query:

```
with cte as (select distinct
extract(month from order_purchase_timestamp) as mon, extract(year from
order_purchase_timestamp) as year,
count(order_id) over(partition by extract(month from
order_purchase_timestamp),extract(year from order_purchase_timestamp))
as count_order from `scaler_project1.orders`
)
select * from cte
order by year,count_order desc
limit 10
```

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	mon	year	count_order			
4	11	2017	7544			
5	12	2017	5673			
6	10	2017	4631			
7	8	2017	4331			
8	9	2017	4285			
9	7	2017	4026			
10	5	2017	3700			

2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

Query:

```
WITH
cte AS(
SELECT
CASE
WHEN EXTRACT(hour FROM order_purchase_timestamp)<12 THEN 'morni
ng'
WHEN EXTRACT(hour
FROM
order_purchase_timestamp)<17 THEN 'afternoon'
ELSE
'night'
END
AS day_time
FROM
`scaler_project1.orders` )
SELECT
DISTINCT *,
COUNT(*)OVER(PARTITION BY day_time)
FROM
cte
```

3. Evolution of E-commerce orders in the Brazil region:
 1. Get month on month orders by states

Query:

```
WITH
cte AS(
SELECT
  DISTINCT c.customer_state,
  FORMAT_DATETIME("%B", DATETIME(order_purchase_timestamp)) AS month
h,
COUNT(*) AS orders_count
FROM
  `scaler_project1.orders` o
JOIN
  `scaler_project1.customers` c
ON
  c.customer_id=o.customer_id
GROUP BY
  c.customer_state,
  FORMAT_DATETIME("%B", DATETIME(order_purchase_timestamp)) )
SELECT
  *
FROM
  cte
ORDER BY
  orders_count DESC,
  customer_state,
  month
```

Result:

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	customer_state	month	orders_count			
1	SP	August	4982			
2	SP	May	4632			
3	SP	July	4381			
4	SP	June	4104			
5	SP	March	4047			
6	SP	April	3967			
7	SP	February	3357			
8	SP	January	3351			
9	SP	November	3012			
10	SP	December	2357			

2. Distribution of customers across the states in Brazil

Query:

```
WITH
cte AS(
SELECT
  DISTINCT c.customer_state,
```

```

COUNT(c.customer_id) OVER(PARTITION BY c.customer_state) AS custo
mer_count
FROM
`scaler_project1.orders` o
JOIN
`scaler_project1.customers` c
ON
c.customer_id=o.customer_id)
SELECT
*
FROM
cte
ORDER BY
customer_count DESC,
customer_state
LIMIT
10

```

Result:

Row	customer_state	customer_count
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

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

1. Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use "payment_value" column in payments table

Query:

```

WITH
cte AS(
SELECT
CASE
WHEN EXTRACT(month FROM order_purchase_timestamp)<9 THEN EXTRAC
T(year FROM order_purchase_timestamp)
END
AS year,
payment_value
FROM
`scaler_project1.orders` o
JOIN
`scaler_project1.payments` p

```

```

ON
    o.order_id=p.order_id),
cte2 AS(
SELECT
    DISTINCT year,
    SUM(payment_value) OVER(PARTITION BY year) AS sell_amount
FROM
    cte
WHERE
    year=2017
    OR year=2018)
SELECT
    DISTINCT CONCAT(ROUND(((
        SELECT
            sell_amount
        FROM
            cte2
        WHERE
            year=2018)-(
        SELECT
            sell_amount
        FROM
            cte2
        WHERE
            year=2017)))/(
        SELECT
            sell_amount
        FROM
            cte2
        WHERE
            year=2017)*100,0),'%')
FROM
    cte2

```

Results:

Row	fit_
1	137%

2. Mean & Sum of price and freight value by customer state

Query:

```

SELECT
    DISTINCT customer_state,
    COUNT(o.order_id) OVER(PARTITION BY customer_state) AS state_order_count,
    ROUND(AVG(payment_value) OVER(PARTITION BY customer_state),2) AS state_mean,
    ROUND(SUM(payment_value) OVER(PARTITION BY customer_state),2) AS state_sum_price,
    ROUND(SUM(oi.freight_value) OVER(PARTITION BY customer_state),2) AS state_freight_value
FROM
    `scaler_project1.customers` c
JOIN
    `scaler_project1.orders` o

```

```

ON
    o.customer_id=c.customer_id
JOIN
    `scaler_project1.payments` p
ON
    p.order_id=o.order_id
JOIN
    `scaler_project1.order_items` oi
ON
    oi.order_id=o.order_id

```

Result:

Row	customer_state	state_order_count	state_mean	state_sum_price	state_freight_value
1	MA	844	235.27	198566.27	32290.33
2	ES	2338	173.57	405805.34	51392.57
3	BA	4048	196.99	797410.36	106538.62
4	AM	171	203.24	34753.3	5656.54
5	MS	843	194.94	164337.28	19739.44
6	PA	1116	234.58	261788.35	39881.05
7	PE	1889	199.25	376377.27	61923.56
8	CE	1551	221.69	343847.83	50350.54
9	RS	6486	176.89	1147277.0	141579.69

5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

Query:

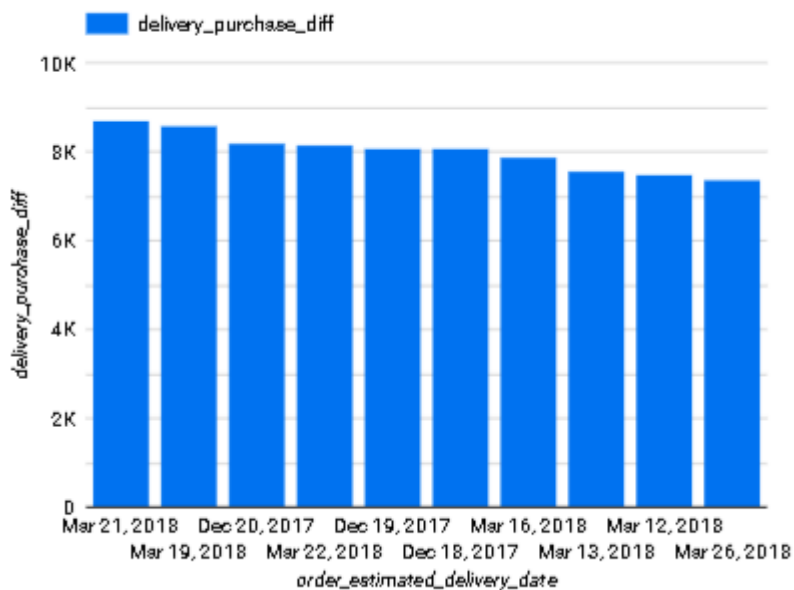
```

SELECT
    DATE(order_purchase_timestamp) AS order_purchase_date,
    DATE(order_delivered_customer_date) AS order_delivered_date,
    DATE(order_estimated_delivery_date) AS order_estimated_delivery_date,
    DATE_DIFF(DATE(order_delivered_customer_date),DATE(order_purchase_timestamp),day) AS delivery_purchase_diff,
    DATE_DIFF(DATE(order_estimated_delivery_date),DATE(order_purchase_timestamp),day) AS estimated_purchase_diff
FROM
    `scaler_project1.orders`
WHERE
    order_status= 'delivered'

```

Results:

Row	order_purchase	order_delivered	order_estimated	delivery_purchase	estimated_purchase
1	2017-03-17	2017-04-07	2017-05-18	21	62
2	2017-03-20	2017-03-30	2017-05-18	10	59
3	2017-03-21	2017-04-18	2017-05-18	28	58
4	2018-08-20	2018-08-29	2018-10-04	9	45
5	2018-08-12	2018-08-23	2018-10-04	11	53
6	2018-08-16	2018-08-23	2018-10-04	7	49
7	2018-08-22	2018-08-29	2018-10-04	7	43
8	2018-08-20	2018-08-29	2018-10-04	9	45
9	2018-08-09	2018-08-22	2018-10-04	13	56
10	2018-08-13	2018-08-29	2018-10-04	16	52
11	2018-08-20	2018-08-30	2018-10-04	10	45



2. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

- time_to_delivery = order_purchase_timestamp - order_delivered_customer_date
- diff_estimated_delivery = order_estimated_delivery_date - order_delivered_customer_date

Query:

```
SELECT
    DATE(order_purchase_timestamp) AS order_purchase_date,
    DATE(order_delivered_customer_date) AS order_delivered_date,
```

```

DATE(order_estimated_delivery_date) AS order_estimated_delivery_date,
DATE_DIFF(DATE(order_delivered_customer_date),DATE(order_purchase_timestamp),day) AS time_to_delivery,
DATE_DIFF(DATE(order_estimated_delivery_date),DATE(order_delivered_customer_date),day ) AS diff_estimated_delivery
FROM
`scaler_project1.orders`
WHERE
order_status= 'delivered'

```

Row	order_purchase	order_delivered	order_estimated	delivery_purchase	estimated_purchase
1	2017-03-17	2017-04-07	2017-05-18	21	62
2	2017-03-20	2017-03-30	2017-05-18	10	59
3	2017-03-21	2017-04-18	2017-05-18	28	58
4	2018-08-20	2018-08-29	2018-10-04	9	45
5	2018-08-12	2018-08-23	2018-10-04	11	53
6	2018-08-16	2018-08-23	2018-10-04	7	49
7	2018-08-22	2018-08-29	2018-10-04	7	43
8	2018-08-20	2018-08-29	2018-10-04	9	45
9	2018-08-09	2018-08-22	2018-10-04	13	56
10	2018-08-13	2018-08-29	2018-10-04	16	52
11	2018-08-20	2018-08-30	2018-10-04	10	45

- Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```

SELECT
customer_state,
ROUND(AVG(DATE_DIFF(DATE(order_delivered_customer_date),DATE(order_purchase_timestamp),day)),2) AS mean_time_to_delivery,
ROUND(AVG(DATE_DIFF(DATE(order_estimated_delivery_date),DATE(order_delivered_customer_date),day)),2) AS mean_diff_estimated_delivery,
ROUND(AVG(freight_value),2) AS mean_freight_value
FROM
`scaler_project1.orders` o
JOIN
`scaler_project1.order_items` oi
ON
oi.order_id=o.order_id
JOIN
`scaler_project1.customers` c
ON
c.customer_id=o.customer_id
WHERE
o.order_status= 'delivered'
GROUP BY
customer_state

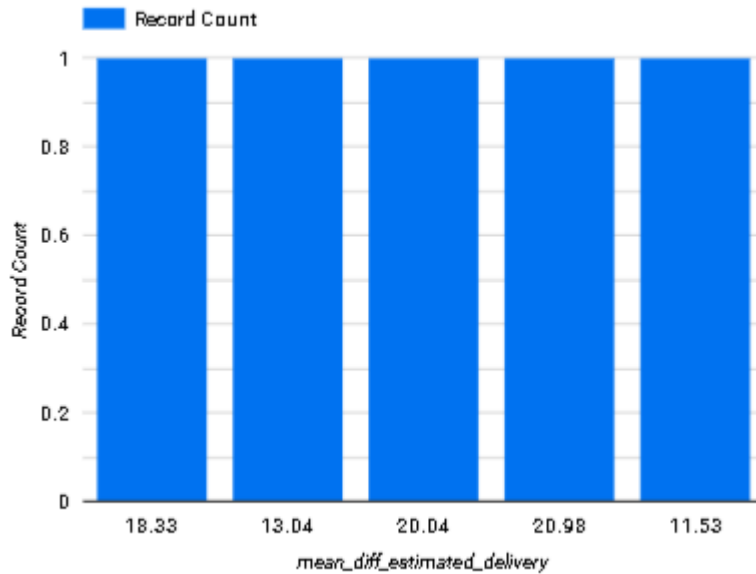
```


JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
row	customer_state	mean_time_to_delivery	mean_diff_estimated_delivery	mean_freight_value		
1	GO	15.34	12.29	22.56		
2	SP	8.66	11.21	15.12		
3	RS	15.13	14.13	21.61		
4	BA	19.19	10.98	26.49		
5	MG	11.92	13.34	20.63		
6	MT	17.91	14.57	28.0		
7	RJ	15.07	12.01	20.91		
8	SC	14.95	11.57	21.51		
9	SE	21.42	10.0	36.57		
10	PE	18.22	13.45	32.69		

4. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
Query:

```
WITH
  cte AS(
  SELECT
    customer_state,
    ROUND(AVG(DATE_DIFF(DATE(order_delivered_customer_date),DATE(order_purchase_timestamp),day)),2) AS mean_time_to_delivery,
    ROUND(AVG(DATE_DIFF(DATE(order_estimated_delivery_date),DATE(order_delivered_customer_date),day)),2) AS mean_diff_estimated_delivery,
    ROUND(AVG(freight_value),2) AS mean_freight_value
  FROM
    `scaler_project1.orders` o
  JOIN
    `scaler_project1.order_items` oi
  ON
    oi.order_id=o.order_id
  JOIN
    `scaler_project1.customers` c
  ON
    c.customer_id=o.customer_id
  WHERE
    o.order_status= 'delivered'
  GROUP BY
    customer_state)
SELECT
  *
FROM
  cte
ORDER BY
  mean_freight_value DESC
LIMIT
  5
```

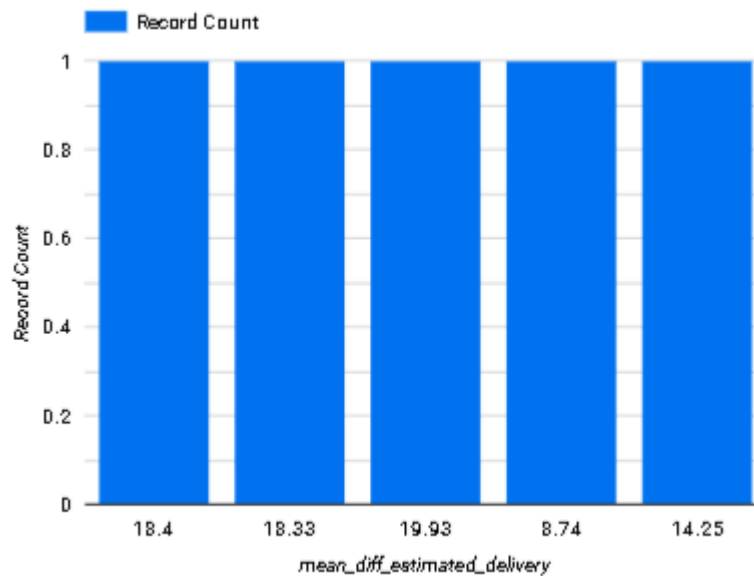
Row	customer_state	mean_time_to_delivery	mean_diff_estimated_delivery	mean_freight_value
1	RR	28.17	18.33	43.09
2	PB	20.55	13.04	43.09
3	RO	19.66	20.04	41.33
4	AC	20.68	20.98	40.05
5	PI	19.32	11.53	39.12



5. Top 5 states with highest/lowest average time to delivery
Query:

```
WITH
cte AS(
SELECT
customer_state,
ROUND(AVG(DATE_DIFF(DATE(order_delivered_customer_date),DATE(order_purchase_timestamp),day)),2) AS mean_time_to_delivery,
ROUND(AVG(DATE_DIFF(DATE(order_estimated_delivery_date),DATE(order_delivered_customer_date),day)),2) AS mean_diff_estimated_delivery,
ROUND(AVG(freight_value),2) AS mean_freight_value
FROM
`scaler_project1.orders` o
JOIN
`scaler_project1.order_items` oi
ON
oi.order_id=o.order_id
JOIN
`scaler_project1.customers` c
ON
c.customer_id=o.customer_id
WHERE
o.order_status= 'delivered'
GROUP BY
customer_state)
SELECT
*
FROM
cte
ORDER BY
mean_time_to_delivery DESC
LIMIT
5
```

Row	customer_state	mean_time_to_d	mean_diff_estim	mean_freight_va
1	AP	28.22	18.4	34.16
2	RR	28.17	18.33	43.09
3	AM	26.34	19.93	33.31
4	AL	24.45	8.74	35.87
5	PA	23.7	14.25	35.63



6. Top 5 states where delivery is really fast/ not so fast compared to estimated date

Query:

```
WITH
cte AS(
SELECT
customer_state,
ROUND(AVG(DATE_DIFF(DATE(order_delivered_customer_date),DATE(order_purchase_timestamp),day)),2) AS mean_time_to_delivery,
ROUND(AVG(DATE_DIFF(DATE(order_estimated_delivery_date),DATE(order_delivered_customer_date),day)),2) AS mean_diff_estimated_delivery,
ROUND(AVG(freight_value),2) AS mean_freight_value
FROM
`scaler_project1.orders` o
JOIN
`scaler_project1.order_items` oi
ON
oi.order_id=o.order_id
JOIN
`scaler_project1.customers` c
ON
c.customer_id=o.customer_id
```

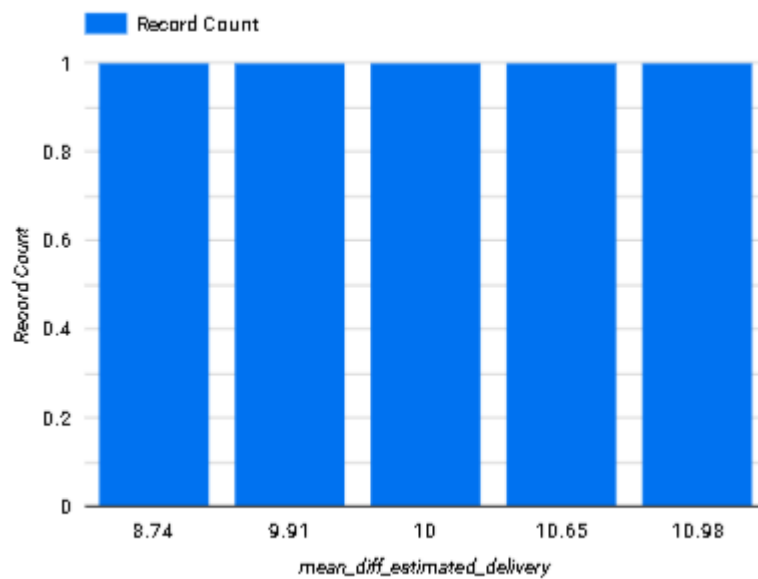
```

WHERE
    o.order_status= 'delivered'
GROUP BY
    customer_state)
SELECT
    *
FROM
    cte
ORDER BY
    mean_diff_estimated_delivery
LIMIT
    5

```

Result:

Row	customer_state	mean_time_to_d	mean_diff_estim	mean_freight_va
1	AL	24.45	8.74	35.87
2	MA	21.59	9.91	38.49
3	SE	21.42	10.0	36.57
4	ES	15.59	10.65	22.03
5	BA	19.19	10.98	26.49



6. Payment type analysis:

1. Month over Month count of orders for different payment types

Query:

```

WITH
    cte AS(
SELECT

```

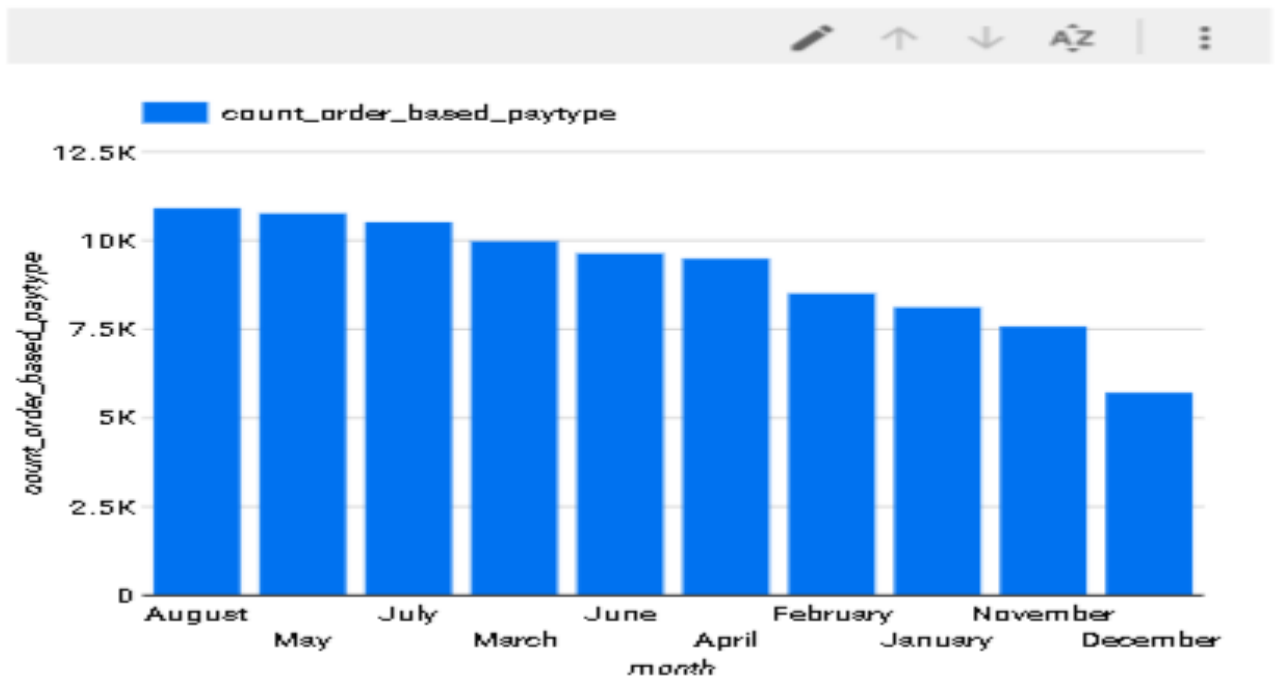
```

        DISTINCT FORMAT_DATETIME("%B", DATETIME(order_purchase_timestamp)) AS month,
        payment_type,
        COUNT(o.order_id)OVER(PARTITION BY EXTRACT(month FROM order_purchase_timestamp),
        payment_type) AS count_order_based_paytype
FROM
    `scaler_project1.orders` o
JOIN
    `scaler_project1.payments` p
ON
    o.order_id=p.order_id
WHERE
    o.order_status= 'delivered')
SELECT
    *
FROM
    cte

```

Results:

Row	month	payment_type	count_order_based_paytype
1	November	voucher	367
2	March	voucher	578
3	August	credit_card	8090
4	February	voucher	408
5	April	debit_card	119
6	December	credit_card	4246
7	June	UPI	1778
8	June	credit_card	7133
9	December	voucher	288
10	August	debit_card	303
11	January	credit_card	5910



2. Count of orders based on the no. of payment instalments

Query:

```
WITH
cte AS(
SELECT
    DISTINCT payment_installments,
    COUNT(o.order_id)OVER(PARTITION BY payment_installments) AS count_order_
based_payment_installments
FROM
    `scaler_project1.orders` o
JOIN
    `scaler_project1.payments` p
ON
    o.order_id=p.order_id
WHERE
    o.order_status= 'delivered')
SELECT
    *
FROM
    cte
ORDER BY
    count_order_based_payment_installments DESC
```

Results:

Row	payment_installments	count_order_based
1	1	50929
2	2	12075
3	3	10164
4	4	6891
5	10	5150
6	5	5095
7	8	4136
8	6	3804
9	7	1562

