




























Business Case: Target SQL









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









- Data type of columns in a table









	customer_table	 QUERY ▾	 SHARE	 COPY	 SNAPSHOT	 DELETE	 EXPORT ▾
<div>SCHEMADETAILSPREVIEW</div>							
<div> Filter Enter property name or value</div>							
<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags 	Description
<input type="checkbox"/>	customer_id	STRING	NULLABLE				
<input type="checkbox"/>	customer_unique_id	STRING	NULLABLE				
<input type="checkbox"/>	customer_zip_code_prefix	INTEGER	NULLABLE				
<input type="checkbox"/>	customer_city	STRING	NULLABLE				
<input type="checkbox"/>	customer_state	STRING	NULLABLE				

	order_items_table	 QUERY ▾	 SHARE	 COPY	 SNAPSHOT	 DELETE	 EXPORT ▾
<div>SCHEMADETAILSPREVIEW</div>							
<div> Filter Enter property name or value</div>							
<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags 	Description
<input type="checkbox"/>	order_id	STRING	NULLABLE				
<input type="checkbox"/>	order_item_id	INTEGER	NULLABLE				
<input type="checkbox"/>	product_id	STRING	NULLABLE				
<input type="checkbox"/>	seller_id	STRING	NULLABLE				
<input type="checkbox"/>	shipping_limit_date	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	price	FLOAT	NULLABLE				
<input type="checkbox"/>	freight_value	FLOAT	NULLABLE				

	orders_table	 QUERY ▾	 SHARE	 COPY	 SNAPSHOT	 DELETE	 EXPORT ▾
<div>SCHEMADETAILSPREVIEW</div>							
<div> Filter Enter property name or value</div>							
<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags 	Description
<input type="checkbox"/>	order_id	STRING	NULLABLE				
<input type="checkbox"/>	customer_id	STRING	NULLABLE				
<input type="checkbox"/>	order_status	STRING	NULLABLE				
<input type="checkbox"/>	order_purchase_timestamp	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	order_approved_at	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	order_delivered_carrier_date	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	order_delivered_customer_date	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	order_estimated_delivery_date	TIMESTAMP	NULLABLE				

	payments_table	 QUERY ▾	 SHARE	 COPY	 SNAPSHOT	 DELETE	 EXPORT ▾
<div> <div>SCHEMA</div> <div>DETAILS</div> <div>PREVIEW</div> </div>							
<div> <div>Filter</div> <div>Enter property name or value</div> </div>							
<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags 	Description
<input type="checkbox"/>	order_id	STRING	NULLABLE				
<input type="checkbox"/>	payment_sequential	INTEGER	NULLABLE				
<input type="checkbox"/>	payment_type	STRING	NULLABLE				
<input type="checkbox"/>	payment_installments	INTEGER	NULLABLE				
<input type="checkbox"/>	payment_value	FLOAT	NULLABLE				

	products_table	 QUERY ▾	 SHARE	 COPY	 SNAPSHOT	 DELETE	 EXPORT ▾
<div> <div>SCHEMA</div> <div>DETAILS</div> <div>PREVIEW</div> </div>							
<div> <div>Filter</div> <div>Enter property name or value</div> </div>							
<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags 	Description
<input type="checkbox"/>	product_id	STRING	NULLABLE				
<input type="checkbox"/>	product_category	STRING	NULLABLE				
<input type="checkbox"/>	product_name_length	INTEGER	NULLABLE				
<input type="checkbox"/>	product_description_length	INTEGER	NULLABLE				
<input type="checkbox"/>	product_photos_qty	INTEGER	NULLABLE				
<input type="checkbox"/>	product_weight_g	INTEGER	NULLABLE				
<input type="checkbox"/>	product_length_cm	INTEGER	NULLABLE				
<input type="checkbox"/>	product_height_cm	INTEGER	NULLABLE				
<input type="checkbox"/>	product_width_cm	INTEGER	NULLABLE				

	sellers_table	 QUERY ▾	 SHARE	 COPY	 SNAPSHOT	 DELETE	 EXPORT ▾
<div> <div>SCHEMA</div> <div>DETAILS</div> <div>PREVIEW</div> </div>							
<div> <div>Filter</div> <div>Enter property name or value</div> </div>							
<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags 	Description
<input type="checkbox"/>	seller_id	STRING	NULLABLE				
<input type="checkbox"/>	seller_zip_code_prefix	INTEGER	NULLABLE				
<input type="checkbox"/>	seller_city	STRING	NULLABLE				
<input type="checkbox"/>	seller_state	STRING	NULLABLE				

- Time period for which the data is given

```
SELECT min(order_purchase_timestamp) AS first_order_date,
max(order_purchase_timestamp) AS last_order_date FROM Target.orders_table;
```

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

- **Cities and States of customers ordered during the given period**

`SELECT c.customer_city, c.customer_state FROM Target.customer_table c JOIN Target.orders_table o ON c.customer_id = o.customer_id GROUP BY 1,2 ORDER BY 1,2;`

Row	customer_city	customer_state
1	abadia dos dourados	MG
2	abadiania	GO
3	abaete	MG
4	abaetetuba	PA
5	abaiara	CE
6	abaira	BA
7	abare	BA
8	abatia	PR
9	abdon batista	SC
10	abelardo luz	SC
11	abranes	BA
12	abre campo	MG

2. In-depth Exploration:

- **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(month from order_purchase_timestamp) AS month, extract(year from order_purchase_timestamp) AS year, count(distinct order_id) AS no_of_orders FROM Target.orders_table GROUP BY 1,2 ORDER BY 2,1;`

Row	month	year	no_of_orders
1	9	2016	4
2	10	2016	324
3	12	2016	1
4	1	2017	800
5	2	2017	1780
6	3	2017	2682
7	4	2017	2404
8	5	2017	3700
9	6	2017	3245
10	7	2017	4026
11	8	2017	4331
12	9	2017	4285
13	10	2017	4631

Row	month	year	no_of_orders
13	10	2017	4631
14	11	2017	7544
15	12	2017	5673
16	1	2018	7269
17	2	2018	6728
18	3	2018	7211
19	4	2018	6939
20	5	2018	6873
21	6	2018	6167
22	7	2018	6292
23	8	2018	6512
24	9	2018	16
25	10	2018	4

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

```
SELECT time_of_the_day, count(*) AS count FROM (SELECT hour, CASE WHEN hour>=0 AND hour<6 THEN 'Dawn' WHEN hour>=6 AND hour<12 THEN 'Morning' WHEN hour>=12 AND hour<18 THEN 'Afternoon' WHEN hour>=18 AND hour<=23 THEN 'Night' END AS time_of_the_day FROM (SELECT extract(hour from order_purchase_timestamp) AS hour FROM `Target.orders_table`) abc) mno GROUP BY 1 ORDER BY 2 DESC;
```

Row	time_of_the_day	count
1	Afternoon	38361
2	Night	34100
3	Morning	22240
4	Dawn	4740

From the above result, we can safely say that the Brazilian customers tend to buy mostly in Afternoon (in between 12 noon and 6 pm), followed by in Night, and they buy the least in Dawn.

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

- **Get month on month orders by states**

```
SELECT extract(month from o.order_purchase_timestamp) AS month, extract(year from o.order_purchase_timestamp) AS year, c.customer_state, count(distinct o.order_id) AS no_of_orders FROM Target.orders_table o JOIN Target.customer_table c ON o.customer_id = c.customer_id GROUP BY 1,2,3 ORDER BY 2,1,3;
```

Row	month	year	customer_state	no_of_orders
1	9	2016	RR	1
2	9	2016	RS	1
3	9	2016	SP	2
4	10	2016	AL	2
5	10	2016	BA	4
6	10	2016	CE	8
7	10	2016	DF	6
8	10	2016	ES	4
9	10	2016	GO	9
10	10	2016	MA	4
11	10	2016	MG	40
12	10	2016	MT	3

- **Distribution of customers across the states in Brazil**

```
SELECT c.customer_state, count(distinct c.customer_id) AS no_of_customers FROM
Target.customer_table c JOIN Target.orders_table o ON c.customer_id = o.customer_id
GROUP BY 1 ORDER BY 1;
```

Row	customer_state	no_of_customers
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747
11	MG	11635
12	MS	715

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

- 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 round((((tcop18-tcop17)/tcop17)*100,2) AS percent_increase FROM (SELECT tcop17,
lead(tcop17,1) over (order by year asc) AS tcop18 FROM (SELECT year, sum(payment_value)
AS tcop17 FROM (SELECT *, extract(month from o.order_purchase_timestamp) AS month,
extract(year from o.order_purchase_timestamp) AS year FROM `Target.payments_table` p
JOIN `Target.orders_table` o ON p.order_id = o.order_id) abc WHERE (month>=1 AND
month<=8) AND (year IN (2017, 2018)) GROUP BY 1 ORDER BY 1) mno LIMIT 1) pqr
```

Row	percent_increase
1	136.98

- Mean & Sum of price and freight value by customer state

```
SELECT c.customer_state, avg(ot.price) AS mop, sum(ot.price) AS sop, avg(ot.freight_value)
AS mofv, sum(ot.freight_value) AS sofv FROM `Target.order_items_table` ot JOIN
`Target.orders_table` o ON ot.order_id = o.order_id JOIN `Target.customer_table` c ON
c.customer_id = o.customer_id GROUP BY 1 ORDER BY 1;
```

Row	customer_state	mop	sop	mofv	sofv
1	AC	173.727717...	15982.9499...	40.0733695...	3686.74999...
2	AL	180.889211...	80314.81	35.8436711...	15914.5899...
3	AM	135.496000...	22356.8400...	33.2053939...	5478.88999...
4	AP	164.320731...	13474.2999...	34.0060975...	2788.50000...
5	BA	134.601208...	511349.990...	26.3639589...	100156.679...
6	CE	153.758261...	227254.709...	32.7142016...	48351.5899...
7	DF	125.770548...	302603.939...	21.0413549...	50625.4999...
8	ES	121.913701...	275037.309...	22.0587765...	49764.5999...
9	GO	126.271731...	294591.949...	22.7668152...	53114.9799...
10	MA	145.204150...	119648.219...	38.2570024...	31523.7700...
11	MG	120.748574...	1585308.02...	20.6301668...	270853.460...
12	MS	142.628376...	116812.639...	23.3748840...	19144.0300...

5. Analysis on sales, freight and delivery time

- Calculate days between purchasing, delivering and estimated delivery

➤ Find time_to_delivery & diff_estimated_delivery.

```
SELECT timestamp_diff(order_purchase_timestamp, order_delivered_customer_date, day)
AS time_to_delivery, timestamp_diff(order_estimated_delivery_date,
order_delivered_customer_date, day) AS diff_estimated_delivery FROM
`Target.orders_table`;
```

Row	time_to_delivery	diff_estimated_delivery
1	-30	-12
2	-30	28
3	-35	16
4	-30	1
5	-32	0
6	-29	1
7	-43	-4
8	-40	-4
9	-37	-1
10	-33	-5
11	-38	-6
12	-36	-2
13	-34	0

- Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
SELECT customer_state, avg(freight_value) AS mofv, avg(time_to_delivery) AS mttdd,
avg(diff_estimated_delivery) AS mded FROM (SELECT *,
timestamp_diff(o.order_purchase_timestamp, o.order_delivered_customer_date, day) AS
time_to_delivery, timestamp_diff(o.order_estimated_delivery_date,
o.order_delivered_customer_date, day) AS diff_estimated_delivery FROM
`Target.order_items_table` ot JOIN `Target.orders_table` o ON o.order_id = ot.order_id JOIN
`Target.customer_table` c ON c.customer_id = o.customer_id) abc GROUP BY 1 ORDER BY 1;
```

Row	customer_state	mofv	mttd	mded
1	AC	40.0733695...	-20.329670...	20.0109890...
2	AL	35.8436711...	-23.992974...	7.97658079...
3	AM	33.2053939...	-25.963190...	18.9754601...
4	AP	34.0060975...	-27.753086...	17.4444444...
5	BA	26.3639589...	-18.774640...	10.1194678...
6	CE	32.7142016...	-20.537166...	10.2566619...
7	DF	21.0413549...	-12.501486...	11.2747346...
8	ES	22.0587765...	-15.192808...	9.76853932...
9	GO	22.7668152...	-14.948177...	11.3728590...
10	MA	38.2570024...	-21.203750...	9.10999999...
11	MG	20.6301668...	-11.515522...	12.3971510...
12	MS	23.3748840...	-15.107274...	10.3378545...

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

Highest

```
SELECT c.customer_state, avg(ot.freight_value) AS average_freight_value FROM
`Target.order_items_table` ot JOIN `Target.orders_table` o ON o.order_id = ot.order_id JOIN
`Target.customer_table` c ON c.customer_id = o.customer_id GROUP BY 1 ORDER BY 2 DESC
LIMIT 5;
```

Row	customer_state	average_freight_value
1	RR	42.984423076923093
2	PB	42.723803986710941
3	RO	41.069712230215842
4	AC	40.073369565217405
5	PI	39.147970479704767

Lowest

```
SELECT c.customer_state, avg(ot.freight_value) AS average_freight_value FROM
`Target.order_items_table` ot JOIN `Target.orders_table` o ON o.order_id = ot.order_id JOIN
```

```
`Target.customer_table` c ON c.customer_id = o.customer_id GROUP BY 1 ORDER BY 2 LIMIT 5;
```

Row	customer_state	average_freight_value
1	SP	15.147275390419132
2	PR	20.531651567944269
3	MG	20.630166806306651
4	RJ	20.960923931682483
5	DF	21.041354945968422

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

Highest

```
SELECT c.customer_state, avg(timestamp_diff(o.order_purchase_timestamp,
o.order_delivered_customer_date, day)) AS average_time_to_delivery_days FROM
`Target.order_items_table` ot JOIN `Target.orders_table` o ON o.order_id = ot.order_id JOIN
`Target.customer_table` c ON c.customer_id = o.customer_id GROUP BY 1 ORDER BY 2 LIMIT 5;
```

Row	customer_state	average_time_to_delivery_days
1	RR	-27.826086956521738
2	AP	-27.753086419753075
3	AM	-25.963190184049076
4	AL	-23.992974238875881
5	PA	-23.301707779886126

Lowest

```
SELECT c.customer_state, avg(timestamp_diff(o.order_purchase_timestamp,
o.order_delivered_customer_date, day)) AS average_time_to_delivery_days FROM
`Target.order_items_table` ot JOIN `Target.orders_table` o ON o.order_id = ot.order_id JOIN
`Target.customer_table` c ON c.customer_id = o.customer_id GROUP BY 1 ORDER BY 2 DESC
LIMIT 5;
```

Row	customer_state	average_time_to_delivery_days
1	SP	-8.25960855241909
2	PR	-11.480793060718735
3	MG	-11.515522180072811
4	DF	-12.501486199575384
5	SC	-14.520985846754517

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

Fast

```
SELECT c.customer_state, avg(timestamp_diff(o.order_estimated_delivery_date,
o.order_delivered_customer_date, day)) AS time_days FROM `Target.order_items_table` ot
JOIN `Target.orders_table` o ON o.order_id = ot.order_id JOIN `Target.customer_table` c ON
c.customer_id = o.customer_id GROUP BY 1 ORDER BY 2 DESC LIMIT 5;
```

Row	customer_state	time_days
1	AC	20.0109890...
2	RO	19.0805860...
3	AM	18.9754601...
4	AP	17.4444444...
5	RR	17.4347826...

Not so fast

```
SELECT c.customer_state, avg(timestamp_diff(o.order_estimated_delivery_date,
o.order_delivered_customer_date, day)) AS time_days FROM `Target.order_items_table` ot
JOIN `Target.orders_table` o ON o.order_id = ot.order_id JOIN `Target.customer_table` c ON
c.customer_id = o.customer_id GROUP BY 1 ORDER BY 2 LIMIT 5;
```

Row	customer_state	time_days
1	AL	7.97658079...
2	MA	9.10999999...
3	SE	9.16533333...
4	ES	9.76853932...
5	BA	10.1194678...

6. Payment type analysis:

- **Month over Month count of orders for different payment types**

```
SELECT extract(month from o.order_purchase_timestamp) AS month, extract(year from
o.order_purchase_timestamp) AS year, p.payment_type, count(o.order_id) AS
count_of_orders FROM `Target.orders_table` o JOIN `Target.payments_table` p ON
o.order_id = p.order_id GROUP BY 1,2,3 ORDER BY 2,1;
```

Row	month	year	payment_type	count_of_orders
1	9	2016	credit_card	3
2	10	2016	credit_card	254
3	10	2016	UPI	63
4	10	2016	voucher	23
5	10	2016	debit_card	2
6	12	2016	credit_card	1
7	1	2017	credit_card	583
8	1	2017	UPI	197
9	1	2017	voucher	61
10	1	2017	debit_card	9
11	2	2017	credit_card	1356
12	2	2017	UPI	398

- Count of orders based on the no. of payment installments

```
SELECT p.payment_installments, count(*) AS count_of_orders FROM
`Target.payments_table` p JOIN `Target.orders_table` o ON p.order_id = o.order_id GROUP
BY 1 ORDER BY 1;
```

Row	payment_installments	count_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

Actionable Insights:

From the given dataset, we can draw some interesting actionable insights regarding the sales at Target (a US based retail industry). The dataset requires us to study the orders at Target for the Brazilian market between the time period September 2016 and October 2018.

For the given data, the first order was placed on September 4th, 2016 followed by 3 more orders in the same month, following which we can see an upward trend in the number of orders placed by distinct customers every month till the second half of 2018. The final order was placed on October 17th, 2018. The customers range from 4310 different cities in Brazil, which in-turn includes several municipalities. These cities form a part of all 27 states in Brazil. Therefore, the customers from every state of Brazil has placed an order at least once in the given timeframe.

There is a robust growing trend of e-commerce in Brazil as observed from the dataset. The number of customers have increased considerably as witnessed by the increase in the number of distinct order ids every subsequent month. There is a dramatic increase in the orders placed in the month of November in 2017 to 7544 orders, mainly because of the Christmas season next month all the way up to the new year, when the company also introduces offers to increase their sales. Alternatively, in the month of March 2017 we can clearly see the peak orders placed, crossing 7k mark yet again. This is because the month of April houses various public holidays and festivals in Brazil such as Good Friday, Orthodox Easter, and Tiradentes Day. The maximum orders were placed in the Afternoon (during the time between 12 noon and 6pm) followed by in the Night, which indicates that customers are usually active in the second half of the day as during this period they mostly get some leisure time from their daily work. At Dawn, the customers are usually inactive, which is evident by the fact that the lowest number of orders are placed in between 12am and 6am.

The state that recorded the highest number of orders placed in any given month is Sao Paulo (SP). Sao Paulo is a major industrial hub of Brazil and holds the 33% share in Brazil's GDP. Therefore, it can be understood that due to the increased economic activity, Sao Paulo's population would have greater share in retail shopping. Apart from that, the state of Minas Gerais (MG) holds the largest customer base of Target, accounting to 11635 customers. One possible reason is that MG is the most populous state of Brazil, and though the economic activity is less compared to that of Sao Paulo, the customer base is large.

Alternatively, there is a sharp increase of approximately 137% in the cost of orders from 2017 first half to 2018 first half. As more number of customers have joined in, leading to the large number of orders placed, we can conclude that the Target's recognition and trust has risen steeply in Brazil. The mean freight value in the state of Roraima (RR) is highest among all the states in Brazil. This is because Roraima is the most logistically and geographically isolated state in the country, thereby increasing the logistics cost.

The orders on average are delivered in 2 weeks from the time of order placed. Also, the orders are delivered almost 10 days prior to the estimated delivery date. This shows that Target is working efficiently in Brazil and is catering to the customer needs in a well organized manner. The lowest average freight value and average time to delivery is of the Sao Paulo state largely because of the high economic activity in the state rendering low logistics cost, and being the business hub of Brazil.

Finally on part of payment analysis, most of the customers of Target preferred payments through credit card, followed by UPI. Credit card seems to be the most feasible option for payments all over the world and the case of Brazil is no different. In the recent years, we

have seen a tremendous rise of UPI payments as it is fast, secure and almost effortless. Therefore, it is no surprise that Target has received a lot of orders through UPI payments as well. A large number of orders are made through first installment of payments signifying that for retail shopping customers prefer having no due as the cost of most orders is low. As the cost increases, the payment installment also increases.

Recommendations

Target has most of its bases covered but the retail giant can further increase its revenue by filling up some grey areas. During the festive season in Brazil such as the months of December and April, the month prior to these observe a large influx of orders. To further accommodate such large orders, it would be conducive to restock the products, add up new maintenance lines, increase the manpower, etc. during this time of the year. Apart from that, we also saw that the majority of orders were placed in the second half of the day. To get more orders, Target can try sending recommendations to its customers during the second half of the day, mainly because during that time the customers sign off from their work and enjoy their personal time online. This measure will proportionally boost Target's sales.

Target can build more stores in the states which have high economic activity largely in Sao Paulo and Minas Gerais. These states have a large customer base, which is constantly increasing with time. Adding more stores will also create new supply chains which will reduce the logistics cost, thereby further increasing the new customers in these states. The cost of orders have risen sharply by 137% from the first half of 2017 to the first half of 2018. To add more value to this, Target should launch online marketing campaigns and advertisement so that more and more people can become aware of Target in Brazil.

Target should build more warehouses in Brazil to cater large demand of its products. This will bring down the delivery time from 2 weeks to less especially in the border states. The 2 weeks time to product delivery is comparatively higher for retail businesses, and therefore it can only be reduced if Target has more warehouses.