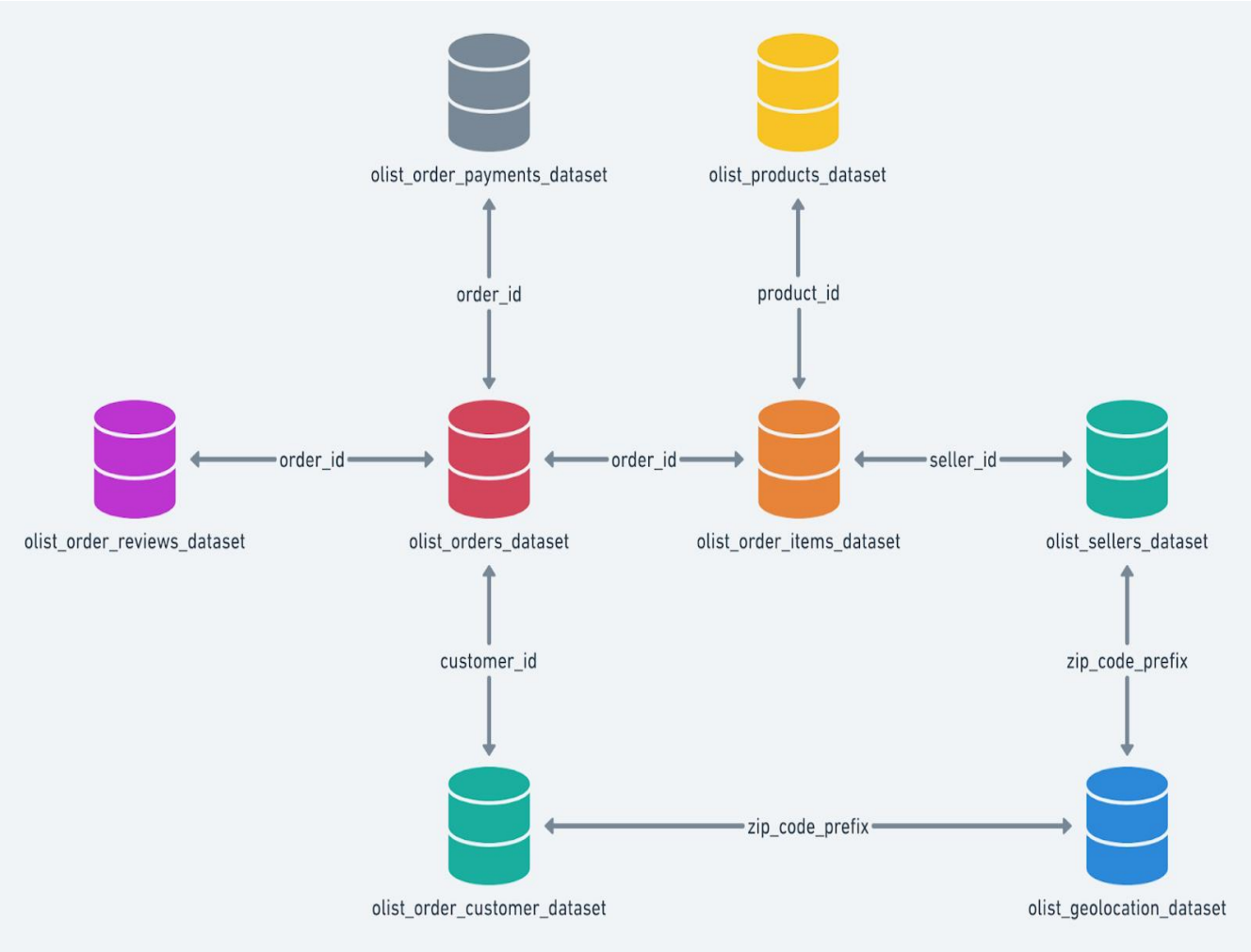
Schema


olist_order_payments_dataset
olist_products_dataset
order_id
product_id
olist_order_reviews_dataset
order_id
olist_orders_dataset
order_id
olist_order_items_dataset
seller_id
olist_sellers_dataset
customer_id
zip_code_prefix
olist_order_customer_dataset
zip_code_prefix
olist_geolocation_dataset

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

1. Data type of all columns in the "customers" table

   Ans:

```
SELECT column_name,data_type from  target_sql.INFORMATION_SCHEMA.COLUMNS
   where table_name='customers'
```

Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

   Insight:Most of the data types are of the string type

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

   Ans:

```
select min(order_purchase_timestamp) as first_order,
max(order_purchase_timestamp)as last_order
from `bigquerydemo-392806.target_sql.orders`
```

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

   Insight: **First order** was placed on **September 4th 2016** and the **last order** was placed on October **17th 2018.**

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

   Ans

```
select  count(distinct customer_city)as customer_city,
count(distinct customer_state)as cutomer_state
from `bigquerydemo-392806.target_sql.customers`c
```

```
join `bigquerydemo-392806.target_sql.orders` o
on c.customer_id=o.customer_id
```

Row	customer_city	cutomer_state
1	4119	27

Insight:Customers belong to 4119 cities and 27 states ordered

2. In-depth Exploration

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

   Ans:

```
select count(case when order_purchase_timestamp >= '2016-09-04 00:00:00' and order_purchase_timestamp
<= '2017-08-31 23:59:59' then order_id end) as sep_2016toaug_2017,
 count(case when order_purchase_timestamp >= '2017-09-01 00:00:00' and order_purchase_timestamp <=
'2018-08-31 23:59:59' then order_id end) as sep_2017toaug_2018,
round((count(case when order_purchase_timestamp >= '2017-09-01 00:00:00' and
order_purchase_timestamp <= '2018-08-31 23:59:59' then order_id end) * 100.0) /
 count(case when order_purchase_timestamp >= '2016-09-04 00:00:00' and order_purchase_timestamp <=
'2017-08-31 23:59:59' then order_id end) - 100, 2
) as growth_percentage from `bigquerydemo-392806.target_sql.orders`;
```

Row	Sep_2016toAug_2017	Sep_2017toAug_2018	growth_percentage
1	23297	76124	226.75

Insight:Taken date from September 2016 to August 2018.There is a growing trend in the no of orders placed .We can see there is a significant jump from 2016 to 2017

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

   Ans

```
with cte as (
        select 'january' as month_name, 1 as month_order
       union all select 'february', 2
       union all select 'march', 3
       union all select 'april', 4
       union all select 'may', 5
       union all select 'june', 6
       union all select 'july', 7
       union all select 'august', 8
       union all select 'september', 9
       union all select 'october', 10
```

```
        union all select 'november', 11
        union all select 'december', 12
        ),
y as(
        select month_order,month_name,
        count(case when extract(year from order_purchase_timestamp) = 2016 then order_id end) as count_at_2016,
        count(case when extract(year from order_purchase_timestamp) = 2017 then order_id end) as count_at_2017,
        count(case when extract(year from order_purchase_timestamp) = 2018 then order_id end) as count_at_2018 from cte
        join `bigquerydemo-392806.target_sql.orders` o
        on cte.month_order=extract(month from o.order_purchase_timestamp)
        group by 1,2
        order by 1)
select month_name,count_at_2016,count_at_2017,count_at_2018 from y;
```

Row	month_name	count_at_2016	count_at_2017	count_at_2018
1	January	0	800	7269
2	February	0	1780	6728
3	March	0	2682	7211
4	April	0	2404	6939
5	May	0	3700	6873
6	June	0	3245	6167
7	July	0	4026	6292
8	August	0	4331	6512
9	September	4	4285	16
10	October	324	4631	4
11	November	0	7544	0
12	December	1	5673	0

Insight:Maximum order happened on **November 2017** and the **least** happened on **November 2016**.

Month wise order in 2018 is much higher than in 2017 and 2016 except for September and October

3. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
    - 0-6 hrs : Dawn
    - 7-12 hrs : Mornings
    - 13-18 hrs : Afternoon
    - 19-23 hrs : Night

Ans

```
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'
when extract(hour from order_purchase_timestamp) between 19 and 23 then 'night'end as
time_of_the_day,count(distinct customer_id)as counting
from `bigquerydemo-392806.target_sql.orders`
group by 1
order by 2 desc limit 1;
```

Row	Time_of_the_day	counting
1	Afternoon	38135

Insight: Found out that out of all the orders Brazilian citizen like to order most in the afternoon

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

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

   Ans

```
with cte as (
        select 'january' as month_name, 1 as month_order
        union all select 'february', 2
        union all select 'march', 3
        union all select 'april', 4
        union all select 'may', 5
        union all select 'june', 6
        union all select 'july', 7
        union all select 'august', 8
        union all select 'september', 9
        union all select 'october', 10
        union all select 'november', 11
        union all select 'december', 12
        )
select
customer_state,
extract(year from order_purchase_timestamp) as year_name,
format_date('%b', order_purchase_timestamp) as month_name,
count(*) as order_count
from `bigquerydemo-392806.target_sql.customers` c
join `bigquerydemo-392806.target_sql.orders` o
on c.customer_id = o.customer_id
join cte mo
```

```
on format_date('%b', order_purchase_timestamp) = mo.month_name
group by 1, 2, 3, mo.month_order
order by 2,month_order;
```

Row	customer_state	year_name	month_name	order_count
1	RR	2016	September	1
2	RS	2016	September	1
3	SP	2016	September	2
4	SP	2016	October	113
5	RS	2016	October	24
6	BA	2016	October	4
7	PR	2016	October	19
8	RJ	2016	October	56
9	RN	2016	October	4
10	MT	2016	October	3

Insight:SP state has the highest number of orders

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

Ans:

```
select  customer_state,count(distinct c.customer_id)as customer_count from `bigquerydemo-392806.target_sql.customers`c
join  `bigquerydemo-392806.target_sql.orders` o
on c.customer_id=o.customer_id
group by 1
order by 2 desc;
```

Row	customer_state	customer_count
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

Insight:SP has most number of customers while RR has least number of customers

4. **Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.**

1. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

   You can use the "payment_value" column in the payments table to get the cost of orders.

   Ans:

```
with x as (
        select  extract(year from o.order_purchase_timestamp)as
        year_of_purchase,round(sum(p.payment_value),2)as summing
        from `bigquerydemo-392806.target_sql.orders` o
        join `bigquerydemo-392806.target_sql.payments` p
        using(order_id)
        where extract(month from order_purchase_timestamp) between 1 and 8
        group by 1
)

select  year_of_purchase,summing as total_payment,round(coalesce((summing-lag(summing,1)over(order by
year_of_purchase ))*100/lag(summing,1)over(order by year_of_purchase ),0),2) as percentage from x
order by 1;
```

Row	year_of_purchase	total_payment	percentage
1	2017	3669022.12	0.0
2	2018	8694733.84	136.98

Insight: We can see that there is **136.98%** increase in the order in 2018 when compared to 2017

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

   Ans:

```
select c.customer_state,round(sum(p.payment_value),2) Total,
round(avg(p.payment_value),2) Average
from `bigquerydemo-392806.target_sql.customers` c
join `bigquerydemo-392806.target_sql.orders` o
using(customer_id)
join `bigquerydemo-392806.target_sql.payments` p
using(order_id)
group by 1
order by 1 desc;
```

Row	customer_state	Total	Average
1	SP	5998226.96	137.5
2	RJ	2144379.69	158.53
3	MG	1872257.26	154.71
4	RS	890898.54	157.18
5	PR	811156.38	154.15
6	SC	623086.43	165.98
7	BA	616645.82	170.82
8	DF	355141.08	161.13
9	GO	350092.31	165.76
10	ES	325967.55	154.71

Insight:SP has **highest total price** and **least average price**.

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

Ans

```
select customer_state,round(sum(freight_value),2) as total_freight_value,round(avg(freight_value),2)as
average_freight from `bigquerydemo-392806.target_sql.orders`o
join `bigquerydemo-392806.target_sql.customers`c using(customer_id)
join `bigquerydemo-392806.target_sql.order_items`od using(order_id)
group by 1
order by 2 desc;
```

Row	customer_state	total_freight_value	average_freight
1	SP	718723.07	15.15
2	RJ	305589.31	20.96
3	MG	270853.46	20.63
4	RS	135522.74	21.74
5	PR	117851.68	20.53
6	BA	100156.68	26.36
7	SC	89660.26	21.47
8	PE	59449.66	32.92
9	GO	53114.98	22.77
10	DF	50625.5	21.04

Insight: The data shows that **SP** has **highest total** but **least average freight value** and **RR** has **lowest total** and **highest average freight_value.**

5. Analysis based on sales, freight and delivery time

1. Find the no. of days taken to deliver each order from the order's purchase date as delivery time. Also, calculate the difference (in days) between the estimated & actual delivery date of an order. Do this in a single query.

   You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

   a. **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
   b. **diff_estimated_delivery** = order_estimated_delivery_date - order_delivered_customer_date

Ans:

```
select  order_id,  date_diff(order_delivered_customer_date, order_purchase_timestamp, day) as time_to_deliver
,date_diff(order_estimated_delivery_date , order_delivered_customer_date,day) as diff_estimated_delivery from `bigquerydemo-392806.target_sql.orders`
where order_delivered_customer_date is not null
order by 2 desc;
```

Row	order_id	time_to_deliver	diff_estimated_deliv
1	ca07593549f1816d26a572e06...	209	-181
2	1b3190b2dfa9d789e1f14c05b...	208	-188
3	440d0d17af552815d15a9e41a...	195	-165
4	0f4519c5f1c541ddec9f21b3bd...	194	-161
5	285ab9426d6982034523a855f...	194	-166
6	2fb597c2f772eca01b1f5c561b...	194	-155
7	47b40429ed8cce3aee9199792...	191	-175
8	2fe324febf907e3ea3f2aa9650...	189	-167
9	2d7561026d542c8dbd8f0daea...	188	-159
10	437222e3fd1b07396f1d9ba8c...	187	-144

Insight: The delivery time of the products varies widely, ranging from **0** to **209** days. The shortest delivery time indicates that some products are eligible for **one day delivery**, while the longest delivery time suggests that some products may face **significant delays**. The minimum difference between the actual and estimated delivery time is **-188** days, which means that some products are delivered almost **six months** after the estimated date. On the other hand, the maximum difference

is **146** days, which means that some products are delivered more than **four months** before the estimated date.

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

Ans:

```
with x as (
        select customer_state,avg(freight_value)as average_freight,row_number()over(order by
        avg(freight_value)desc)as rnk from `bigquerydemo-392806.target_sql.orders`o
        join `bigquerydemo-392806.target_sql.customers`c using(customer_id)
        join `bigquerydemo-392806.target_sql.order_items`od using(order_id)
        group by 1),
y as (
        select customer_state,avg(freight_value)as average_freight,row_number()over(order by
        avg(freight_value)asc)as rnk from `bigquerydemo-392806.target_sql.orders`o
        join `bigquerydemo-392806.target_sql.customers`c using(customer_id)
        join `bigquerydemo-392806.target_sql.order_items`od using(order_id)
        group by 1
        )
        select x.customer_state as upper_5,y.customer_state as lower_5 from x join y using(rnk)
        where x.rnk<=5;
```

Row	upper_5	lower_5
1	RR	SP
2	PB	PR
3	RO	MG
4	AC	RJ
5	PI	DF

Insight: The data has shown that the state where order number is higher the average freight value is lower.I can find a negative correlation between these 2

3. Find out the top 5 states with the highest & lowest average delivery time

Ans:

```
with cte as (
        select *,date_diff(order_delivered_customer_date, order_purchase_timestamp, day) as
        time_to_deliver from `bigquerydemo-392806.target_sql.orders`
        where order_delivered_customer_date is not null),
x as (
        select customer_state,avg(time_to_deliver)as average_time_to_deliver,row_number()over(order by
        avg(time_to_deliver)desc)as rnk from cte
        join `bigquerydemo-392806.target_sql.customers`c using(customer_id)
        group by 1),
y as (
```

```
        select customer_state,avg(time_to_deliver)as average_time_to_deliver,row_number()over(order by
        avg(time_to_deliver)asc)as rnk from cte
        join `bigquerydemo-392806.target_sql.customers`c using(customer_id)
        group by 1
        )
select x.customer_state as upper_5,y.customer_state as lower_5 from x join y using(rnk)
where x.rnk<=5
```

Row	upper_5	lower_5
1	RR	SP
2	AP	PR
3	AM	MG
4	AL	DF
5	PA	SC

Insight:**RR** has the highest average delivery time while **SP** has the least average delivery time

4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
   You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state

   Ans:

```
with cte as (
      select customer_state, round(avg(DATE_DIFF(order_estimated_delivery_date,
      order_delivered_customer_date,DAY)),2) as fast_delivery
      from `bigquerydemo-392806.target_sql.orders` o
      join `bigquerydemo-392806.target_sql.customers` c
      using(customer_id)
      where o.order_status ='delivered'
      group by 1
            ),
x as(
      select customer_state,fast_delivery,dense_rank() over(order by fast_delivery
      desc)as rnk from cte )
select customer_state,fast_delivery from x
where rnk<=5
order by 2 desc
```

Row	customer_state	fast_delivery
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

Insight:On average state AC has the fastest delivery

6. Analysis based on the payments

1. Find the month on month no. of orders placed using different payment types

Ans:

```
with cte as (
        select 'january' as month_name, 1 as month_order
        union all select 'february', 2
        union all select 'march', 3
        union all select 'april', 4
        union all select 'may', 5
        union all select 'june', 6
        union all select 'july', 7
        union all select 'august', 8
        union all select 'september', 9
        union all select 'october', 10
        union all select 'november', 11
        union all select 'december', 12
        ),
y as (
        select extract(year from  o.order_purchase_timestamp ) year_of_order,month_order,month_name,
        sum(case when payment_type='credit_card' then 1 else 0 end)as count_credit_card,
        sum(case when payment_type='voucher' then 1 else 0 end)as count_voucher,
        sum(case when payment_type='debit_card' then 1 else 0 end)as count_debit_card,
        sum(case when payment_type='upi' then 1 else 0 end)as count_upi,
        sum(case when payment_type='not_defined' then 1 else 0 end)as count_not_defined
        from  `bigquerydemo-392806.target_sql.orders` o
        join cte on extract(month from  o.order_purchase_timestamp)=month_order
        join `bigquerydemo-392806.target_sql.payments` using(order_id)

        group by 1,2,3
        order by 1,2
        )
select year_of_order,month_name,count_credit_card,count_voucher,
count_debit_card,count_upi,count_not_defined from y;
```

Row	year_of_order	month_name	count_credit_card	count_voucher	count_debit_card	count_UPI	count_not_defined
1	2016	september	3	0	0	0	0
2	2016	october	254	23	2	63	0
3	2016	december	1	0	0	0	0
4	2017	january	583	61	9	197	0
5	2017	february	1356	119	13	398	0
6	2017	march	2016	200	31	590	0
7	2017	april	1846	202	27	496	0
8	2017	may	2853	289	30	772	0
9	2017	june	2463	239	27	707	0
10	2017	july	3086	364	22	845	0

Insight: People prefer using **credit cards** more as the payment method as it can give flexibility in paying and cashbacks. Least preferred payment method is **debit cards** . Identified that there were some payments made in August and September 2018 that did not belong to any of the four categories of credit cards, debit cards, voucher, and upi.

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

Ans:

```
with cte as (
        select o.order_id,max(p.payment_installments) as max_installment
         from `bigquerydemo-392806.target_sql.orders` o
         join `bigquerydemo-392806.target_sql.payments` p
         on o.order_id = p.order_id
        where p.payment_installments <> 0 and o.order_status<>'canceled'
         group by o.order_id
        )

select max_installment as installment,count(distinct order_id) as order_count from cte
group by 1
order by 1;
```

Row	installment	order_count
1	1	48268
2	2	12363
3	3	10429
4	4	7070
5	5	5227
6	6	3908
7	7	1622
8	8	4251
9	9	644
10	10	5315

Insight: Most orders are done within **3 instalments**, which could suggest that customers prefer to pay in smaller amounts over a shorter period of time.

7.Customer Satisfaction

1. Determine the state with the highest customer satisfaction percentage

Ans:

```
with cte as (select  seller_state,review_score,order_id
             from `bigquerydemo-392806.target_sql.order_items`
             join `bigquerydemo-392806.target_sql.order_reviews`using(order_id)
             join `bigquerydemo-392806.target_sql.sellers`  using(seller_id)
            join `bigquerydemo-392806.target_sql.orders` using(order_id)
            where order_status='delivered'),
x as(select seller_state,sum(case when review_score>2 then 1 else 0 end)/count(review_score) as percentage from cte
   group by 1)
select seller_state,percentage from x
order by 2 desc;
```

Row	seller_state	percentage
1	PA	100.0
2	MT	92.36111111111...
3	MS	91.83673469387...
4	PI	90.90909090909...
5	GO	89.96062992125...
6	RS	89.43955534969...
7	RN	89.28571428571...
8	CE	88.88888888888...
9	BA	88.62179487179...
10	MG	87.44901526628...

Insight:All the ordered delivered in the state PA has 100% satisfaction.(Assuming score>2)

Inference:

1. No of orders in 2018 increased by a amount of 136.98% compared to 2017.
2. No of orders most in SP It can be because of the lowest average freight value, price and the delivery time
3. People prefer credit cards as their payment methods.So adding attractive discounts and cashbacks can attract more customers.
4. All the delivered order in PA got 100% customer satisfaction.
5. It can be seen that customer satisfaction in SP is less than expected .Need to look into this issue since it is most revenue generating state for the company.