

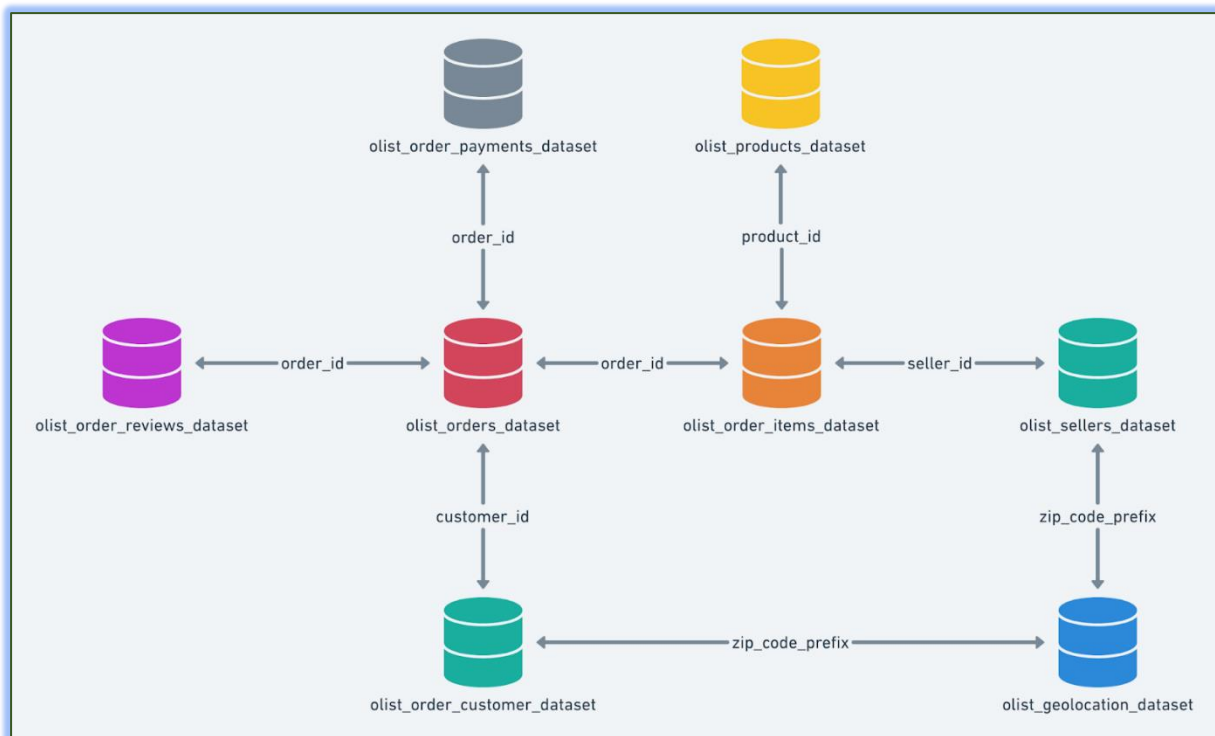


BUSINESS CASE STUDY -
OPERATIONS OF
TARGETS IN BRAZIL



BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

DATA SCHEMA



BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

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.
2. Get the time range between which the orders were placed.
3. Count the Cities & States of customers who ordered during the given period.

2. In-depth Exploration:

1. Is there a growing trend in the no. of orders placed over the past years?
2. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?
3. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
 1. 0-6 hrs : Dawn
 2. 7-12 hrs : Mornings
 3. 13-18 hrs : Afternoon
 4. 19-23 hrs : Night

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

1. Get the month on month no. of orders placed in each state.
2. How are the customers distributed across all the states?

4. 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.
2. Calculate the Total & Average value of order price for each state.
3. Calculate the Total & Average value of order freight for each state.

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:

- o **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
- o **diff_estimated_delivery** = order_estimated_delivery_date - order_delivered_customer_date

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

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

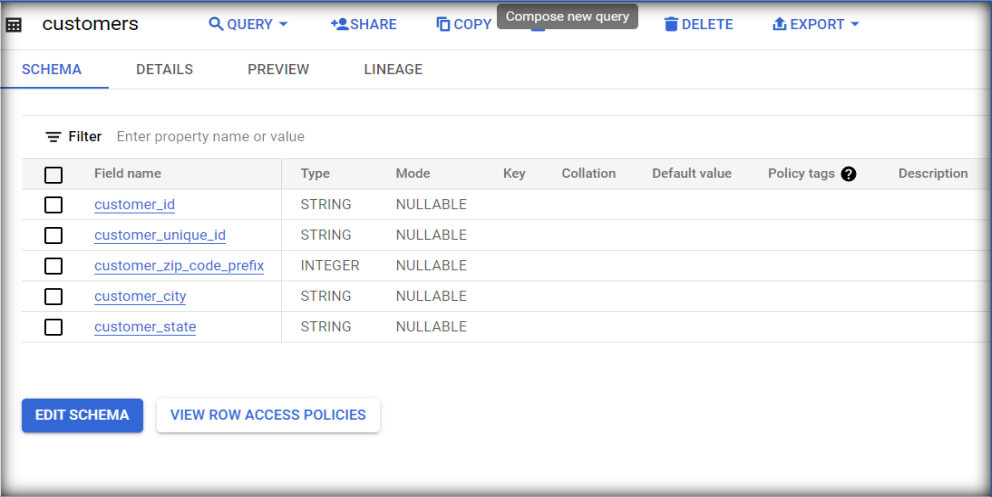
6. Analysis based on the payments:

1. Find the month on month no. of orders placed using different payment types.
2. Find the no. of orders placed on the basis of the payment installments that have been paid.

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.

ANSWER:



Field name	Type	Mode	Key	Collation	Default value	Policy tags	Description
<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					

Table Field features and description

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

Features	Description
customer_id	ID of the consumer who made the purchase
customer_unique_id	Unique ID of the consumer
customer_zip_code_prefix	Zip Code of consumer's location
customer_city	Name of the City from where order is made
customer_state	State Code from where order is made (Eg. são paulo - SP)

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

ANSWER(1):

```
SELECT
max(extract(date from
order_purchase_timestamp)) as endDate,
min(extract(date from
order_purchase_timestamp)) as startDate,
FROM `target_sql.orders`
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	endDate	startDate		
1	2018-10-17	2016-09-04		

ANSWER(2)

```
SELECT
max(order_purchase_timestamp) as
lastOrder,
min(order_purchase_timestamp) as
firstOrder,
FROM `target_sql.orders`
```

Query results

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

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

```
SELECT
count(distinct cus.customer_city) as
total_city,
cus.customer_state as state_Name
FROM `target_sql.customers` as cus
inner join `target_sql.orders` as ord
on cus.customer_id= ord.customer_id
group by cus.customer_state
order by total_city desc
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	total_city	total_state		
1	4119	27		

```
SELECT
count(distinct cus.customer_city) as
total_city,
count(distinct cus.customer_state)as
total_state,
FROM `target_sql.customers` as cus
inner join `target_sql.orders` as ord
on cus.customer_id= ord.customer_id
```

Query results

JOB INFORMATION		RESULTS	JSON
Row	total_city	state_Name	
1	745	MG	
2	629	SP	

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

SELECT

```
count(distinct ord.order_id) as totalOrder,  
cus.customer_city as customerCity  
FROM `target_sql.customers` as cus  
inner join `target_sql.orders` as ord  
on cus.customer_id= ord.customer_id  
group by cus.customer_city  
order by totalOrder desc
```

Query results

JOB INFORMATION		RESULTS	JSON
Row	totalOrder	customerCity	
1	15540	sao paulo	
2	6882	rio de janeiro	

SELECT

```
count(distinct cus.customer_unique_id) as  
customerCount,  
cus.customer_city as customerCity  
FROM `target_sql.customers` as cus  
inner join `target_sql.orders` as ord  
on cus.customer_id= ord.customer_id  
group by cus.customer_city  
order by customerCount desc
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	customerCount	customerCity		
1	14984	sao paulo		
2	6620	rio de janeiro		
3	2672	belo horizonte		

INSIGHTS

1. Its found that using 27 states there were 4119 cities who ordered in between 4-09-2016 to 17-10-2018.
2. It is recorded that highest count of customer's city in state named 'MG'
3. It is recorded that lowest count of customer city in state named 'RR'.
4. Highest orders were recorded from city 'Sao paulo'.
5. Lowest orders were recorded from city 'vila bela da santissima trindade'
6. Highest customers count were recorded from city 'Sao paulo'.
7. Lowest customer count were recorded from city 'vila bela da santissima trindade'.

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

2. In-depth Exploration:

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

SELECT

```
count (distinct order_id) order_count,  
extract(year from  
ord.order_purchase_timestamp) as  
year  
from  
`target_sql.orders` as ord  
group by year order by year
```

Query results

JOB INFORMATION		RESULTS	JSON
Row	order_count	year	
1	329	2016	
2	45101	2017	
3	54011	2018	

INSIGHTS:

Yes count of order is increasing yearly where lowest order count was in 2016 and highest order count was in 2018

SELECT

```
count (distinct cus.customer_unique_id)  
customer_count,  
extract(year from  
ord.order_purchase_timestamp) as  
year  
FROM `target_sql.customers` as cus  
inner join `target_sql.orders` as ord  
on cus.customer_id= ord.customer_id  
group by year  
order by customer_count desc
```

Query results

JOB INFORMATION		RESULTS	JSON
Row	customer_count	year	
1	52749	2018	
2	43713	2017	
3	326	2016	

INSIGHTS:

It is recorded that highest customer count was in 2018

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

```
SELECT
count(distinct
ord.order_id) as
totalOrder,
extract(DAY from
order_purchase_timestamp)
as DAY,
FROM `target_sql.orders` as ord
group by DAY
order by totalOrder desc
```

Query results

JOB INFORMATION		RESULTS	JSON
Row	totalOrder	DAY	
1	3877	24	
2	3581	16	
3	3524	15	

Query results

JOB INFORMATION		RESULTS	JSON
Row	totalOrder	DAY	
28	3011	28	
29	2557	29	
30	2534	30	
31	1680	31	

INSIGHTS:

1. It is recorded that highest order were placed on 24th Day of the month
2. It is recorded that lowest order were placed last day of the month

```
SELECT
count(distinct
ord.order_id) as
totalOrder,
extract(MONTH from
order_purchase_timestamp)
as MONTH,
FROM `target_sql.orders` as ord
group by MONTH
order by totalOrder desc
```

Query results

JOB INFORMATION		RESULTS	JSON
Row	totalOrder	MONTH	
1	10843	8	
2	10573	5	
3	10318	7	
4	9893	3	

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

INSIGHTS:

1. It is recorded that highest order were placed in month of August
2. It is recorded that lowest order were placed in month of September\
3. It is recorded that highest order were placed in quarter 2 nd and lowest were in quarter 4 th

SELECT

```
count(distinct ord.order_id) as totalOrder,  
FORMAT_DATE("%A",order_purchase_timesta  
mp)  
DayName  
FROM `target_sql.orders` as ord  
group by DayName  
order by totalOrder desc
```

Query results

JOB INFORMATION		RESULTS	JSON
Row	totalOrder	DayName	
1	16196	Monday	
2	15963	Tuesday	
3	15552	Wednesday	
4	14761	Thursday	
5	14122	Friday	

INSIGHTS :

1. Highest order were recorded on Monday
2. Lowest order were recorded on Saturday

SELECT

```
count(distinct ord.order_id) as  
totalOrder,  
EXTRACT(HOUR from  
order_purchase_timestamp)  
HOUR  
FROM `target_sql.orders` as ord  
group by HOUR  
order by totalOrder desc
```

Query results

JOB INFORMATION		RESULTS	JSON
Row	totalOrder	HOUR	
1	6675	16	
2	6578	11	
3	6569	14	
4	6518	13	

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

INSIGHTS;

1. Highest order were recorded at 4.00PM
2. Lowest order were 5.00 AM

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

```
SELECT  
count(distinct  
ord.order_id) as  
totalOrder,  
extract(DAY from  
order_purchase_timestamp)  
as DAY,  
FROM `target_sql.orders` as ord  
group by DAY  
order by totalOrder desc
```

Query results

JOB INFORMATION		RESULTS	JSON
Row	totalOrder	DAY	
1	3877	24	
2	3581	16	
3	3524	15	
4	3483	4	

Query results

JOB INFORMATION		RESULTS	JSON
Row	totalOrder	DAY	
23	3168	10	
24	3128	23	
25	3122	27	
26	3116	21	
27	3101	1	
28	3011	28	
29	2557	29	
30	2534	30	
31	1680	31	

INSIGHTS :

1. It is recorded that highest order were placed on 24th Day of the month
2. It is recorded that lowest order were placed last day of the month

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

```
SELECT
count(distinct ord.order_id) as totalOrder,
CASE
WHEN EXTRACT(HOUR from
order_purchase_timestamp)>=0 and EXTRACT(HOUR from
order_purchase_timestamp)<=6
THEN 'Dawn'
WHEN EXTRACT(HOUR from
order_purchase_timestamp)>=7 and EXTRACT(HOUR from
order_purchase_timestamp)<=12
THEN 'Morning'
WHEN EXTRACT(HOUR from
order_purchase_timestamp)>=13 and EXTRACT(HOUR from
order_purchase_timestamp)<=18
THEN 'Afternoon'
ELSE 'Night'
END AS HOUR ,
CASE
WHEN EXTRACT(MONTH from
order_purchase_timestamp) in(12,1,2,3)
THEN 'Summer'
WHEN EXTRACT(MONTH from
order_purchase_timestamp) in(4,5,6,7)
THEN 'Mid Winter'
ELSE 'Winter'
END AS season
FROM `target_sql.orders` as
ord
group by HOUR, season
order by totalOrder desc
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	totalOrder	HOUR		season
1	15208	Afternoon		Mid Winter
2	12463	Afternoon		Summer
3	11352	Night		Mid Winter
4	11130	Morning		Mid Winter
5	10464	Afternoon		Winter
6	9039	Night		Summer
7	8949	Morning		Summer
8	7940	Night		Winter
9	7654	Morning		Winter

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

INSIGHTS:

- 1. Highest order takes place in mid winter's and summer's afternoon hour .**
- 2. Lowest orders are recorded in mid winter and summer's Dawn hour**

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

```
SELECT
count(distinct ord.order_id) as
totalOrder,
EXTRACT(HOUR from
order_purchase_timestamp)
HOUR
FROM `target_sql.orders` as ord
group by HOUR
order by totalOrder desc
```

```
SELECT
count(distinct ord.order_id) as totalOrder,
CASE
WHEN EXTRACT(HOUR from
order_purchase_timestamp)>=0 and
EXTRACT(HOUR from
order_purchase_timestamp)<=6
THEN 'Dawn'
WHEN EXTRACT(HOUR from
order_purchase_timestamp)>=7 and
EXTRACT(HOUR from
order_purchase_timestamp)<=12
THEN 'Morning'
WHEN EXTRACT(HOUR from
order_purchase_timestamp)>=13 and
EXTRACT(HOUR from
order_purchase_timestamp)<=18
THEN 'Afternoon'
ELSE 'Night'
END AS HOUR
FROM `target_sql.orders` as ord
group by HOUR
order by totalOrder desc
```

Query results			
JOB INFORMATION		RESULTS	JSON
Row	totalOrder	HOUR	
1	6675	16	
2	6578	11	
3	6569	14	
4	6518	13	
5	6454	15	

19	1170	1	
20	510	2	
21	502	6	
22	272	3	
23	206	4	
24	188	5	

Query results			
JOB INFORMATION		RESULTS	JSON
Row	totalOrder	HOUR	
1	38135	Afternoon	
2	28331	Night	
3	27733	Morning	
4	5242	Dawn	

INSIGHTS

- Highest order were recorded at Afternoon time it is active hour
- Dawn time is lazy hour for order placing

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

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

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

```
select
count(distinct ord.order_id) as order_count,
FORMAT_DATETIME("%B",ord.order_purchase_timestamp)
as MonthName,
extract(MONTH from
ord.order_purchase_timestamp) as
MonthNumber,
cus.customer_state as state_Name
FROM `target_sql.customers` as
cus
inner join `target_sql.orders`
as ord
on cus.customer_id= ord.customer_id
group by cus.customer_state,
MonthName,MonthNumber
order by order_count desc,MonthNumber asc
```

Query results			
JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS
Row	order_count	MonthName	MonthNumber
1	4982	August	8
2	4632	May	5
3	4381	July	7
4	4104	June	6
5	4047	March	3
6	3967	April	4
7	3357	February	2
8	3351	January	1
9	3012	November	11
10	2357	December	12
11	1908	October	10

INSIGHTS:

Highest order of each month were recorded in state named SP.

Query results			
JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS
Row	O_CN	HOURL	state_Name
1	16104	Afternoon	SP
2	11720	Night	SP
3	11664	Morning	SP
4	4941	Afternoon	RJ
5	4417	Afternoon	MG
6	3629	Night	RJ
7	3515	Morning	RJ
8	3388	Morning	MG
9	3357	Night	MG

INSIGHTS:

Highest order of Afternoon, Morning and Night is in state SP

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

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

select

```
count(distinct ord.order_id) as O_CN,
CASE
WHEN EXTRACT(HOUR from
order_purchase_timestamp)>=0 and
EXTRACT(HOUR from
order_purchase_timestamp)<=6
THEN 'Dawn'
WHEN EXTRACT(HOUR from
order_purchase_timestamp)>=7 and
EXTRACT(HOUR from
order_purchase_timestamp)<=12
THEN 'Morning'
WHEN EXTRACT(HOUR from
order_purchase_timestamp)>=13 and
EXTRACT(HOUR from
order_purchase_timestamp)<=18
THEN 'Afternoon'
ELSE 'Night'
END AS HOUR ,
cus.customer_state as state_Name
FROM `target_sql.customers` as cus
inner join `target_sql.orders` as ord
on cus.customer_id= ord.customer_id
group by cus.customer_state,HOUR
order by O_CN desc
```

Query results

JOB INFORMATION		RESULTS	JSON
Row	count	state	
1	40302	SP	
2	12384	RJ	
3	11259	MG	
4	5277	RS	
5	4882	PR	

22	273	TO	
23	240	RO	
24	143	AM	
25	77	AC	
26	67	AP	
27	45	RR	

```
select count(distinct
customer_unique_id) as count,
customer_state as state
from
`target_sql.customers`
group by customer_state
order by count desc
```

2. How are the customers distributed across all the states

```
select count(distinct cus.customer_unique_id) as
count,
customer_state as state,
CASE
WHEN EXTRACT(MONTH from
ord.order_purchase_timestamp)
in(12,1,2,3)
THEN 'Summer'
WHEN EXTRACT(MONTH from
ord.order_purchase_timestamp)
in(4,5,6,7)
THEN 'Mid Winter'
ELSE 'Winter'
END AS season
FROM `target_sql.customers` as cus
inner join `target_sql.orders` as ord
on cus.customer_id= ord.customer_id
group by customer_state,season
order by count desc
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	count	state		season
1	16729	SP		Mid Winter
2	12806	SP		Summer
3	11305	SP		Winter
4	4794	RJ		Mid Winter
5	4361	MG		Mid Winter
6	4154	RJ		Summer
7	3869	MG		Summer
8	3611	RJ		Winter
9	3169	MG		Winter

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

INSIGHTS

Customer are distributed as per weather season across all states of brazil where state SP is having highest customer count in mid summer, summer and winter

4. 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).

```
with
y2017 as
(select
extract(month from ord.order_purchase_timestamp) monthNumber,
FORMAT_DATETIME("%B",ord.order_purchase_timestamp) as month,
round(sum(pay.payment_value)) as payment
from `target_sql.orders` as ord
inner join `target_sql.payments` as pay
on ord.order_id = pay.order_id
where extract(year from ord.order_purchase_timestamp)=2017
and extract(month from ord.order_purchase_timestamp) between 1 and 8
group by month, monthNumber
order by monthNumber
),
y2018 as
(select
extract(month from ord.order_purchase_timestamp) monthNumber,
FORMAT_DATETIME("%B",ord.order_purchase_timestamp) as month,
round(sum(pay.payment_value)) as payment
from `target_sql.orders` as ord
inner join `target_sql.payments` as pay
on ord.order_id = pay.order_id
where extract(year from ord.order_purchase_timestamp)=2018
and extract(month from ord.order_purchase_timestamp) between 1 and 8
group by month, monthNumber
order by monthNumber
)
select
a.month,
a.payment as Year2017, b.payment as year2018,
concat(round(((b.payment/a.payment)*100),2),"%") as percentageIncrease
from y2017 as a inner join y2018 b
on a.month=b.month
order by a.monthNumber
```

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

Query results

JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW
Row	month	Year2017	year2018	percentageIncrease	
1	January	138488.0	1115004.0	805.13%	
2	February	291908.0	992463.0	339.99%	
3	March	449864.0	1159652.0	257.78%	
4	April	417788.0	1160785.0	277.84%	
5	May	592919.0	1153982.0	194.63%	
6	June	511276.0	1023880.0	200.26%	
7	July	592383.0	1066541.0	180.04%	
8	August	674396.0	1022425.0	151.61%	

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

```
select
cus.customer_state as state,
round(avg(pay.payment_value),2) as avgPrice,
round(sum(pay.payment_value),2) as totPrice
from `target_sql.customers` as cus
inner join `target_sql.orders` as ord
on ord.customer_id=cus.customer_id
inner join `target_sql.payments` as pay
on ord.order_id=pay.order_id
group by cus.customer_state
order by totPrice
```

Query results

JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS
Row	state	avgPrice	totPrice
1	RR	218.8	10064.62
2	AP	232.33	16262.8
3	AC	234.29	19680.62
4	AM	181.6	27966.93
5	RO	233.2	60866.2
6	TO	204.27	61485.33
7	SE	208.44	75246.25
8	AL	227.08	96962.06

INSIGHTS

Highest total price of order is with state PB and lowest Total price is with SP

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

```
select
cus.customer_state as state,
round(avg(oi.freight_value),2) as avgFreight,
round(sum(oi.freight_value),2) as totFreight
from `target_sql.customers` as cus
inner join `target_sql.orders` as ord
on ord.customer_id=cus.customer_id
inner join `target_sql.order_items` as oi
on ord.order_id=oi.order_id
group by cus.customer_state
order by totFreight desc
```

Query results

JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS
Row	state	avgFreight	totFreight
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
24	AM	33.21	5478.89
25	AC	40.07	3686.75
26	AP	34.01	2788.5
27	RR	42.98	2235.19

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

select

```
cus.customer_state as state,
round(avg(oi.freight_value),2) as avgFreight,
round(sum(oi.freight_value),2) as totFreight
from `target_sql.customers` as cus
inner join `target_sql.orders` as ord
on ord.customer_id=cus.customer_id
inner join `target_sql.order_items` as oi
on ord.order_id=oi.order_id
group by cus.customer_state
order by avgFreight desc
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	state	avgFreight	totFreight	
1	RR	42.98	2235.19	
2	PB	42.72	25719.73	
3	RO	41.07	11417.38	
4	AC	40.07	3686.75	
5	PI	39.15	21218.2	
6	MA	38.26	31523.77	
7	TO	37.25	11732.68	
25	MG	20.63	270853.46	
26	PR	20.53	117851.68	
27	SP	15.15	718723.07	

INSIGHTS

Highest Average freight value for order with state RR and lowest Average freight value is with SP

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.

select

```
order_purchase_timestamp dateofOrder,
order_estimated_delivery_date expec_Date,
order_delivered_customer_date date_delivered,
DATE_DIFF(order_estimated_delivery_date,order_purchase_timestamp,Day) as expec_Days,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,Day) as
diff_Delivery_from_Order, DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date,Day) as diff_from_Estimation
from `target_sql.orders`
where order_delivered_customer_date is NOT NULL
AND order_estimated_delivery_date is NOT NULL
order by order_purchase_timestamp
```

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	dateofOrder	expec_Date	date_delivered	expec_Days	diff_Delivery_from_0	diff_from_Estimation	
1	2016-09-15 12:16:38 UTC	2016-10-04 00:00:00 UTC	2016-11-09 07:47:38 UTC	18	54	-36	
2	2016-10-03 09:44:50 UTC	2016-10-27 00:00:00 UTC	2016-10-26 14:02:13 UTC	23	23	0	
3	2016-10-03 16:56:50 UTC	2016-11-07 00:00:00 UTC	2016-10-27 18:19:38 UTC	34	24	10	
4	2016-10-03 21:01:41 UTC	2016-11-25 00:00:00 UTC	2016-11-08 10:58:34 UTC	52	35	16	
5	2016-10-03 21:13:36 UTC	2016-11-29 00:00:00 UTC	2016-11-03 10:58:07 UTC	56	30	25	
6	2016-10-03 22:06:03 UTC	2016-11-23 00:00:00 UTC	2016-10-31 11:07:42 UTC	50	27	22	
7	2016-10-03 22:31:31 UTC	2016-11-23 00:00:00 UTC	2016-10-14 16:08:00 UTC	50	10	39	

A Project Report By TWINKLE PANDEY

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

INSIGHTS

Here negative Values in column diff_from_estimation represent delay in days from expected and positive values presents before estimated and 0

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

```
with query1 as
(select
cus.customer_state as state,
round(avg(oi.freight_value),2) as avgFreight
from `target_sql.customers` as cus
inner join `target_sql.orders` as ord
on ord.customer_id=cus.customer_id
inner join `target_sql.order_items` as oi
on ord.order_id=oi.order_id
group by cus.customer_state
order by avgFreight limit 5),
query2 as
(select
cus.customer_state as state,
round(avg(oi.freight_value),2) as avgFreight
from `target_sql.customers` as cus
inner join `target_sql.orders` as ord
on ord.customer_id=cus.customer_id
inner join `target_sql.order_items` as oi
on ord.order_id=oi.order_id
group by cus.customer_state
order by avgFreight desc limit 5)
select *, "Lowest" as category from query1
union ALL
select *, "Highest" from query2
order by category
```

```
with query1 as
(select
cus.customer_state as state,
round(Avg(
DATE_DIFF(ord.order_estimated_delivery_date,
ord.order_delivered_customer_date,Day)
),2)Avg_Delivery_Time_Difference
from `target_sql.customers` as cus
inner join `target_sql.orders` as ord
on ord.customer_id=cus.customer_id
group by cus.customer_state
order by Avg_Delivery_Time_Difference limit 5),
query2 as
(select
cus.customer_state as state,
round(Avg(
DATE_DIFF(ord.order_estimated_delivery_date,
ord.order_delivered_customer_date,Day)
),2)Avg_Delivery_Time_Difference
from `target_sql.customers` as cus
inner join `target_sql.orders` as ord
on ord.customer_id=cus.customer_id
group by cus.customer_state
order by Avg_Delivery_Time_Difference desc limit 5)
select *, "Lowest" as category from query1
union ALL
select *, "Highest" from query2
order by category
```

Query results

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

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	state	Avg_Delivery_Time_Difference	category	
1	AC	19.76	Highest	
2	RO	19.13	Highest	
3	AP	18.73	Highest	
4	AM	18.61	Highest	
5	RR	16.41	Highest	
6	AL	7.95	Lowest	
7	MA	8.77	Lowest	
8	SE	9.17	Lowest	

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

4. 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
count(ord.order_id) totalOrder,
cus.customer_state as state,
round(Avg(
DATE_DIFF(ord.order_estimated_delivery_date,
ord.order_delivered_customer_date,Day)
),2)Avg_actual_difference,
from `target_sql.customers` as
cus
inner join `target_sql.orders` as ord
on ord.customer_id=cus.customer_id
where order_delivered_customer_date is NOT NULL
AND order_estimated_delivery_date is NOT NULL
group by cus.customer_state
order by Avg_actual_difference desc
```

Query results			
JOB INFORMATION		RESULTS	EXECUTION DETAILS
Row	totalOrder	state	Avg_actual_difference
1	80	AC	19.76
2	243	RO	19.13
3	67	AP	18.73
4	145	AM	18.61
5	41	RR	16.41
6	886	MT	13.43
7	946	PA	13.19
8	5344	RS	12.98
25	335	SE	9.17
26	717	MA	8.77
27	397	AL	7.95

INSIGHTS

1. Top 5 state where orders are deliver on average of 16-19 days before are AC,RO,AP,AM,RR
2. Where comparatively BA,ES,SE,MA,Al states having delivery average before 7 to 9 Days of expected date

```
select
count(order_id) totalOrder,
CASE
when DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date,Day) =0
Then 'As Expected'
when DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date,Day) >0
Then 'Before'
when DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date,Day) <0
Then 'Delayed'
ELSE 'Not a case'
END as cases
from `target_sql.orders`
where order_delivered_customer_date is NOT NULL
AND order_estimated_delivery_date is NOT NULL
group by cases
order by totalOrder desc
```

Query results			
JOB INFORMATION		RESULTS	EXECUTION DETAILS
Row	totalOrder	cases	
1	87187	Before	
2	6535	Delayed	
3	2754	As Expected	

INSIGHTS

Maximum orders are
deliver before expected

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

6. Analysis based on the payments

```
Select count(pay.order_id) orderCount,
pay.payment_type ,
FORMAT_DATETIME("%B",ord.order_purchase_time
stamp) as month
from `target_sql.payments`
as pay
inner join `target_sql.orders` as
ord
on pay.order_id=ord.order_id
group by payment_type,month
order by orderCount desc
```

INSIGHTS

1. It is recorded that in every month maximum order's payment are done with credit card 2. Where lowest order's payment are maid through debit card only.

Query results			
JOB INFORMATION		RESULTS	JSON
EXECUTION DETAILS			
Row	orderCount	payment_type	month
1	8350	credit_card	May
2	8269	credit_card	August
3	7841	credit_card	July
4	7707	credit_card	March
5	7301	credit_card	April
6	7276	credit_card	June
7	6609	credit_card	February
8	6103	credit_card	January
9	5897	credit_card	November
46	64	debit_card	December
47	54	debit_card	October
48	43	debit_card	September
49	2	not_defined	August
50	1	not_defined	September

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

```
with q1 as
(select count(order_id) orderCount
from `target_sql.payments`
where payment_installments>0),
q2 as
(select count(order_id)
from `target_sql.payments`
where payment_installments>0 and
payment_sequential=payment_installments),
q3 as
(select count(order_id)
from `target_sql.payments`
where payment_installments>0 and
payment_sequential>payment_installments),
q4 as
(select count(order_id)
from `target_sql.payments`
where payment_installments>0
and payment_sequential<payment_installments),
q5 as
(select count(order_id)
from `target_sql.payments`
where payment_installments <=0)

select *, "Total orders on Installments" from q1
union all
select *, "No installments" from q5
union all
```

Query results		
JOB INFORMATION		RESULTS
JSON		
Row	orderCount	f0_
1	103884	Total orders on Installments
2	51284	Under paid
3	48290	Dues clear
4	4310	Over paid
5	2	No installments

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

```

with dueHighest as (
select count(ord.order_id) orderCount,
round(sum(pay.payment_value),2) as dueAmount,
cus.customer_state
from `target_sql.orders` as ord
inner join `target_sql.payments` pay
on ord.order_id=pay.order_id
inner join `target_sql.customers` as cus
on ord.customer_id=cus.customer_id
where payment_installments>0
and payment_sequential<payment_installments
group by cus.customer_state
order by orderCount limit 5),
dueLowest as (
select count(ord.order_id) orderCount,
round(sum(pay.payment_value),2) as dueAmount,
cus.customer_state
from `target_sql.orders` as ord
inner join `target_sql.payments` pay
on ord.order_id=pay.order_id
inner join `target_sql.customers` as cus
on ord.customer_id=cus.customer_id
where payment_installments>0
and payment_sequential<payment_installments
group by cus.customer_state
order by orderCount desc limit 5)
select *, "Lowest Dues" as category from dueHighest
union all
select *, "Highest Dues" from dueLowest
    
```

Query results

JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW
Row	orderCount	dueAmount	customer_state	category	
1	22	6171.83	RR	Lowest Dues	
2	38	11089.27	AP	Lowest Dues	
3	52	13399.06	AC	Lowest Dues	
4	81	19075.2	AM	Lowest Dues	
5	147	38406.23	RO	Lowest Dues	
6	19955	3598305.36	SP	Highest Dues	
7	7052	1391285.85	RJ	Highest Dues	
8	6308	1202963.75	MG	Highest Dues	
9	2773	539920.93	RS	Highest Dues	
10	2498	501884.48	PR	Highest Dues	

Find the count of orders of due payments in most 10 cities

```

select count(ord.order_id)
orderCount,
round(sum(pay.payment_value),2)
as dueAmount,
cus.customer_city as
customer_city
from `target_sql.orders` as ord
inner join `target_sql.payments` pay
on ord.order_id=pay.order_id
inner join `target_sql.customers` as cus
on
ord.customer_id=cus.customer_id
where payment_installments>0
and
payment_sequential<payment_installments
group by customer_city
order by orderCount desc
    
```

Query results

JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS
Row	orderCount	dueAmount	customer_city
1	7143	1299895.5	sao paulo
2	3651	736161.97	rio de janeiro
3	1503	270340.4	belo horizonte
4	1057	221788.75	brasilia
5	729	150963.35	salvador
6	705	140876.86	porto alegre
7	699	149492.89	curitiba
8	646	131320.6	campinas
9	587	101018.53	guarulhos
10	439	82135.62	niteroi

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

Find top 10 cities of highly satisfied customer (5 marks review)

```
select
count(review.review_id) as
reviewCount,
cus.customer_city as
customer_city
from `target_sql.orders` ord
inner join `target_sql.customers` as cus
on
ord.customer_id=cus.customer_id
inner join `target_sql.orders_reviews`
review
on review.order_id=ord.order_id
where review.review_score=5
group by customer_city
order by reviewCount desc limit 10
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	reviewCount	customer_city		
1	9314	sao paulo		
2	3801	rio de janeiro		
3	1614	belo horizonte		
4	1204	brasilia		
5	943	curitiba		
6	814	campinas		
7	776	porto alegre		
8	671	guarulhos		
9	597	sao bernardo do campo		

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	reviewCount	customer_city		
1	2044	sao paulo		
2	1400	rio de janeiro		
3	382	belo horizonte		
4	321	brasilia		
5	277	salvador		
6	236	porto alegre		
7	199	campinas		
8	187	curitiba		
9	176	guarulhos		
10	165	niteroi		

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

Find customer whose order's dues are clear and not satisfied in feedback (below 3 rating)

```
select
distinct cus.customer_unique_id as
customers
from `target_sql.orders` ord
inner join `target_sql.customers` as cus
on ord.customer_id=cus.customer_id
inner join `target_sql.orders_reviews` review
on review.order_id=ord.order_id
inner join `target_sql.payments` pay
on pay.order_id=ord.order_id
where review.review_score <3 and
pay.payment_installments>0 and
pay.payment_sequential>=payment_installments
group by customers
```

Query results

SAVE RESULTSEXPLORE DATA

JOB INFORMATIONRESULTSJSONEXECUTION DETAILSCHARTPREVIEWEXECUTION GRAPH

Row	customers
1	050d050af169f87216bbe2475...
2	22f5499bc63c8ebcacf51cf025...
3	5efea094993d7cad5dcf72012...
4	c6be127fa6e30c6f705a20523...
5	acf5cef98a4798f470a2bb70a3...
6	3306ce16186d1c3938adbfc02...
7	00e336519abc0012c8711db21...
8	df5e44ce11714dec60482a8bd...
9	eefe1ea56c1116469d9f0a77b...
10	dd8253fe17486f2e2cbff105e3...
11	1743631071053f26bb95b0ca2...

Results per page: 501 – 50 of 6670

INSIGHTS

These customers has cleared their dues but not satisfied with orders these customers must be followed up by sales team.

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

Find customer whose order's dues are clear and not satisfied in feedback (includes and above 4 rating)

```
select
distinct cus.customer_unique_id as
customers
from `target_sql.orders` ord
inner join `target_sql.customers` as cus
on ord.customer_id=cus.customer_id
inner join `target_sql.orders_reviews` review
on review.order_id=ord.order_id
inner join `target_sql.payments` pay
on pay.order_id=ord.order_id
where review.review_score >=4 and
pay.payment_installments>0 and
pay.payment_sequential>=payment_installments
group by customers
```

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

JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	customers					
1	05d1d2d9f0161c5f397ce7fc77...					
2	54fc4ff419d5e05db5fe42906b...					
3	587482ee4b3da3583df4057f5...					
4	041d483c228fc3802663dca1b...					
5	b8cb40ec83dd073753f79b286...					
6	5373d3bd71b05d85cc1b76a1...					
7	fc26fd74a5d2f4716a5e7e8208...					
8	b56105b59a417275e17fde4a8...					
9	85844faa46c21288b566e98c3...					
10	35510879ba54fa9a954e79e76...					
11	21b9c0c96acc58e73fd6a01ed...					

Results per page: 50 1 - 50 of 37192

INSIGHTS

These customers have cleared their dues and satisfied with order rating they may be prime customers

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

Find customer whose order's dues are not clear and not satisfied in feedback (includes and above 4 rating)

```
select
distinct cus.customer_unique_id as
customers
from `target_sql.orders` ord
inner join `target_sql.customers` as cus
on ord.customer_id=cus.customer_id
inner join `target_sql.orders_reviews` review
on review.order_id=ord.order_id
inner join `target_sql.payments` pay
on pay.order_id=ord.order_id
where review.review_score >=4 and
pay.payment_installments>0 and
pay.payment_sequential<payment_installments
group by customers
```

Query results

SAVE RESULTS

EXPLORE DATA

JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

CHART

PREVIEW

EXECUTION GRAPH

Row	customers
1	161e5b2b4513edb336986a89...
2	c8d8b460d18c2896b78bcc324...
3	4f51b1fb01815cc981a1e3393...
4	59d83a6ad4e3ddf1d98331914...
5	4df2fb3faf5f1af030298a009f9...
6	b2a10ac7df4d1c125611afeef7...
7	a5908de40ac33ac0c752190b5...
8	60ec651482858c327c177cf93...
9	12c4c68f9af94cf0051ae70682...
10	22c8d7801061f27eff57f37ac4...

Results per page:

50

1 – 50 of 37621

INSIGHTS


These Category of orders are having highly dues even after there satisfaction is more then 4 star these customers must be followed up to clear theis dues.


BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL


Find customer whose order's dues are not clear and not satisfied in feedback (below 3 rating)

```
select
distinct cus.customer_unique_id as
customers
from `target_sql.orders` ord
inner join `target_sql.customers` as cus
on ord.customer_id=cus.customer_id
inner join `target_sql.orders_reviews` review
on review.order_id=ord.order_id
inner join `target_sql.payments` pay
on pay.order_id=ord.order_id
where review.review_score <3 and
pay.payment_installments>0 and
pay.payment_sequential<payment_installments
group by customers
```

Query results

 SAVE RESULTS

 EXPLORE DATA



JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

CHART

PREVIEW

EXECUTION GRAPH

Row	customers
1	fc003b1bdc0df64b4d065d9b...
2	402cce5c0509000eed9e77fec...
3	e093c00d1a7972c2a479ed40e...
4	aac060762fc0944f10caa1f269...
5	5bed15735f2d1bdfb2ba82869...
6	c74f33f8921f86511371450f06...
7	a6ecedfb81724dfa96774f35c0...

INSIGHTS

Total 7756 customers rated their order below 3 star and there dues are also not cleared

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

Top 10 category of order which order quantity is high

```
select
count(ord.order_id) totalOrder,
pro.product_category as category
from
`target_sql.orders` ord
inner join `target_sql.order_items` as oi
on oi.order_id=ord.order_id
inner join `target_sql.products` as pro
on pro.product_id=oi.product_id
group by category
order by totalOrder desc limit 10
```

Query results

SAVE RESULTS

EXPLORE DATA

<

JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

CHART

PREVIEW

EXECUTION

>

Row	totalOrder	category
1	11115	bed table bath
2	9670	HEALTH BEAUTY
3	8641	sport leisure
4	8334	Furniture Decoration
5	7827	computer accessories
6	6964	housewares
7	5991	Watches present
8	4545	telephony
9	4347	Garden tools
10	4235	automotive

INSIGHTS

These category are ordered high in quantity

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

Top 10 category of dues

```
select
count(ord.order_id) totalOrder,
pro.product_category as category
from
`target_sql.orders`
ord
inner join `target_sql.order_items`
as oi
on oi.order_id=ord.order_id
inner join `target_sql.products` as
pro
on pro.product_id=oi.product_id
inner join `target_sql.payments`
pay
on pay.order_id=ord.order_id
where pay.payment_installments>0 and
pay.payment_sequential<payment_installments
group by category
order by totalOrder desc limit 10
```

Query results

SAVE RESULTS

EXPLORE DATA

<

JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

CHART

PREVIEW

EXECUTION

>

Row	totalOrder	category
1	7131	bed table bath
2	5533	HEALTH BEAUTY
3	4499	Furniture Decoration
4	4009	Watches present
5	3956	sport leisure
6	3776	housewares
7	3050	computer accessories
8	2308	Cool Stuff
9	2214	Garden tools
10	2151	toys

INSIGHTS

These Category of orders are having highly dues need to take action

CONCLUSION

ACTIONABLE INSIGHTS

> As the completion of this business case study that is operations of targets in Brazil, it can be concluded that we have 99,441 customers of data available. and we have 96096 number of Unique Customers ids.

> Also there is a increasing trend in orders , trend sustains during 2018 and also there a slight fall we can observe during october 2017 following with a great hike in november month and again a fall in end of december 2017 and january 2018.

> Also we can observe the trend of increasing orders with time and also for revenue.

> The growth rate for july and august in 2017 to 2018 is relatively very low!

> In products Data , total 32951 different products available in Target with 73 different product_category. health and beauty, Watches present, bed table bath, sport leisure, computer accessories, Furniture Decoration, housewares, Automotive are some of the top selling product categories.

>Health and beauty products are top selling having highest orders.

>PCs and Musical Instruments category have relatively less number of products , but contributes in a high revenue.

> So when it comes to the number of orders , Average number of order are higher during November month , september and october month average orders are comparatively low , in may and july and august have higher average orders compare to other months.

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL


>If we analysis state and region wise then states São Paulo ,Paraná, Minas Gerais, Distrito Federal ,Santa Catarina and Rio de Janeiro are some of the states having faster delivery time relatively And São Paulo ,Rio de Janeiro , Minas Gerais ,Rio Grande do Sul and Paraná are top 5 highest orders states and also generating highest revenue. São Paulo state has the highest numbers of sellers in country.


>Customers are from different 4119 cities and 27 states from Brazil.

>PCs and Musical Instruments category have relatively less number of products , but contributes in a high revenue.




RECOMMENDATIONS

 IT WAS OBSERVED AN INCREASING TREND IN REVENUE AND ORDERS OVER TIME , YET DURING OCTOBER AND JANUARY SALES ARE DECREASING PROBABLY SO AS BY INTRODUCING POSSIBLE DISCOUNT ON NOT SO RUNNING PRODUCT CAN HELP SELL MORE PRODUCTS DURING THOSE LOW GOING MONTHS.

 DELIVERY IIS QUITE SLOW IN THE TOP STATES THAT IS THE NORTH REGION OF THE BRAZIL AS COMPARED TO THE ESTIMATED DATES SO THIS COULD BE SOLVED BY FAST DELIVERY IN THE NORTH REGIONS SO AS TO INCREASE NEW CUSTOMERS AND REVENUE FROM NORTH.

 AS NORTH BRAZIL HAS THE WORLDS LARGEST RIVER AND MOST EXTESIVE RAIN FOREST, MUST BE A GOOD TRAVEL DESTINATION, INTRODUCING NECESSARY SURVIVAL/ CAMPING/ADVENTURE PRODUCTS CAN HELP INCREASE REVENUE AND ORDER FROM NORTHEN REGION

 WE CAN OBSERVE THE AVERAGE TIME TO COMPLETE THE DELIVERY IS 12 DAYS . WHICH SHOULD BE REDUCED TO ATLEAST HALF , AS DUE TO HIGH COMPETITION IN E-COMMERCE MARKET , ITS IS VITAL TO DO SO.



THE END

A Project Report

By:-

TWINKLE PANDEY

Scaler Learner

BUSINESS CASE STUDY -OPERATIONS OF TARGETS IN BRAZIL

A Project Report By TWINKLE PANDEY