DESCRIPTION:

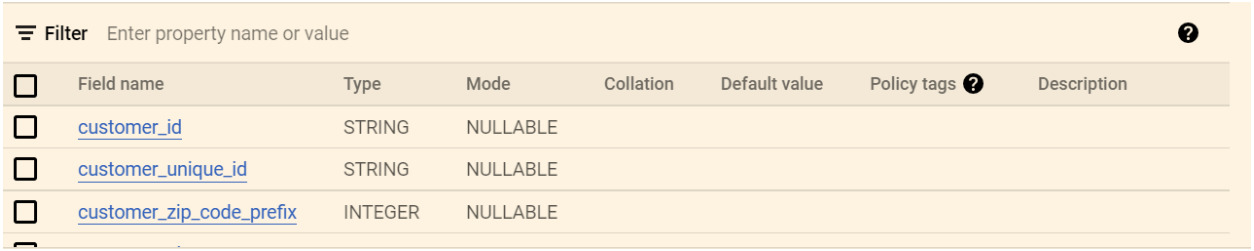
This business case has information of 100k orders from 2016 to 2018 made at AMERICAN RETAIL CORPORATION in Brazil. Its features allow viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers. Performing Exploratory analysis and giving insights and recommendation from the data.

ANALYSIS OF DATASET:

PERFORMING EXPLORATORY ANALYSIS

1. **Data type of columns in a table**:

The schema part of the big query gives us the datatype of any table, attached the screenshot for reference:

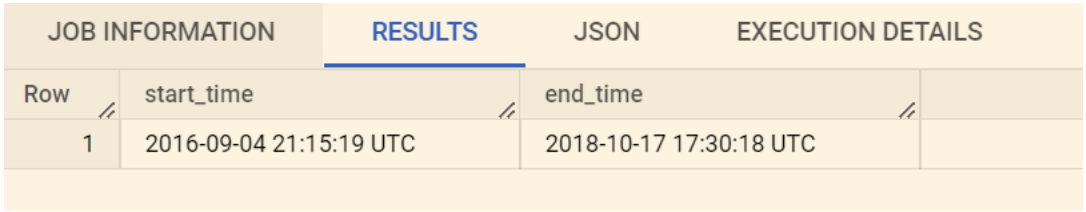


2. **Time period for which the data is given**:

Query:

```
SELECT

min(order_purchase_timestamp) as start_time ,

max(order_purchase_timestamp) as end_time

FROM

target_brazil.orders
```

Output:



3.1 States covered in the dataset:

Query:

```
SELECT
geolocation_state FROM  target_brazil.geolocation GROUP BY geolocation_state
```

Output:

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

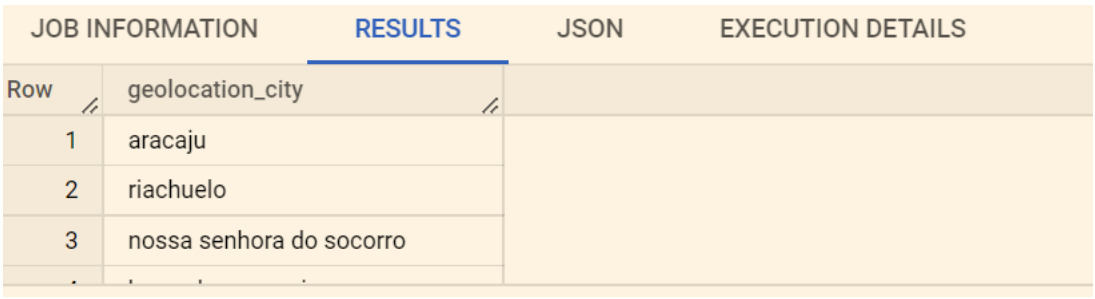
ow	geolocation_state
1	SE
2	AL
3	PI

3.2 Cities covered in the data set:

Query:

```
SELECT
distinct geolocation_city
FROM
target_brazil.geolocation
```

Output:



JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	geolocation_city
1	aracaju
2	riachuelo
3	nossa senhora do socorro

In-depth Exploration:

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

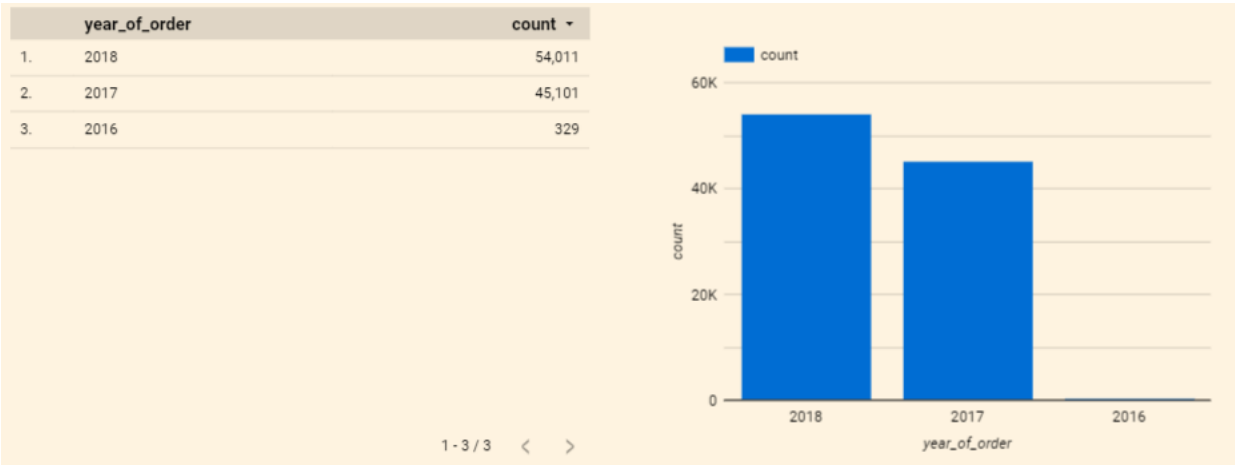
Query: Year wise

```
SELECT

count(order_id) as count ,

extract(year from order_purchase_timestamp) as year_of_order,

FROM

target_brazil.orders

group by order_purchase_timestamp
```

output:

Row	count	year_of_order
1	1	2017
2	1	2017
3	1	2017
4	1	2018

Chart: Year wise



Query: Month wise

```
SELECT

count(order_id) as count ,

 extract(MONTH from order_purchase_timestamp) as odr_month,

FROM

target_brazil.orders

group by extract(MONTH from order_purchase_timestamp)
```

Output:

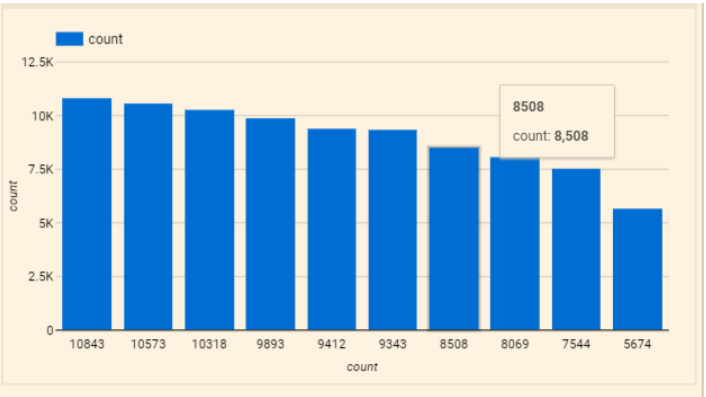
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	count	odr_month
1	7544	11
2	5674	12
3	8508	2
4	9343	4
5	10318	7
6	10573	5
7	4959	10
8	8069	1
9	9412	6
10	4305	9
11	9893	3
12	10843	8

Chart: Month wise

	odr_month	count
1.	August	10,843
2.	May	10,573
3.	July	10,318
4.	March	9,893
5.	June	9,412
6.	April	9,343
7.	February	8,508
8.	January	8,069
9.	November	7,544
10.	December	5,674
11.	October	4,959

1 - 12 / 12



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

4.1 Get month on month orders by states:

Query:

```
SELECT  count(o.order_id) as monthlyorders_perstate,

EXTRACT(month FROM order_purchase_timestamp) as month,c.customer_state,

FROM target_brazil.customers as c LEFT JOIN target_brazil.orders as o

ON c.customer_id=o.customer_id

GROUP BY month ,c.customer_state

ORDER BY month  desc
```

Row	monthlyord...	month	customer_state
1	30	12	RN
2	193	12	SC
3	2357	12	SP
4	691	12	MG
5	283	12	RS
6	127	12	GO
7	192	12	BA
8	271	12	PR
9	783	12	RJ
10	58	12	PA
11	81	12	CE
12	37	12	PB
13	50	12	MT
14	113	12	ES
15	41	12	MA
16	11	12	RO

Chart : For the total number of orders as per state

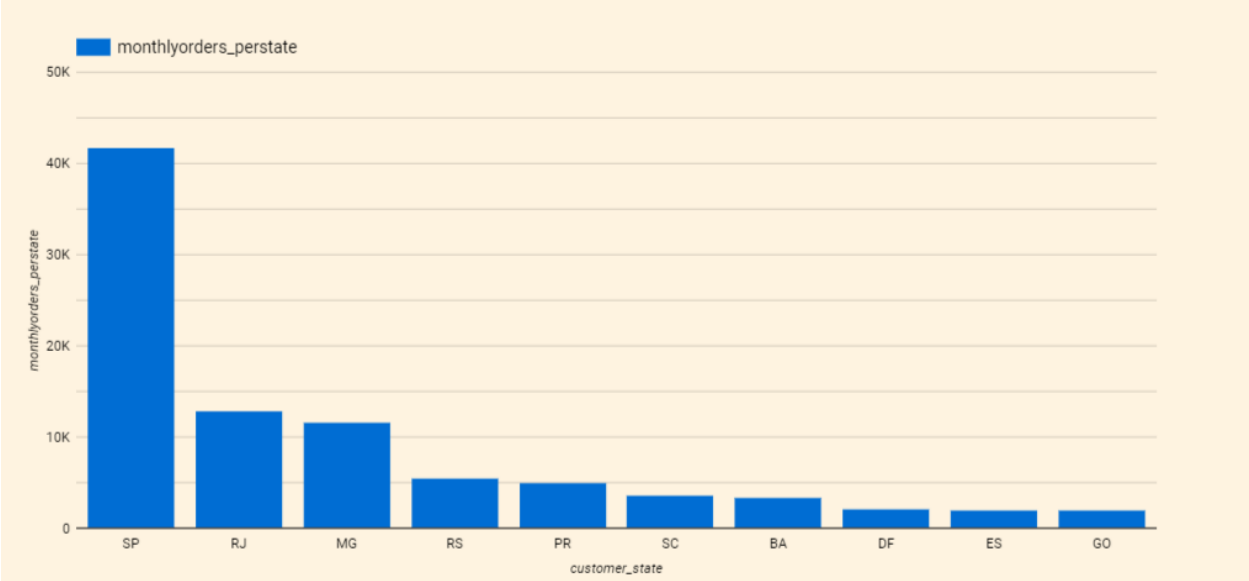
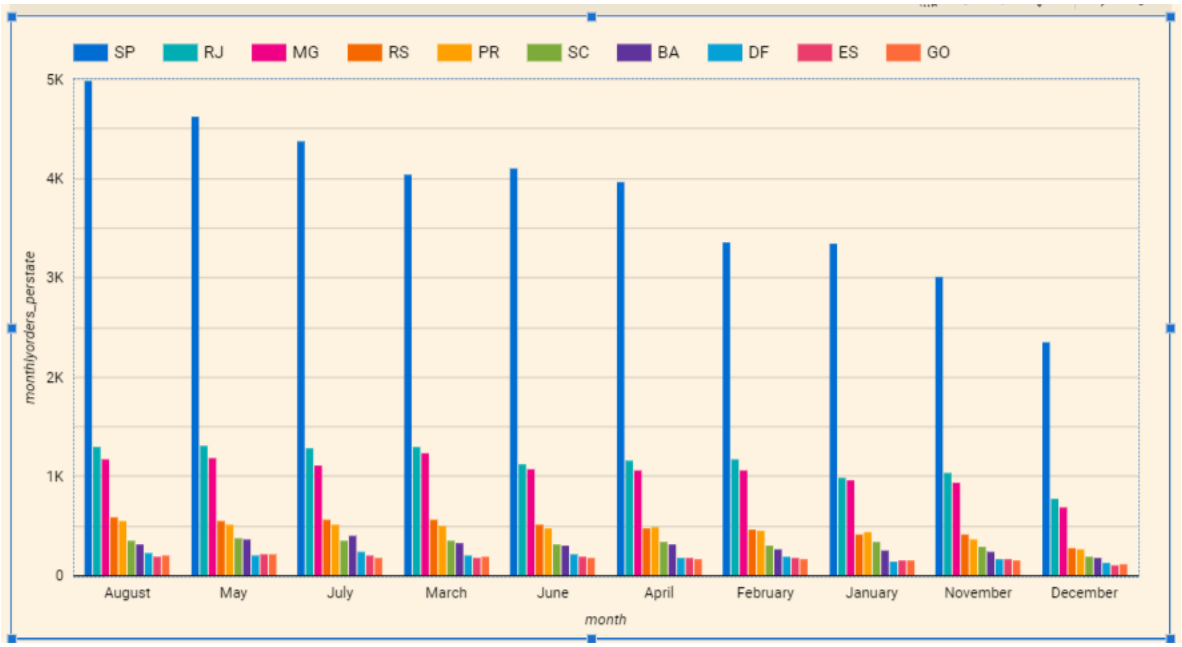


Chart: For state and month wise orders



Insights:

1. Clearly from the graph the “State of São Paulo” has more number of orders .Notable fact is that the SP has as many orders as all other states combined from Brazil.

4.2 Get month on month orders by cities:

Query:

```
SELECT count(o.order_id) as monthlyorders_percity,
EXTRACT(month FROM order_purchase_timestamp)as month,
c.customer_city
FROM target_brazil.customers as c left join target_brazil.orders as o
on c.customer_id=o.customer_id
GROUP BY month ,c.customer_city
order by month  desc
```

The above query gives the number of orders order city wise per every month

output:

Row	monthlyord...	month	customer_city
29	6	12	avare
30	17	12	bauru
31	34	12	belem
32	16	12	betim
33	1	12	caibi
34	1	12	cambe
35	11	12	cotia
36	2	12	crato
37	1	12	cuite
38	2	12	edeia
39	1	12	galia
40	2	12	garca
41	1	12	goias
42	1	12	guara
43	2	12	icara

4.3 How are customers distributed in Brazil?

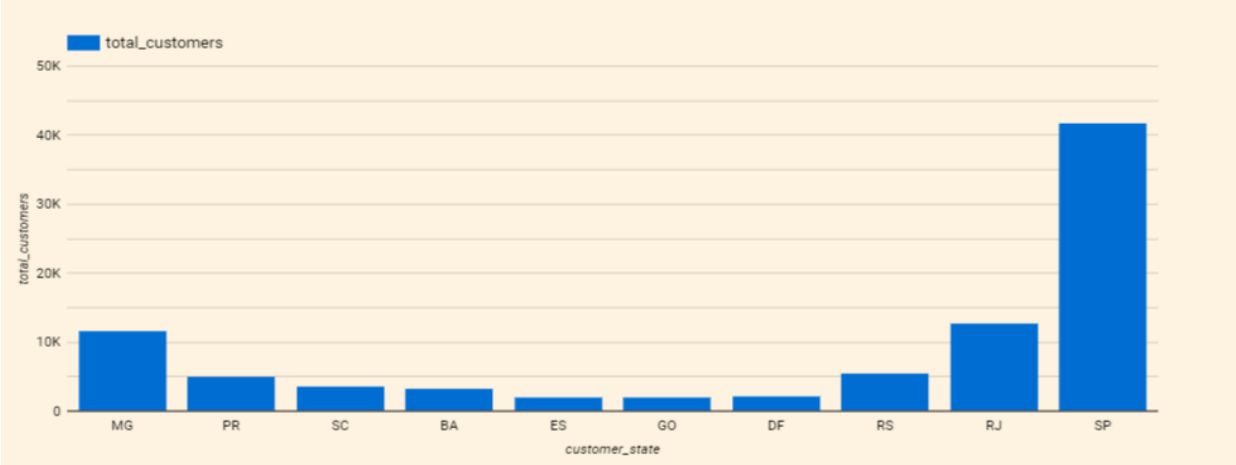
Query:

```
SELECT customer_state, count(customer_id) as total_customers

FROM target_sql.customers

GROUP BY customer_state

ORDER BY count(customer_id) desc
```

Output:

Row	customer_state	total_custo...
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

Chart: percantage of customers per state



Insights:

1. We can clearly see that the customers are more from state “SP”

5. **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

Query:

```
SELECT
 z.*,
 ROUND((total_per_year-prv_year)/prv_year*100,2) AS YOY
FROM (
 SELECT
  SUM(y.total) AS total_per_year,
  y.order_year,
  LAG(SUM(y.total),1) OVER(ORDER BY y.order_year ) AS prv_year
 FROM (
  SELECT
   x.order_id,
   SUM(x.price+x.freight_value) AS total,
   x.order_year,
   x.order_month
  FROM (
   SELECT
    ord.order_id,
    EXTRACT(month
    FROM
     ord.order_purchase_timestamp) AS order_month,
    EXTRACT(year
    FROM
     ord.order_purchase_timestamp) AS order_year,
    ord_it.price,
    ord_it.freight_value
   FROM
    target_brazil.orders AS ord
   LEFT JOIN
    target_brazil.order_items AS ord_it
   ON
    ord.order_id = ord_it.order_id
   WHERE
    EXTRACT(year
    FROM
     ord.order_purchase_timestamp) IN(2017,
     2018)
    AND EXTRACT(month
    FROM
     ord.order_purchase_timestamp) IN(1,
     2,
     3,
```

```
        4,
        5,
        6,
        7,
        8)) AS x
  GROUP BY
    x.order_month,
    x.order_year,
    x.order_id
  ORDER BY
    x.order_year DESC,
    x.order_month DESC) AS y
 GROUP BY
  order_year)AS z
```

output:

Row	total_per_ye...	order_year	prv_year	YOY
1	3610270.14...	2017	null	null
2	8643531.13...	2018	3610270.14...	139.42

Insight:
From the table we can say there is 139.42 % increase in sales from 2017 to 2018 (with Jan to Aug months)

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

Query:

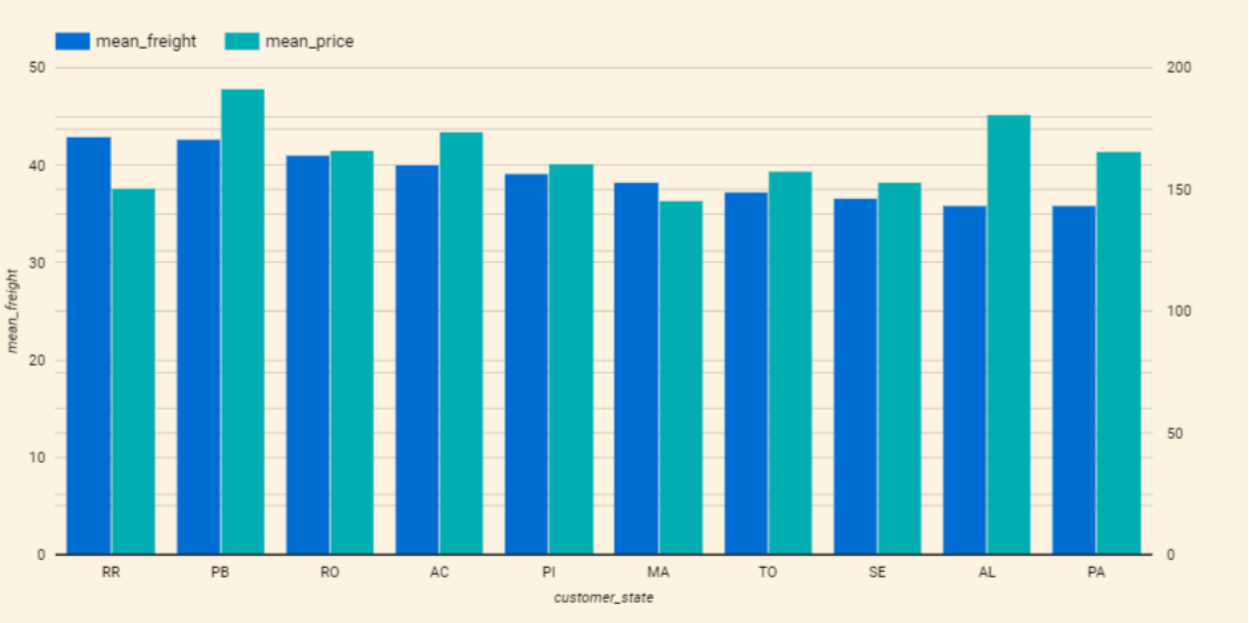
```
SELECT
  o.order_id,
  c.customer_state,
  SUM(price) AS price_per_state,
  SUM(freight_value) AS freight_per_state,
  AVG(price) AS mean_price,
  AVG(freight_value) AS mean_freight
FROM
  target_brazil.customers AS c
LEFT JOIN
  target_brazil.orders AS o
ON
```

```
 c.customer_id=o.customer_id
LEFT JOIN
 target_brazil.order_items AS ot
ON
 o.order_id=ot.order_id
GROUP BY
 o.order_id,
 c.customer_state
```

output:

Row	order_id	customer_state	price_per_st...	freight_per_...	mean_price	mean_freight
1	bf74f34eea55f16dd17b621231...	RN	157.49	38.65	157.49	38.65
2	667fc0af3acc404a6ef971908b...	RN	105.99	48.18	105.99	48.18
3	9f738fc8b806bc3d86ccf78855...	RN	349.6	49.68	174.8	24.84
4	9fd3d5bb20296499ef3fbcaa4d...	CE	149.99	27.59	149.99	27.59
5	ecf6789fa93718435fc6279a4c...	CE	79.99	27.1	79.99	27.1
6	9b41629ccbc3ae4be489cb815...	CE	572.0	90.62	572.0	90.62

chart:



Insights:

1. We can clearly see that the mean freight price of states RR and PB are highest while PA the lowest freight charges.
2. The average price of products is highest In the state of PB followed by AL.

Recommendations:

May be the people in the state of RR and are willing to buy good quality products as seen that the avg price is higher in that state . So we can recommend good quality brands in that state

6. Analysis on sales, freight and delivery time

6.1 Calculate days between purchasing, delivering and estimated delivery:

Query:

```
SELECT
 *,
 x.estimated_days-x.actual_days AS differnce
FROM (
 SELECT
  order_id,
  DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp,day)
AS estimated_days,
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,day)
AS actual_days,
 FROM
  target_brazil.orders
 WHERE
  order_delivered_customer_date IS NOT NULL) AS x
```

output:

Row	order_id	estimated_d...	actual_days	differnce
1	1950d777989f6a877539f5379...	17	30	-13
2	2c45c33d2f9cb8ff8b1c86cc28...	59	30	29
3	65d1e226dfaeb8cdc42f66542...	52	35	17
4	635c894d068ac37e6e03dc54e...	32	30	2
5	3b97562c3aee8bdedcb5c2e45...	33	32	1
6	68f47f50f04c4cb6774570cfde...	31	29	2
7	276e9ec344d3bf029ff83a161c...	39	43	-4
8	54e1a3c2b97fb0809da548a59...	36	40	-4
9	fd04fa4105ee8045f6a0139ca5...	35	37	-2
10	302bb8109d097a9fc6e9cefc5...	28	33	-5

Insights:

From the above table the negative values gives the early delivery and the positive values gives the delay in the delivery

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

Query:

```
select cust.customer_state,count(order_id) as Total_orders,

round(avg(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)),0) As

Avg_time_to_delivery,

round(avg(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,  DAY)),0) as

Avg_diff_estimated_delivery

from target_brazil.orders o inner join target_brazil.customers cust ON

o.customer_id= cust.customer_id

where order_status='delivered'

group by customer_state

order by customer_state
```

output:

Row	customer_state	Total_orders	Avg_time_to...	Avg_diff_est...
1	AC	80	21.0	20.0
2	AL	397	24.0	8.0
3	AM	145	26.0	19.0
4	AP	67	27.0	19.0
5	BA	3256	19.0	10.0
6	CE	1279	21.0	10.0
7	DF	2080	13.0	11.0
8	ES	1995	15.0	10.0
9	GO	1957	15.0	11.0
10	MA	717	21.0	9.0

6.3 Sort the data to get the following:

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

Query:

```
SELECT
  cust.customer_state,
  COUNT(order_id) AS Total_orders,
  ROUND(AVG(DATE_DIFF(order_delivered_customer_date,  order_purchase_timestamp, DAY)),0) AS Avg_time_to_delivery,
  ROUND(AVG(DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date, DAY)),0) AS Avg_diff_estimated_delivery
FROM
  target_brazil.orders o
INNER JOIN
  target_brazil.customers cust
ON
  o.customer_id= cust.customer_id
WHERE
  order_status='delivered'
```

```
GROUP BY
  customer_state
ORDER BY
  Avg_time_to_delivery DESC
LIMIT
  5
```

Output:

Row	customer_state	Total_orders	Avg_time_to...	Avg_diff_est...
1	RR	41	29.0	-16.0
2	AP	67	27.0	-19.0
3	AM	145	26.0	-19.0
4	AL	397	24.0	-8.0
5	PA	946	23.0	-13.0

Insights:

1. From the table we can say that top 5 countries have deliveries delivered within estimated time of delivery. Negative number indicates the difference of days

6.3.2 Top 5 states where delivery is really fast/ not so fast compared to estimated date:

Query:

```
SELECT
  cust.customer_state,
  COUNT(order_id) AS Total_orders,
  ROUND(AVG(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp, DAY)),0) AS Avg_time_to_delivery,
  ROUND(AVG(DATE_DIFF(order_delivered_customer_date,
order_estimated_delivery_date, DAY)),0) AS Avg_diff_estimated_delivery
FROM
  target_brazil.orders o
INNER JOIN
```

```
 target_brazil.customers cust
ON
 o.customer_id= cust.customer_id
WHERE
 order_status='delivered'
GROUP BY
 customer_state
ORDER BY
 Avg_diff_estimated_delivery ASC
LIMIT
 5
```

Output:

Row	customer_state	Total_orders	Avg_time_to...	Avg_diff_est...
1	RR	41	29.0	-16.0
2	AP	67	27.0	-19.0
3	AM	145	26.0	-19.0
4	AL	397	24.0	-8.0
5	PA	946	23.0	-13.0

Insight:

1.

7. Payment type analysis:

7.1 Month and year of count of orders for different payment types:

Query:

```
SELECT
 COUNT(o.order_id) AS total_orders,
 p.payment_type,
 EXTRACT(YEAR
 FROM
  o.order_purchase_timestamp) AS per_year,
 EXTRACT(MONTH
 FROM
  order_purchase_timestamp) AS per_month,
```

```
FROM
  target_brazil.orders o,
  target_brazil.payments p
WHERE
  p.order_id = o.order_id
GROUP BY
  EXTRACT(YEAR
  FROM
    order_purchase_timestamp),
  EXTRACT(MONTH
  FROM
    order_purchase_timestamp),

  p.payment_type
```

Output:

Row	total_orders	payment_type	per_year	per_month
1	1509	UPI	2017	11
2	4377	credit_card	2017	12
3	1325	UPI	2018	2
4	5897	credit_card	2017	11
5	202	voucher	2017	4
6	3086	credit_card	2017	7

7.2 Month on month count of orders;

Query:

```
SELECT
  COUNT(o.order_id) AS total_orders,
  p.payment_type,
  EXTRACT(MONTH
  FROM
    order_purchase_timestamp) AS month,
FROM
  target_brazil.orders o,
  target_brazil.payments p
WHERE
  p.order_id = o.order_id
GROUP BY
```

```
 EXTRACT(MONTH
 FROM
  order_purchase_timestamp),
 p.payment_type
```

output:

Row	total_orders	payment_type	month
1	1509	UPI	11
2	4378	credit_card	12
3	1723	UPI	2
4	5897	credit_card	11
5	572	voucher	4

7.3 Distribution of payment installments and count of orders:

Query:

```
SELECT
 p.payment_installments,
 COUNT(o.order_id) total_orders
FROM
 target_brazil.orders o
RIGHT JOIN
 target_brazil.payments p
ON
 o.order_id = p.order_id
GROUP BY
 p.payment_installments
ORDER BY
 COUNT(o.order_id) desc
```

Output:

Row	payment_in...	total_orders
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920
9	7	1626
10	9	644

Insights:

1. Most customers prefer to pay mostly in lesser installments , might be due to interest getting added