

BUSINESS CASE TARGET SQL

Target is a globally renowned brand and a prominent retailer in the United States. 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 particular business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.

By analyzing this extensive dataset, it becomes possible to gain valuable insights into Target's operations in Brazil. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels.

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

The data is available in 8 csv files:

1. customers.csv
 2. sellers.csv
 3. order_items.csv
 4. geolocation.csv
 5. payments.csv
 6. reviews.csv
 7. orders.csv
 8. products.csv
-
-

The column description for these csv files is given below.

The **customers.csv** contain following features:

Features	Description
customer_id	ID of the consumer who made the purchase
customer_unique_id	Unique ID of the consumer
customer_zip_code_prefix	Zip Code of consumer's location
customer_city	Name of the City from where order is made
customer_state	State Code from where order is made (Eg. são paulo - SP)

The **sellers.csv** contains following features:

Features	Description
seller_id	Unique ID of the seller registered
seller_zip_code_prefix	Zip Code of the seller's location
seller_city	Name of the City of the seller
seller_state	State Code (Eg. são paulo - SP)

The **order_items.csv** contain following features:

Features	Description
order_id	A Unique ID of order made by the consumers
order_item_id	A Unique ID given to each item ordered in the order
product_id	A Unique ID given to each product available on the site
seller_id	Unique ID of the seller registered in Target
shipping_limit_date	The date before which the ordered product must be shipped
price	Actual price of the products ordered
freight_value	Price rate at which a product is delivered from one point to another

The **geolocations.csv** contain following features:

Features	Description
geolocation_zip_code_prefix	First 5 digits of Zip Code

geolocation_lat	Latitude
geolocation_lng	Longitude
geolocation_city	City
geolocation_state	State

The **payments.csv** contain following features:

Features	Description
order_id	A Unique ID of order made by the consumers
payment_sequential	Sequences of the payments made in case of EMI
payment_type	Mode of payment used (Eg. Credit Card)
payment_installments	Number of installments in case of EMI purchase
payment_value	Total amount paid for the purchase order

The **orders.csv** contain following features:

Features	Description
order_id	A Unique ID of order made by the consumers
customer_id	ID of the consumer who made the purchase
order_status	Status of the order made i.e. delivered, shipped, etc.
order_purchase_timestamp	Timestamp of the purchase
order_delivered_carrier_date	Delivery date at which carrier made the delivery
order_delivered_customer_date	Date at which customer got the product
order_estimated_delivery_date	Estimated delivery date of the products

The **reviews.csv** contain following features:

Features	Description
review_id	ID of the review given on the product ordered by the order id
order_id	A Unique ID of order made by the consumers
review_score	Review score given by the customer for each order on a scale of 1-5
review_comment_title	Title of the review

review_comment_message	Review comments posted by the consumer for each order
review_creation_date	Timestamp of the review when it is created
review_answer_timestamp	Timestamp of the review answered

The **products.csv** contain following features:

Features	Description
product_id	A Unique identifier for the proposed project.
product_category_name	Name of the product category
	Length of the string which specifies the name
product_name_length	given to the products ordered
	Length of the description written for each product
product_description_length	ordered on the site
	Number of photos of each product ordered available
product_photos_qty	on the shopping portal
product_weight_g	Weight of the products ordered in grams
product_length_cm	Length of the products ordered in centimeters
product_height_cm	Height of the products ordered in centimeters
product_width_cm	Width of the product ordered in centimeters

Problem Statement:

Assuming you are a data analyst/ scientist at Target, you have been assigned the task of analyzing the given dataset to extract valuable insights and provide actionable recommendations

SOLUTION:

Understanding the data in the dataset

- Customers Table

select * from Target.customers limit 5;

Row	customer_id	customer_unique_id	customer_zip_code	customer_city	customer_state
1	0735e7e4298a2ebbb46649346...	fc003b1bdc0df64b4d065d9b...	59650	acu	RN
2	903b3d86e3990db01619a4ebe...	46824822b15da44e983b021d...	59650	acu	RN
3	38c97666e962d4fea7fd6a83e...	b6108acc674ae5c99e29adc10...	59650	acu	RN
4	77c2f46cf580f4874c9a5751c2...	402cce5c0509000eed9e77fec...	63430	ico	CE
5	4d3ef4cfff8ad4767c199c36a...	6ba00666ab7eada5ceec279b2...	63430	ico	CE

- Geolocation Table

select * from Target.geolocation limit 5;

Row	geolocation_zip_code	geolocation_lat	geolocation_lng	geolocation_city	geolocation_state
1	49010	-10.9105145187...	-37.0524007769...	aracaju	SE
2	49047	-10.9268145	-37.0710630000...	aracaju	SE
3	49030	-10.9701647943...	-37.0616438307...	aracaju	SE
4	49048	-10.9401835317...	-37.0708502427...	aracaju	SE
5	49050	-10.9271573528...	-37.0630786896...	aracaju	SE

- Order_items Table

select * from Target.order_items limit 5;

Row	order_id	order_item_id	product_id	seller_id	shipping_limit_date	price	freight_value
1	f09e36e258656850b92657ac5...	1	44d53f1240d6332232e4393c0...	b64d51f0435e884e8de603b16...	2018-07-09 13:31:36 UTC	3.0	12.79
2	f9ccaff7267fd0cf076e795b1fa...	1	44d53f1240d6332232e4393c0...	b64d51f0435e884e8de603b16...	2018-08-14 14:04:44 UTC	3.0	15.23
3	c79bdf061e22288609201ec60...	1	5304ff3fa35856a156e1170a60...	cf6f6bc4df3999b9c6440f124f...	2017-05-12 19:05:20 UTC	3.5	8.72
4	37193e64eb9a46b7f3197762f...	1	98224bfc1eaadb3a394ec334c...	ce616e1913288884e7742faac...	2018-06-28 01:30:49 UTC	3.5	7.39
5	95d6357ffe41aa6d2998852a7...	1	98224bfc1eaadb3a394ec334c...	ce616e1913288884e7742faac...	2018-06-12 19:15:14 UTC	3.5	18.23

- Order_reviews Table

select * from Target.order_reviews limit 5;

Row	review_id	order_id	review_score	review_comment_title	review_creation_date	review_answer_timestamp
1	be7e2989673cb2a147b87ba73...	777c67eab7c0712ccde8ffbb2...	1	null	0001-04-17 00:00:00 UTC	0001-04-17 07:40:00 UTC
2	e12151267e4594d69eda14a87...	4338a4463f7f9193d2a03a83a...	1	null	0001-04-17 00:00:00 UTC	0001-04-17 09:04:00 UTC
3	41d614b133efebcd10352001b...	b8aaeda740b17cf925d3f2a13...	1	null	0001-04-17 00:00:00 UTC	0002-04-17 03:48:00 UTC
4	c950324a42c5796d06f569f77...	b159d0ce7cd881052da94fa16...	1	null	0001-04-17 00:00:00 UTC	0001-04-17 10:24:00 UTC
5	76823ada94c8861ecebcfb7c...	2a3007ed051b02a0e0dd0709c...	1	null	0001-04-17 00:00:00 UTC	0002-04-17 13:58:00 UTC

- Order Table

select * from Target.orders limit 5;

Row	order_id	customer_id	order_status	order_purchase_timestamp	order_approved_at	order_delivered_carrier_date
1	7a4df5d8cff4090e541401a20a...	725e9c75605414b21fd8c8d5a...	created	2017-11-25 11:10:33 UTC	null	null
2	35de4050331c6c644cddc86f4...	4ee64f4bfc542546f422da0aeb...	created	2017-12-05 01:07:58 UTC	null	null
3	b5359909123fa03c50bcb0cfe...	438449d4af8980d107bf04571...	created	2017-12-05 01:07:52 UTC	null	null
4	dba5062fbda3af4fb6c33b1e04...	964a6df3d9bdf60fe3e7b8bb69...	created	2018-02-09 17:21:04 UTC	null	null
5	90ab3e7d52544ec7bc3363c82...	7d61b9f4f216052ba664f22e9c...	created	2017-11-06 13:12:34 UTC	null	null

- Payment Table

select * from Target.payments limit 5;

Row	order_id	payment_sequential	payment_type	payment_installment	payment_value
1	1a57108394169c0b47d8f876a...	2	credit_card	0	129.94
2	744bade1fcf9ff3f31d860ace07...	2	credit_card	0	58.69
3	8bcbe01d44d147f901cd31926...	4	voucher	1	0.0
4	fa65dad1b0e818e3ccc5cb0e3...	14	voucher	1	0.0
5	6ccb433e00daae1283ccc9561...	4	voucher	1	0.0

- Products Table

select * from Target.products limit 5;

Row	product_id	product_category	product_name_length	product_description	product_photos_qty	product_weight_g	product_length_cm	product_height_cm	product_width_cm
1	5eb564652db742ff8f28759cd8...	null	null	null	null	null	null	null	null
2	09ff539a621711667c43eba6a...	babies	60	865	3	null	null	null	null
3	2f763ba79d9cd987b2034aac7...	electronics	45	1198	2	595	8	6	6
4	a69f15dfb803d485e8933e80b...	Watches present	53	506	6	150	11	16	6
5	e1cfc87f543782b8a78b59fc85...	Garden tools	39	524	4	369	26	7	7

- Sellers Table

select * from Target.sellers limit 5;

Row	seller_id ▼	seller_zip_code_prefix	seller_city ▼	seller_state ▼
1	4be2e7f96b4fd749d52dff41f8...	69900	rio branco	AC
2	327b89b872c14d1c0be7235ef...	69005	manaus	AM
3	4221a7df464f1fe2955934e30f...	48602	bahia	BA
4	651530bf5c607240ccdd89a30...	44600	ipira	BA
5	2b402d5dc42554061f8ea98d1...	44900	irece	BA

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

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

```
SELECT TABLE_NAME, COLUMN_NAME, DATA_TYPE FROM Target.INFORMATION_SCHEMA.COLUMNS;
```

Row	TABLE_NAME ▼	COLUMN_NAME ▼	DATA_TYPE ▼
1	order_items	order_id	STRING
2	order_items	order_item_id	INT64
3	order_items	product_id	STRING
4	order_items	seller_id	STRING
5	order_items	shipping_limit_date	TIMESTAMP
6	order_items	price	FLOAT64
7	order_items	freight_value	FLOAT64
8	sellers	seller_id	STRING
9	sellers	seller_zip_code_prefix	INT64
10	sellers	seller_city	STRING

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

```
SELECT MIN(order_purchase_timestamp) AS min_time_of_purchase,
MAX(order_estimated_delivery_date) AS max_time_of_purchase
FROM Target.orders, Target.order_items;
```

Row	min_time_of_purchase ▼	max_time_of_purchase ▼
1	2016-09-04 21:15:19 UTC	2018-11-12 00:00:00 UTC

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

```
SELECT DISTINCT customer_state, customer_city
FROM Target.customers
GROUP BY 1, 2 ORDER BY 1, 2;
```

Row	customer_state	customer_city
1	AC	brasileia
2	AC	cruzeiro do sul
3	AC	epitaciolandia
4	AC	manoel urbano
5	AC	porto acre
6	AC	rio branco
7	AC	senador guiomard
8	AC	xapuri
9	AL	agua branca

2. In-depth Exploration:

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

```
SELECT *, ROUND(((orders_count - prev_order_count) / prev_order_count) * 100, 2) AS order_growth_rate_percent FROM
(SELECT *, LAG(orders_count) OVER(ORDER BY YEAR, MONTH) AS prev_order_count FROM
(SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS YEAR, EXTRACT(MONTH FROM
order_purchase_timestamp) AS MONTH, COUNT(*) AS orders_count
FROM Target.orders
WHERE order_status = 'delivered'
GROUP BY 1, 2 ORDER BY 1, 2) AS BASE1 ORDER BY YEAR, MONTH) AS BASE2;
```

Row	YEAR	MONTH	orders_count	prev_order_count	order_growth_rate_percent
1	2016	9	1	null	null
2	2016	10	265	1	26400.0
3	2016	12	1	265	-99.62
4	2017	1	750	1	74900.0
5	2017	2	1653	750	120.4
6	2017	3	2546	1653	54.02

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

```
select count(order_id) as total_orders,
```


EXTRACT(Month FROM order_purchase_timestamp) AS month,
from `Target.orders` group by month order by month asc;

Row	total_orders	month
1	8069	1
2	8508	2
3	9893	3
4	9343	4
5	10573	5
6	9412	6
7	10318	7

2.3 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 rs : Afternoon

13-19 19-23 hrs : Night

```
SELECT COUNTIF((TIME(order_purchase_timestamp) >= '05:00:00' AND TIME(order_purchase_timestamp) < '06:00:00')) AS
dawn_orders_count_5am_6am,
COUNTIF((TIME(order_purchase_timestamp) >= '06:00:00' AND TIME(order_purchase_timestamp) < '12:00:00')) AS
morning_orders_count_6am_12pm,
COUNTIF((TIME(order_purchase_timestamp) >= '12:00:00' AND TIME(order_purchase_timestamp) < '18:00:00')) AS
afternoon_orders_count_12pm_6pm,
COUNTIF((TIME(order_purchase_timestamp) >= '18:00:00' AND TIME(order_purchase_timestamp) <= '23:59:59') OR
(TIME(order_purchase_timestamp) >= '00:00:00' AND TIME(order_purchase_timestamp) < '05:00:00')) AS
night_orders_count_6pm_5am
FROM Target.orders;
```

Row	dawn_orders_count	morning_orders_count	afternoon_orders_count	night_orders_count
1	188	22240	38361	38652

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

3.1 Get the month on month no. of orders placed in each state.

```

select state, time_period, total_orders,
LAG(total_orders) OVER(PARTITION BY state ORDER BY year, month ) AS prev_month_orders_count,
ROUND((((total_orders - LAG(total_orders) OVER(PARTITION BY state ORDER BY year, month )) /
LAG(total_orders) OVER(PARTITION BY state ORDER BY year, month))* 100,2) AS MoM_percent_growth
FROM (
SELECT
state, time_period,year,month,COUNT(*) AS total_orders
FROM (
SELECT o.order_id, o.order_purchase_timestamp,
EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
EXTRACT(Month FROM order_purchase_timestamp) AS month,
FORMAT_DATE('%b %Y', DATE(ORDER_PURCHASE_TIMESTAMP)) AS time_period,
c.customer_state AS state
FROM
`Target.orders` o JOIN `Target.customers` c
USING (customer_id) ORDER BY year, month) t1 GROUP BY state, time_period,year, month) t2;

```

Row	customer_state	customer_city	YEAR	MONTH	orders_count	prev_orders_count	orders_count_growth
1	AL	belem	2017	4	1	null	null
2	AL	belem	2018	2	1	1	0.0
3	AP	oiapoque	2017	5	1	null	null
4	BA	cairu	2017	10	1	null	null
5	BA	masquete	2017	7	1	null	null
6	BA	masquete	2018	1	1	1	0.0
7	BA	retirolandia	2017	3	1	null	null
8	CE	coreau	2018	5	1	null	null
9	CE	eusebio	2017	8	2	null	null
10	CE	eusebio	2017	9	2	2	0.0

3.2 How are the customers distributed across all the states?

```

SELECT customer_state, customer_city ,
COUNT(DISTINCT customer_id) AS count_customer_id,
COUNT(DISTINCT customer_unique_id) AS count_customer_unique_id
FROM Target.customers
GROUP BY 1, 2
ORDER BY 1, 2;

```

Row	customer_state ▼	customer_city ▼	count_customer_id	count_customer_uniq
1	AC	brasileia	1	1
2	AC	cruzeiro do sul	3	3
3	AC	epitaciolandia	1	1
4	AC	manoel urbano	1	1
5	AC	porto acre	1	1
6	AC	rio branco	70	66
7	AC	senador guiomard	2	2
8	AC	xapuri	2	2
9	AL	agua branca	1	1
10	AL	anadia	2	2

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

- 4.1** 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.

WITH TABLE1 AS

(SELECT ROUND(SUM(price + freight_value), 2) AS total_cost_2017

FROM (SELECT O.*, OI.* FROM `Target.orders` AS O JOIN `Target.order_items` AS OI ON O.order_id = OI.order_id
WHERE O.order_status = 'delivered' AND (EXTRACT(YEAR FROM O.order_purchase_timestamp) = 2017) AND
EXTRACT(MONTH FROM O.order_purchase_timestamp) BETWEEN 1 AND 8)),

TABLE2 AS (SELECT ROUND(SUM(price + freight_value), 2) AS total_cost_2018

FROM (SELECT O.*, OI.* FROM `Target.orders` AS O JOIN `Target.order_items` AS OI ON O.order_id = OI.order_id
WHERE O.order_status = 'delivered' AND (EXTRACT(YEAR FROM O.order_purchase_timestamp) = 2018) AND
EXTRACT(MONTH FROM O.order_purchase_timestamp) BETWEEN 1 AND 8))

SELECT T1.total_cost_2017, T2.total_cost_2018, ROUND(((T2.total_cost_2018 - T1.total_cost_2017) / T1.total_cost_2017) *
100, 2) AS cost_growth_rate

FROM TABLE1 AS T1 CROSS JOIN TABLE2 AS T2;

Row	total_cost_2017 ▼	total_cost_2018 ▼	cost_growth_rate ▼
1	3472898.25	8451584.77	143.36

- 4.2** Calculate the Total & Average value of order price for each state.

SELECT C.customer_state, ROUND(AVG(OI.price + OI.freight_value), 2) AS avg_cost, ROUND(SUM(price + freight_value),
2) AS sum_cost

FROM `Target.customers` AS C JOIN `Target.orders` AS O ON C.customer_id = O.customer_id
JOIN

`Target.order_items` AS OI ON O.order_id = OI.order_id

WHERE O.order_status = 'delivered'

GROUP BY 1;

Row	customer_state	avg_cost	sum_cost
1	GO	146.78	334212.35
2	SP	124.22	5769703.15
3	RS	140.44	861472.79
4	BA	160.5	591137.81
5	MG	140.82	1818891.67
6	MT	174.76	181224.42
7	RJ	145.33	2055401.57
8	SC	145.26	595127.78
9	SE	187.44	70289.13
10	PE	176.96	308972.05

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

```
select seller_state,avg(Freight_value) as avg_freight
from `Target.order_items` join
`Target.sellers` using (seller_id) group by seller_state;
```

Row	seller_state	avg_freight
1	SP	18.45221266585...
2	MG	24.08463351081...
3	PR	22.72096874639...
4	SC	26.14651779141...
5	RS	26.03141882673...
6	DF	20.57181312569...
7	ES	32.71809139784...
8	RJ	19.47486508924...
9	GO	24.16442307692...
10	PA	19.38874999999...

5. Analysis based on sales, freight and delivery time.

5.1 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:

- **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
- **diff_estimated_delivery** = order_estimated_delivery_date - order_delivered_customer_date

```
SELECT order_id, order_purchase_timestamp, order_delivered_customer_date, order_estimated_delivery_date,
(TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase_timestamp, DAY)) AS time_to_delivery,
(TIMESTAMP_DIFF(order_estimated_delivery_date, order_purchase_timestamp, DAY)) AS diff_estimated_delivery,
(TIMESTAMP_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY)) AS diff_estdel_actdel
FROM Target.orders
WHERE order_status = 'delivered';
```

Row	order_id	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	time_to_delivery	diff_estimated_delivery	diff_estdel_actdel
1	cec8f5f7a13e5ab934a486ec9e...	2017-03-17 15:56:47 UTC	2017-04-07 13:14:56 UTC	2017-05-18 00:00:00 UTC	20	61	40
2	58527ee4726911bee84a0f42c...	2017-03-20 11:01:17 UTC	2017-03-30 14:04:04 UTC	2017-05-18 00:00:00 UTC	10	58	48
3	10ed5499d1623638ee810eff1...	2017-03-21 13:38:25 UTC	2017-04-18 13:52:43 UTC	2017-05-18 00:00:00 UTC	28	57	29
4	818996ea247803ddc123789f2...	2018-08-20 15:56:23 UTC	2018-08-29 22:52:40 UTC	2018-10-04 00:00:00 UTC	9	44	35
5	d195cac9ccaa1394ede717d38...	2018-08-12 18:14:29 UTC	2018-08-23 02:08:44 UTC	2018-10-04 00:00:00 UTC	10	52	41
6	64eeb35d3ade7fcdff9fbb1ca5...	2018-08-16 07:55:32 UTC	2018-08-23 00:09:45 UTC	2018-10-04 00:00:00 UTC	6	48	41
7	2691ae869f13b10f3d356461b...	2018-08-22 22:39:54 UTC	2018-08-29 19:11:48 UTC	2018-10-04 00:00:00 UTC	6	42	35
8	1cd147d1c0fe18f3b742a3533...	2018-08-20 17:04:34 UTC	2018-08-29 16:41:59 UTC	2018-10-04 00:00:00 UTC	8	44	35
9	b36d2e6b1781d380e140608a...	2018-08-09 19:17:50 UTC	2018-08-22 18:04:27 UTC	2018-10-04 00:00:00 UTC	12	55	42

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

```
SELECT order_id, order_purchase_timestamp, order_delivered_customer_date, order_estimated_delivery_date,
(TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase_timestamp, DAY)) AS time_to_delivery,
(TIMESTAMP_DIFF(order_estimated_delivery_date, order_purchase_timestamp, DAY)) AS diff_estimated_delivery,
(TIMESTAMP_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY)) AS diff_estdel_actdel
FROM Target.orders
WHERE order_status = 'delivered';
```

Row	order_id	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	time_to_delivery	diff_estimated_delivery	diff_estdel_actdel
1	cec8f5f7a13e5ab934a486ec9e...	2017-03-17 15:56:47 UTC	2017-04-07 13:14:56 UTC	2017-05-18 00:00:00 UTC	20	61	40
2	58527ee4726911bee84a0f42c...	2017-03-20 11:01:17 UTC	2017-03-30 14:04:04 UTC	2017-05-18 00:00:00 UTC	10	58	48
3	10ed5499d1623638ee810eff1...	2017-03-21 13:38:25 UTC	2017-04-18 13:52:43 UTC	2017-05-18 00:00:00 UTC	28	57	29
4	818996ea247803ddc123789f2...	2018-08-20 15:56:23 UTC	2018-08-29 22:52:40 UTC	2018-10-04 00:00:00 UTC	9	44	35
5	d195cac9ccaa1394ede717d38...	2018-08-12 18:14:29 UTC	2018-08-23 02:08:44 UTC	2018-10-04 00:00:00 UTC	10	52	41
6	64eeb35d3ade7fcdff9fbb1ca5...	2018-08-16 07:55:32 UTC	2018-08-23 00:09:45 UTC	2018-10-04 00:00:00 UTC	6	48	41
7	2691ae869f13b10f3d356461b...	2018-08-22 22:39:54 UTC	2018-08-29 19:11:48 UTC	2018-10-04 00:00:00 UTC	6	42	35
8	1cd147d1c0fe18f3b742a3533...	2018-08-20 17:04:34 UTC	2018-08-29 16:41:59 UTC	2018-10-04 00:00:00 UTC	8	44	35
9	b36d2e6b1781d380e140608a...	2018-08-09 19:17:50 UTC	2018-08-22 18:04:27 UTC	2018-10-04 00:00:00 UTC	12	55	42
10	88ab6b0ede7f19c65b5b71771...	2018-08-13 12:12:46 UTC	2018-08-29 20:58:39 UTC	2018-10-04 00:00:00 UTC	16	51	35

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

```
SELECT C.customer_state,
AVG(OI.freight_value) AS avg_freight_value,
AVG(BASE1.time_to_delivery) AS avg_time_to_delivery,
```

```

AVG(BASE1.diff_estimated_delivery) AS avg_diff_estimated_delivery
FROM `Target.customers` AS C
JOIN
(SELECT *,
TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS time_to_delivery,
TIMESTAMP_DIFF(order_estimated_delivery_date, order_purchase_timestamp, DAY) AS diff_estimated_delivery,
FROM `Target.orders`
WHERE order_status = 'delivered') AS BASE1 ON C.customer_id = BASE1.customer_id
JOIN
`Target.order_items` AS OI ON BASE1.order_id = OI.order_id
GROUP BY C.customer_state;

```

Row	customer_state	avg_freight_value	avg_time_to_delivery	avg_diff_estimated_delivery
1	RN	35.71808061420...	18.87332053742...	32.22456813819...
2	CE	32.73449509116...	20.53716690042...	31.02524544179...
3	RS	21.61319204434...	14.70829936409...	28.26866644929...
4	SC	21.50735904320...	14.51720771296...	25.51476690261...
5	SP	15.11518235446...	8.259662797958...	18.86938081295...
6	MG	20.62634252090...	11.51409104986...	24.26393620315...
7	BA	26.48755633994...	18.77464023893...	29.17512897094...
8	RJ	20.91143604610...	14.68882132503...	26.08406985788...
9	GO	22.56286780851...	14.94817742643...	26.62977602108...

5.4 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.

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

```

SELECT C.customer_state, ROUND(AVG(OI.freight_value), 2) AS avg_freight_value,
FROM `Target.customers` AS C
JOIN
(SELECT *,
FROM `Target.orders`
WHERE order_status = 'delivered') AS BASE1 ON C.customer_id = BASE1.customer_id
JOIN
`Target.order_items` AS OI ON BASE1.order_id = OI.order_id
GROUP BY C.customer_state
ORDER BY AVG(OI.freight_value) DESC LIMIT 5;

```

Row	customer_state	avg_freight_value
1	PB	43.09
2	RR	43.09
3	RO	41.33
4	AC	40.05
5	PI	39.12

5.4.2 QUERY FOR Top 5 states with lowest avg_freight_value:

```
SELECT C.customer_state, ROUND(AVG(OI.freight_value), 2) AS avg_freight_value,
FROM `Target.customers` AS C
JOIN
(SELECT *,
FROM `Target.orders`
WHERE order_status = 'delivered') AS BASE1 ON C.customer_id = BASE1.customer_id
JOIN
`Target.order_items` AS OI ON BASE1.order_id = OI.order_id
GROUP BY C.customer_state
ORDER BY AVG(OI.freight_value) ASC
LIMIT 5;
```

Row	customer_state	avg_freight_value
1	SP	15.12
2	PR	20.47
3	MG	20.63
4	RJ	20.91
5	DF	21.07

5.4.3 Top 5 states with highest/lowest average time to delivery

QUERY for the top 5 states with the highest average time to delivery:

```
SELECT C.customer_state, ROUND(AVG(BASE1.time_to_delivery), 2) AS avg_time_to_delivery,
FROM `Target.customers` AS C
JOIN
(SELECT *,
TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS time_to_delivery,
```

```
FROM `Target.orders`
WHERE order_status = 'delivered') AS BASE1 ON C.customer_id = BASE1.customer_id
GROUP BY C.customer_state
ORDER BY AVG(BASE1.time_to_delivery) DESC limit 5;
```

Row	customer_state	avg_time_to_delivery
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32

5.4.4 QUERY for the top 5 states with the lowest average time to delivery:

```
SELECT C.customer_state, ROUND(AVG(BASE1.time_to_delivery), 2) AS avg_time_to_delivery,
FROM `bcs1target.customer` AS C
JOIN (SELECT *,
TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS time_to_delivery
FROM `bcs1target.orders`
WHERE order_status = 'delivered') AS BASE1 ON C.customer_id = BASE1.customer_id
GROUP BY C.customer_state
ORDER BY AVG(BASE1.time_to_delivery) ASC
LIMIT 5;
```

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

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

query for the top 5 states where delivery is really fast compared to the estimated date

```
SELECT C.customer_state, ROUND(AVG(BASE1.diff_estdel_actdel), 2) AS avg_daydiff_estdel_actdel
FROM `Target.customers` AS C
JOIN
(SELECT *,
TIMESTAMP_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS diff_estdel_actdel
FROM `Target.orders`
```



```

WHERE order_status = 'delivered') AS BASE1 ON C.customer_id = BASE1.customer_id
GROUP BY C.customer_state
ORDER BY AVG(BASE1.diff_estdel_actdel) DESC
LIMIT 5;

```

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

QUERY for top 5 states where delivery is NOT SO FAST compared to the estimated date

```

SELECT C.customer_state, ROUND(AVG(BASE1.diff_estdel_actdel), 2) AS avg_daydiff_estdel_actdel
FROM `Target.customers` AS C
JOIN
(SELECT *,
TIMESTAMP_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS diff_estdel_actdel
FROM `Target.orders`
WHERE order_status = 'delivered') AS BASE1 ON C.customer_id = BASE1.customer_id
GROUP BY C.customer_state
ORDER BY AVG(BASE1.diff_estdel_actdel) ASC
LIMIT 5;

```

Row	customer_state	avg_daydiff_estdel_a
1	AL	7.95
2	MA	8.77
3	SE	9.17
4	ES	9.62
5	BA	9.93

6. Analysis based on the payments:

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

```

SELECT time_period, payment_type, COUNT(*) AS total_orders
FROM (SELECT
p.order_id, p.payment_type,
EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
EXTRACT(Month FROM order_purchase_timestamp) AS month,
FORMAT_DATE('%b %Y', DATE(ORDER_PURCHASE_TIMESTAMP)) AS time_period
FROM
`Target.payments` p JOIN `Target.orders` o
USING

```

```

(order_id)) t1
GROUP BY time_period, payment_type, t1.YEAR, t1.month
ORDER BY t1.YEAR, t1.month;

```

Row	time_period	payment_type	total_orders
1	Sep 2016	credit_card	3
2	Oct 2016	credit_card	254
3	Oct 2016	UPI	63
4	Oct 2016	voucher	23
5	Oct 2016	debit_card	2
6	Dec 2016	credit_card	1
7	Jan 2017	credit_card	583
8	Jan 2017	UPI	197
9	Jan 2017	voucher	61
10	Jan 2017	debit_card	9

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

```

SELECT P.payment_installments, COUNT(*) AS orders_count
FROM `Target.payments` AS P
JOIN
(SELECT *
FROM `Target.orders`
WHERE order_status = 'delivered') AS O ON P.order_id = O.order_id
group by 1;

```

Row	payment_installment	orders_count
1	0	2
2	1	50929
3	2	12075
4	3	10164
5	4	6891
6	5	5095
7	6	3804
8	7	1563
9	8	4136

7. Actionable Insights and Recommendations

1. Most of the orders are placed in the august

2. Most of the reviews are least rating, those items need to be reevaluated
3. We can see how the orders trajectory is showing very abrupt increase in orders volume with in very short time. Looking at overall trend, it is seen that business is picking up very fast in brazil so company has to be ready with extra workforce. To avoid high risk, it can consider hiring contractual employees
4. As Brazilian customers usually tend to buy in afternoon and night, we can increase staff in during this time frame in order to manage the customers' requests, and services better during this time by reducing workforce of morning and dawn
5. We can see, only 3 state contribute for maximum volume, and rest of the state need to be focused for improving the business.
6. The monthly orders growth rate can be as high as 62.77% (Nov 2017) during the peak seasons. So to meet high demands, inventory levels must be adjusted accordingly to ensure that all the items are stocked at optimal levels. Else Target will lose sales from potential customers and the consumers will likely seek competitors to get what they need. By losing these sales, Target will also lose out on profits
7. As a significant number of orders (66.55% of total orders) come from three south-eastern states of Brazil i.e. Sao Paulo, Rio de Janeiro, and Minas Gerais so to reduce the freight cost and delivery time of the products, Target has to continue building a strong seller network to improve customer buying experience in such regions
8. Target has to develop a good social omnipresence. It should have footprints across all social media platforms to reach new potential customers and sellers. Since the count of customers and sellers is very less in most of the northern regions of Brazil.