

OVERVIEW: This particular business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews,By analyzing this extensive dataset, it becomes possible to gain valuable insights into Target's operations in Brazil

Problem Statement: To analyze the sales and operations of Target in Brazil and to extract valuable insights from it and provide actionable recommendations.

21`zQ1-Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

Q1.1-Data type of all columns in the "customers" table.

```
select
column_name, data_type
from "target_scaler".information_schema.COLUMNS
where table_name="customers"
```

Filter Enter property name or value

<input type="checkbox"/>	Field name	Type	Mode	Key	Collation	Default Va
<input type="checkbox"/>	customer_id	STRING	NULLABLE			
<input type="checkbox"/>	customer_unique_id	STRING	NULLABLE			
<input type="checkbox"/>	customer_zip_code_prefix	INTEGER	NULLABLE			
<input type="checkbox"/>	customer_city	STRING	NULLABLE			
<input type="checkbox"/>	customer_state	STRING	NULLABLE			

EDIT SCHEMA

VIEW ROW ACCESS POLICIES

Q1.2-Get the time range between which the orders were placed.

TARGET CASE STUDY

Search (/) for resources, docs, products, and more

Search

1

Set up billing to upgrade to the full BigQuery experience. [Learn more](#)

DISMISS

UPGRADE

*Untitled 2

*Untitled 3

Query results

SAVE RESULTS
 EXPLORE DATA

JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

EXECUTION GRAPH

PREVIEW

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

```
select distinct
customer_city, customer_state
from `target_scaler.customers`
```

Q 2 In-depth Exploration:

Q-2.1Is there a growing trend in the no. of orders placed over the past years? Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

```
select year, month, cnt
from
(select
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
count(order_id) as cnt
from `target_scaler.orders`
group by extract(year from order_purchase_timestamp),
          extract(month from order_purchase_timestamp)
) T
order by year, month;
```

Google Cloud TARGET CASE STUDY Search (/) for resources, docs, products, and more

SANDBOX Set up billing to upgrade to the full BigQuery experience. [Learn more](#) DISMISS UPGRADE

Explorer + ADD

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW Expand

Row	year	month	cnt
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
11	2017	8	4331
12	2017	9	4285

Results per page: 50 1 - 25 of 25

PERSONAL HISTORY PROJECT HISTORY REFRESH

INFERENCE:

-> So from the above insights it is very noticeable that the trend for the number of orders being placed is on rise beside the months of sept. , oct. , Dec , a considerable dip in orders is seen from oct to dec

RECOMMENDATION:

-> since the peoples of brazil are showing interest in e-commerce , target should try to diversify their products portfolio more and more, ensuring that the person of every age group coming over the website cannot go empty handed. apart from it they should move on with such tricks and ideas which ensures that count of product being ordered should always be more than the previous month, because it can be seen that the increase in orders is not that much consistent, they can move on by providing extra discount on every last sunday of the month and using social media to advertise that offers so that maximum amount of traffic can be catered , stock clearance sales can also be organised in the months in which the sales are lowest i.e in the last quarter of the year.

Q-2.2 During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

```
select count(distinct order_id) as order_count, classification_of_day
from
(select order_id,
CASE
when extract(hour from order_purchase_timestamp) >= 5 and extract(hour from
order_purchase_timestamp) <= 7 then "DAWN"
when extract(hour from order_purchase_timestamp) >= 7 and extract(hour from
order_purchase_timestamp) <= 12 then "MORNING"
when extract(hour from order_purchase_timestamp) >= 12 and extract(hour from
order_purchase_timestamp) <= 17 then "AFTERNOON"
ELSE "NIGHT"
END as classification_of_day
from `target_scaler.orders`)
group by classification_of_day
order by count(distinct order_id);
```

Query results				
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	order_count	classification_of_day		
1	1921	DAWN		
2	26502	MORNING		
3	32366	AFTERNOON		
4	38652	NIGHT		

INFERENCE:

-> it can be seen that night is the most preferred time for shopping in brazil followed by afternoon , morning and dawn

RECOMMENDATION:

-> since the brazilians visiting the e-commerce stores is more in the night we can try to encash this time to go for more and more advertising of the products we want to make live in the future, and try to gain feedback from the peoples of different age group regarding their needs and their recommendations about that product

Apart from it we have to keep a close eye upon the products being purchased at that time by the peoples of different age and ensuring their availability is must since the traffic is more we can try to understand the purchasing habits of the customers and implementing those insights to increase the overall sales throughout the day

Q-3 Evolution of E-commerce orders in the Brazil region:

Q-3.1 Get the month on month no. of orders placed in each state.

```
select
year, month, count(order_id)
from
(select
extract(month from o.order_purchase_timestamp) as month,
extract(year from o.order_purchase_timestamp) as year,
count(order_id) as cnt
from `target_scaler.orders` as o inner join `target_scaler.customers` as c
on o.customer_id = c.customer_id)
```

Q-3.2 How are the customers distributed across all the states?

```
select customer_state as state , count(customer_id) as No_of_customers
from `target_scaler.customers`
group by customer_state
order by No_of_customers desc
```

The screenshot shows the Google Cloud BigQuery interface. The 'Query results' tab is active, displaying a table with 12 rows of customer counts by state. The table has columns for 'Row', 'state', and 'No_of_customers'. The states are listed in descending order of customer count: SP (41746), RJ (12852), MG (11635), RS (5466), PR (5045), SC (3637), BA (3380), DF (2140), ES (2033), GO (2020), PE (1652), and CE (1336). The interface includes a sidebar with the Explorer view showing the project structure, a top navigation bar with search and settings, and a bottom status bar with history and refresh options.

Row	state	No_of_customers
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
11	PE	1652
12	CE	1336

INFERENCE:

-> From the above result it is clear that the state "SP" have the most number of customers

RECOMMENDATION:

-> I would firstly recommend that the target must conduct a detailed study on the state "SP" and try to understand the positive behavior of their customers in that state and what is compelling them to purchase from the target and what more improvements can be done in it, and those insights from the state SP can be tried to get adjusted in other states as well to boost the sales

Apart from it we should also analyze the age group of the customers buying the most number of products and ensure that the strategies and product portfolio that is compelling them for the purchase, if can be used in other groups as well

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

Q4.1 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.

```
with money_move_analysis as
(select p.payment_value,
extract(month from o.order_purchase_timestamp) as month,
extract(year from o.order_purchase_timestamp) as year
from `target_scaler.payments` as p inner join `target_scaler.orders` as o
on p.order_id = o.order_id),
monthly_purchase as
(select year, month, count(payment_value) as No_of_orders,
avg(payment_value) as cost from money_move_analysis
group by year, month
having month <= 8 and year between 2017 and 2018)
select
a.month, a.year, b.year, a.cost, b.cost, a.No_of_orders, b.No_of_orders,
((b.cost-a.cost)/a.cost)*100 as percentage_change
from monthly_purchase a inner join monthly_purchase b
on a.month = b.month and a.year <> b.year
order by a.year, a.month
limit 10
```

Query results

SAVE RESULTS

EXPLORE DATA

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		EXECUTION GRAPH		Expand	
Row		year	year_1	cost	cost_1	No_of_orders	No_of_orders_1	percentage_change	
1	1	2017	2018	162.9271058823...	147.4288218960...	850	7563	-9.51240366196...	
2	2	2017	2018	154.7762513255...	142.7593987341...	1886	6952	-7.76401578954...	
3	3	2017	2018	158.57017976736	154.3732854100...	2837	7512	-2.64671097901...	
4	4	2017	2018	162.5002061454...	161.0189318906...	2571	7209	-0.91155223117...	
5	5	2017	2018	150.3343864097...	161.7354099509...	3944	7135	7.583776282650...	
6	6	2017	2018	148.7998777648...	159.5077893752...	3436	6419	7.196183069029...	
7	7	2017	2018	137.2209682649...	163.9066774243...	4317	6507	19.44725321262...	
8	8	2017	2018	148.2189714285...	152.6463601074...	4550	6698	2.987059373204...	
9	1	2018	2017	147.4288218960...	162.9271058823...	7563	850	10.51238406911...	
10	2	2018	2017	142.7593987341...	154.7762513255...	6952	1886	8.417556180490...	

INFERENCE:

-> so from the above result it can be derived that there is a negative change in the cost of orders from JAN to APR, slightly showing a positive trend afterwards

RECOMMENDATION:

-> it can be seen that when the cost of orders decreased from 2017 to 2018 for the first 4 months, the sales boost and a drastic change in the number of orders being placed can be noticed. The same trend continues but not with that pace, so target can look after giving quarterly heavy discount sales. It will help in two ways, firstly in those sales they can clear off their obsolete stock or the one which is going to be out of market soon. Secondly, these quarterly sales will start making the customer loyal for target because we will be giving the product which will be value for money in those sales, ensuring that the quality of the product is not hampered.

Q4.2 Calculate the Total & Average value of order price for each state. And also Calculate the Total & Average value of order freight for each state.

```

select customer_state,
sum(price) as total_sum, avg(price) as avg_price,
sum(freight_value) as total_freight, avg(freight_value) as avg_freight
from `target_scaler.order_items` as o inner join `target_scaler.orders` as o1
on o.order_id = o1.order_id
inner join `target_scaler.customers` as c
on o1.customer_id = c.customer_id
group by customer_state

```


Google Cloud TARGET CASE STUDY Search (/) for resources, docs, products, and more

SANDBOX Set up billing to upgrade to the full BigQuery experience. [Learn more](#) DISMISS UPGRADE

Explorer + ADD |< Q3.2 X Q4.1 X Q *Untitled 3 X

Q Type to search

Viewing workspace resources. SHOW STARRED ONLY

- Project queries
 - Q 1.3
 - Q 2.1
 - Q 2.3
 - Q 3.2
 - Q 4.1
- target_scaler
 - customers
 - geolocation
 - order_items
 - order_reviews
 - orders
 - payments

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	customer_state	total_sum	avg_price	total_freight	avg_freight	
1	PB	115268.0799999...	191.4752159468...	25719.73000000...	42.72380398671...	
2	AL	80314.81	180.8892117117...	15914.58999999...	35.84367117117...	
3	AC	15982.94999999...	173.7277173913...	3686.749999999...	40.07336956521...	
4	RO	46140.64000000...	165.9735251798...	11417.37999999...	41.06971223021...	
5	PA	178947.8099999...	165.6924166666...	38699.30000000...	35.83268518518...	
6	AP	13474.29999999...	164.3207317073...	2788.500000000...	34.00609756097...	
7	PI	86914.08000000...	160.3580811808...	21218.20000000...	39.14797047970...	
8	TO	49621.74000000...	157.5293333333...	11732.68000000...	37.24660317460...	
9	RN	83034.98000000...	156.9659357277...	18860.10000000...	35.65236294896...	
10	CE	227254.7099999...	153.7582611637...	48351.58999999...	32.71420162381...	
11	SE	58920.85000000...	153.0411688311...	14111.46999999...	36.65316883116...	
12	RR	7829.429999999...	150.5659615384...	2235.19	42.98442307692...	

Results per page: 50 1 ~ 27 of 27 |< < > >|

PERSONAL HISTORY PROJECT HISTORY REFRESH

INFERENCE:

-> from the above result we get the information about the top 5 states in terms of order cost and its freight

RECOMMENDATION:

-> Since we know our top 5 best performing states and it shows that the customers are picking the products from the target in those states, we can move on with minute increase in overall price of the product in these states and using that increased margin to reduce the cost in the next five states , target can also think of creating a benchmark of the price and according to that benchmark the low performing states will be given discounts.

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

Q 5.1 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.

```
select
c.customer_state, avg(freight_value) as avg_freight,
```

```

avg(order_delivered_customer_date - order_purchase_timestamp) as delivery_time,
avg(order_estimated_delivery_date - order_delivered_customer_date) as
est_del_diff
from `target_scaler.customers` as c inner join `target_scaler.orders` as o
on c.customer_id = o.customer_id
inner join `target_scaler.order_items` as oi
on oi.order_id = o.order_id
group by c.customer_state
order by avg_freight desc;

```

TUDY Search (/) for resources, docs, products, and more Search 1 ? : [Profile]

Full BigQuery experience. [Learn more](#) DISMISS [UPGRADE](#)

Query results [SAVE RESULTS](#) [EXPLORE DATA](#)

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	customer_state	avg_freight	delivery_time	est_del_diff		
3	RO	41.06971223021...	0-0 0 473:44:41.212454212	0-0 0 464:11:10.890109890		
4	AC	40.07336956521...	0-0 0 497:10:23.516483516	0-0 0 487:59:17.450549450		
5	PI	39.14797047970...	0-0 0 465:13:58.927342256	0-0 0 260:27:6.619502868		
6	MA	38.25700242718...	0-0 0 519:34:4.800	0-0 0 221:24:4.645		
7	TO	37.24660317460...	0-0 0 418:45:40.490322580	0-0 0 279:36:27.777419354		
8	SE	36.65316883116...	0-0 0 515:12:59.317333333	0-0 0 223:49:3.408		
9	AL	35.84367117117...	0-0 0 587:44:21.852459016	0-0 0 193:22:34.871194379		
10	PA	35.83268518518...	0-0 0 570:5:50.211574952	0-0 0 325:39:52.679316888		
11	RN	35.65236294896...	0-0 0 463:48:23.767754318	0-0 0 318:23:59.376199616		
12	AP	34.00609756097...	0-0 0 676:56:34.925925925	0-0 0 426:26:28.518518518		
13	AM	33.20539393939...	0-0 0 633:22:59.564417177	0-0 0 461:25:29.638036809		
14	PE	32.91786267995...	0-0 0 438:42:8.667239404	0-0 0 306:20:47.015463917		

Results per page: 50 1 - 27 of 27 < > |

PERSONAL HISTORY PROJECT HISTORY [REFRESH](#)

35°C Haze 22:52 08-06-2023

INFERENCE:

-> insights are self explanatory here

RECOMMENDATION:

-> the best thing target can do is to remain committed to there delivery date ensuring that there are no unnecessary delays, this will frustrate the customer and will definitely refrain them from further orders, target should find the loop holes which are leading to these delays and try to solve them accordingly

Q5.2 Find out the top 5 states with the highest & lowest average freight value.

```
(select c.customer_state, avg(freight_value) as avg_freight
from `target_scaler.orders` as o inner join `target_scaler.order_items` as oi
on o.order_id = oi.order_id
inner join `target_scaler.customers` as c
on o.customer_id = c.customer_id
group by c.customer_state
order by avg_freight desc
limit 5)
union all
(select c.customer_state, avg(freight_value) as avg_freight
from `target_scaler.orders` as o inner join `target_scaler.order_items` as oi
on o.order_id = oi.order_id
inner join `target_scaler.customers` as c
on o.customer_id = c.customer_id
group by c.customer_state
order by avg_freight asc
limit 5)
```

the full BigQuery experience. [Learn more](#)

DISMISS

UPGRADE

Row	customer_state	avg_freight
1	RR	42.98442307692...
2	PB	42.72380398671...
3	RO	41.06971223021...
4	AC	40.07336956521...
5	PI	39.14797047970...
6	SP	15.14727539041...
7	PR	20.53165156794...
8	MG	20.63016680630...
9	RJ	20.96092393168...
10	DF	21.04135494596...

INFERENCE:

-> from the above result we can see the average freight value from the top 5 states to the lowest 5 states

RECOMMENDATIONS:

-> target should do something to keep the freight cost uniform one way to do so is by setting up its warehouses in the states with higher freight costs so that the cost may go down to some extent, also they can try to source the suppliers in each city of the state which will ensure fast and cost effective delivery

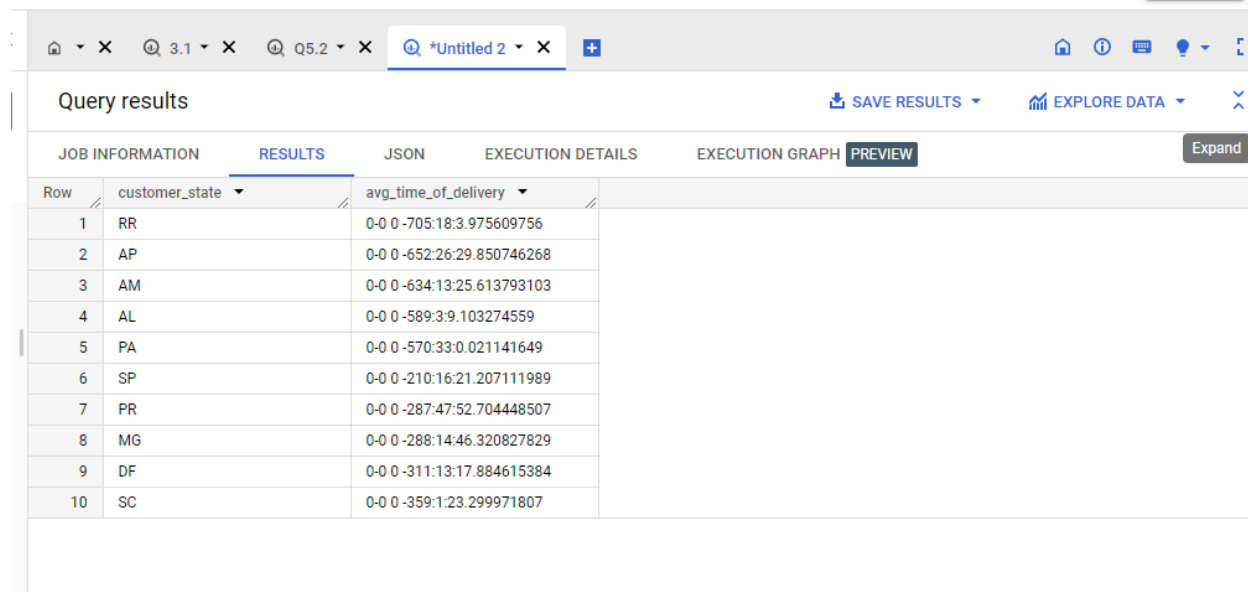
Q5.3 Find out the top 5 states with the highest & lowest average delivery time.

```
(select c.customer_state, avg(order_purchase_timestamp -  
order_delivered_customer_date) as avg_time_of_delivery  
from `target_scaler.orders` as o inner join `target_scaler.customers` as c  
on o.customer_id = c.customer_id  
group by c.customer_state  
order by avg_time_of_delivery asc  
limit 5)  
union all  
(select c.customer_state, avg(order_purchase_timestamp -  
order_delivered_customer_date) as avg_time_of_delivery  
from `target_scaler.orders` as o inner join `target_scaler.customers` as c  
on o.customer_id = c.customer_id  
group by c.customer_state  
order by avg_time_of_delivery desc  
limit 5)
```

he full BigQuery experience. [Learn more](#)

DISMISS

UPGRADE



The screenshot shows the Google Cloud BigQuery interface. At the top, there's a navigation bar with tabs for '3.1', 'Q5.2', and '*Untitled 2'. Below this, the 'Query results' section is active, displaying a table with two columns: 'customer_state' and 'avg_time_of_delivery'. The table contains 10 rows of data, sorted by delivery time. The 'customer_state' values are RR, AP, AM, AL, PA, SP, PR, MG, DF, and SC. The 'avg_time_of_delivery' values are in HH:MM:SS.SSS format. The interface also includes tabs for 'JOB INFORMATION', 'RESULTS', 'JSON', 'EXECUTION DETAILS', 'EXECUTION GRAPH', and 'PREVIEW'. A 'SAVE RESULTS' button and an 'EXPLORE DATA' link are visible at the top right of the results section.

Row	customer_state	avg_time_of_delivery
1	RR	0-0 0 -705:18:3.975609756
2	AP	0-0 0 -652:26:29.850746268
3	AM	0-0 0 -634:13:25.613793103
4	AL	0-0 0 -589:3:9.103274559
5	PA	0-0 0 -570:33:0.021141649
6	SP	0-0 0 -210:16:21.207111989
7	PR	0-0 0 -287:47:52.704448507
8	MG	0-0 0 -288:14:46.320827829
9	DF	0-0 0 -311:13:17.884615384
10	SC	0-0 0 -359:1:23.299971807

INFERENCE:

-> insights are self explanatory here

RECOMMENDATION:

-> Target should primarily focus on the states with the highest delivery time , this will surely gonna be affecting their sales, we can use the same approach here as well i.e setting up the local suppliers or warehouses in the states only , ensuring fast and cost effective delivery.

Q5.4 Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

```
(select c.customer_state, o.order_id,
date_diff(order_delivered_customer_date, order_estimated_delivery_date, day) as
actual_del_time
from `target_scaler.orders` as o inner join `target_scaler.customers` as c
on o.customer_id = c.customer_id
where order_delivered_customer_date is not null and order_estimated_delivery_date is
not null
order by actual_del_time desc
limit 5)
```

union all

```
(select c.customer_state, o.order_id,  
date_diff(order_delivered_customer_date, order_estimated_delivery_date, day) as  
actual_del_time  
from `target_scaler.orders` as o inner join `target_scaler.customers` as c  
on o.customer_id = c.customer_id  
where order_delivered_customer_date is not null and order_estimated_delivery_date is  
not null  
order by actual_del_time  
limit 8)
```

The screenshot shows a web-based query results interface. At the top, there's a header with 'Query results' and buttons for 'SAVE RESULTS' and 'EXPLORE DATA'. Below this is a tabbed interface with 'RESULTS' selected. The table has columns: 'Row', 'customer_state', 'order_id', and 'actual_del_time'. The data is sorted by 'actual_del_time' in descending order. The interface also shows 'Results per page: 30' and '1 - 13 of 13'.

Row	customer_state	order_id	actual_del_time
1	RJ	1b3190b2dfa9d789e1f14c05b...	188
2	ES	ca07593549f1816d26a572e06...	181
3	SP	47b40429ed8cce3aee9199792...	175
4	SP	2fe324febf907e3ea3f2aa9650...	167
5	SE	285ab9426d6982034523a855f...	166
6	SP	0607f0efea4b566f1eb8f7d3c2...	-146
7	MA	c72727d29cde4cf870d569bf6...	-139
8	RS	eec7f369423b033e549c02f3c...	-134
9	SP	c2bb89b5c1dd978d507284be...	-123
10	RJ	40dc2ba6f322a17626aac6244...	-108
11	SP	1a695d543b7302aa9446c8d5f...	-83
12	SP	39e0115911bf404857e14baa7...	-82

INFERENCE:

-> insights can be gained directly from the query

RECOMMENDATIONS:


-> target should try to gain insights from the states in which the order delivery is the fastest and try to implement the same in the other states as well this will ensure that at least the order will be delivered on time

Q6-Analysis based on the payments:

Q6.1 Find the month on month no. of orders placed using different payment types.

```
(select
extract(year from o.order_purchase_timestamp) as year,
extract(month from o.order_purchase_timestamp) as month,
count(o.order_id) as total_orders,
p.payment_type
from `target_scaler.orders` as o inner join `target_scaler.payments` as p
on o.order_id = p.order_id
group by year,month,p.payment_type
order by year, month)
```

Query results


SAVE RESULTS

JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

EXECUTION GRAPH

PREVIEW

Row	year	month	total_orders	payment_type
1	2016	9	3	credit_card
2	2016	10	254	credit_card
3	2016	10	23	voucher
4	2016	10	2	debit_card
5	2016	10	63	UPI
6	2016	12	1	credit_card
7	2017	1	61	voucher
8	2017	1	197	UPI
9	2017	1	583	credit_card
10	2017	1	9	debit_card
11	2017	2	1356	credit_card
12	2017	2	119	voucher

Results per page:

50

1 - 50

INFERENCE:

-> from the above results credit cards are the most preferred payment options

RECOMMENDATION:

-> Since credit card is the most preferred payment options, target should tie up with various credit card companies and come with the options in which more discount can be given to the customers based on the credit card they are using, it will be a win win situation for both of the credit card companies and target, if the sale from a particular credit card is good enough target can even move on with a profit sharing model with that credit card company , periodic offers can also be given to the vouchers and UPI using customers.

Q6.2 Find the no. of orders placed on the basis of the payment installments that have been paid.

```
select p.payment_installments, count(o.order_id) as No_of_orders
from `target_scaler.orders` as o inner join `target_scaler.payments` as p
on o.order_id = p.order_id
group by p.payment_installments
order by p.payment_installments
```

CASE STUDY Search (/) for resources, docs, products, and more Search 1

ide to the full BigQuery experience. [Learn more](#) DISMISS

Query results SAVE RESULTS EXPLOR

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

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
10	9	644
11	10	5328
12	11	23

Results per page: 50 1 - 24 of 24

INFERENCE:

-> maximum number of orders are fully paid in 1st installment only

RECOMMENDATION:

-> one of the good sign for target is that in most of the orders its payment is not floating in the market it shows that the cash flow is going alright, most of the orders being placed are paid off in the first installment only but it also depends upon the cist of the orders, it might be possible that those orders are of lower value then others hence to increase the cash flow in the high tickets orders market as well

target can try to give its customers some extra discount if they are willing to pay in one go.