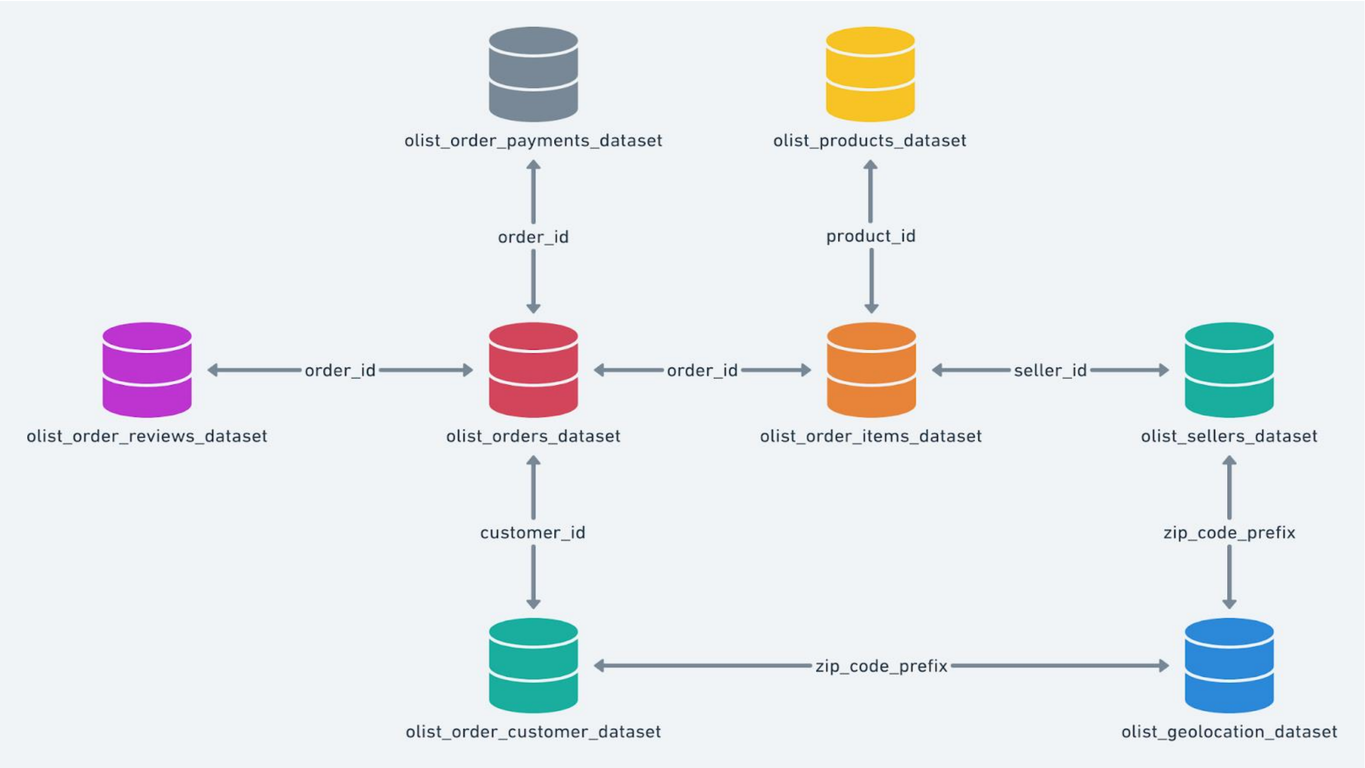
Target – case study using SQL

Context

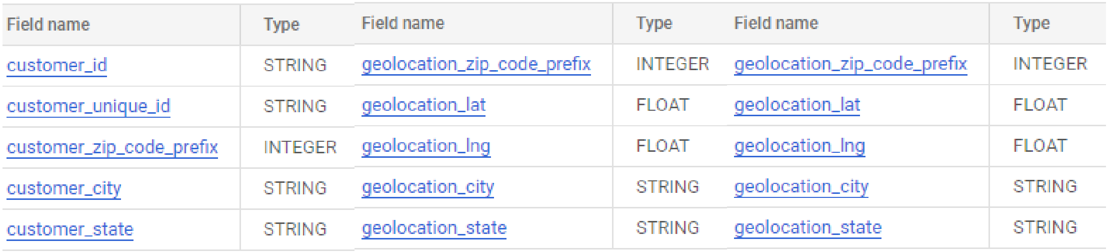
Target is one of the world's most recognized brands and one of America's leading retailers. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

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.

High level overview of relationship between datasets:



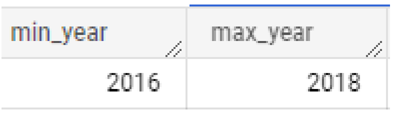
1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset
    1. Data type of columns in a table



Field name	Type	Field name	Type	Field name	Type
customer_id	STRING	geolocation_zip_code_prefix	INTEGER	geolocation_zip_code_prefix	INTEGER
customer_unique_id	STRING	geolocation_lat	FLOAT	geolocation_lat	FLOAT
customer_zip_code_prefix	INTEGER	geolocation_lng	FLOAT	geolocation_lng	FLOAT
customer_city	STRING	geolocation_city	STRING	geolocation_city	STRING
customer_state	STRING	geolocation_state	STRING	geolocation_state	STRING

2. Time period for which the data is given

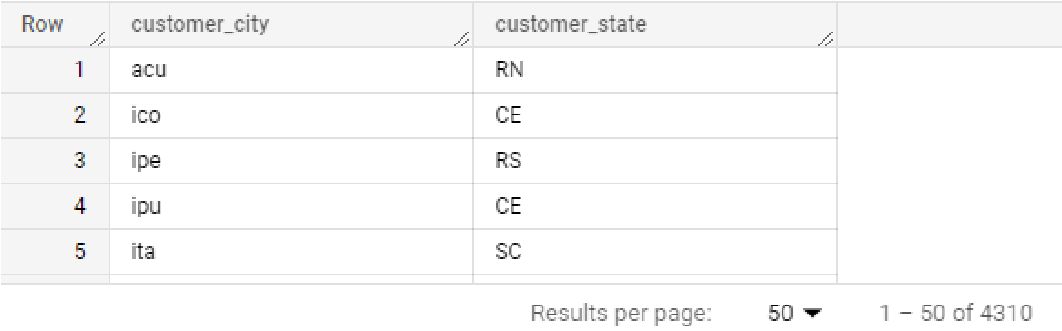
```
select extract(year from min(order_purchase_timestamp)) as min_year,
extract(year from max(order_purchase_timestamp)) as max_year from `my-project-target-382813.Target.orders` ;
```



min_year	max_year
2016	2018

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

```
SELECT distinct customer_city, customer_state FROM `my-project-target-382813.Target.customers` ;
```



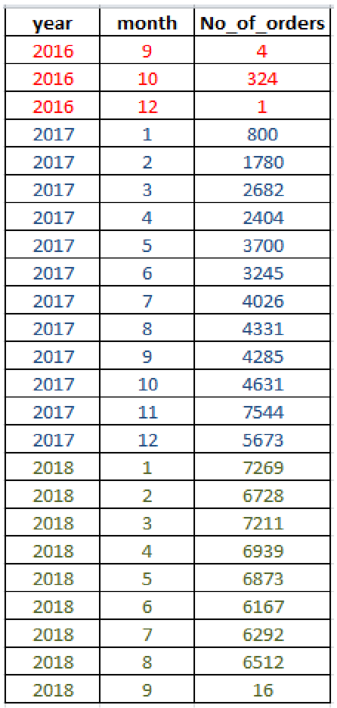
Row	customer_city	customer_state
1	acu	RN
2	ico	CE
3	ipe	RS
4	ipu	CE
5	ita	SC

Results per page: 50 ▾ 1 – 50 of 4310

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?

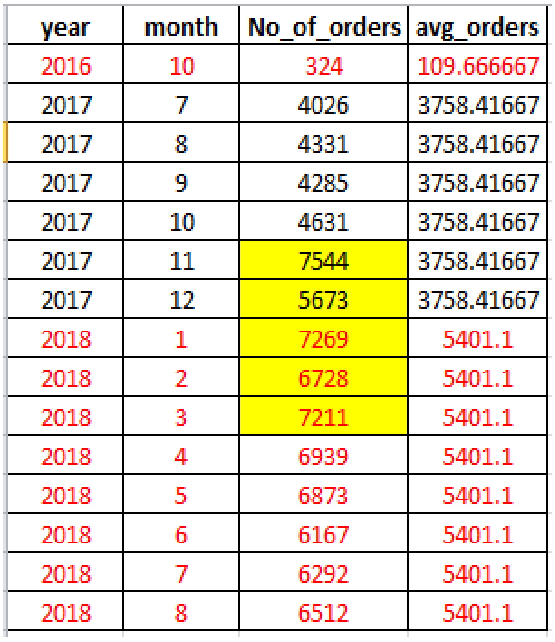
```
SELECT  extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month , count(order_id) as No_of_orders  FROM `my-project-target-382813.Target.orders` group by year, month order by  year,month;
```



year	month	No_of_orders
2016	9	4
2016	10	324
2016	12	1
2017	1	800
2017	2	1780
2017	3	2682
2017	4	2404
2017	5	3700
2017	6	3245
2017	7	4026
2017	8	4331
2017	9	4285
2017	10	4631
2017	11	7544
2017	12	5673
2018	1	7269
2018	2	6728
2018	3	7211
2018	4	6939
2018	5	6873
2018	6	6167
2018	7	6292
2018	8	6512
2018	9	16

Yes, There's a Increased growth on e-commerce in brazil. In the year 2016 there was a slow growth and later at the start of 2017 there was good increase in growth when compared to previous year and then in the year 2018 there was a static no. of orders until august.

```
        select u.* from (select  t.*, avg(No_of_orders)over(partition by year) as avg_orders
        from (SELECT  extract(year from order_purchase_timestamp) as year,
        extract(month from order_purchase_timestamp) as month , count(order_id) as No_of_orde
rs,
        FROM `my-project-target-
382813.Target.orders` group by year, month order by  year,month) t)u
        where No_of_orders > avg_orders order by year, month ;
```

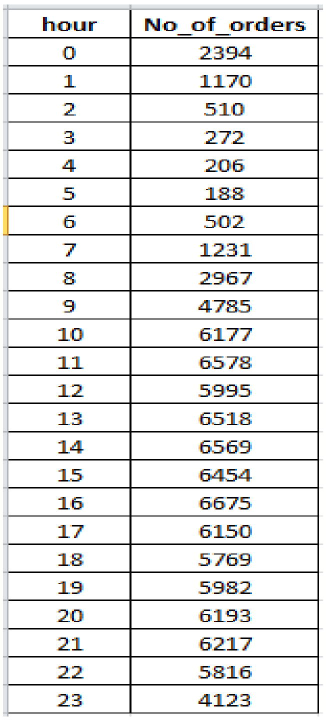


year	month	No_of_orders	avg_orders
2016	10	324	109.666667
2017	7	4026	3758.41667
2017	8	4331	3758.41667
2017	9	4285	3758.41667
2017	10	4631	3758.41667
2017	11	7544	3758.41667
2017	12	5673	3758.41667
2018	1	7269	5401.1
2018	2	6728	5401.1
2018	3	7211	5401.1
2018	4	6939	5401.1
2018	5	6873	5401.1
2018	6	6167	5401.1
2018	7	6292	5401.1
2018	8	6512	5401.1

Yes, At specific months like November and December of 2017, January, February and march of 2018 are peak in no. of orders.

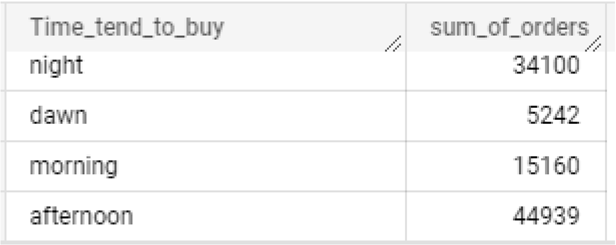
2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

```
SELECT  extract(hour from order_purchase_timestamp) as hour,
count(order_id) as No_of_orders  FROM `my-project-target-
382813.Target.orders`
group by hour order by  hour;
```



hour	No_of_orders
0	2394
1	1170
2	510
3	272
4	206
5	188
6	502
7	1231
8	2967
9	4785
10	6177
11	6578
12	5995
13	6518
14	6569
15	6454
16	6675
17	6150
18	5769
19	5982
20	6193
21	6217
22	5816
23	4123

```
select distinct Time_tend_to_buy,sum(No_of_orders)over(partition by Time_tend_to_buy)
as sum_of_orders
from (with c as (SELECT  extract(hour from order_purchase_timestamp) as hour,
count(order_id) as No_of_orders  FROM `my-project-target-382813.Target.orders`
group by hour order by  hour)
select *, case when hour >= 0 and hour < 7 then 'dawn'
            when hour >= 7 and hour < 11 then 'morning'
            when hour >= 11 and hour < 18 then 'afternoon'
            when hour >=18 and hour <= 23 then 'night'
            end as Time_tend_to_buy from c) u;
```



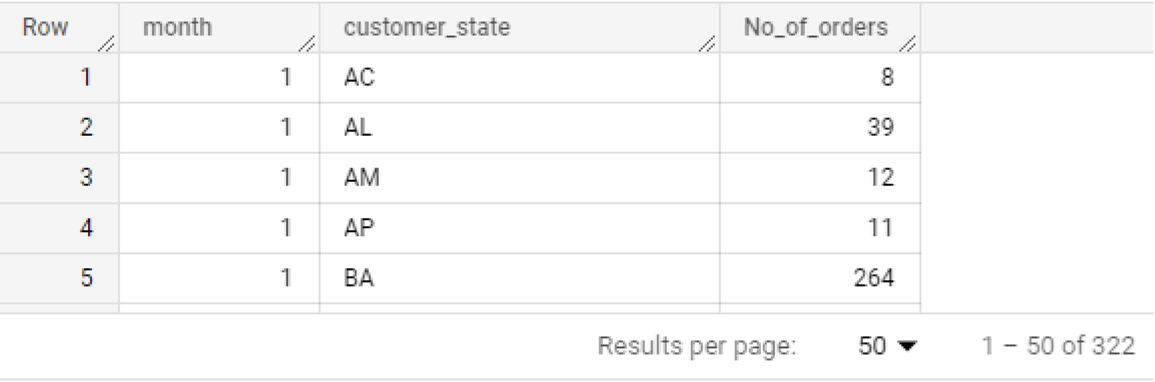
Time_tend_to_buy	sum_of_orders
night	34100
dawn	5242
morning	15160
afternoon	44939

Therefore, Afternoon between 11:00 to 18:00 customers tend to buy more.

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

1. Get month on month orders by states

```
SELECT  extract(month from order_purchase_timestamp) as month , B.customer_state,
count(A.order_id) as No_of_orders  FROM `my-project-target-382813.Target.orders` A
join `my-project-target-382813.Target.customers` B on A.customer_id = B.customer_id
group by B.customer_state, month order by  month, B.customer_state;
```

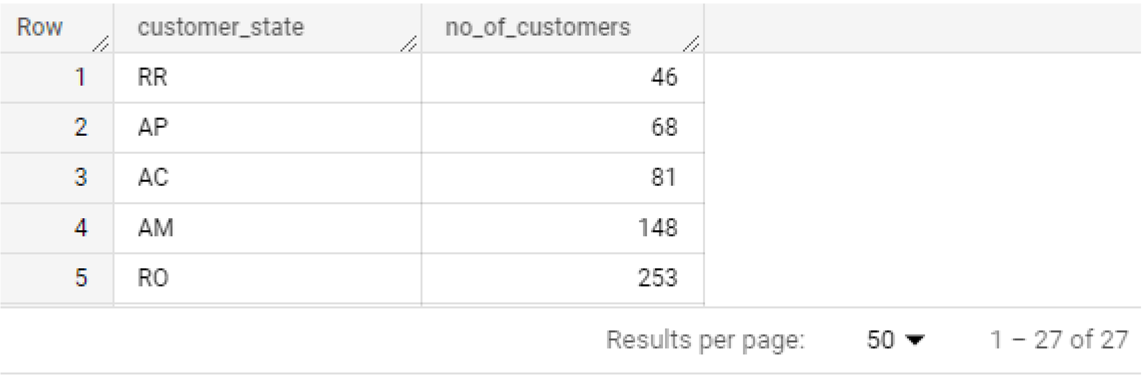


Row	month	customer_state	No_of_orders
1	1	AC	8
2	1	AL	39
3	1	AM	12
4	1	AP	11
5	1	BA	264

Results per page: 50 ▾ 1 – 50 of 322

2. Distribution of customers across the states in Brazil

```
select customer_state, count(customer_id) as no_of_customers from `my-project-target-382813.Target.customers` group by customer_state order by no_of_customers;
```

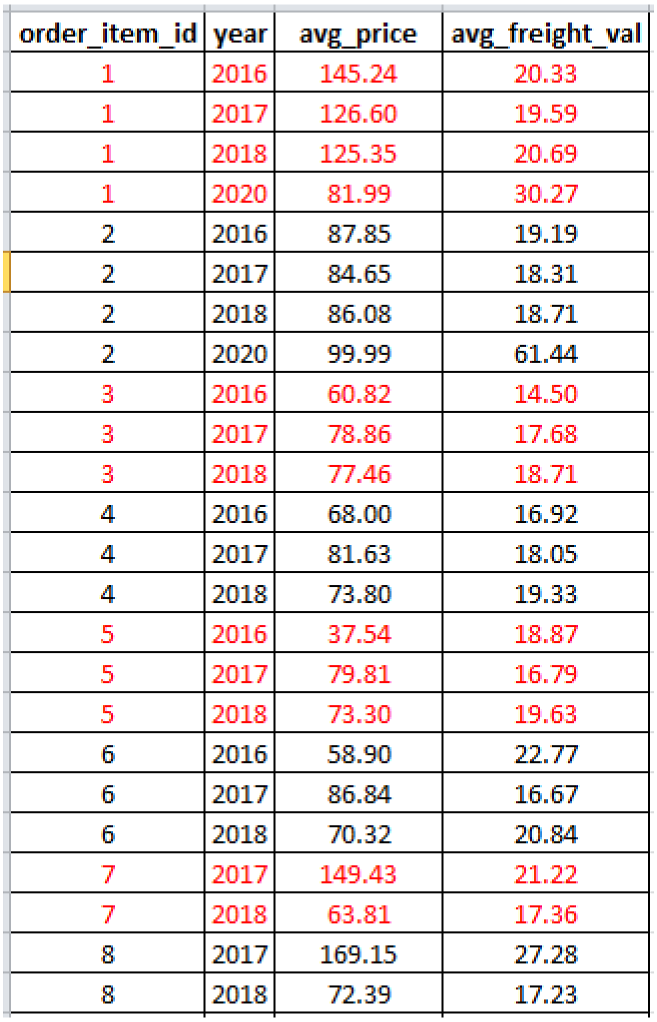


Row	customer_state	no_of_customers
1	RR	46
2	AP	68
3	AC	81
4	AM	148
5	RO	253

Results per page: 50 ▾ 1 – 27 of 27

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

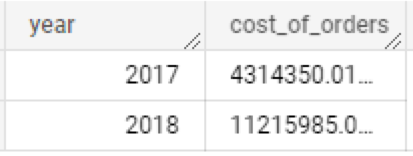
```
select order_item_id, extract(year from shipping_limit_date) as year,  avg(price) as avg_price, avg(freight_value) as avg_freight_val from `Target.order_items` group by year, order_item_id order by order_item_id, year;
```



order_item_id	year	avg_price	avg_freight_val
1	2016	145.24	20.33
1	2017	126.60	19.59
1	2018	125.35	20.69
1	2020	81.99	30.27
2	2016	87.85	19.19
2	2017	84.65	18.31
2	2018	86.08	18.71
2	2020	99.99	61.44
3	2016	60.82	14.50
3	2017	78.86	17.68
3	2018	77.46	18.71
4	2016	68.00	16.92
4	2017	81.63	18.05
4	2018	73.80	19.33
5	2016	37.54	18.87
5	2017	79.81	16.79
5	2018	73.30	19.63
6	2016	58.90	22.77
6	2017	86.84	16.67
6	2018	70.32	20.84
7	2017	149.43	21.22
7	2018	63.81	17.36
8	2017	169.15	27.28
8	2018	72.39	17.23

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

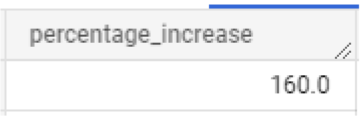
```
select year, sum(payment_value) as cost_of_orders  from (select extract (year from shipping_limit_date) as year,extract (month from shipping_limit_date) as month, p.payment_value from `Target.payments` p join `Target.order_items` o on o.order_id = p.order_id) t where (month >=1 and month<= 8) and year<=2018 group by year order by year;
```



year	cost_of_orders
2017	4314350.01...
2018	11215985.0...

```
with c as(select year, sum(payment_value) as cost_of_orders  from (select extract (year from shipping_limit_date) as year,extract (month from shipping_limit_date) as month, p.payment_value from `Target.payments` p join `Target.order_items` o on o.order_id = p.order_id) t where (month >=1 and month<= 8) and year<=2018 group by year order by year)
```

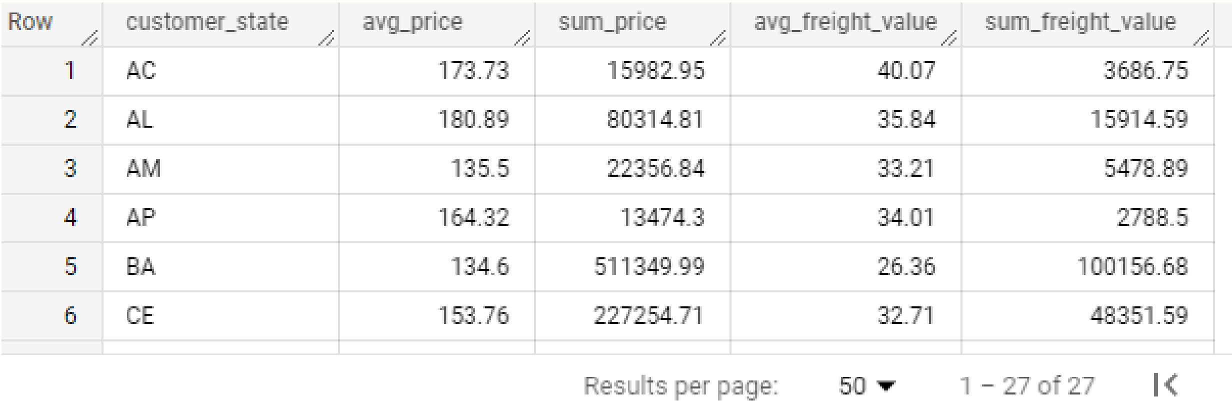
```
selectround(((cost_of_orders -
lag(cost_of_orders)over(order by year asc ))/lag(cost_of_orders)over(order by
 year asc )) * 100)
as percentage_increase from c  ;
```



percentage_increase
160.0

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

```
select distinct customer_state, round(avg(price)over(partition by customer_state),2) as avg_
price, round(sum(price)over(partition by customer_state),2) as sum_price, round(avg(freight
_value)over(partition by customer_state),2) as avg_freight_value, round(sum(freight_value)
over(partition by customer_state),2) as sum_freight_value from Target.customers c join Tar
get.orders o on c.customer_id = o.customer_id join Target.order_items oi on o.order_id = oi.
order_id order by customer_state;
```



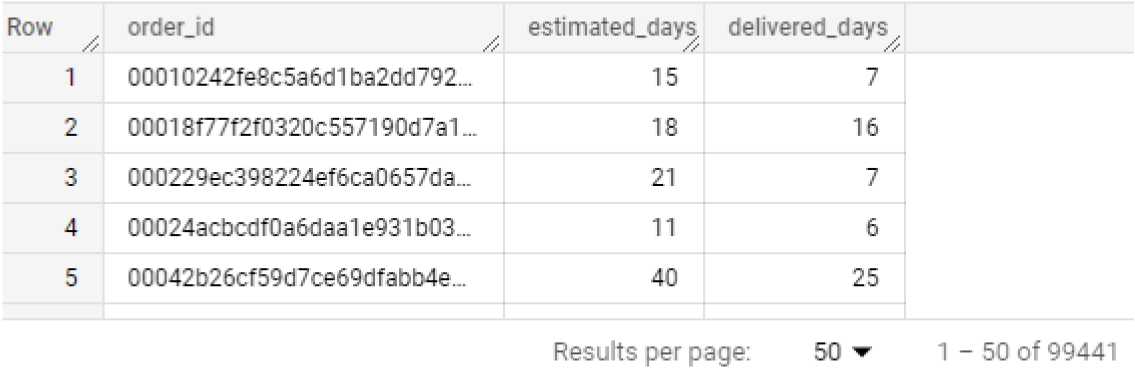
Row	customer_state	avg_price	sum_price	avg_freight_value	sum_freight_value
1	AC	173.73	15982.95	40.07	3686.75
2	AL	180.89	80314.81	35.84	15914.59
3	AM	135.5	22356.84	33.21	5478.89
4	AP	164.32	13474.3	34.01	2788.5
5	BA	134.6	511349.99	26.36	100156.68
6	CE	153.76	227254.71	32.71	48351.59

Results per page: 50 ▾ 1 – 27 of 27

5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

```
select order_id,date_diff( order_estimated_delivery_date , order_purchase_ti
mestamp, day) as estimated_days, date_diff(order_delivered_customer_date
, order_purchase_timestamp, day) as delivered_days  from Target.orders;
```



Row	order_id	estimated_days	delivered_days
1	00010242fe8c5a6d1ba2dd792...	15	7
2	00018f77f2f0320c557190d7a1...	18	16
3	000229ec398224ef6ca0657da...	21	7
4	00024acbcdf0a6daa1e931b03...	11	6
5	00042b26cf59d7ce69dfabb4e...	40	25

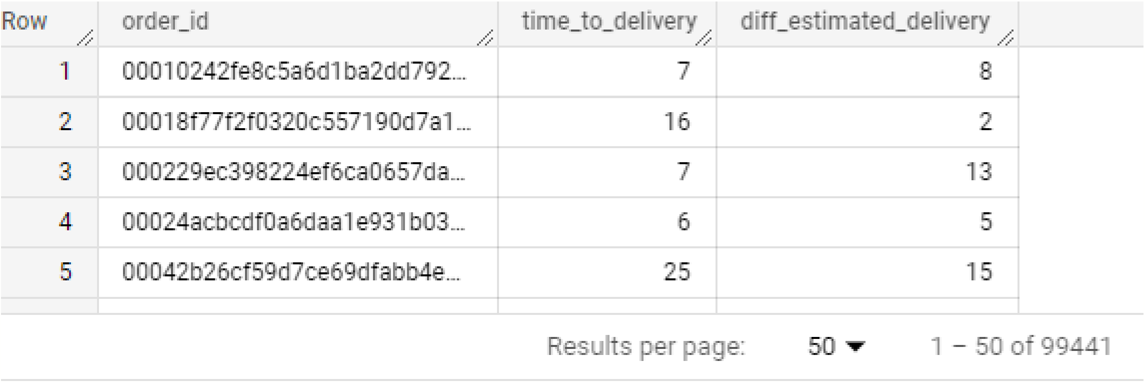
Results per page: 50 ▾ 1 – 50 of 99441

2. Find time_to_delivery & diff_estimated_delivery.

   Formula for the same given below:

   - time_to_delivery = order_purchase_timestamp - order_delivered_customer_date
   - diff_estimated_delivery = order_estimated_delivery_date - order_delivered_customer_date

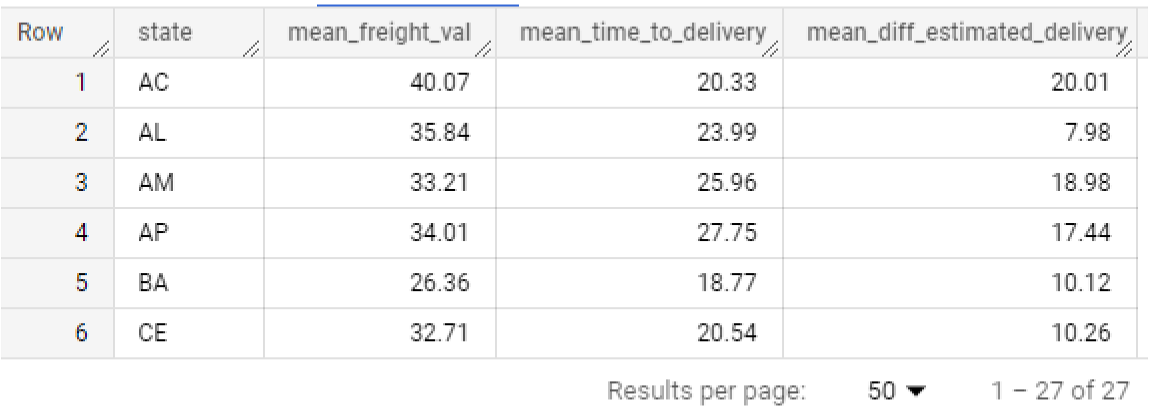
```
select order_id, date_diff( order_delivered_customer_date , order_purchase_timestamp, day) as time_to_delivery, date_diff(order_estimated_delivery_date , order_delivered_customer_date, day) as diff_estimated_delivery  from Target.orders order by order_id;
```



Row	order_id	time_to_delivery	diff_estimated_delivery
1	00010242fe8c5a6d1ba2dd792...	7	8
2	00018f77f2f0320c557190d7a1...	16	2
3	000229ec398224ef6ca0657da...	7	13
4	00024acbcdf0a6daa1e931b03...	6	5
5	00042b26cf59d7ce69dfabb4e...	25	15

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

```
select distinct state, mean_freight_val, round(avg(time_to_delivery)over(partition by state),2) as mean_time_to_delivery, round(avg(diff_estimated_delivery)over(partition by state),2) as mean_diff_estimated_delivery from
 (select customer_state as state, round(avg(freight_value)over(partition by customer_state),2) as mean_freight_val,
date_diff( order_delivered_customer_date , order_purchase_timestamp, day) as time_to_delivery, date_diff(order_estimated_delivery_date , order_delivered_customer_date, day) as diff_estimated_delivery
from Target.customers c join Target.orders o on c.customer_id = o.customer_id join Target.order_items oi on o.order_id = oi.order_id order by customer_state)t order by state;
```



Row	state	mean_freight_val	mean_time_to_delivery	mean_diff_estimated_delivery
1	AC	40.07	20.33	20.01
2	AL	35.84	23.99	7.98
3	AM	33.21	25.96	18.98
4	AP	34.01	27.75	17.44
5	BA	26.36	18.77	10.12
6	CE	32.71	20.54	10.26

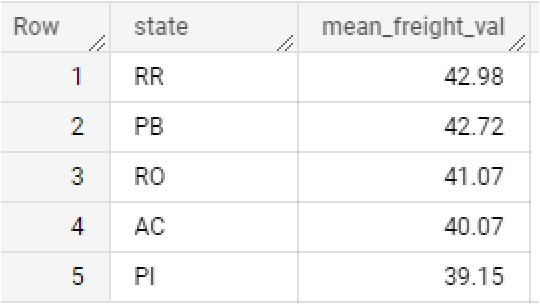
4. Sort the data to get the following:

************ can also be done by considering above table as cte ************

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

Highest average freight value :

```
select distinct state, mean_freight_val  from
(select customer_state as state, round(avg(freight_value)over(partition by customer_state),2) as mean_freight_valfrom Target.customers c join Target.orders o on c.customer_id = o.customer_id join Target.order_items oi on o.order_id = oi.order_id order by customer_state) t
 order by mean_freight_val desc limit 5;
```

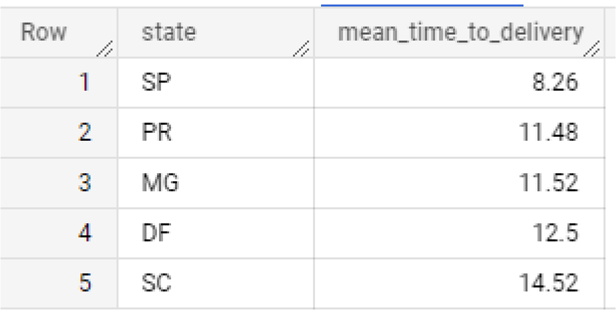


Row	state	mean_freight_val
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

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

Lowest average time to delivery :

```
select distinct state, round(avg(time_to_delivery)over(partition by state),2) as mean_time_to_delivery from
 (select customer_state as state, date_diff( order_delivered_customer_date , order_purchase_timestamp, day) as time_to_delivery
  from Target.customers c join Target.orders o on c.customer_id = o.customer_id join Target.order_items oi on o.order_id = oi.order_id order by customer_state)t
order by mean_time_to_delivery asc limit 5;
```

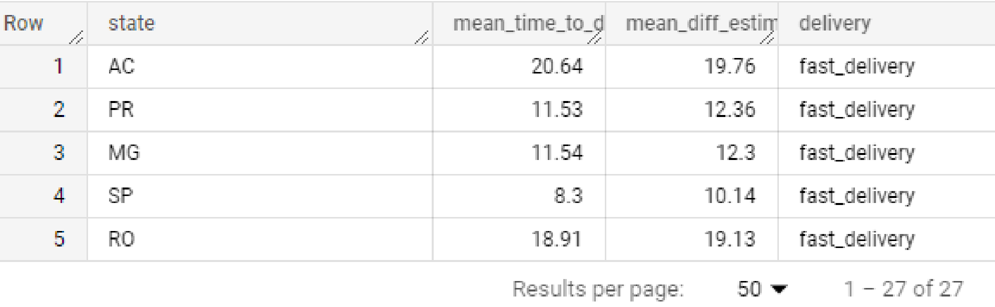


Row	state	mean_time_to_delivery
1	SP	8.26
2	PR	11.48
3	MG	11.52
4	DF	12.5
5	SC	14.52

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

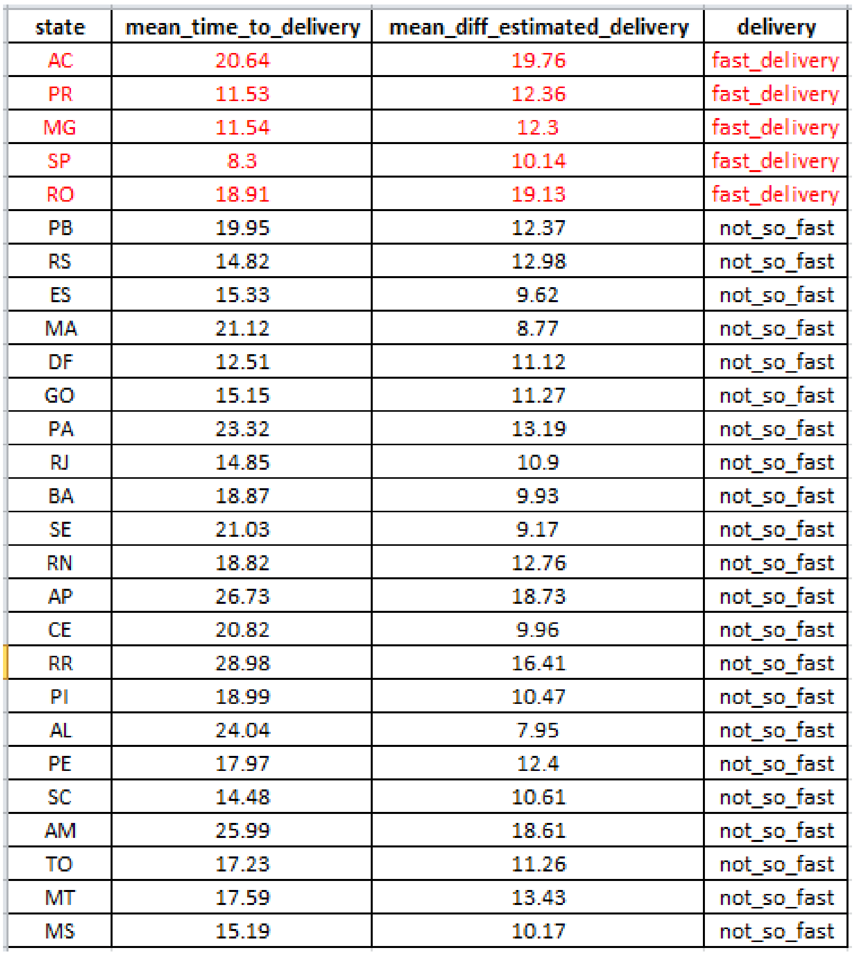
Delivery is really fast when then mean_time_to_delivery is less than the mean_diff_estimated_delivery or when difference between them is less than 1.

```
select state , mean_time_to_delivery, mean_diff_estimated_delivery, case when (mean_time_to_delivery < mean_diff_estimated_delivery) or (mean_time_to_delivery -
mean_diff_estimated_delivery)<1
then "fast_delivery" else "not_so_fast" end as delivery
 from (select distinct state, round(avg(time_to_delivery)over(partition by state),2) as mean_time_to_delivery, round(avg(diff_estimated_delivery)over(partition by state),2) as mean_diff_estimated_delivery from
 (select customer_state as state, date_diff( order_delivered_customer_date , order_purchase_timestamp, day)  as time_to_delivery, date_diff(order_estimated_delivery_date , order_delivered_customer_date, day) as diff_estimated_delivery
 from Target.customers c join Target.orders o on c.customer_id = o.customer_id  order by customer_state)) order by delivery;
```



Row	state	mean_time_to_d	mean_diff_estim	delivery
1	AC	20.64	19.76	fast_delivery
2	PR	11.53	12.36	fast_delivery
3	MG	11.54	12.3	fast_delivery
4	SP	8.3	10.14	fast_delivery
5	RO	18.91	19.13	fast_delivery

Results per page: 50 ▾ 1 – 27 of 27



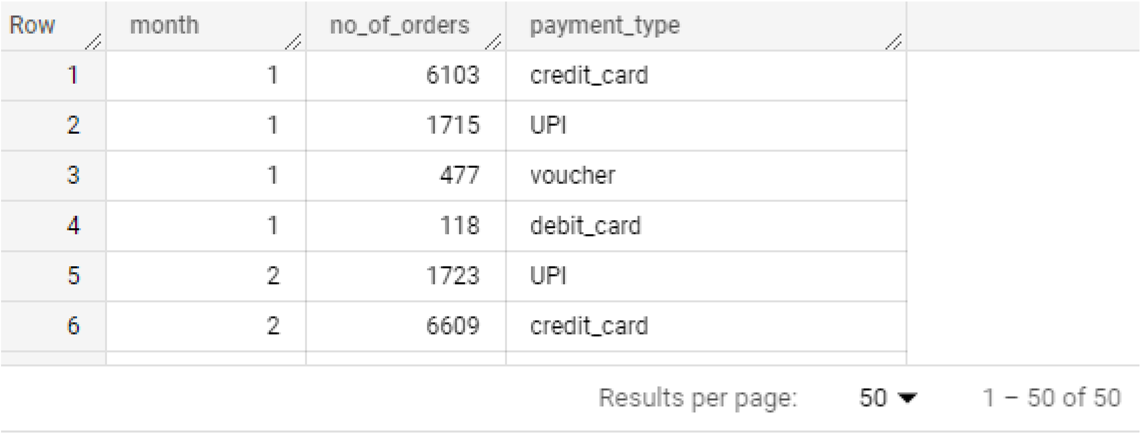
state	mean_time_to_delivery	mean_diff_estimated_delivery	delivery
AC	20.64	19.76	fast_delivery
PR	11.53	12.36	fast_delivery
MG	11.54	12.3	fast_delivery
SP	8.3	10.14	fast_delivery
RO	18.91	19.13	fast_delivery
PB	19.95	12.37	not_so_fast
RS	14.82	12.98	not_so_fast
ES	15.33	9.62	not_so_fast
MA	21.12	8.77	not_so_fast
DF	12.51	11.12	not_so_fast
GO	15.15	11.27	not_so_fast
PA	23.32	13.19	not_so_fast
RJ	14.85	10.9	not_so_fast
BA	18.87	9.93	not_so_fast
SE	21.03	9.17	not_so_fast
RN	18.82	12.76	not_so_fast
AP	26.73	18.73	not_so_fast
CE	20.82	9.96	not_so_fast
RR	28.98	16.41	not_so_fast
PI	18.99	10.47	not_so_fast
AL	24.04	7.95	not_so_fast
PE	17.97	12.4	not_so_fast
SC	14.48	10.61	not_so_fast
AM	25.99	18.61	not_so_fast
TO	17.23	11.26	not_so_fast
MT	17.59	13.43	not_so_fast
MS	15.19	10.17	not_so_fast

Top 5 states with fast_delivery marked in RED color out of 27 states.

6. Payment type analysis:

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

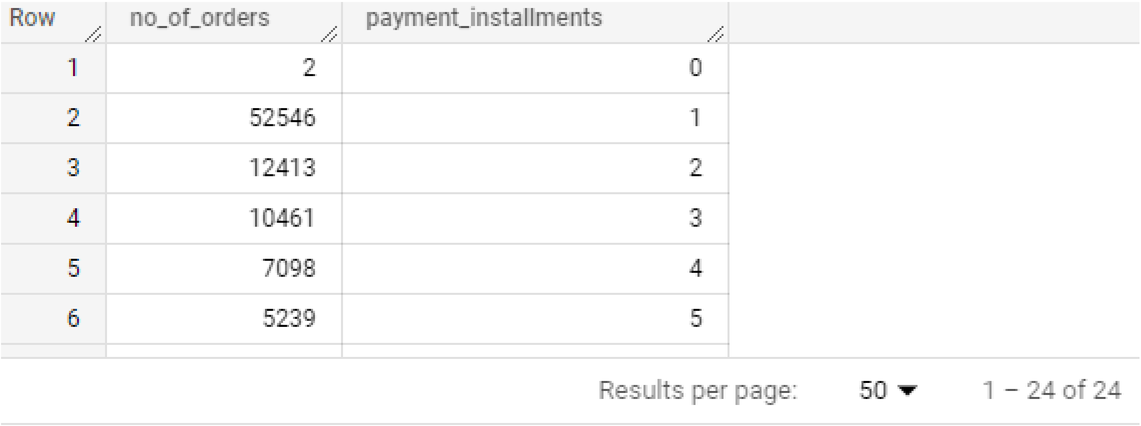
```
select extract(month from(order_purchase_timestamp)) as month, count(o.order_id) as no_of_orders,payment_type from Target.orders o join Target.payments p on o.order_id = p.order_id group by month,payment_type order by month;
```



Row	month	no_of_orders	payment_type
1	1	6103	credit_card
2	1	1715	UPI
3	1	477	voucher
4	1	118	debit_card
5	2	1723	UPI
6	2	6609	credit_card

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

```
select  count(o.order_id) as no_of_orders, payment_installments from Target.orders o join Target.payments p on o.order_id = p.order_id group by payment_installments order by payment_installments;
```



Row	no_of_orders	payment_installments
1	2	0
2	52546	1
3	12413	2
4	10461	3
5	7098	4
6	5239	5

Analysis by,
Name : Meghana S
Batch : DSML Feb23 Beginner