Context

Target is one of the world's most recognized brands and one of America's leading retailers. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

This business case has information of 100k orders from 2016 to 2018 made at Target in Brazil. Its features allows viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers.

Dataset: https://drive.google.com/drive/folders/1TGEc66YKbD443nslRi1bWgVd238gJCnb

Data is available in 8 csv files:

1. customers.csv

2. geolocation.csv

3. order_items.csv

4. payments.csv

5. reviews.csv

6. orders.csv

7. products.csv

8. sellers.csv

Each feature or columns of different CSV files are described below:

The **customers.csv** contain following features:

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 the location of the consumer.
customer_city	Name of the City from where order is made.
customer_state	State Code from where order is made(Ex- sao paulo-SP).

The **sellers.csv** contains following features:

Features	Description
seller_id	Unique Id of the seller registered
seller_zip_code_prefix	Zip Code of the location of the seller.
seller_city	Name of the City of the seller.
seller_state	State Code (Ex- sao paulo-SP)

The **order_items.csv** contain following features:

Features	Description
order_id	A unique id of order made by the consumers.
order_item_id	A Unique id given to each item ordered in the order.
product_id	A unique id given to each product available on the site.
seller_id	Unique Id of the seller registered in Target.
shipping_limit_date	The date before which shipping of the ordered product must be completed.
price	Actual price of the products ordered .
freight_value	Price rate at which a product is delivered from one point to another.

The **geolocations.csv** contain following features:

Features	Description
geolocation_zip_code_prefix	first 5 digits of zip code
geolocation_lat	latitude
geolocation_lng	longitude
geolocation_city	city name
geolocation_state	state

The **payments.csv** contain following features:

Features	Description
order_id	A unique id of order made by the consumers.
payment_sequential	sequences of the payments made in case of EMI.
payment_type	mode of payment used.(Ex-Credit Card)
payment_installments	number of installments in case of EMI purchase.
payment_value	Total amount paid for the purchase order.

The **orders.csv** contain following features:

Features	Description

order_id	A unique id of order made by the consumers.
customer_id	Id of the consumer who made the purchase.
order_status	status of the order made i.e delivered, shipped etc.
order_purchase_timestamp	Timestamp of the purchase.
order_delivered_carrier_date	delivery date at which carrier made the delivery.
order_delivered_customer_date	date at which customer got the product.
order_estimated_delivery_date	estimated delivery date of the products.

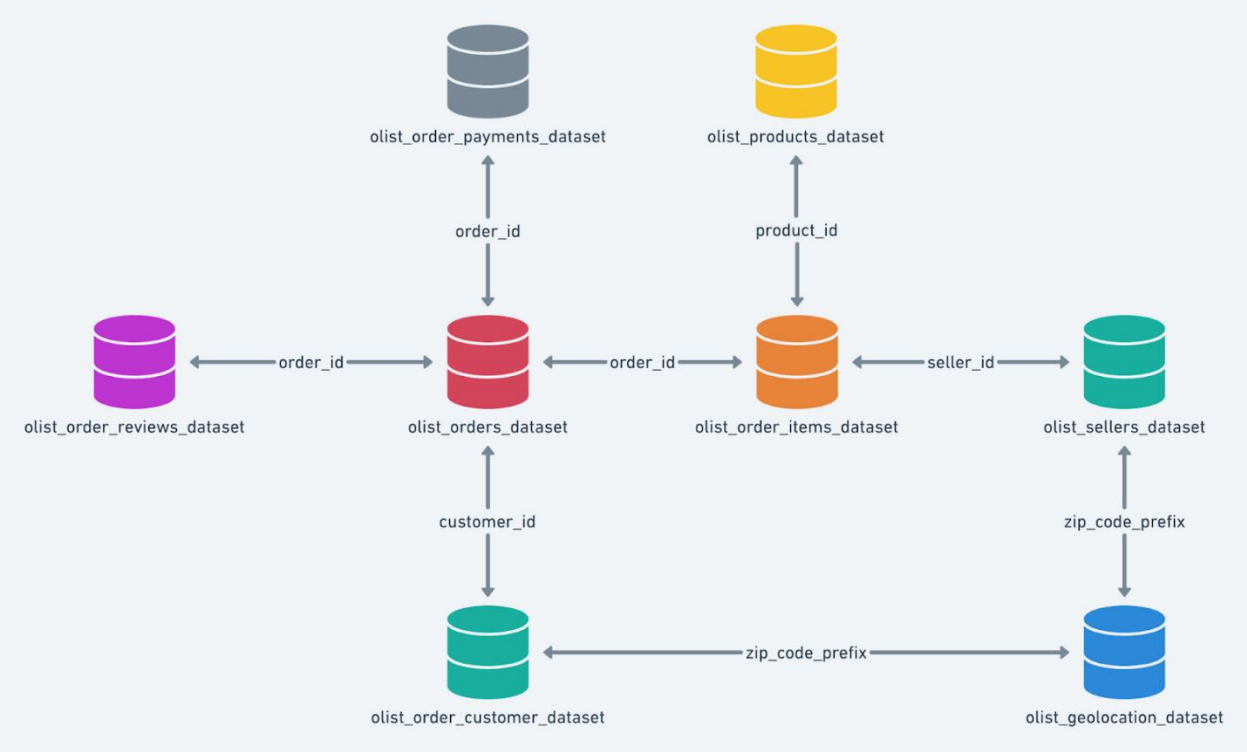
The **reviews.csv** contain following features:

Features	**Description**
review_id	Id of the review given on the product ordered by the order id.
order_id	A unique id of order made by the consumers.
review_score	review score given by the customer for each order on the scale of 1–5.
review_comment_title	Title of the review
review_comment_message	Review comments posted by the consumer for each order.
review_creation_date	Timestamp of the review when it is created.
review_answer_timestamp	Timestamp of the review answered.

The **products.csv** contain following features:

Features	**Description**
product_id	A unique identifier for the proposed project.
product_category_name	Name of the product category
product_name_lenght	length of the string which specifies the name given to the products ordered.
product_description_lenght	length of the description written for each product ordered on the site.
product_photos_qty	Number of photos of each product ordered available on the shopping portal.
product_weight_g	Weight of the products ordered in grams.
product_length_cm	Length of the products ordered in centimeters.
product_height_cm	Height of the products ordered in centimeters.
product_width_cm	width of the product ordered in centimeters.

High level overview of relationship between datasets:



Assume you are a data scientist at Target, and are given this data to analyze and provide some insights and recommendations from it.

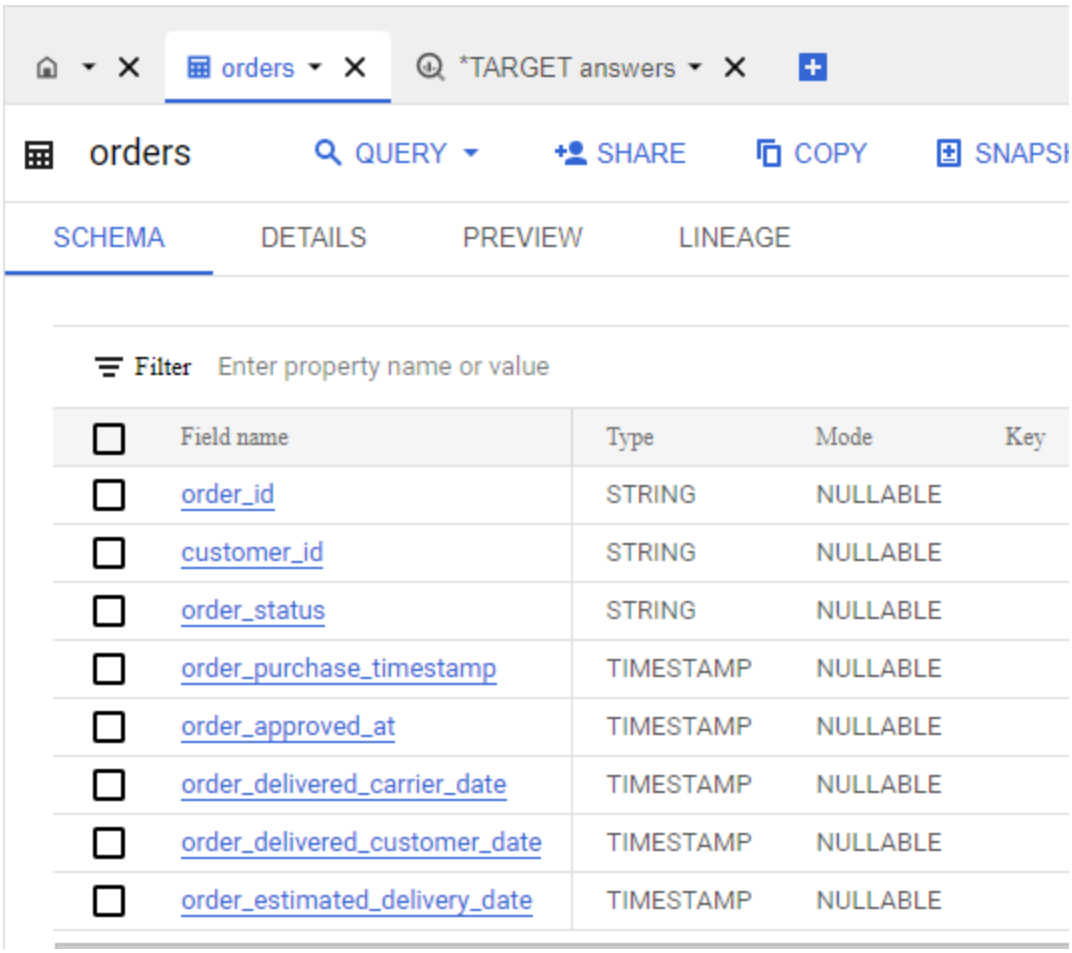
What does ‘good’ look like?

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

1.Data type of columns in a table

Ans 1.1

In the table schema option in big query, the data type for each column is clearly visible.



Field name	Type	Mode	Key
order_id	STRING	NULLABLE	
customer_id	STRING	NULLABLE	
order_status	STRING	NULLABLE	
order_purchase_timestamp	TIMESTAMP	NULLABLE	
order_approved_at	TIMESTAMP	NULLABLE	
order_delivered_carrier_date	TIMESTAMP	NULLABLE	
order_delivered_customer_date	TIMESTAMP	NULLABLE	
order_estimated_delivery_date	TIMESTAMP	NULLABLE	

INSIGHTS

The data type depiction can help in interlinking based on the similarities on columns between the different tables which will further help in performing joins, sub-queries to depict required results.

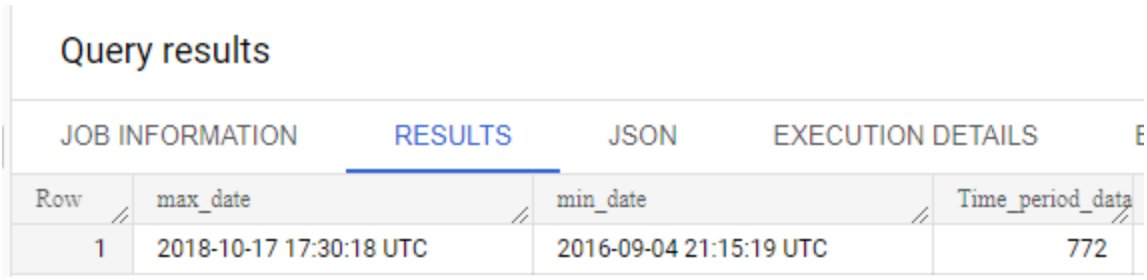
2.Time period for which the data is given

Ans-1.2

SQL QUERY

```
select MAX(order_purchase_timestamp) as max_date, MIN(order_purchase_timestam
p) as min_date,
DATE_DIFF(MAX(order_purchase_timestamp), MIN(order_purchase_timestamp), day)
AS Time_period_data
from `Target_CS.orders`;
```

RESULTS



ASSUMPTIONS

Considered column 'order_purchase_timestamp' from orders table as the time period in which the orders have been placed.

INSIGHTS

The data depicted above provides insights regarding for the time period in which orders have been made which is from Sept'2016 to Oct'2017. This data can help in performing various comparisons like ordering habits, depicting peaks in different years, months and different time frames.

RECOMMENDATIONS

This data can be further categorized on the basis of order status excluding the orders which are cancelled or unavailable which will help in identifying the orders completed or in transit.

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

Ans-1.3

SQL QUERY

```
select c.customer_city, c.customer_state,
from Target_CS.customer c
JOIN `Target_CS.orders` o ON c.customer_id = o.customer_id
order by o.order_purchase_timestamp;
```

RESULTS

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION D

Row	customer_city	customer_state
1	boa vista	RR
2	passo fundo	RS
3	sao jose dos campos	SP
4	sao joaquim da barra	SP
5	sao paulo	SP
6	sao paulo	SP
7	panambi	RS
8	rio de janeiro	RJ
9	porto alegre	RS
10	hortolandia	SP

ASSUMPTIONS

Considered column 'order_purchase_timestamp' from orders table as the time period in which the orders have been placed.

INSIGHTS

This data provides details regarding the orders placed by customers, their city and states as per the time frame in column 'order_purchase_timestamp' when linked with orders table.
This data helps in identifying the cities and states where we can focus.

RECOMMENDATIONS

This data can help in highlighting those regions which are not covered yet as well depicting distinct orders count would help in examining the cities and states which lower count where we can strategize and get to know the requirement or any difficulties in delivering the service.

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?

Ans 2.1

SQL QUERY

```
select extract(month from o.order_purchase_timestamp) as month_n
o,count(distinct p.order_id) as order_id,
sum(p.payment_value) as order_value
from Target_CS.payments p
JOIN `Target_CS.orders` o
ON p.order_id=o.order_id
group by month_no
order by order_id DESC, order_value DESC
```

RESULTS

Query results

JOB INFORMATION | RESULTS | JSON

Row	month_no	order_id	order_value
1	8	10843	1696821.64...
2	5	10573	1746900.97...
3	7	10318	1658923.67...
4	3	9893	1609515.72...
5	6	9412	1535156.88...
6	4	9343	1578573.51...
7	2	8508	1284371.35...
8	1	8069	1253492.22...
9	11	7544	1194882.80...
10	12	5674	878421.100...

ASSUMPTIONS

Considered column 'order_purchase_timestamp' from orders table as the time period on the basis of months in which the orders have been placed.

INSIGHTS

Growing trend can be clearly observed in the results as the orders_count is highest in August month and orders_value is highest in May month compared to other months figures. August, May, June are the peaks months.

RECOMMENDATIONS

This data can help in identifying the months in which there is low count which can further help in detail analyzing of the cause for such behaviors like understanding cultural aspects in different regions, product behavior, geographical limits, inventory management subject to requirement in different months etc.

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

ANS-2.2

SQL QUERY

```
select
case when EXTRACT (HOUR from order_purchase_timestamp) between 0
 AND 6 then 'DAWN'
when EXTRACT (HOUR from order_purchase_timestamp) between 7 AND
12 then 'MORNING'
when EXTRACT (HOUR from order_purchase_timestamp) between 13 AND
 18 then 'AFTERNOON'
else 'NIGHT'
end as Buying_patterns,
count (distinct order_id) as orders_count
from `Target_CS.orders`
group by Buying_patterns
order by orders_count;
```

RESULTS

Query results

JOB INFORMATION | RESULTS | JSON

Row	Buying_patterns	orders_count
1	AFTERNOON	38135
2	NIGHT	28331
3	MORNING	27733
4	DAWN	5242

ASSUMPTIONS

Considered column 'order_purchase_timestamp' from orders table as the time period on the basis of day categorization (Dawn, Morning, Afternoon or Night) in which the orders have been placed.

INSIGHTS

Helps in identifying the buying patterns of the customers with reference to different time frame of a single day. The time frame from 13:00 to 18:00 is having the highest numbers of order count.

RECOMMENDATIONS

The timings can be used for providing different offers or promotion in this various time frame for attracting the customers to buy a particular product they are in need.

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

1.Get month on month orders by states

Ans-3.1
SQL QUERY

```
select  c.customer_state, extract (month from o.order_purchase_t
imestamp) as month_no,
count(distinct o.order_id) as orders_count
from `Target_CS.orders`o
JOIN `Target_CS.customer` c
ON o.customer_id = c.customer_id
group by month_no, c.customer_state
order by month_no;
```

RESULTS

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION D

Row	customer_state	month_no	orders_count
1	RJ	1	990
2	SP	1	3351
3	DF	1	151
4	RS	1	427
5	CE	1	99
6	PE	1	113
7	PR	1	443
8	BA	1	264
9	MG	1	971
10	RN	1	51

ASSUMPTIONS

Considered column 'order_purchase_timestamp' from orders table as the time period on the basis of months in which the orders have been placed.

INSIGHTS

This result depicts the state-wise orders count as per the provided purchase data month-wise considered in assumption.
It helps in identifying the states with maximum orders count and minimum order count on monthly basis. This data can be used to focus on the state's behavior on different months

RECOMMENDATIONS

This data can help in identifying the states with minimum or below average order count which further help in identifying the cause for such lower figures, delivery time taken, warehouse availability in the states.

2.Distribution of customers across the states in Brazil

Ans 3.2

SQL QUERY

```
select c.customer_state,Count( distinct c.customer_id) as custom
er_count
from `Target_CS.customer` c
JOIN `Target_CS.geolocation` g
ON c.customer_zip_code_prefix= g.geolocation_zip_code_prefix
group by c.customer_state
order by customer_count;
```

RESULTS

Query results

JOB INFORMATION | RESULTS | JSON

Row	customer_state	customer_count
1	RR	46
2	AP	68
3	AC	81
4	AM	148
5	RO	251
6	TO	279
7	SE	349
8	AL	412
9	RN	483
10	PI	492

ASSUMPTIONS

Considered column 'geoloaction_zip_code_prefix' as the basis to capture the data belonging to the country Brazil.

INSIGHTS

This data provides insights regarding the customers distribution in the different states of Brazil. Maximum, average and minimum customers count in states.

RECOMMENDATIONS

This data can help in identifying the customers counts which are below average which can further help in examining the causes for such lower count of customers, delivery behavior, review table of customers can be used to capture customer needs, help in fabricating the offers for lower count states to attract more and more customer. Advertisement department can focus such areas for attracting the public which are not included yet.

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

Ans 4.1

SQL Query

```
select ROUND(cy_sales,2) as CY_2018_sales, ROUND(ly_sales,2) as CY_201
7_sales,
round(((cy_sales-ly_sales)/(ly_sales))*100,0) as percentage_increase
from
(select
sum(case when year=2018 and month between 1 and 8 then payment_value e
nd)as cy_sales,
sum(case when year=2017 and month between 1 and 8 then payment_value e
nd)as ly_sales,
from
(select
extract (month from o.order_purchase_timestamp) as month,
extract (year from o.order_purchase_timestamp) as year,
p.payment_value
from `Target_CS.payments` p
JOIN `Target_CS.orders` o
ON p.order_id =o.order_id ));
```

RESULTS

Query results

JOB INFORMATION | RESULTS | JSON

Row	CY_2018_sales	CY_2017_sales	percentage_increa
1	8694733.84	3669022.12	137.0

ASSUMPTIONS

Considered column 'order_purchase_timestamp', fetched month from Jan to Aug for the year 2017 and 2018 as the basis to capture the data of sales earned (considered payment_value column from payments), keeping 2017 as the base year for depicting the percentage increase.

INSIGHTS

The result captures the increase in percentage volume of sales from 2017 to 2018, keeping base year as 2017.

RECOMMENDATIONS

This data helps in monitoring the total revenue generated from one year to another and the percentage increase over the same.
This method will help in keeping close check over the year-to- year performance in the available market. Trend fixture over the years can be attained.

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

Ans 4.2

SQL QUERY

```
select c.customer_state, ROUND(avg(ot.price),0) as Avg_price,ROUND(sum
(ot.price),0) as Total_price,
ROUND(avg(ot.freight_value),0) as Avg_freight, ROUND(sum(ot.freight_va
lue),0) as Total_freight
from `Target_CS.orders` o
JOIN Target_CS.customer c
ON o.customer_id = c.customer_id
JOIN `Target_CS.order_items` ot
ON o.order_id = ot.order_id
group by c.customer_state;
```

RESULTS

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRA

Row	customer_state	Avg_price	Total_price	Avg_freight	Total_freight
1	MT	148.0	156454.0	28.0	29715.0
2	MA	145.0	119648.0	38.0	31524.0
3	AL	181.0	80315.0	36.0	15915.0
4	SP	110.0	5202955.0	15.0	718723.0
5	MG	121.0	1585308.0	21.0	270853.0
6	PE	146.0	262788.0	33.0	59450.0
7	RJ	125.0	1824093.0	21.0	305589.0
8	DF	126.0	302604.0	21.0	50625.0
9	RS	120.0	750304.0	22.0	135523.0
10	SE	153.0	58921.0	37.0	14111.0

ASSUMPTIONS

Considered 'order_id' column from orders table and 'customer_id' from customer as the basis to include the orders which were actually placed.

INSIGHTS

This data provides insights regarding the state-wise distribution of cost which include the parameters of price and freight value incurred for delivering the ordered product.
It helps in identifying the regions where there is high price and freight value with which they can investigate over those regional areas.

RECOMMENDATIONS

This data can help in detail analyzation of those regions where there is high price and freight value, examining the warehouse availability, a comparison between freight value and the delivery time taken.

5.Analysis on sales, freight and delivery time

1.Calculate days between purchasing, delivering and estimated delivery

Ans 5.1

SQL QUERY

```
select ifnull(abs(DATE_DIFF(DATE(order_purchase_timestamp),DATE(order_
delivered_customer_date),DAY)),0) as day_diff,
```

```
ifnull(abs(DATE_DIFF(DATE(order_purchase_timestamp),DATE(order_estimat
ed_delivery_date),DAY)),0) as day_diff_1,
ifnull(abs(DATE_DIFF(DATE(order_delivered_customer_date),DATE(order_es
timated_delivery_date),DAY)),0) as day_diff_2
from `Target_CS.orders`
```

RESULTS

Query results

JOB INFORMATION | RESULTS | JSON

Row	day_diff	day_diff_1	day_diff_2
1	0	51	0
2	0	7	0
3	0	45	0
4	0	55	0
5	0	57	0
6	0	55	0
7	0	57	0
8	0	42	0
9	0	4	0
10	0	4	0

ASSUMPTIONS

Considered 'order_purchase_timestamp', 'order_delivered_customer_date' and 'order_estimated_delivery_date' from orders table as the sources for finding the days between purchasing – delivery - estimated delivery.

INSIGHTS

The data provides the days between purchase and actual delivery, purchase and estimated delivery, delivery date and estimated delivery. It gives a clear insight over how the product when ordered takes what amount of time to reach the ultimate customer.

RECOMMENDATIONS

This data can help in understanding the order delivery cycle, if clubbed with customer states can help in examining the states where there is high delivery time for correction purposes.

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

time_to_delivery = order_delivered_customer_date-order_purchase_timestampx`

diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

Ans 5.2

SQL QUERY

```
select ifnull(abs(DATE_DIFF(DATE(order_delivered_customer_date),DATE(o
rder_purchase_timestamp),DAY)),0) as time_to_delivery ,
ifnull(abs(DATE_DIFF(DATE(order_estimated_delivery_date),DATE(order_de
livered_customer_date),DAY)),0) as diff_estimated_deliveryfrom `Target
_CS.orders`
```

RESULTS

Query results

JOB INFORMATION | RESULTS

Row	time_to_delivery	diff_estimated_de
1	30	12
2	31	29
3	36	17
4	31	2
5	33	1
6	30	2
7	44	4
8	41	4
9	37	1
10	34	5

ASSUMPTIONS

Considered 'order_purchase_timestamp', 'order_delivered_customer_date' and 'order_estimated_delivery_date' from orders table as the sources for finding the days between purchasing – delivery - estimated delivery.

INSIGHTS

This data helps in understanding the time taken for each order when purchased takes how much to time to deliver at the customer end which is depicted by 'time_to_delivery' column in result.
In the similar manner, the difference of days between estimated delivery date to the actual delivery date.

RECOMMENDATIONS

This data can help in understanding the order delivery cycle, computation of average time taken which can help in identifying and focusing on the orders where time taken is more than the average, accordingly a further analysis on the data over state- wise can target those more time taking orders.

3.Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

Ans5.3

SQL QUERY

```
select c.customer_state,
ROUND(avg(ot.freight_value),2) as avg_freight_val,
ROUND(AVG(ifnull(abs(DATE_DIFF(DATE(o.order_delivered_customer_date),D
ATE(o.order_purchase_timestamp),DAY)),0)),2) as time_to_delivery ,
ROUND(AVG(ifnull(abs(DATE_DIFF(DATE(o.order_estimated_delivery_date),D
ATE(o.order_delivered_customer_date),DAY)),0)),2) as diff_estimated_de
livery
from `Target_CS.orders` o
JOIN Target_CS.order_items ot
ON o.order_id =ot.order_id
JOIN Target_CS.customer c
ON o.customer_id = c.customer_id
group by c.customer_state;
```

RESULTS

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	avg_freight_val	time_to_delivery	diff_estimated_de
1	MT	28.17	17.6	15.55
2	MA	38.26	20.96	13.07
3	AL	35.84	23.51	12.33
4	SP	15.15	8.48	11.67
5	MG	20.63	11.73	13.85
6	PE	32.92	17.62	15.07
7	RJ	20.96	14.63	14.65
8	DF	21.04	12.62	12.87
9	RS	21.74	14.89	15.14
10	SE	36.65	20.86	14.36

ASSUMPTIONS

Considered 'order_purchase_timestamp', 'order_delivered_customer_date' and 'order_estimated_delivery_date' from orders and customer_id from customer table to establish a link between orders and customer tables as the sources for finding the days between purchasing – delivery – estimated delivery.

INSIGHTS

Customer state-wise break-up of average freight_value, time_to_delivery and diff_estimated_delivery which helps in clear understanding of the 3 aspects in different states.

RECOMMENDATIONS

Customer state-wise break-up can help in focusing on the states where there is more freight value, high time in delivery and difference between the commitment to customer and actual deliverance is lacking. We can study the different state prospects, the logistic they have, the warehouse capabilities present in the respective states, the delivery mechanism followed by different delivery firms. These measures can help in reducing the addressed factors.

4.Sort the data to get the following:

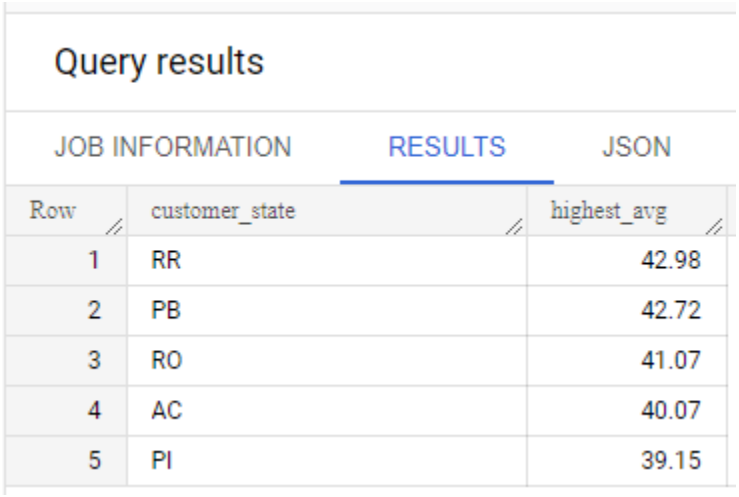
5.Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

Ans5.5

PART ONE SQL QUERY

```
select c.customer_state,
ROUND(avg(ot.freight_value),2) as highest_avg,
from `Target_CS.orders` o
JOIN Target_CS.order_items ot
ON o.order_id =ot.order_id
JOIN Target_CS.customer c
ON o.customer_id = c.customer_id
group by c.customer_state
order by highest_avg DESC
LIMIT 5;
```

RESULTS



Query results

JOB INFORMATION | RESULTS | JSON

Row	customer_state	highest_avg
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

INSIGHTS

This data presents the top 5 customer states with highest average freight value.

RECOMMENDATIONS

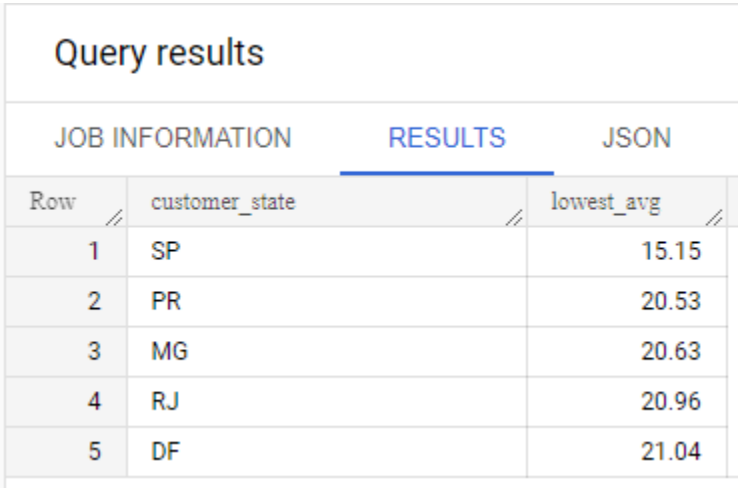
Taking decisions over repositioning or setup of new warehouses to help control the delivery time in these cities (states).

PART TQO SQL QUERY

```
select c.customer_state,
ROUND(avg(ot.freight_value),2) as lowest_avg
from `Target_CS.orders` o
JOIN Target_CS.order_items ot
ON o.order_id =ot.order_id
JOIN Target_CS.customer c
ON o.customer_id = c.customer_id
group by c.customer_state
```

```
order by lowest_avg
LIMIT 5;
```

RESULTS



Query results

JOB INFORMATION | RESULTS | JSON

Row	customer_state	lowest_avg
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04

INSIGHTS

This data presents the top 5 customer states with lowest average freight value.

RECOMMENDATIONS

Comparison of the data with the delivery time taken, the customer count in these cities with respect to the states can help in understanding of the logistic structure.

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

Ans 5.6

PART ONE SQL QUERY

```
select c.customer_state, ROUND(AVG(ifnull(abs(DATE_DIFF(DATE(o.order_d
elivered_customer_date),DATE(o.order_purchase_timestamp),DAY)),0)),2)
as time_to_delivery
from `Target_CS.orders` o
JOIN Target_CS.customer c
ON o.customer_id= c.customer_id
group by c.customer_state
order by time_to_delivery DESC
LIMIT 5;
```

RESULTS

Query results

JOB INFORMATION | RESULTS | JSON

Row	customer_state	time_to_delivery
1	AP	26.78
2	RR	26.15
3	AM	25.82
4	AL	23.55
5	PA	23.02

INSIGHTS

This data presents the top 5 customer states with highest average time to delivery.

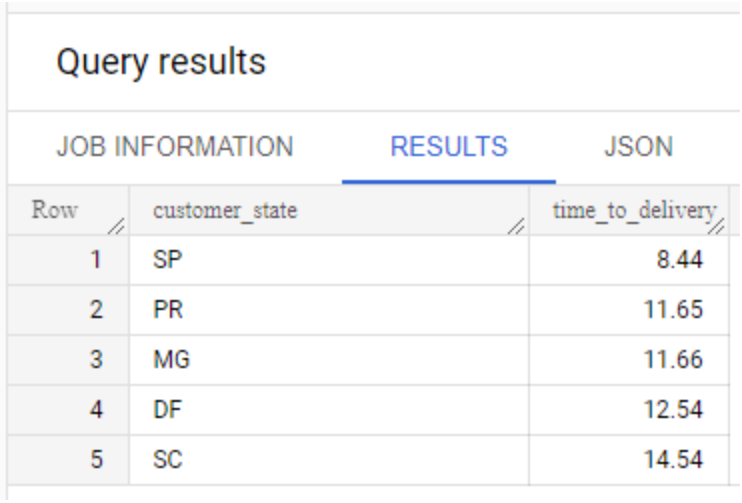
RECOMMENDATIONS

This data helps in analysis of logistic structure setup in these states. Detailed breaking of states into cities over the customer count and orders placed in past can help in re-structuring of the logistic laid.

PART TWO SQL QUERY

```
select c.customer_state, ROUND(AVG(ifnull(abs(DATE_DIFF(DATE(o.order_d
elivered_customer_date),DATE(o.order_purchase_timestamp),DAY)),0)),2)
as time_to_delivery
from `Target_CS.orders` o
JOIN Target_CS.customer c
ON o.customer_id= c.customer_id
group by c.customer_state
order by time_to_delivery ASC
LIMIT 5;
```

RESULTS



Query results

JOB INFORMATION | RESULTS | JSON

Row	customer_state	time_to_delivery
1	SP	8.44
2	PR	11.65
3	MG	11.66
4	DF	12.54
5	SC	14.54

INSIGHTS

This data presents the top 5 customer states with lowest average time to delivery.

RECOMMENDATIONS

These logistic structures can be examined closely to integrate the same in other states especially with highest time to delivery.

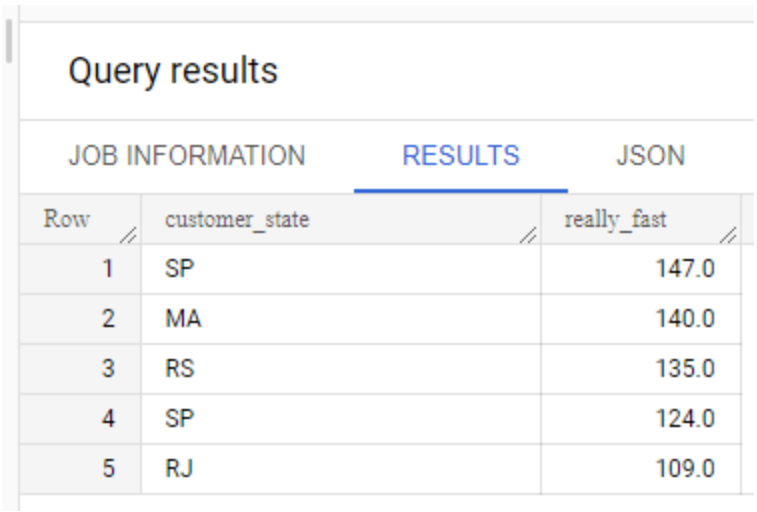
7.Top 5 states where delivery is really fast/ not so fast compared to estimated date

Ans5.7

PART ONE SQL QUERY

```
select c.customer_state,
ROUND(ifnull(DATE_DIFF(DATE(order_estimated_delivery_date),DATE(order_
delivered_customer_date),DAY),0),2) as really_fast
from `Target_CS.orders` o
JOIN Target_CS.customer c
ON o.customer_id=c.customer_id
group by c.customer_state, really_fast
order by really_fast DESC
LIMIT 5;
```

RESULTS



Query results

JOB INFORMATION | RESULTS | JSON

Row	customer_state	really_fast
1	SP	147.0
2	MA	140.0
3	RS	135.0
4	SP	124.0
5	RJ	109.0

INSIGHTS

The top 5 customer states which are really fast in delivery aspect.

RECOMMENDATIONS

These logistic structures can be examined closely to integrate the same in other states especially with highest time to delivery.

PART TWO SQL QUERY

```
select c.customer_state,
ROUND(ifnull(DATE_DIFF(DATE(order_estimated_delivery_date),DATE(order_
delivered_customer_date),DAY),0),2) as not_so_fast
from `Target_CS.orders` o
JOIN Target_CS.customer c
ON o.customer_id=c.customer_id
group by c.customer_state, not_so_fast
order by not_so_fast
LIMIT 5;
```

RESULTS

Query results

JOB INFORMATION | RESULTS | JSON

Row	customer_state	not_so_fast
1	RJ	-188.0
2	ES	-181.0
3	SP	-175.0
4	SP	-167.0
5	SE	-166.0

INSIGHTS

The top 5 customer states which are not so fast in delivery aspect. They effect the freight value.

RECOMMENDATIONS

These states need a change in their logistic structure to overcome the lack in delivery prospect.

6. Payment type analysis:

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

Ans 6.1

SQL QUERY

```
select p.payment_type, extract (month from o.order_purchase_timestamp)
 as month_no,
count(distinct p.order_id) as orders_count
from `Target_CS.payments` p
JOIN `Target_CS.orders` o
ON p.order_id= o.order_id
group by p.payment_type, month_no
order by p.payment_type, month_no;
```

RESULTS

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION D

Row	payment_type	month_no	orders_count
1	UPI	1	1715
2	UPI	2	1723
3	UPI	3	1942
4	UPI	4	1783
5	UPI	5	2035
6	UPI	6	1807
7	UPI	7	2074
8	UPI	8	2077
9	UPI	9	903
10	UPI	10	1056

ASSUMPTIONS

Considered column 'order_purchase_timestamp' from orders table as the time period on the basis of months in which the orders have been placed.

INSIGHTS

This data explains the order count on monthly basis categorize over the different payment types available in the payments table. It helps in identifying the most used payment method.

RECOMMENDATIONS

Payment types which are most used can be treated as a medium to create some good payment offers which will further attract the customers to place an order. Least use methods can also be given a chance by a way of promoting them with an offer. On the monthly basis also, these above steps can be taken.

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

Ans 6.2

SQL QUERY

```
select payment_installments, count(distinct order_id) as
orders_count from `Target_CS.payments`
group by payment_installments;
```

RESULTS

Query results

JOB INFORMATION | RESULTS

Row	payment_installm	orders_count
1	0	2
2	1	49060
3	2	12389
4	3	10443
5	4	7088
6	5	5234
7	6	3916
8	7	1623
9	8	4253
10	9	644
11	10	5315

INSIGHTS

Payment installments count categorized on the basis of order count to understand the category of different installments available and how much orders are fitted in each installment pattern.

RECOMMENDATIONS

These patterns can help determine the orders related to customers which tend to have different payment patterns. We can relate the same to build an EMI mechanism which can help in monitoring of payment realization.