TARGET SQL PROJECT

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

    Oder_items table

Field name	Type
order_id	STRING
order_item_id	INTEGER
product_id	STRING
seller_id	STRING
shipping_limit_date	TIMESTAMP
price	FLOAT
freight_value	FLOAT

String type:- order_id, product_id,seller_id

Integer type:- order_item_id.

Timestamp type:- shipping_limit_date

Float type: price, fright_value

2. **Time period for which the data is given**

   We obtain the result by using the min and max functions on order_purchase_timestamp column in the orders table.

   Query:

```
SELECT
      min(order_purchase_timestamp) as start_time_period,
      max(order_purchase_timestamp) as end_time_period
FROM `my-sql-project-387303.Brazil_data.orders`
```

   Query result:

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

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

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

   We can solve it by joining the customers and orders table and get customer_id, city, state, for the given time period.

   Query:

```
SELECT
      c.customer_id,
      c.customer_city,
      c.customer_state
FROM `my-sql-project-387303.Brazil_data.Customer` c
left join `my-sql-project-387303.Brazil_data.orders` o on c.customer_id = o.customer_id;
```

   Query result:

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	customer_id	customer_city	customer_state
1	0735e7e4298a2ebbb4664934...	acu	RN
2	903b3d86e3990db01619a4eb...	acu	RN
3	38c97666e962d4fea7fd6a83e...	acu	RN
4	77c2f46cf580f4874c9a5751c2...	ico	CE
5	4d3ef4cfffb8ad4767c199c36a...	ico	CE
6	3000841b86e1fbe9493b52324...	ico	CE
7	3c325415ccc7e622c66dec4bc...	ico	CE
8	04f3a7b250e3be964f01bf22bc...	ico	CE
9	894202b8ef01f4719a4691e79...	ico	CE
10	9d715b9fb75a9d081c14126c0...	ico	CE

Insights:

Most Customer orders from the following top 5 cities:

```
SELECT customer_city, COUNT(*) as total_customers
FROM my-sql-project-387303.Brazil_data.Customer
GROUP BY customer_city
ORDER BY total_customers DESC
LIMIT 5;
```

Press Alt+F1 for Accessibility

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	customer_city	total_customers
1	sao paulo	15540
2	rio de janeiro	6882
3	belo horizonte	2773
4	brasilia	2131
5	curitiba	1521

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?
    a) We can answer the first part of question by looking at the trend starting from 2016 till 2018.

Query:

```
SELECT t.Year,count(*) Ord_count , Round(sum(t.price)) Ord_value
FROM
(
  select *,
         extract(year from o.order_purchase_timestamp) as Year
  from `my-sql-project-387303.Brazil_data.orders` o
  inner join `my-sql-project-387303.Brazil_data.order_items` oi on o.order_id=oi.order_id
) t
group by t.year
order by t.year;
```

Query Result:

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	Year	Ord_count	Ord_value
1	2016	370	49786.0
2	2017	50864	6155807.0
3	2018	61416	7386051.0

Insights:

We can observe inside the results that there is an increase in the no.of orders and value of orders.

b) We can answer he second part of the question by replicating the same for months over the time period.

Query:

```
SELECT t.Month,count(*) Ord_count , Round(sum(t.price)) Ord_value
FROM
(
  select *,
         extract(month from o.order_purchase_timestamp) as Month
  from `my-sql-project-387303.Brazil_data.orders` o
  inner join `my-sql-project-387303.Brazil_data.order_items` oi on
o.order_id=oi.order_id
) t
group by t.month
order by t.month;
```

Query Result:

Row	Month	Ord_count	Ord_value
1	1	9163	1070343.0
2	2	9623	1091482.0
3	3	11217	1357558.0
4	4	10659	1356575.0
5	5	12061	1502589.0
6	6	10661	1298163.0
7	7	11611	1393539.0
8	8	12158	1428658.0
9	9	4838	624814.0
10	10	5685	713727.0
11	11	8665	1010271.0
12	12	6309	743925.0

Insights:

As we see that order count is maximum in the following months : march, April, may, june,july,augest.

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

We can get the order count in the specific timing of the day by using CTE, case end and group functions on orders tables.

Query:

```
with hour_cte as
(
  select *,
         extract(hour from order_purchase_timestamp) as Hour
  from `my-sql-project-387303.Brazil_data.orders`
)

select t.Time_of_day, count(*) Ord_count
from
(
  select *,
       case
           when Hour between 0 and 6 then 'Dawn'
           when Hour between 7 and 12 then 'Morning'
           when Hour between 13 and 18 then 'Afternoon'
           when Hour between 19 and 23 then 'Night'
```

```
        end as Time_of_day
  from hour_cte
) t
group by t.time_of_day
order by Ord_count desc;
```

Query Result:

Row	Time_of_day	Ord_count
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

Insights:

We can see that from the query results Brazilians tend to order more during afternoon time followed by the night times.

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

1) Get month on month orders by states

This can be obtained from the Customer,orders,Order_items table and using group by states and months of purchase time_stamp

Query:

```
select c.customer_state,
      extract (month from o.order_purchase_timestamp) as Month,
      count(*) Ord_count,
      round(sum(oi.price)) ord_value
from `my-sql-project-387303.Brazil_data.Customer` c
inner join `my-sql-project-387303.Brazil_data.orders` o on c.customer_id =
o.customer_id
inner join `my-sql-project-387303.Brazil_data.order_items` oi on
o.order_id=oi.order_id
group by c.customer_state,Month
order by c.customer_state,Month
```

Query result:

Row	customer_state	Month	Ord_count	ord_value
1	AC	1	10	1473.0
2	AC	2	9	652.0
3	AC	3	4	554.0
4	AC	4	9	1585.0
5	AC	5	10	3459.0
6	AC	6	7	814.0
7	AC	7	14	2089.0
8	AC	8	7	1085.0
9	AC	9	5	1787.0
10	AC	10	6	735.0

2) Distribution of customers across the states in Brazil

This can be obtained by joining customers,order,order_items table and using group by states and aggregating on count of orders and sum of price of orders placed.

Query:

```
select
      c.customer_state,
      count(*) Ord_count,
      round(sum(oi.price)) Ord_value
from `my-sql-project-387303.Brazil_data.Customer` c
inner join `my-sql-project-387303.Brazil_data.orders` o on c.customer_id=o.customer_id
inner join `my-sql-project-387303.Brazil_data.order_items` oi on o.order_id =
oi.order_id
group by c.customer_state
order by Ord_count desc, Ord_value desc;
```

Query Results:

Row	customer_state	Ord_count	Ord_value
1	SP	47449	5202955.0
2	RJ	14579	1824093.0
3	MG	13129	1585308.0
4	RS	6235	750304.0
5	PR	5740	683084.0
6	SC	4176	520553.0
7	BA	3799	511350.0
8	DF	2406	302604.0
9	GO	2333	294592.0
10	ES	2256	275037.0

Insights:

As we can see from the results most orders have came from SP then RJ followed by MG.

Q4). 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:

```
  with yr17_cte as
(
  select t.year,sum(t.payment_value) ord_cost
  from
  (
    select
          extract (month from o.order_purchase_timestamp) Month,
          extract (year from o.order_purchase_timestamp) Year,
          p.payment_value
    from `my-sql-project-387303.Brazil_data.orders` o
    inner join `my-sql-project-387303.Brazil_data.payments` p on o.order_id =
p.order_id
  )t
  where t.year = 2017 and t.month <=8
  group by t.year
```

```
),

yr18_cte as
(
  select x.year,sum(x.payment_value) ord_cost
  from
  (
    select
          extract (month from o.order_purchase_timestamp) Month,
          extract (year from o.order_purchase_timestamp) Year,
          p.payment_value
    from `my-sql-project-387303.Brazil_data.orders` o
    inner join `my-sql-project-387303.Brazil_data.payments` p on o.order_id =
p.order_id
  )x
  where x.year = 2018 and x.month <=8
  group by x.year
)

select y17.Year Yr2017,
       y18.Year Yr2018,
       round (((y18.ord_cost - y17.ord_cost)/y17.ord_cost)*100,2) per_incr_ord_cost
from yr17_cte as y17,yr18_cte as y18;
```

Query result:

Row	Yr2017	Yr2018	per_incr_ord_cost
1	2017	2018	136.98

Insights:

There is 137% increase in cost of orders from 2017 to 2018.

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

This can be obtained by joining customers, orders, order_items table and using group by states and aggregating on avg.sum of price placed and avg.sum of frieght_value.

Query:

```
select
      c.customer_state,
      round(avg(oi.price)) Mean_price,
      round(sum(oi.price)) Sum_price,
      round(avg(oi.freight_value)) Mean_freight_value,
      round(sum(oi.freight_value)) sum_freight_value
from `my-sql-project-387303.Brazil_data.Customer` c
inner join `my-sql-project-387303.Brazil_data.orders` o on c.customer_id =
o.customer_id
inner join `my-sql-project-387303.Brazil_data.order_items` oi on o.order_id =
oi.order_id
group by c.customer_state
order by c.customer_state;
```

Query Results:

Row	customer_state	Mean_price	Sum_price	Mean_freight_value	sum_freight_value
1	AC	174.0	15983.0	40.0	3687.0
2	AL	181.0	80315.0	36.0	15915.0
3	AM	135.0	22357.0	33.0	5479.0
4	AP	164.0	13474.0	34.0	2789.0
5	BA	135.0	511350.0	26.0	100157.0
6	CE	154.0	227255.0	33.0	48352.0
7	DF	126.0	302604.0	21.0	50625.0
8	ES	122.0	275037.0	22.0	49765.0
9	GO	126.0	294592.0	23.0	53115.0
10	MA	145.0	119648.0	38.0	31524.0

Insights:

Mean price is highest for the state with code PB,
Sum price is highest for the state code SP,
Mean freight value is highest for the state code PB,
Sum freight value is highest for state code SP

Q5) Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

This can be calculated using the orders table and the columns purchase_timestamp, delivery_date, estimated_delivery_date.

Query:

```
select
      order_id,
      order_purchase_timestamp,
      order_estimated_delivery_date,
      order_delivered_customer_date,
      timestamp_diff(order_estimated_delivery_date,order_purchase_timestamp,day) as
ETA_days,
      timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
No_delivery_days,
      timestamp_diff(order_delivered_customer_date,order_estimated_delivery_date,day)
as Late_days
from `my-sql-project-387303.Brazil_data.orders`;
```

Query Result:

Row	order_id	order_purchase_timestamp	order_estimated_delivery_date	order_delivered_customer_date	ETA_days	No_delivery_days	Late_days
201	3c3c6b986a4d292ad99cabd0f...	2017-08-28 15:47:18 UTC	2017-10-09 00:00:00 UTC	2017-09-09 19:16:39 UTC	41	12	-29
202	354ca44327bb555017ef42fd9...	2017-01-30 16:33:23 UTC	2017-03-15 00:00:00 UTC	2017-01-31 19:13:10 UTC	43	1	-42
203	aac01d5ce6fab5aa0e191d71a...	2017-01-28 13:24:20 UTC	2017-03-15 00:00:00 UTC	2017-02-09 11:03:47 UTC	45	11	-33
204	c7da1b57f7d2251252370bc7d...	2017-01-27 21:27:17 UTC	2017-03-15 00:00:00 UTC	2017-02-09 13:21:35 UTC	46	12	-33
205	9ee391a01004a9f330f16ebf9f...	2017-01-27 15:28:18 UTC	2017-03-15 00:00:00 UTC	2017-02-07 14:51:41 UTC	46	10	-35
206	c3ad507aba1f6b47354085e7e...	2017-01-21 19:46:23 UTC	2017-03-15 00:00:00 UTC	2017-02-09 15:44:10 UTC	52	18	-33
207	e3f7a673dd1981e0d7b330c8a...	2017-01-24 09:04:09 UTC	2017-03-15 00:00:00 UTC	2017-02-06 11:23:58 UTC	49	13	-36
208	4d7b93709f08f674ff9d2185fd...	2017-01-31 17:11:49 UTC	2017-03-15 00:00:00 UTC	2017-02-07 14:06:10 UTC	42	6	-35
209	df537c849af44beef86a7ef7de...	2017-01-19 21:48:41 UTC	2017-03-15 00:00:00 UTC	2017-01-30 11:41:52 UTC	54	10	-43
210	5836a9fd173a29689e12bf421...	2017-01-30 14:19:23 UTC	2017-03-15 00:00:00 UTC	2017-02-08 10:12:18 UTC	43	8	-34

Insights:

The following are the top 10 orders where delivery delay was close to 6 months than the expected delivery days.

Row	order_id	order_purchase_timestamp	order_estimated_delivery_date	order_delivered_customer_date	ETA_days	No_delivery_days	Late_days
1	ca07593549f1816d26a572e06...	2017-02-21 23:31:27 UTC	2017-03-22 00:00:00 UTC	2017-09-19 14:36:39 UTC	28	209	181
2	1b3190b2dfa9d789e1f14c05b...	2018-02-23 14:57:35 UTC	2018-03-15 00:00:00 UTC	2018-09-19 23:24:07 UTC	19	208	188
3	440d0d17af552815d15a9e41a...	2017-03-07 23:59:51 UTC	2017-04-07 00:00:00 UTC	2017-09-19 15:12:50 UTC	30	195	165
4	0f4519c5f1c541ddec9f21b3bd...	2017-03-09 13:26:57 UTC	2017-04-11 00:00:00 UTC	2017-09-19 14:38:21 UTC	32	194	161
5	285ab9426d6982034523a855f...	2017-03-08 22:47:40 UTC	2017-04-06 00:00:00 UTC	2017-09-19 14:00:04 UTC	28	194	166
6	2fb597c2f772eca01b1f5c561b...	2017-03-08 18:09:02 UTC	2017-04-17 00:00:00 UTC	2017-09-19 14:33:17 UTC	39	194	155
7	47b40429ed8cce3aee9199792...	2018-01-03 09:44:01 UTC	2018-01-19 00:00:00 UTC	2018-07-13 20:51:31 UTC	15	191	175
8	2fe324febf907e3ea3f2aa9650...	2017-03-13 20:17:10 UTC	2017-04-05 00:00:00 UTC	2017-09-19 17:00:07 UTC	22	189	167
9	2d7561026d542c8dbd8f0daea...	2017-03-15 11:24:27 UTC	2017-04-13 00:00:00 UTC	2017-09-19 14:38:18 UTC	28	188	159

2. Find time_to_delivery & diff_estimated_delivery.

   This can be calculated using the orders table and the columns purchase_timestamp, delivery_date, estimated_delivery_date.

   Query:

```
select
       order_id,
       timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_delivery,
       timestamp_diff(order_delivered_customer_date,order_estimated_delivery_date,
day) as diff_estimated_delivery
from `my-sql-project-387303.Brazil_data.orders`
```

   Query Results:

Row	order_id	time_to_delivery	diff_estimated_delive
1	1950d777989f6a877539f5379...	30	12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	-28
3	65d1e226dfaeb8cdc42f66542...	35	-16
4	635c894d068ac37e6e03dc54e...	30	-1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	-1
7	276e9ec344d3bf029ff83a161c...	43	4
8	54e1a3c2b97fb0809da548a59...	40	4

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

This can be obtained by joining customers, orders, order_items table and using group by states and aggregating on avg of freight_value, time_to_delivery, diff_estimated_delivery.

Query:

```
select
      t.customer_state,
      round(avg(t.freight_value))Mean_freight_value,
      round(avg(t.time_to_delivery))Mean_time_to_delivery,
      round(avg(t.diff_estimated_delivery)) Mean_diff_estimated_delivery
from
(
  select *,
        timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_delivery,
        timestamp_diff(order_delivered_customer_date,order_estimated_delivery_date,
day) as diff_estimated_delivery
  from `my-sql-project-387303.Brazil_data.Customer` c
  inner join `my-sql-project-387303.Brazil_data.orders` o on
c.customer_id=o.customer_id
  inner join `my-sql-project-387303.Brazil_data.order_items` oi on o.order_id =
oi.order_id
) t
group by t.customer_state
order by t.customer_state
```

Query results:

Row	customer_state	Mean_freight_value	Mean_time_to_delive	Mean_diff_estimated
1	AC	40.0	20.0	-20.0
2	AL	36.0	24.0	-8.0
3	AM	33.0	26.0	-19.0
4	AP	34.0	28.0	-17.0
5	BA	26.0	19.0	-10.0
6	CE	33.0	21.0	-10.0
7	DF	21.0	13.0	-11.0
8	ES	22.0	15.0	-10.0
9	GO	23.0	15.0	-11.0
10	MA	38.0	21.0	-9.0

Insights:

Mean freight value is highest for state with code PB

Mean time to delivery is highest for state with code RR

Mean difference in estimated delivery is highest for state with code AL

Mean freight value is lowest for state with code SP

Mean time to delivery is lowest for state with code SP

Mean difference in estimated delivery is lowest for state with code AC.

4 Sort the data to get the following:

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

Highest Avg freight value

Query:

```
select
      t.customer_state,
      round(avg(t.freight_value)) Mean_freight_value,
      round(avg(t.freight_value)) Mean_time_to_delivery,
      round(avg(t.diff_estimated_delivery)) Mean_diff_estimated_delivery
from
(
  select *,
        timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_delivery,
        timestamp_diff(order_delivered_customer_date,order_estimated_delivery_date,
day) as diff_estimated_delivery
  from `my-sql-project-387303.Brazil_data.Customer` c
  inner join `my-sql-project-387303.Brazil_data.orders` o on
c.customer_id=o.customer_id
  inner join `my-sql-project-387303.Brazil_data.order_items` oi on o.order_id =
oi.order_id
)t
group by t.customer_state
order by Mean_freight_value desc
limit 5;
```

Query result:

Row	customer_state	Mean_freight_value	Mean_time_to_delive	Mean_diff_estimated
1	PB	43.0	43.0	-12.0
2	RR	43.0	43.0	-17.0
3	RO	41.0	41.0	-19.0
4	AC	40.0	40.0	-20.0
5	PI	39.0	39.0	-11.0

Lowest Avg Freight Value:

Query:

```
select
      t.customer_state,
      round(avg(t.freight_value)) Mean_freight_value,
      round(avg(t.freight_value)) Mean_time_to_delivery,
      round(avg(t.diff_estimated_delivery)) Mean_diff_estimated_delivery
from
(
  select *,
        timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_delivery,
        timestamp_diff(order_delivered_customer_date,order_estimated_delivery_date,
day) as diff_estimated_delivery
  from `my-sql-project-387303.Brazil_data.Customer` c
  inner join `my-sql-project-387303.Brazil_data.orders` o on
c.customer_id=o.customer_id
  inner join `my-sql-project-387303.Brazil_data.order_items` oi on o.order_id =
oi.order_id
)t
group by t.customer_state
order by Mean_freight_value
limit 5;
```

Query result:

Row	customer_state	Mean_freight_value	Mean_time_to_delive	Mean_diff_estimated
1	SP	15.0	15.0	-10.0
2	PR	21.0	21.0	-13.0
3	RJ	21.0	21.0	-11.0
4	DF	21.0	21.0	-11.0
5	MG	21.0	21.0	-12.0

b) Top 5 states with highest/lowest average time to delivery

Highest Avg time to delivery

Query:

```
select
      t.customer_state,
      round(avg(t.freight_value)) Mean_freight_value,
      round(avg(t.freight_value)) Mean_time_to_delivery,
      round(avg(t.diff_estimated_delivery)) Mean_diff_estimated_delivery
from
(
  select *,
        timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_delivery,
        timestamp_diff(order_delivered_customer_date,order_estimated_delivery_date,
day) as diff_estimated_delivery
  from `my-sql-project-387303.Brazil_data.Customer` c
  inner join `my-sql-project-387303.Brazil_data.orders` o on
c.customer_id=o.customer_id
  inner join `my-sql-project-387303.Brazil_data.order_items` oi on o.order_id =
oi.order_id
)t
group by t.customer_state
order by Mean_time_to_delivery desc
limit 5;
```

Query Result:

Row	customer_state	Mean_freight_value	Mean_time_to_delive	Mean_diff_estimated
1	PB	43.0	43.0	-12.0
2	RR	43.0	43.0	-17.0
3	RO	41.0	41.0	-19.0
4	AC	40.0	40.0	-20.0
5	PI	39.0	39.0	-11.0

Lowest Avg time to delivery

Query:

```
select
      t.customer_state,
      round(avg(t.freight_value)) Mean_freight_value,
      round(avg(t.freight_value)) Mean_time_to_delivery,
      round(avg(t.diff_estimated_delivery)) Mean_diff_estimated_delivery
from
(
  select *,
        timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_delivery,
```

```
        timestamp_diff(order_delivered_customer_date,order_estimated_delivery_date,
day) as diff_estimated_delivery
  from `my-sql-project-387303.Brazil_data.Customer` c
  inner join `my-sql-project-387303.Brazil_data.orders` o on
c.customer_id=o.customer_id
  inner join `my-sql-project-387303.Brazil_data.order_items` oi on o.order_id =
oi.order_id
)t
group by t.customer_state
order by Mean_time_to_delivery
limit 5;
```

Query Result:

Row	customer_state	Mean_freight_value	Mean_time_to_delive	Mean_diff_estimated
1	SP	15.0	15.0	-10.0
2	PR	21.0	21.0	-13.0
3	RJ	21.0	21.0	-11.0
4	DF	21.0	21.0	-11.0
5	MG	21.0	21.0	-12.0

C) Top 5 states where delivery is really fast/ not so fast compared to estimated date

Highest Avg diff_estimated_delivery

Query:

```
select
      t.customer_state,
      round(avg(t.freight_value)) Mean_freight_value,
      round(avg(t.freight_value)) Mean_time_to_delivery,
      round(avg(t.diff_estimated_delivery)) Mean_diff_estimated_delivery
from
(
  select *,
        timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_delivery,
        timestamp_diff(order_delivered_customer_date,order_estimated_delivery_date,
day) as diff_estimated_delivery
  from `my-sql-project-387303.Brazil_data.Customer` c
  inner join `my-sql-project-387303.Brazil_data.orders` o on
c.customer_id=o.customer_id
  inner join `my-sql-project-387303.Brazil_data.order_items` oi on o.order_id =
oi.order_id
)t
group by t.customer_state
order by Mean_diff_estimated_delivery desc
limit 5;
```

Query Results:

Row	customer_state	Mean_freight_value	Mean_time_to_delive	Mean_diff_estimated
1	AL	36.0	36.0	-8.0
2	SE	37.0	37.0	-9.0
3	MA	38.0	38.0	-9.0
4	SP	15.0	15.0	-10.0
5	BA	26.0	26.0	-10.0

Lowest Avg time to delivery

Query:

```
select
      t.customer_state,
      round(avg(t.freight_value)) Mean_freight_value,
      round(avg(t.freight_value)) Mean_time_to_delivery,
      round(avg(t.diff_estimated_delivery)) Mean_diff_estimated_delivery
from
(
  select *,
        timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_delivery,
        timestamp_diff(order_delivered_customer_date,order_estimated_delivery_date,
day) as diff_estimated_delivery
  from `my-sql-project-387303.Brazil_data.Customer` c
  inner join `my-sql-project-387303.Brazil_data.orders` o on
c.customer_id=o.customer_id
  inner join `my-sql-project-387303.Brazil_data.order_items` oi on o.order_id =
oi.order_id
)t
group by t.customer_state
order by Mean_diff_estimated_delivery
limit 5;
```

Query Result:

Row	customer_state	Mean_freight_value	Mean_time_to_delive	Mean_diff_estimated
1	AC	40.0	40.0	-20.0
2	AM	33.0	33.0	-19.0
3	RO	41.0	41.0	-19.0
4	RR	43.0	43.0	-17.0
5	AP	34.0	34.0	-17.0

Q6. Payment type analysis:

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

   This can be obtained by joining orders and payments table and extracting month from purchase timestamp column to group by payment type and month and aggregate on count.

   Query:

```
select t.payment_type, t.month,count(*) as Ord_count
from
(
  select *,
        extract(month from o.order_purchase_timestamp) as month
  from `my-sql-project-387303.Brazil_data.orders` o
  inner join `my-sql-project-387303.Brazil_data.payments` p on o.order_id = p.order_id
) t
group by t.payment_type, t.month
order by t.payment_type, t.month;
```

Query result:

Row	payment_type	month	Ord_count
5	UPI	5	2035
6	UPI	6	1807
7	UPI	7	2074
8	UPI	8	2077
9	UPI	9	903
10	UPI	10	1056
11	UPI	11	1509
12	UPI	12	1160
13	credit_card	1	6103

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

   This can be obtained by joining orders and payments table and grouping by payment installments and aggregating on count.

Query:

```
select p.payment_installments, count(*) as Ord_count
from `my-sql-project-387303.Brazil_data.orders` o
inner join `my-sql-project-387303.Brazil_data.payments` p on o.order_id=p.order_id
group by p.payment_installments
order by p.payment_installments;
```

Query Results:

Row	payment_installment	Ord_count
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

Insights:

Payments installments periods of 22,23 21 months seems to be the least popular choices.

Row	payment_installment	Ord_count
1	22	1
2	23	1
3	0	2
4	21	3
5	16	5
6	17	8
7	14	15
8	13	16
9	20	17
10	24	18

Recommendations derived from insights:

1. As most customer orders came from Sao Paulo and Rio de Janeiro, opening more retail stores in these cities will help serve the demand.

2. From 2016-2018, we can clearly see an exponential increasing trend of orders, to generalize upon this investing more in Brazil would make sense.

3. The sales seem to be soaring in the summer months. The products which are much in demand during this period can identified and inventory for the same can be replenished on a timely basis.

4. The orders seem to be at their peak during afternoon times. If it's an online website that's picking up the order details, we need to make sure that the servers are up on running during these busy times to avoid customer dissatisfaction.

5. Business has grown by 137% between 2017 and 2018, this clearly indicates the large scope for Target as a retailer in Brazil.

6. Mean price of orders is highest in Paraiba state. This means more value products seem to sell more in this state. Hence luxury goods sales can be pushed in this region.

7. There are some exception cases where delay in delivery date was close to 6 months more than the promised date. More probe into such orders and products has to be conducted in order to understand the reason behind the delay. We need to identify whether it's normal for certain kinds of goods or is it a one-time delay resulting from supply chain management delays.

8. Orders that had the fastest delivery times got good feedback from customers. This clearly emphasizes the need to improve delivery times for all the products across all regions to penetrate the market better.

9. Average time to deliver orders is highest in Roraima state. Trying to setup a warehouse stocking all the most sold inventory items can reduce this number.

10. Average difference in time between estimated delivery date and actual delivery date is highest in Alagoas state. Probe must be conducted in understanding what went wrong between the time period where goods have already been shipped from the main warehouse and reaching the local warehouse & from local warehouse to the customer through delivery agent.

11. The payment type of credit card has the most transactions. Care must be taken to maintain the payment gateway servers functioning at all times in order for the customer to have a smooth payment experience.

12. Least popular plans for the installments are for between 21 and 23 months. Target can try to partner with credit card firms to offer more incentive to customer for buying products with installments between the above mentioned period.