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

1.1. The dataset given contains columns which include datatypes such as String, Integer, Float and Timestamp.

1.2. The time period for which the data is given ranges from **2016-09-04 to 2018-10-17.**

1.3. SELECT DISTINCT customer_city, customer_state FROM `Target_case.customers_table` JOIN `Target_case.orders_table` USING (customer_id);

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

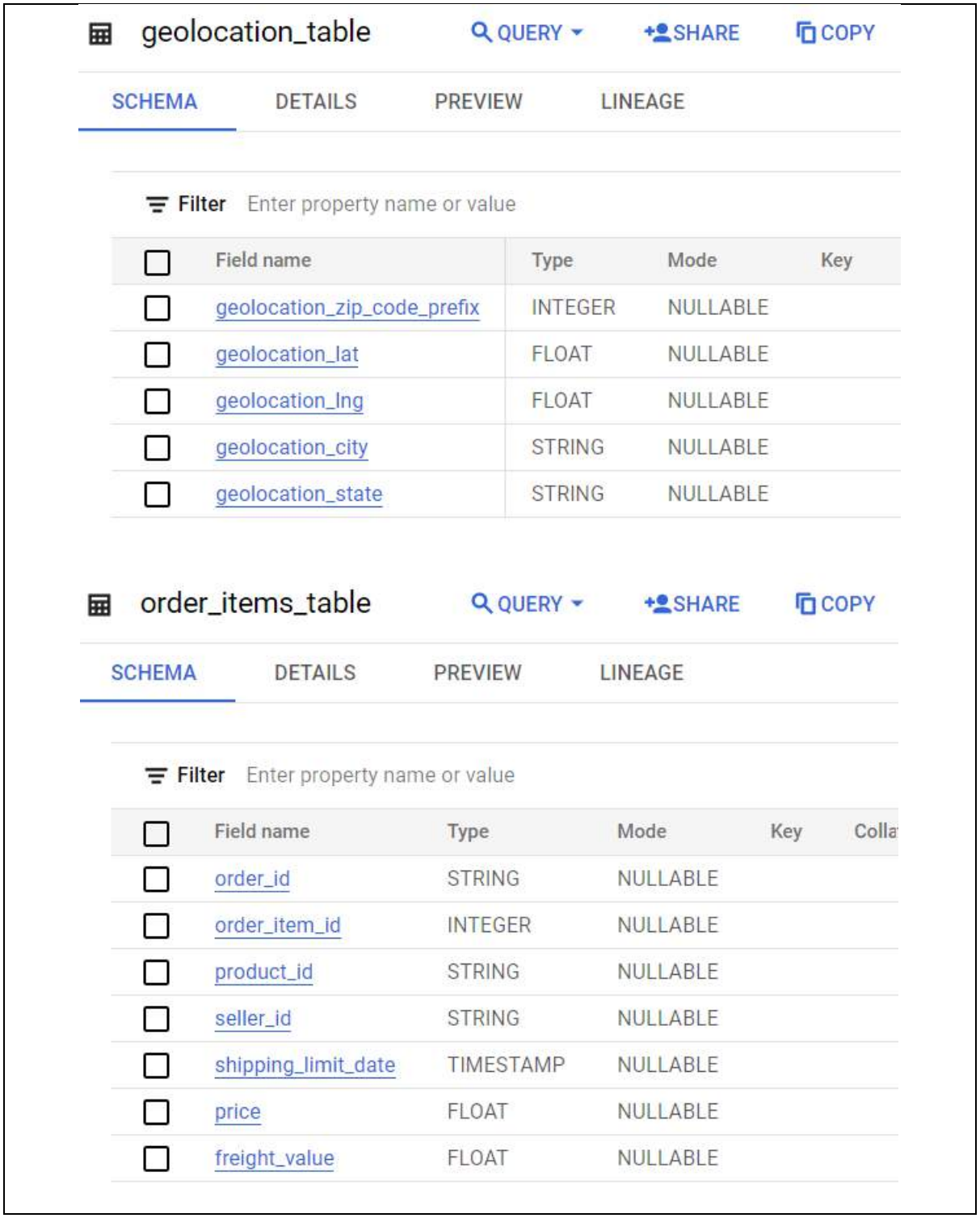
Row	customer_state	customer_city
1	RN	acu
2	CE	ico
3	RS	ipe
4	CE	ipu
5	SC	ita
6	SP	itu
7	SP	jau
8	MG	luz
9	SP	poa

customers_table QUERY SHARE COPY

SCHEMA | DETAILS | PREVIEW | LINEAGE

Filter Enter property name or value

Field name	Type	Mode	Key
customer_id	STRING	NULLABLE	
customer_unique_id	STRING	NULLABLE	
customer_zip_code_prefix	INTEGER	NULLABLE	
customer_city	STRING	NULLABLE	
customer_state	STRING	NULLABLE	



geolocation_table

QUERY SHARE COPY

SCHEMA DETAILS PREVIEW LINEAGE

Filter Enter property name or value

Field name	Type	Mode	Key
geolocation_zip_code_prefix	INTEGER	NULLABLE	
geolocation_lat	FLOAT	NULLABLE	
geolocation_lng	FLOAT	NULLABLE	
geolocation_city	STRING	NULLABLE	
geolocation_state	STRING	NULLABLE	

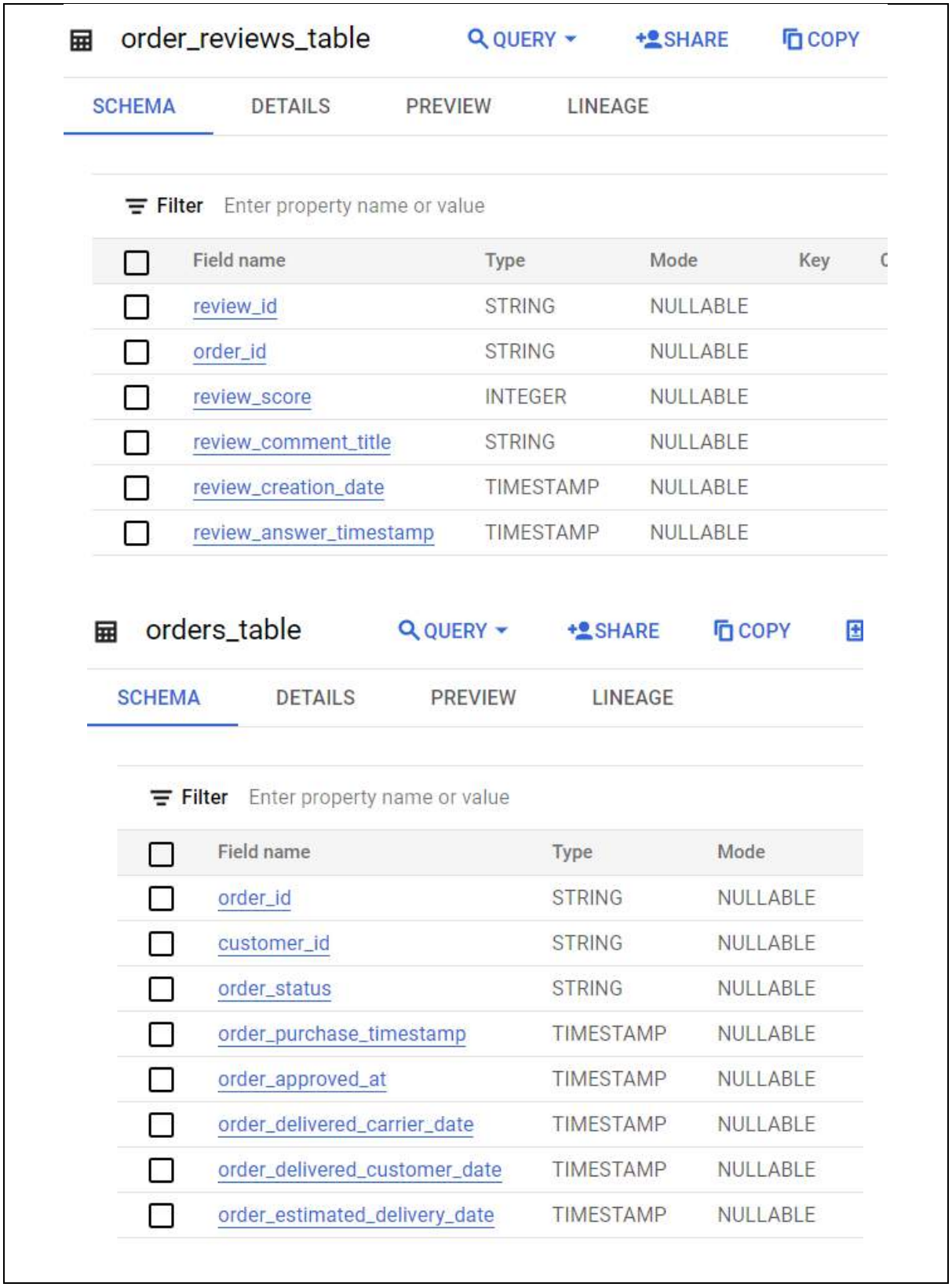
order_items_table

QUERY SHARE COPY

SCHEMA DETAILS PREVIEW LINEAGE

Filter Enter property name or value

Field name	Type	Mode	Key	Colla
order_id	STRING	NULLABLE		
order_item_id	INTEGER	NULLABLE		
product_id	STRING	NULLABLE		
seller_id	STRING	NULLABLE		
shipping_limit_date	TIMESTAMP	NULLABLE		
price	FLOAT	NULLABLE		
freight_value	FLOAT	NULLABLE		



order_reviews_table

QUERY SHARE COPY

SCHEMA DETAILS PREVIEW LINEAGE

Filter Enter property name or value

Field name	Type	Mode	Key
review_id	STRING	NULLABLE	
order_id	STRING	NULLABLE	
review_score	INTEGER	NULLABLE	
review_comment_title	STRING	NULLABLE	
review_creation_date	TIMESTAMP	NULLABLE	
review_answer_timestamp	TIMESTAMP	NULLABLE	

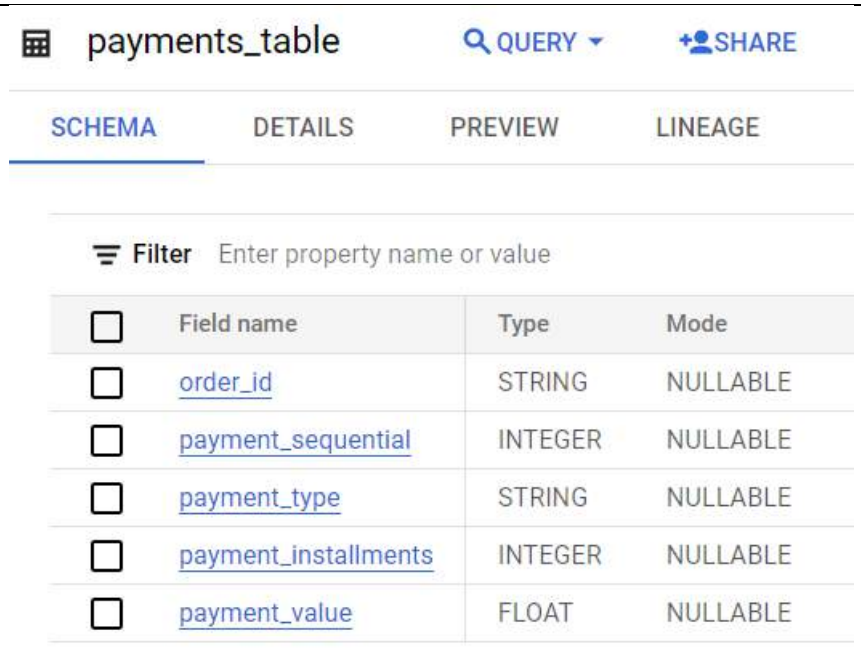
orders_table

QUERY SHARE COPY

SCHEMA DETAILS PREVIEW LINEAGE

Filter Enter property name or value

Field name	Type	Mode
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



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?

```
SELECT
  EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
  COUNT(DISTINCT o.order_id) AS order_count,
  round(sum(p.payment_value),3) as sales_value
FROM
  `Target_case.orders_table` as o
  join `Target_case.payments_table` as p on o.order_id = p.order_id
GROUP BY
  order_year,
  order_month
ORDER BY
  order_year, order_month;
```

order_year	order_month	order_count	sales_value
2016	9	3	252.24
2016	10	324	59090.48
2016	12	1	19.62
2017	1	800	138488.04
2017	2	1780	291908.01
2017	3	2682	449863.6
2017	4	2404	417788.03
2017	5	3700	592918.82
2017	6	3245	511276.38
2017	7	4026	592382.92
2017	8	4331	674396.32

The above query and table represent the order_count and their corresponding payment_value across month on month. For the year 2017, there is gradual increase in count of orders month on month which peaks around the month of November. The payment_value is the highest as well in the month of November for the year 2017 which is in 7-figures.

The year 2018 shows a marginal increase in sales as each month reaches a payment_value in 7 figures compared to only 1 month from the year 2017 and the count of orders is above 6000+ for each month of 2018 up until month 9 which sees a sharp drop to almost 2 digit and a single digit order count for the month 10 (October).

The sales from September of 2016 to October of 2018 follows a curve with a slow and gradual increase in sales month on month with a sharp drop in the month of September and October in 2018.

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

```
SELECT
  CASE
    WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) >= 0 AND EXTRACT(HOUR FROM
o.order_purchase_timestamp) < 6 THEN 'Dawn'
    WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) >= 6 AND EXTRACT(HOUR FROM
o.order_purchase_timestamp) < 12 THEN 'Morning'
    WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) >= 12 AND EXTRACT(HOUR FROM
o.order_purchase_timestamp) < 18 THEN 'Afternoon'
    ELSE 'Night'
  END AS purchase_time_segment,
  COUNT(DISTINCT o.order_id) AS order_count
FROM
  `Target_case.orders_table` as o
GROUP BY
  purchase_time_segment;
```

purchase_time_segment	order_count
Morning	22240
Dawn	4740
Afternoon	38361
Night	34100

Based on the order count, Brazilian customers tend to place the highest orders in the 'afternoon' between 4-5 PM.

Get month on month orders by states

```
SELECT EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
    EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
    c.customer_state,
    COUNT(*) AS order_count
FROM `Target_case.orders_table` as o  join `Target_case.customers_table` as c on  o.customer_id
= c.customer_id
GROUP BY customer_state, year, month
ORDER BY customer_state, year, month;
```

year	month	customer_state	order_count
2017	1	AC	2
2017	2	AC	3
2017	3	AC	2
2017	4	AC	5
2017	5	AC	8
2017	6	AC	4
2017	7	AC	5
2017	8	AC	4
2017	9	AC	5
2017	10	AC	6

Distribution of customers across the states in Brazil

```
select  customer_state , count(*) as count_customers from `Target_case.customers_table`
group by customer_state
```

customer_state	count_customers
RN	485
CE	1336
RS	5466
SC	3637
SP	41746
MG	11635
BA	3380
RJ	12852
GO	2020
MA	747
PE	1652

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 orders_2017 AS (
  SELECT
    SUM(p.payment_value) AS total_payment_2017
  FROM
    `Target_case.orders_table` o
    JOIN `Target_case.payments_table` p ON o.order_id = p.order_id
  WHERE
    EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017
    AND EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
),
orders_2018 AS (
  SELECT
    SUM(p.payment_value) AS total_payment_2018
  FROM
    `Target_case.orders_table` o
    JOIN `Target_case.payments_table` p ON o.order_id = p.order_id
  WHERE
    EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2018
    AND EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
)
SELECT
  2018 AS year,
  ((o18.total_payment_2018 - o17.total_payment_2017) / o17.total_payment_2017) * 100 AS
percentage_increase
FROM
  orders_2018 o18,
```

```
orders_2017 o17;
```

year	percentage_increase
2018	136.9768716466...

Group Data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
SELECT
  customer_state AS state,
  round(AVG(freight_value),2) AS mean_freight_value,
  round(AVG(time_to_delivery),2) AS mean_time_to_delivery,
  round(AVG(diff_estimated_delivery),2) AS mean_diff_estimated_delivery
FROM (
  SELECT
    o.order_id,
    c.customer_state,
    TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY) AS
time_to_delivery,
    TIMESTAMP_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY)
AS diff_estimated_delivery,
    oi.freight_value
  FROM
    `Target_case.orders_table` AS o
  JOIN
    `Target_case.customers_table` AS c ON o.customer_id = c.customer_id
  JOIN
    `Target_case.order_items_table` AS oi ON o.order_id = oi.order_id
) AS subquery
GROUP BY
  state;
```

state	mean_freight_value	mean_time_to_delive	mean_diff_estimated
MT	28.17	17.51	13.64
MA	38.26	21.2	9.11
AL	35.84	23.99	7.98
SP	15.15	8.26	10.27
MG	20.63	11.52	12.4
PE	32.92	17.79	12.55
RJ	20.96	14.69	11.14
DF	21.04	12.5	11.27
RS	21.74	14.71	13.2
SE	36.65	20.98	9.17
PR	20.53	11.48	12.53

Mean & Sum of price and freight value by customer state

```
select round(sum(oi.price),2) as sum_price, round(avg(oi.price),2) as mean_price ,
round(sum(oi.freight_value),2) as sum_freight , round(avg(oi.freight_value),2) as
mean_freight,c.customer_state
from `Target_case.order_items_table` as oi
join `Target_case.orders_table` as o on oi.order_id = o.order_id
join `Target_case.customers_table` as c on o.customer_id=c.customer_id
group by c.customer_state
```

sum_price	mean_price	sum_freight	mean_freight	customer_state
5202955.05	109.65	718723.07	15.15	SP
1824092.67	125.12	305589.31	20.96	RJ
683083.76	119.0	117851.68	20.53	PR
520553.34	124.65	89660.26	21.47	SC
302603.94	125.77	50625.5	21.04	DF
1585308.03	120.75	270853.46	20.63	MG
178947.81	165.69	38699.3	35.83	PA
511349.99	134.6	100156.68	26.36	BA
294591.95	126.27	53114.98	22.77	GO
750304.02	120.34	135522.74	21.74	RS
49621.74	157.53	11732.68	37.25	TO

Calculate days between purchasing, delivering and estimated delivery

```
select order_id,
ifnull(date_diff(order_delivered_customer_date , order_purchase_timestamp , DAY),0) as
delivery_timeline, date_diff(order_estimated_delivery_date,order_purchase_timestamp,day) as
estimated_delivery_timeline
from `Target_case.orders_table`
order by 2 desc;
```

order_id	delivery_timeline	estimated_delivery_t
ca07593549f1816d26a572e...	209	28
1b3190b2dfa9d789e1f14c0...	208	19
440d0d17af552815d15a9e4...	195	30
0f4519c5f1c541ddec9f21b3...	194	32
285ab9426d6982034523a8...	194	28
2fb597c2f772eca01b1f5c56...	194	39
47b40429ed8cce3aee91997...	191	15
2fe324febf907e3ea3f2aa96...	189	22
2d7561026d542c8dbd8f0da...	188	28
437222e3fd1b07396f1d9ba...	187	42
c27815f7e3dd0b926b58552...	187	25

Find time_to_delivery & diff_estimated_delivery

```
SELECT
  order_id,
  TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
time_to_delivery,
  TIMESTAMP_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS
diff_estimated_delivery
FROM
 `Target_case.orders_table`;
```

order_id	time_to_delivery	diff_estimated_delive
1950d777989f6a877539f5379...	30	-12
2c45c33d2f9cb8ff8b1c86cc28...	30	28
65d1e226dfaeb8cdc42f66542...	35	16
635c894d068ac37e6e03dc54e...	30	1
3b97562c3aee8bdedcb5c2e45...	32	0
68f47f50f04c4cb6774570cfde...	29	1
276e9ec344d3bf029ff83a161c...	43	-4
54e1a3c2b97fb0809da548a59...	40	-4
fd04fa4105ee8045f6a0139ca5...	37	-1
302bb8109d097a9fc6e9cefc5...	33	-5
66057d37308e787052a32828...	38	-6

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

Descending

```
  SELECT
  customer_state AS state,
  round(AVG(freight_value),2) AS mean_freight_value,
  round(AVG(time_to_delivery),2) AS mean_time_to_delivery,
  round(AVG(diff_estimated_delivery),2) AS mean_diff_estimated_delivery
FROM (
  SELECT
    o.order_id,
    c.customer_state,
    TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY) AS
time_to_delivery,
    TIMESTAMP_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY) AS
diff_estimated_delivery,
    oi.freight_value
  FROM
    `Target_case.orders_table` AS o
```

```
 JOIN
   `Target_case.customers_table` AS c ON o.customer_id = c.customer_id
 JOIN
   `Target_case.order_items_table` AS oi ON o.order_id = oi.order_id
) AS subquery
GROUP BY
 state
order by
 mean_freight_value desc
limit 5
```

state	mean_freight_value	mean_time_to_delive	mean_diff_estimated
RR	42.98	27.83	17.43
PB	42.72	20.12	12.15
RO	41.07	19.28	19.08
AC	40.07	20.33	20.01
PI	39.15	18.93	10.68

Ascending

```
SELECT
 customer_state AS state,
 round(AVG(freight_value),2) AS mean_freight_value,
 round(AVG(time_to_delivery),2) AS mean_time_to_delivery,
 round(AVG(diff_estimated_delivery),2) AS mean_diff_estimated_delivery
FROM (
 SELECT
   o.order_id,
   c.customer_state,
   TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY) AS
time_to_delivery,
   TIMESTAMP_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY) AS
diff_estimated_delivery,
   oi.freight_value
 FROM
   `Target_case.orders_table` AS o
 JOIN
   `Target_case.customers_table` AS c ON o.customer_id = c.customer_id
 JOIN
   `Target_case.order_items_table` AS oi ON o.order_id = oi.order_id
) AS subquery
GROUP BY
 state
order by
 mean_freight_value asc
limit 5
```

state	mean_freight_value	mean_time_to_delive	mean_diff_estimatec
SP	15.15	8.26	10.27
PR	20.53	11.48	12.53
MG	20.63	11.52	12.4
RJ	20.96	14.69	11.14
DF	21.04	12.5	11.27

Top 5 states with highest/lowest average time to delivery

Highest-

```
SELECT
  customer_state AS state,
  round(AVG(freight_value),2) AS mean_freight_value,
  round(AVG(time_to_delivery),2) AS mean_time_to_delivery,
  round(AVG(diff_estimated_delivery),2) AS mean_diff_estimated_delivery
FROM (
  SELECT
    o.order_id,
    c.customer_state,
    TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY) AS
time_to_delivery,
    TIMESTAMP_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY)
AS diff_estimated_delivery,
    oi.freight_value
  FROM
    `Target_case.orders_table` AS o
  JOIN
    `Target_case.customers_table` AS c ON o.customer_id = c.customer_id
  JOIN
    `Target_case.order_items_table` AS oi ON o.order_id = oi.order_id
) AS subquery
GROUP BY
  state
order by
mean_time_to_delivery desc
```

state	mean_freight_value	mean_time_to_delive
RR	42.98	27.83
AP	34.01	27.75
AM	33.21	25.96
AL	35.84	23.99
PA	35.83	23.3

Lowest-

```
SELECT
  customer_state AS state,
  round(AVG(freight_value),2) AS mean_freight_value,
  round(AVG(time_to_delivery),2) AS mean_time_to_delivery,
  round(AVG(diff_estimated_delivery),2) AS mean_diff_estimated_delivery
FROM (
  SELECT
    o.order_id,
    c.customer_state,
    TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY) AS
time_to_delivery,
    TIMESTAMP_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY)
AS diff_estimated_delivery,
    oi.freight_value
  FROM
    `Target_case.orders_table` AS o
  JOIN
    `Target_case.customers_table` AS c ON o.customer_id = c.customer_id
  JOIN
    `Target_case.order_items_table` AS oi ON o.order_id = oi.order_id
) AS subquery
GROUP BY
  state
order by
mean_time_to_delivery asc
```

state	mean_freight_value	mean_time_to_delive
SP	15.15	8.26
PR	20.53	11.48
MG	20.63	11.52
DF	21.04	12.5
SC	21.47	14.52

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

Not so fast-

```
SELECT
  customer_state AS state,
  round(AVG(diff_estimated_delivery),2) AS mean_diff_estimated_delivery
FROM (
  SELECT
    o.order_id,
    c.customer_state,
    TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY) AS
time_to_delivery,
    TIMESTAMP_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY)
AS diff_estimated_delivery,
    oi.freight_value
  FROM
    `Target_case.orders_table` AS o
  JOIN
    `Target_case.customers_table` AS c ON o.customer_id = c.customer_id
  JOIN
    `Target_case.order_items_table` AS oi ON o.order_id = oi.order_id
) AS subquery
GROUP BY
  state
order by
mean_diff_estimated_delivery desc
```

state	mean_diff_estimated
AC	20.01
RO	19.08
AM	18.98
AP	17.44
RR	17.43

Really Fast

```
SELECT
  customer_state AS state,
  round(AVG(diff_estimated_delivery),2) AS mean_diff_estimated_delivery
FROM (
  SELECT
    o.order_id,
    c.customer_state,
    TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY) AS
time_to_delivery,
```

```
   TIMESTAMP_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY)
AS diff_estimated_delivery,
   oi.freight_value
 FROM
   `Target_case.orders_table` AS o
 JOIN
   `Target_case.customers_table` AS c ON o.customer_id = c.customer_id
 JOIN
   `Target_case.order_items_table` AS oi ON o.order_id = oi.order_id
) AS subquery
GROUP BY
 state
order by
mean_diff_estimated_delivery asc
```

state	mean_diff_estimated
AL	7.98
MA	9.11
SE	9.17
ES	9.77
BA	10.12

---

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

```
SELECT
 EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
 EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
 p.payment_type,
 COUNT(o.order_id) AS order_count
FROM
 `Target_case.orders_table` AS o
JOIN
 `Target_case.payments_table` AS p ON o.order_id = p.order_id
GROUP BY
 year,month,
 p.payment_type
ORDER BY
 year,month,
 p.payment_type;
```

year	month	payment_type	order_count
2016	9	credit_card	3
2016	10	UPI	63
2016	10	credit_card	254
2016	10	debit_card	2
2016	10	voucher	23
2016	12	credit_card	1
2017	1	UPI	197
2017	1	credit_card	583
2017	1	debit_card	9
2017	1	voucher	61
2017	2	UPI	398

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

```
select payment_installments, count(distinct order_id) as count_installments from
`Target_case.payments_table`
group by payment_installments
order by 1;
```

payment_installment	count_installments
0	2
1	49060
2	12389
3	10443
4	7088
5	5234
6	3916
7	1623
8	4253
9	644
10	5315

Actionable Insights.

- Growing Trend in E-commerce – The month-on-month analysis shows that the order count and payment_value are showing a steady incline reaching peaks in the last quarters for the respective years. The increase in order count also indicates an adoption of online purchasing with the consumers especially towards the end of the year for each given year. This in-turn can motivate the company to invest more in marketing the company and services to it's consumers leading to further increase in revenue.
- Seasonality – From the data formatted, we can conclude that there are certain times of the year where the value of orders and sales is larger than the rest of the year. For this dataset that time would be in the last quarter i.e November (11^{th} Month) in 2017 and September (9^{th} Month) in 2018. Seasonality can in turn help the company to decide what time of the year to run their promotional and marketing campaigns vigorously.
- We have also concluded that the greatest number of orders are placed by the customers in the 'afternoon' time of day. This information can be used for targeted marketing efforts, localized promotions, and optimizing logistics and delivery networks to serve customers more efficiently.
- This in turn has a net positive impact on the Economy as more money is flowing through different channels. The growth in order_count and payment_value month on month indicates a net growth rate on the economy.
- A section which would need improving would be the order fulfilment once the order is placed. In some states it has been observed that the time difference between the estimated delivery time and the actual delivery time is almost 10x times more. Improvement in the logistics infrastructure can help retain order cancellations.

Recommendations

- Improvements can be made in the logistics infrastructure which has been setup and can be targeted state-wise where delivery times are much higher compared to the estimated delivery times.
- Marketing campaigns can be setup based on the observance of seasonality in trends i.e during the last quarter of the year. This can in turn more customers and increase profits.
- Based on the time of day where sales are at the highest i.e 'Afternoon', customer support availability can be organised to ensure support channels are available and responsive to address customer queries.
- Utilizing the information from customer reviews areas of improvement can be identified and help enhance overall customer experience.
- Having an analytical approach to the data can help derive deeper insights about the business and in-turn promote data driven decision making.
- Adopt a continuous improvement mindset as a business by constantly reviewing the available data, monitoring the key performance indicators (KPIs) and making changes according to the current deductions from the dataset.