Name: Himanshu Sharma

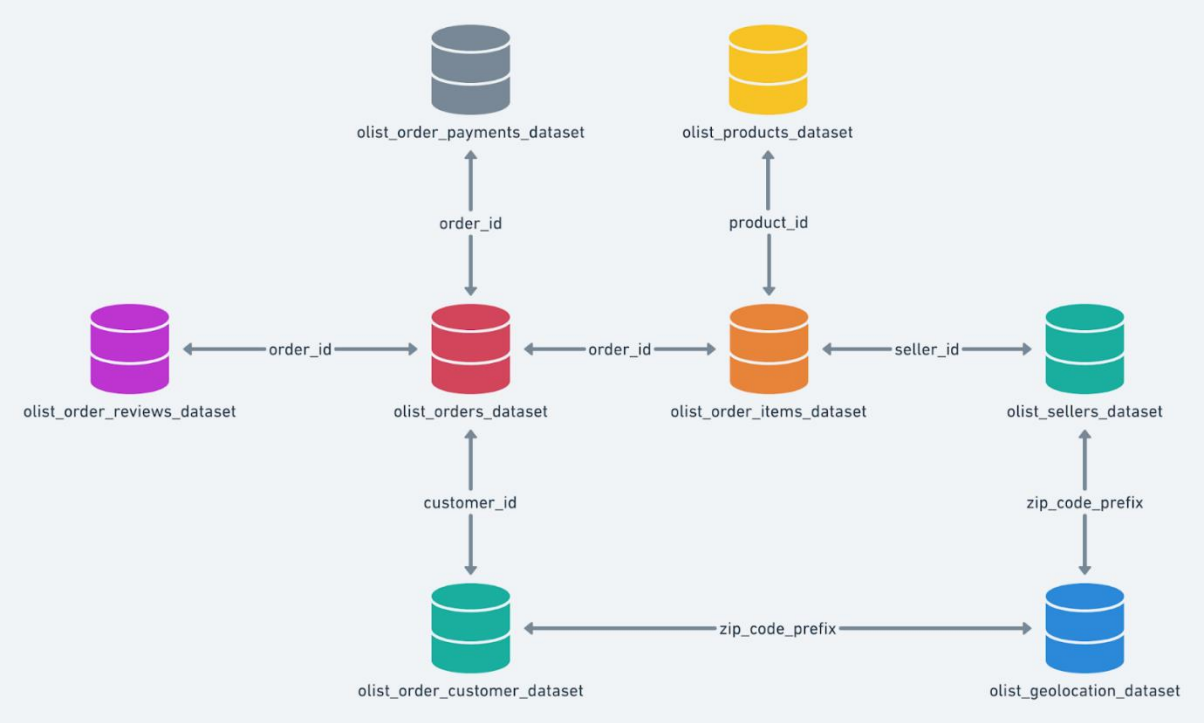
Business Case: Target SQL

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 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.

Dataset: https://drive.google.com/drive/folders/1TGEc66YKbD443nslRi1bWgVd238gJCnb

Data is available in 8 csv files. High level overview of relationship between datasets is as follow:



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

Table name: customers

Field name	Type
customer_id	STRING
customer_unique_id	STRING
customer_zip_code_prefix	INTEGER
customer_city	STRING
customer_state	STRING

Table name: geolocations

Field name	Type
geolocation_zip_code_prefix	INTEGER
geolocation_lat	FLOAT
geolocation_lng	FLOAT
geolocation_city	STRING
geolocation_state	STRING

Table name: payments

Field name	Type
order_id	STRING
payment_sequential	INTEGER
payment_type	STRING
payment_installments	INTEGER
payment_value	FLOAT

Table name: products

Field name	Type
product_id	STRING
product_category	STRING
product_name_length	INTEGER
product_description_length	INTEGER
product_photos_qty	INTEGER
product_weight_g	INTEGER
product_length_cm	INTEGER
product_height_cm	INTEGER
product_width_cm	INTEGER

Table name: order_items

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

Table name: sellers

Field name	Type
seller_id	STRING
seller_zip_code_prefix	INTEGER
seller_city	STRING
seller_state	STRING

Table name: orders

Field name	Type
order_id	STRING
customer_id	STRING
order_status	STRING
order_purchase_timestamp	TIMESTAMP
order_approved_at	TIMESTAMP
order_delivered_carrier_date	TIMESTAMP
order_delivered_customer_date	TIMESTAMP
order_estimated_delivery_date	TIMESTAMP

Table name: order_reviews

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

1.2. Time period for which the data is given.

Query:

```
select
min (order_purchase_timestamp) as Date_First_Transaction,
max (order_purchase_timestamp) as Date_Last_Transaction
from `targetcasestudy-1004.target.orders`
```

Result:

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

Insight: E-commerce is growing fast around the world. The in-dept analysis of the datasets has been carried out to reflect the trend in the Brazilian market. After digging into the datasets, it is found that the data is given for the time period between 2016-2018.

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

Query:

```
select distinct customer_city, customer_state
from `targetcasestudy-1004.target.customers` cust
RIGHT JOIN
`targetcasestudy-1004.target.orders` o
on cust.customer_id = o.customer_id
```

Result:

Row	customer_city	customer_state
1	rio de janeiro	RJ
2	sao leopoldo	RS
3	general salgado	SP
4	brasilia	DF
5	paranavai	PR
6	cuiaba	MT
7	sao luis	MA
8	maceio	AL
9	hortolandia	SP
10	varzea grande	MT
11	belo horizonte	MG
12	sao paulo	SP
13	ipojuca	PE
14	itanhaem	SP
15	porto alegre	RS
16	sao lourenco da mata	PE
17	aracaju	SE
18	ituverava	SP
19	diadema	SP
20	londrina	PR
21	rolante	RS

Insight: It is clear with the data that the customer base of Target is present in all the 27 states of Brazil.

2. In-depth Exploration:

2.1. 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:

```
select
extract(year from order_purchase_timestamp) as years,
count(distinct order_id) as number_of_orders
from `targetcasestudy-1004.target.orders`
group by years
order by years
```

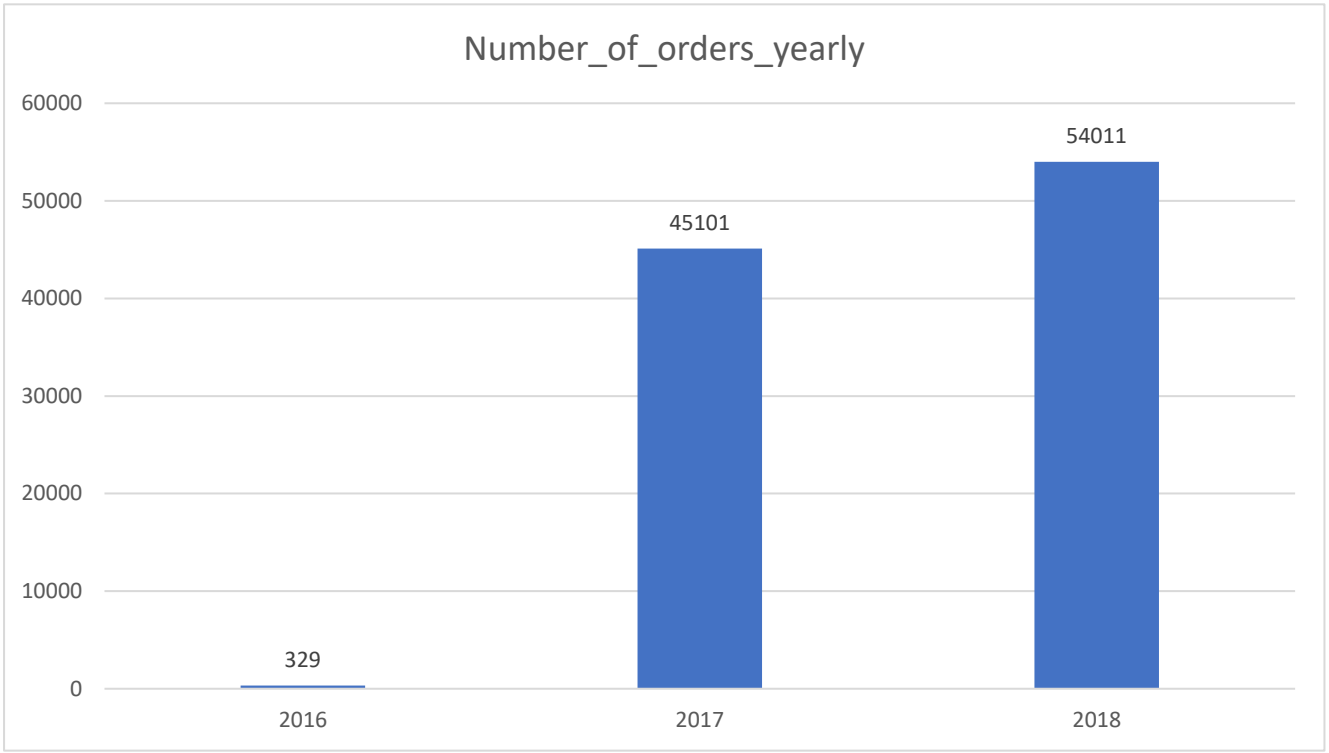
Result:

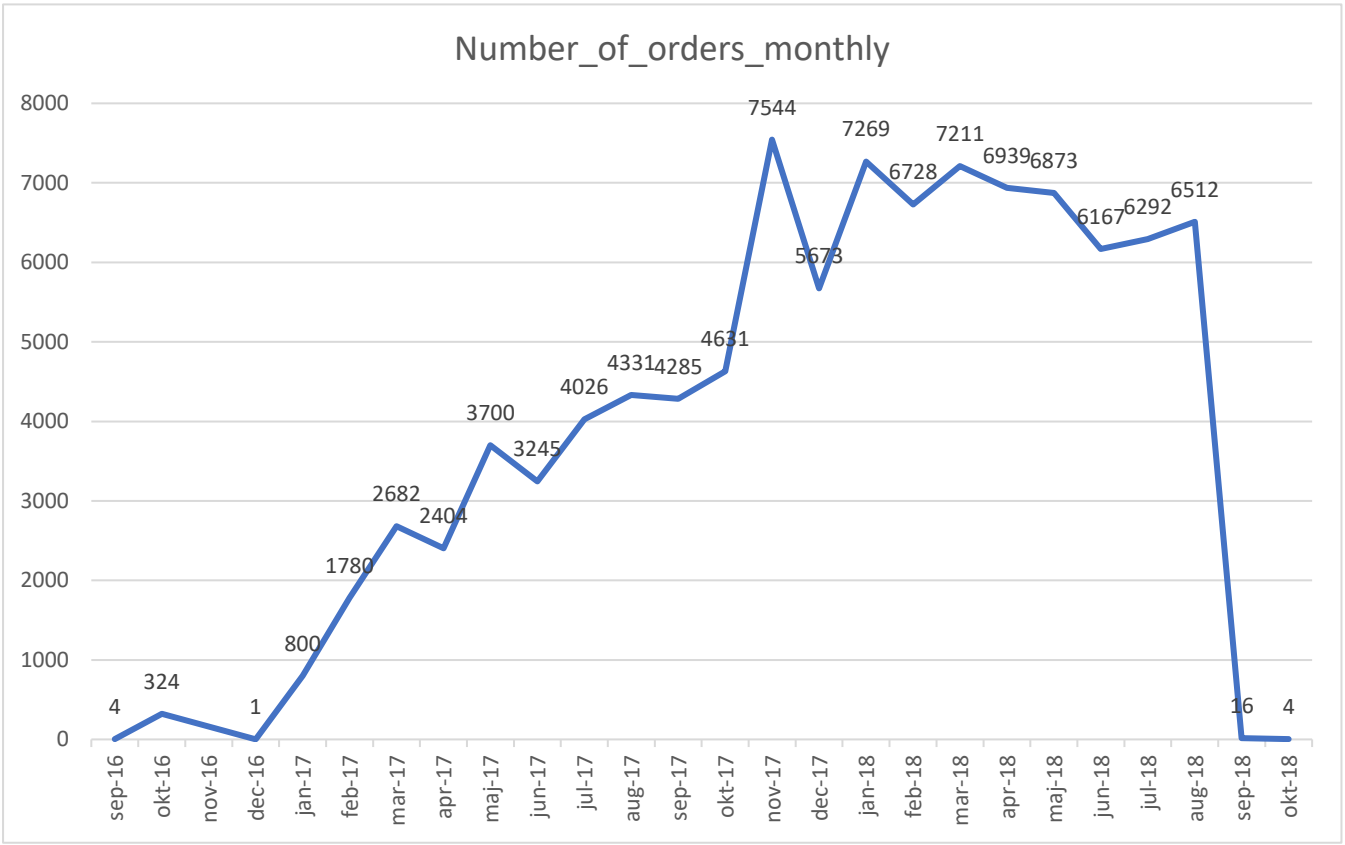
Row	years	number_of_orders
1	2016	329
2	2017	45101
3	2018	54011

Insight: It is clearly visible in the result and graphs that there is increasing trend in the e-commerce in Brazil. As the data is available from September 2016 that's why one can observe the smaller number of orders in this year. In year 2017 and 2018 there is huge jump in terms of sales. It indicates the fast growth of the company in Brazil market.

The monthly data based on order purchase shows that year 2017 was the best year for the sales of the company. They hit the peak of number of orders during the November month in 2017. There was a deep fall in the number of orders in September and October months of 2018.

Both the graphs below indicate the yearly and monthly number of orders for the provided data:


Number_of_orders_yearly
60000
50000
40000
30000
20000
10000
0
329
45101
54011
2016
2017
2018


Number_of_orders_monthly
8000
7000
6000
5000
4000
3000
2000
1000
0
4
324
1
800
1780
2682
2404
3700
3245
4026
4331
4285
4631
7544
5673
7269
6728
7211
6939
6873
6167
6292
6512
16
4
sep-16
okt-16
nov-16
dec-16
jan-17
feb-17
mar-17
apr-17
maj-17
jun-17
jul-17
aug-17
sep-17
okt-17
nov-17
dec-17
jan-18
feb-18
mar-18
apr-18
maj-18
jun-18
jul-18
aug-18
sep-18
okt-18

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

Query:

```
SELECT COUNT(*) as No_of_orders,
CASE
WHEN order_purchase between '03:00:00' and '06:00:00'
THEN 'Dawn'
WHEN order_purchase between '06:00:00' and '12:00:00'
THEN 'Morning'
WHEN order_purchase between '12:00:00' and '17:00:00'
THEN 'Afternoon'
ELSE 'Night'
END AS Time_customer_tend_to_buy
from (select*,extract(time from order_purchase_timestamp)as
order_purchase from `targetcasestudy-1004.target.orders`) t_stamp
GROUP BY Time_customer_tend_to_buy
order by No_of_orders
```

Result:

Row	No_of_orders	Time_customer_tend_to_buy
1	666	Dawn
2	22240	Morning
3	32212	Afternoon
4	44323	Night

Insight: For the given dataset, entire day is divided into four parts Dawn, Morning, Afternoon, Night. It is visible in the result that maximum number of orders were made in night. The company should roll out discounts and offers during mid-day, so that it can attract more customer during that timeframe.

Recommendations: As is it visible that customers tend to order more during the night, the company should focus more on the categorized buyers with some strategies.

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

3.1. Get month on month orders by states.

Query:

```
select extract(month from o.order_purchase_timestamp) as month,
count(o.order_id)as order_id, c.customer_state,
from `targetcasestudy-1004.target.orders`as o
join `targetcasestudy-1004.target.customers` as c
on o.customer_id=c.customer_id
group by month, c.customer_state
order by month
```

Result:

Row	month	order_id	customer_state
1	1	990	RJ
2	1	3351	SP
3	1	151	DF
4	1	427	RS
5	1	99	CE
6	1	113	PE
7	1	443	PR
8	1	264	BA
9	1	971	MG
10	1	51	RN
11	1	82	PA
12	1	66	MA
13	1	345	SC
14	1	19	TO

Insight: The ‘SP’ state attracted the greatest number of customers.

3.2. Distribution of customers across the states in Brazil

Query:

```
select customer_state, count(customer_id) as numberofcust
from `targetcasestudy-1004.target.customers`
group by 1
order by 2 desc
```

Result:

Row	customer_state	numberofcust
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: The 'SP' state attracted 41746 number of customers during the given period.

4. Impact on Economy: Analyse the money movement by e-commerce by looking at order prices, freight, and others.
   4.1. 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.

Query:

```
with cte as

(
select *
from (select round(sum(payment_value), 2) as payment_sum_2017
from `targetcasestudy-1004.target.orders` o
INNER JOIN
`targetcasestudy-1004.target.payments` p
on o.order_id = p.order_id
where extract(year from order_purchase_timestamp) = 2017 and extract(month from
order_purchase_timestamp) between 1 and 8
) h
Cross Join
(
select round(sum(payment_value), 2) as payment_sum_2018
from `targetcasestudy-1004.target.orders` o
INNER JOIN
`targetcasestudy-1004.target.payments` p
on o.order_id = p.order_id
where extract(year from order_purchase_timestamp) = 2018 and extract(month from
order_purchase_timestamp) between 1 and 8
) i
group by payment_sum_2017, payment_sum_2018
```

```
)
select*, round((( payment_sum_2018- payment_sum_2017)/ payment_sum_2017 ),4)
*100 as percent_change
from cte
```

Result:

Row	Sumofpayment_2017	Sumofpayment_2018	percent_change
1	3669022.12	8694733.84	136.98

Insight: Here, the increase in the cost of orders for 2017 and 2018 respectively was analysed. January to August months were considered during this calculation. On this basis of same it was calculated that there was an increase of 137% in number of customers during the given period.

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

Query:

```
select c.customer_state,
round(sum(oi.price),2) as sum_price,
round(avg(oi.price),2) as mean_price,
round(sum(oi.freight_value),2) as sum_freightval,
round(avg(oi.freight_value),2) as mean_freightval
from `targetcasestudy-1004.target.orders` as o
inner join `targetcasestudy-1004.target.order_items` as oi
on o.order_id = oi.order_id
inner join `targetcasestudy-1004.target.customers` as c
on o.customer_id = c.customer_id
group by c.customer_state
order by c.customer_state
```

Result:

Row	customer_state	sum_price	mean_price	sum_freightval	mean_freightval
1	AC	15982.95	173.73	3686.75	40.07
2	AL	80314.81	180.89	15914.59	35.84
3	AM	22356.84	135.5	5478.89	33.21
4	AP	13474.3	164.32	2788.5	34.01
5	BA	511349.99	134.6	100156.68	26.36
6	CE	227254.71	153.76	48351.59	32.71
7	DF	302603.94	125.77	50625.5	21.04
8	ES	275037.31	121.91	49764.6	22.06
9	GO	294591.95	126.27	53114.98	22.77
10	MA	119648.22	145.2	31523.77	38.26
11	MG	1585308.03	120.75	270853.46	20.63

5. Analysis on sales, freight, and delivery time

    1. Calculate days between purchasing, delivering and estimated delivery.

    Query:

```
select
distinct order_id, extract(date from order_purchase_timestamp) as
Date_of_purchase,
extract(date from order_delivered_carrier_date) as
Delivery_Carrier_Date,
extract(date from order_estimated_delivery_date) as
Estimated_date_of_Delivery,
(extract(day from order_delivered_carrier_date) - extract(day from
order_purchase_timestamp)) as pur_delivery_days,
(extract(day from order_delivered_carrier_date) - extract(day from
order_estimated_delivery_date)) as diff_est_delivery
from `targetcasestudy-1004.target.orders`
where order_status = 'delivered'
```

    Result:

Row	order_id	Date_of_purchase	Delivery_Carrier_Dat	Estimated_date_of_D	pur_delivery_days	diff_est_delivery
1	44879a8f19c5e8a5e9278477b...	2018-08-23	2018-08-27	2018-10-04	4	23
2	0562291f2b37f55cc259053d2...	2018-04-30	2018-04-30	2018-06-06	0	24
3	a01f50d51f398895df76f09531...	2018-05-12	2018-05-28	2018-06-06	16	22
4	628923e74a955e432c826a2e0...	2018-05-11	2018-05-30	2018-06-06	19	24
5	ab879558e02a4aec8e7aa5941...	2018-04-21	2018-04-28	2018-06-06	7	22
6	b10350ed7f626af6db79354ad...	2018-05-23	2018-05-28	2018-06-06	5	22
7	656ada45a719393ba3e97d8d...	2018-04-27	2018-04-30	2018-06-06	3	24
8	d1594d3b636b86cf4ce22e2ca...	2018-04-29	2018-05-29	2018-06-06	0	23
9	a8214c9e03a43c8544892568...	2018-05-22	2018-05-30	2018-06-06	8	24
10	7a74611af770d37eb88aa4c25...	2018-05-22	2018-05-29	2018-06-06	7	23
11	0ed4bf3a5a970c8b076ce4cce...	2018-05-22	2018-05-28	2018-06-06	6	22
12	3407bfcbaa0cb49c244ededbc...	2018-04-21	2018-04-30	2018-06-06	9	24
13	4b8bfc23c1469d2169ca53287...	2018-05-12	2018-05-28	2018-06-06	16	22
14	fdeb8e2ed51d3a12a9d6c27a2...	2018-01-21	2018-01-31	2018-02-06	10	25
15	82e77a081a87b4df3966d2fce...	2017-12-29	2017-12-29	2018-02-06	0	23
16	7edf24d26573e36b72c670b38...	2018-01-22	2018-01-31	2018-02-06	9	25
17	68e6025320f22a05b32dd8e3f...	2018-01-22	2018-01-31	2018-02-06	9	25
18	10992b09416c413bd4685017...	2018-01-21	2018-01-31	2018-02-06	10	25
19	13b405c01103fcb4ab199d6a8...	2018-01-22	2018-01-31	2018-02-06	9	25

2. Find time_to_delivery & diff_estimated_delivery.

   Formula for the same given below:

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

Query:

```
select
distinct order_id, extract(datetime from order_purchase_timestamp) as
Purchase_date,
extract(datetime from order_delivered_customer_date) as
Delivery_Cust_Date,
extract(datetime from order_estimated_delivery_date) as Est_Delivery_Date,
(extract(datetime from order_delivered_customer_date) - extract(datetime from
order_purchase_timestamp)) as Time_to_delivery,
(extract(datetime from order_estimated_delivery_date) - extract(datetime from
order_delivered_customer_date)) as Diff_estimated_delivery
from `targetcasestudy-1004.target.orders`
where order_status = 'delivered'
```

Result:

Row	order_id	Purchase_date	Delivery_Cust_Date	Est_Delivery_Date	Time_to_delivery	Diff_estimated_delivery
1	c158e9806f85a33877bdfd4f60...	2017-04-14T22:06:32	2017-05-08T11:10:26	2017-05-18T00:00:00	0-0 23 13:3:54	0-0 9 12:49:34
2	b60b53ad0bb7dacacf2989fe2...	2017-05-10T14:03:27	2017-05-23T13:12:27	2017-05-18T00:00:00	0-0 12 23:9:0	0-0 -5 -13:12:27
3	c830f223aae08493ebecb52f2...	2017-04-22T15:50:30	2017-05-05T13:27:50	2017-05-18T00:00:00	0-0 12 21:37:20	0-0 12 10:32:10
4	a8aa2cd070eeac7e4368cae3d...	2017-05-09T17:42:45	2017-05-16T23:22:20	2017-05-18T00:00:00	0-0 7 5:39:35	0-0 1 0:37:40
5	813c55ce9b6baa8f879e064fbf...	2017-04-26T01:01:39	2017-05-08T08:54:36	2017-05-18T00:00:00	0-0 12 7:52:57	0-0 9 15:5:24
6	44558a1547e448b41c48c408...	2017-05-10T20:47:02	2017-05-12T17:00:05	2017-05-18T00:00:00	0-0 1 20:13:3	0-0 5 6:59:55
7	036b791897847cdb8e39df794...	2017-05-10T15:34:59	2017-05-17T11:14:40	2017-05-18T00:00:00	0-0 6 19:39:41	0-0 0 12:45:20
8	1aba60c04110bdd421b250ea...	2017-04-18T21:20:40	2017-05-10T11:50:00	2017-05-18T00:00:00	0-0 21 14:29:20	0-0 7 12:10:0
9	0312ecf90786def87f98aa19e0...	2017-05-10T22:02:40	2017-05-18T17:09:46	2017-05-18T00:00:00	0-0 7 19:7:6	0-0 0 -17:9:46
10	635c894d068ac37e6e03dc54e...	2017-04-15T15:37:38	2017-05-16T14:49:55	2017-05-18T00:00:00	0-0 30 23:12:17	0-0 1 9:10:5
11	f79bd0b3bacc5142f90f81a15b...	2017-04-22T13:55:16	2017-05-12T13:55:55	2017-05-18T00:00:00	0-0 20 0:0:39	0-0 5 10:4:5
12	3b97562c3aee8bdedcb5c2e45...	2017-04-14T22:21:54	2017-05-17T10:52:15	2017-05-18T00:00:00	0-0 32 12:30:21	0-0 0 13:7:45
13	51a7d7d26116c0eaee6bb35ea...	2017-04-25T08:12:22	2017-05-04T07:18:36	2017-05-18T00:00:00	0-0 8 23:6:14	0-0 13 16:41:24
14	bb0607616e290609d11b67e0...	2017-04-25T09:43:59	2017-05-10T09:19:50	2017-05-18T00:00:00	0-0 14 23:35:51	0-0 7 14:40:10
15	1451f4e1d7e9367d2fe5173de...	2017-04-22T18:45:59	2017-05-15T11:52:19	2017-05-18T00:00:00	0-0 22 17:6:20	0-0 2 12:7:41
16	bf375386af3b1f89486a565f18...	2017-04-17T14:04:59	2017-05-01T12:34:00	2017-05-18T00:00:00	0-0 13 22:29:1	0-0 16 11:26:0

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

Query:

```
select
distinct customer_state,
avg(freight_value) as meanfreight,
avg(extract(datetime from order_delivered_customer_date) -
extract(datetime from order_purchase_timestamp)) as time_to_delivery,
avg(extract(datetime from order_estimated_delivery_date) -
extract(datetime from order_delivered_customer_date)) as diff_estimated_delivery
from `targetcasestudy-1004.target.customers` c
Inner join
`targetcasestudy-1004.target.orders` o
on c.customer_id = o.customer_id
Inner join
`targetcasestudy-1004.target.order_items` oi
on oi.order_id = o.order_id
where order_status = 'delivered'
group by c.customer_state
```

Result:

Row	customer_state	meanfreight	time_to_delivery	diff_estimated_delivery
1	GO	22.5628678085...	0-0 14 33:40:48.375933245	0-0 11 14:18:7.996486605
2	SP	15.1151823544...	0-0 8 17:22:20.194547921	0-0 10 12:17:14.311513533
3	RS	21.6131920443...	0-0 14 28:31:32.063916517	0-0 13 10:22:7.941790314
4	BA	26.4875563399...	0-0 18 29:56:34.191963073	0-0 10 6:55:31.734727124
5	MG	20.6263425209...	0-0 11 23:34:48.313564571	0-0 12 15:23:35.678460823
6	MT	27.9969141755...	0-0 17 23:4:49.308582449	0-0 13 21:30:17.274831243
7	RJ	20.9114360461...	0-0 14 27:32:58.179594145	0-0 11 7:17:3.752386339
8	SC	21.5073590432...	0-0 14 23:56:36.372711740	0-0 10 20:50:21.306809860
9	SE	36.5731733333...	0-0 20 35:12:59.317333333	0-0 9 7:49:3.408
10	PE	32.6933333333...	0-0 17 30:42:8.667239404	0-0 12 18:20:47.015463917
11	TO	37.4350322580...	0-0 17 10:45:40.490322580	0-0 11 15:36:27.777419354
12	CE	32.7344950911...	0-0 20 23:41:56.934081346	0-0 10 9:51:29.995792426
13	PR	20.4718162506...	0-0 11 22:43:59.571074526	0-0 12 19:0:31.719242343
14	PA	35.6290132827...	0-0 23 18:5:50.211574952	0-0 13 13:39:52.679316888
15	MS	23.3509001233...	0-0 15 12:49:3.794081381	0-0 10 12:41:12.300863131
16	ES	22.0289797752...	0-0 15 15:34:39.280898876	0-0 9 22:46:51.880898876

4. Sort the data to get the following:
5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

- Top 5 states with lowest average freight value:

Query:

```
select c.customer_state,
round(avg(oi.freight_value),2) as mean_freightval
from `targetcasestudy-1004.target.orders` as o
join `targetcasestudy-1004.target.order_items` as oi
on o.order_id = oi.order_id
join `targetcasestudy-1004.target.customers` as c
on o.customer_id = c.customer_id
group by 1
order by 2
limit 5
```

Result:

Row	customer_state	mean_freightval
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04

- Top 5 states with highest average freight value:

Query:

```
select c.customer_state,
round(avg(oi.freight_value),2) as mean_freightval
from `targetcasestudy-1004.target.orders` as o
join `targetcasestudy-1004.target.order_items` as oi
on o.order_id = oi.order_id
join `targetcasestudy-1004.target.customers` as c
on o.customer_id = c.customer_id
group by 1
order by 2 desc
limit 5
```

Result:

Row	customer_state	mean_freightval
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

Query:

```
select c.customer_state,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,
day)),2) as time_to_delivery,
from `targetcasestudy-1004.target.orders` o
join `targetcasestudy-1004.target.customers` c
on o.customer_id = c.customer_id
group by 1
order by 2
limit 5
```

Result:

Row	customer_state	time_to_delivery
1	SP	8.3
2	PR	11.53
3	MG	11.54
4	DF	12.51
5	SC	14.48

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

Top 5 states where delivery is really fast

Query:

```
select distinct c.customer_state,
avg(extract(datetime from order_estimated_delivery_date) -
extract(datetime from order_delivered_customer_date)) as delivery_diff
from `targetcasestudy-1004.target.orders` o
left join
`targetcasestudy-1004.target.customers` c
on o.customer_id = c.customer_id
```

```
group by c.customer_state
order by delivery_diff
limit 5
```

Result:

Row	customer_state	delivery_diff
1	AL	0-0 7 24:46:9.886649874
2	MA	0-0 8 21:18:29.762900976
3	SE	0-0 9 7:53:14.623880597
4	ES	0-0 9 19:7:50.208521303
5	BA	0-0 9 26:22:39.872542997

Top 5 states where delivery is not so fast

Query:

```
select distinct c.customer_state,
avg(extract(datetime from order_estimated_delivery_date) -
extract(datetime from order_delivered_customer_date)) as delivery_diff
from `targetcasestudy-1004.target.orders` o
left join
`targetcasestudy-1004.target.customers` c
on o.customer_id = c.customer_id
group by c.customer_state
order by delivery_diff desc
limit 5
```

Result:

Row	customer_state	delivery_diff
1	AC	0-0 19 25:50:53.400
2	RO	0-0 19 9:31:25.802469135
3	AP	0-0 18 25:25:34.119402985
4	AM	0-0 18 20:26:36.986206896
5	RR	0-0 16 14:16:13.243902439

6. Payment type analysis:

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

   Query:

```
select payment_type,
extract(month from o.order_purchase_timestamp) as months_wise_data,
count(*) as no_of_orders
from `targetcasestudy-1004.target.orders`o
inner join `targetcasestudy-1004.target.payments`p
on o.order_id = p.order_id
group by 1,2
order by 2
```

   Result:

Row	payment_type	months_wise_data	no_of_orders
1	credit_card	1	6103
2	UPI	1	1715
3	voucher	1	477
4	debit_card	1	118
5	UPI	2	1723
6	credit_card	2	6609
7	voucher	2	424
8	debit_card	2	82
9	credit_card	3	7707
10	UPI	3	1942
11	debit_card	3	109
12	voucher	3	591
13	voucher	4	572
14	credit_card	4	7301

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

   Query:

```
select payment_installments,
count(*) as no_of_orders
from `targetcasestudy-1004.target.payments`
group by payment_installments
```

Result:

Row	payment_installment	no_of_orders
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
11	10	5328
12	11	23
13	12	133
14	13	16
15	14	15

Recommendations:

1. Different kind of vouchers or gift cards can be sent to the customers who come under medium range buyer's category.

2. Solving the supply change issue in bottom 5 states where the delivery is not so fast.

3. Understanding the needs of the customers by getting in touch with them which belongs to bottom 5 states in terms of number of customers and understanding their needs.