

Target Case Study

1. To get the data types of all the columns in the table, run the following command.

BigQuery Command

```
SELECT DATA_TYPE from scaler-dsml-380510.Target.INFORMATION_SCHEMA.COLUMNS
where table_schema = 'Target' and table_name = 'geolocation';
```

To get the data types of all the tables, change the table_name to get the respective table column's data types.

Google Cloud Scaler-DSML Search (/) for resources, docs, products

SANDBOX Set up billing to upgrade to the full BigQuery experience. [Learn more](#)

*Unsaved query 2

RUN SAVE SHARE SCHEDULE MORE

```
1 SELECT DATA_TYPE from scaler-dsml-380510.Target.INFORMATION_SCHEMA.COLUMNS
2 where table_schema = 'Target' and table_name = 'geolocation';
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION
Row	DATA_TYPE				
1	INT64				
2	FLOAT64				
3	FLOAT64				
4	STRING				
5	STRING				

2. To get the time period of the orders data, we need to get the starting and ending timestamps of the orders from the orders table. Run the following command to get the result.

We need to find the minimum and maximum timestamps to get the starting and ending timestamps.

```
select min(order_purchase_timestamp),max(order_purchase_timestamp) from `Target.orders`;
```

```
1 select min(order_purchase_timestamp),max(order_purchase_timestamp) from `Target.orders`;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	f0_	f1_			
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC			

2016-09-04 is the starting timestamp and 2018-10-17 is the ending timestamp

3. To get the cities and states of the customers ordered in a given period, use a geolocation table to get the result. Run the following commands individually to get distinct states and cities from where orders are placed.

```
select distinct(geolocation_city) from `Target.geolocation`;  
select distinct(geolocation_state) from `Target.geolocation`;
```

```
1 select distinct(geolocation_city) from `Target.geolocation`;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	geolocation_city			
1	aracaju			
2	riachuelo			
3	nossa senhora do socorro			
4	barra dos coqueiros			
5	itaporanga d'ajuda			
6	sao cristovao			

```
1 select distinct(geolocation_state) from `Target.geolocation`;
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	geolocation_state			
1	SE			
2	AL			
3	PI			
4	AP			
5	AM			
6	RR			

2. 1. In Brazil, the number of orders has an upward trend in 2017 and a downward trend in 2018.

```
select count(order_id) as sales,extract(year from order_purchase_timestamp) as year,  
extract(month from order_purchase_timestamp) as month  
from `Target.orders`  
group by year,month  
order by year,month;
```

```

1 select count(order_id) as sales,
2 extract(year from order_purchase_timestamp) as year,
3 extract(month from order_purchase_timestamp) as month
4 from `Target.orders`
5 group by year,month
6 order by year,month;

```

Query results

JOB INFORMATION		RESULTS		JSON	EXECUTION I
Row	sales	year	month		
1	4	2016	9		
2	324	2016	10		
3	1	2016	12		
4	800	2017	1		

2. Brazilian customers tend to buy more during the afternoon, i.e., between 1 pm and 6 pm. Most of the sales(orders) happened during this time.

```

with
a as
(select
CASE
when extract(hour from order_purchase_timestamp) between 0 and 6 then "DAWN"
when extract(hour from order_purchase_timestamp) between 7 and 12 then "MORNING"
when extract(hour from order_purchase_timestamp) between 13 and 18 then "AFTERNOON"
when extract(hour from order_purchase_timestamp) between 19 and 23 then "NIGHT"
END AS time,count(order_id) as sales from `Target.orders`
group by time
order by sales DESC)

select * from a;

```

```

1 with
2 a as
3 (select
4 CASE
5 when extract(hour from order_purchase_timestamp) between 0 and 6 then "DAWN"
6 when extract(hour from order_purchase_timestamp) between 7 and 12 then "MORNING"
7 when extract(hour from order_purchase_timestamp) between 13 and 18 then "AFTERNOON"

```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	time	sales				
1	AFTERNOON	38135				
2	NIGHT	28331				
3	MORNING	27733				
4	DAWN	5242				

3. 1. To get the monthly count of the orders for each state, combine two tables of customers and orders and group them by state of customer and month of purchase.

```

select count(o.order_id) as sales,extract(month from o.order_purchase_timest
amp) as month,c.customer_state as state
from `Target.orders` AS o
join `Target.customers` AS c
on o.customer_id=c.customer_id
group by state,month
order by state,month;

```

Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	sales	month	state
1	8	1	AC
2	6	2	AC
3	4	3	AC

2. The count of customers among states gives the customer partition among states.

```
select customer_state,count(customer_id) as CNT
from `Target.customers`
group by customer_state
order by CNT;
```

Query results

JOB INFORMATION RESULTS JSON EXEC

Row	customer_state	CNT
1	RR	46
2	AP	68
3	AC	81
4	AM	148
5	PO	252


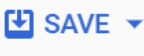


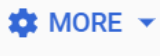
4. 1. % increase in the cost of orders in 2018 respective to 2017
with

```
a as
(select sum(payment_value) as pv1
from `Target.payments` as p
join `Target.orders` as o
on o.order_id=p.order_id
```

```

where extract(year from o.order_purchase_timestamp) = 2017
      and extract(month from o.order_purchase_timestamp) between 1 and 8),
b as
(select sum(payment_value) as pv2
from `Target.payments` as p
join `Target.orders` as o
on o.order_id=p.order_id
where extract(year from o.order_purchase_timestamp) = 2018
      and extract(month from o.order_purchase_timestamp) between 1 and 8)
select round(((pv2-pv1)/pv1)*100,2)
from a
cross join b;

```

```

1 with
2 a as
3 (select sum(payment_value) as pv1
4 from `Target.payments` as p
5 join `Target.orders` as o
6 on o.order_id=p.order_id
7 where extract(year from o.order_purchase_timestamp) = 2017
8      and extract(month from o.order_purchase_timestamp) between 1 and 8),
9 b as
10 [(select sum(payment_value) as pv2
11 from `Target.payments` as p
12 join `Target.orders` as o
13 on o.order_id=p.order_id
14 where extract(year from o.order_purchase_timestamp) = 2018
15      and extract(month from o.order_purchase_timestamp) between 1 and 8)]
16 select round(((pv2-pv1)/pv1)*100,2)
17 from a

```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUT
Row	f0_				
1	136.98				

There was a 136.98% rise in order cost in 2018 compared to 2017.

2. Mean and the sum of price and freight Value

```

select sum(price) AS Price,sum(freight_value) AS Freight_Value,
sum(price)/count(I.order_id) AS Price_Mean,sum(freight_value)/count(I.order_
id) AS Freight_Value_Mean,C.customer_state
from `Target.order_items` AS I
join `Target.orders` AS O
on I.order_id=O.order_id
join `Target.customers` AS C
on O.customer_id=C.customer_id
group by C.customer_state

```

order by C.customer_state;

<div> <div> <div>2023-03-13 21:45:24</div> <div> <div>Unsaved query 2</div> <div>Unsaved query</div> <div>Unsaved query 4</div> <div>Unsaved query 5</div> </div> </div> <div> <div>RUN</div> <div>SAVE</div> <div>SHARE</div> <div>SCHEDULE</div> <div>MORE</div> </div> </div>					
<pre> 1 select sum(price) AS Price,sum(freight_value) AS Freight_Value,sum(price)/count(I.order_id) AS Price_Mean,sum(freight_value)/count(I.order_id) AS Freight_Value_Mean,C.customer_state 2 from `Target.order_items` AS I 3 join `Target.orders` AS O 4 on I.order_id=O.order_id 5 join `Target.customers` AS C 6 on O.customer_id=C.customer_id 7 group by C.customer_state 8 order by C.customer_state; </pre>					
<div>Query results</div> <div> <div>SAVE RESULTS</div> <div>EXPLORE</div> </div>					
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	Price	Freight_Value	Price_Mean	Freight_Value_M	customer_state
1	15982.9499...	3686.75000...	173.727717...	40.0733695...	AC
2	80314.8099...	15914.5899...	180.889211...	35.8436711...	AL
3	22356.8400...	5478.89000...	135.496000...	33.2053939...	AM
4	13474.2999...	2788.50000...	164.320731...	34.0060975...	AP
5	511349.990...	100156.679...	134.601208...	26.3639589...	BA

5. 1. The number of days between purchase and delivery is stored as DeliveryTime
The number of days between purchase and estimated delivery is stored as EstimatedDeliveryTime

```

select DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,Day)
AS DeliveryTime,
DATE_DIFF(order_estimated_delivery_date,order_purchase_timestamp,Day)
AS EstimatedDeliveryTime
from `Target.orders`
where order_status="delivered";

```

<pre> 1 select DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,Day) AS DeliveryTime, 2 DATE_DIFF(order_estimated_delivery_date,order_purchase_timestamp,Day) AS EstimatedDeliveryTime 3 from `Target.orders` 4 where order_status="delivered"; </pre>					
<div>Query results</div>					
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	DeliveryTime	EstimatedDelive			
1	30	32			
2	32	33			
3	29	31			
4	43	39			
5	40	36			
6	37	35			
7	33	28			

- 2.
- ```

select 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`;

```

```

1 select DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,Day) AS time_to_delivery,
2 DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,Day) AS diff_estimated_delivery
3 from `Target.orders`;
4

```

#### Query results

| JOB INFORMATION |                  | RESULTS                 |  | JSON | EXECUTION DETAILS | EXECUTION GRAPH | PREVIEW |
|-----------------|------------------|-------------------------|--|------|-------------------|-----------------|---------|
| 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                      |  |      |                   |                 |         |

3.

```

select avg(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,
Day)) AS time_to_delivery,

```

```

avg(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,Day)) AS diff_estimated_delivery,

```

```

avg(I.freight_value) AS Freight_Value,C.customer_state

```

```

from `Target.orders` AS O

```

```

join `Target.order_items` AS I

```

```

on O.order_id=I.order_id

```

```

join `Target.customers` AS C

```

```

on O.customer_id=C.customer_id

```

```

where order_status="delivered"

```

```

group by C.customer_state;

```

```

1 select avg(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,Day)) AS time_to_delivery,
2 avg(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,Day)) AS diff_estimated_delivery,
3 avg(I.freight_value) AS Freight_Value,C.customer_state
4 from `Target.orders` AS O
5 join `Target.order_items` AS I
6 on O.order_id=I.order_id
7 join `Target.customers` AS C
8 on O.customer_id=C.customer_id
9 where order_status="delivered"
10 group by C.customer_state;

```

#### Query results

| JOB INFORMATION |                  | RESULTS                 |               | JSON           | EXECUTION DETAILS | EXECUTION GRAPH | PREVIEW |
|-----------------|------------------|-------------------------|---------------|----------------|-------------------|-----------------|---------|
| Row             | time_to_delivery | diff_estimated_delivery | Freight_Value | customer_state |                   |                 |         |
| 1               | 14.94817742...   | 11.372859025032927      | 22.5628678... | GO             |                   |                 |         |
| 2               | 8.259662797...   | 10.264141599018073      | 15.1151823... | SP             |                   |                 |         |
| 3               | 14.70829936...   | 13.203000163052323      | 21.6131920... | RS             |                   |                 |         |
| 4               | 18.77464023...   | 10.119467825142538      | 26.4875563... | BA             |                   |                 |         |
| 5               | 11.51409104...   | 12.399039950449046      | 20.6263425... | MG             |                   |                 |         |

5.

```

select avg(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,
Day)) AS time_to_delivery,

```

```

avg(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,Day)) AS diff_estimated_delivery,

```

```

avg(I.freight_value) AS Freight_Value,C.customer_state

```

```

from `Target.orders` AS O

```

```

join `Target.order_items` AS I

```

```

on O.order_id=I.order_id

```

```

join `Target.customers` AS C

```

```

on O.customer_id=C.customer_id
where order_status="delivered"
group by C.customer_state
order by freight_value DESC
LIMIT 5;

```

```

3 avg(I.freight_value) AS Freight_Value,C.customer_state
4 from `Target.orders` AS O
5 join `Target.order_items` AS I
6 on O.order_id=I.order_id
7 join `Target.customers` AS C
8 on O.customer_id=C.customer_id
9 where order_status="delivered"
10 group by C.customer_state
11 order by freight_value DESC
12 LIMIT 5;

```

## Query results

| JOB INFORMATION |                  | RESULTS                 | JSON          | EXECUTION DETAILS | EXECUTION TIME |
|-----------------|------------------|-------------------------|---------------|-------------------|----------------|
| Row             | time_to_delivery | diff_estimated_delivery | Freight_Value | customer_state    |                |
| 1               | 20.1194539...    | 12.1501706...           | 43.0916894... | PB                |                |
| 2               | 27.8260869...    | 17.4347826...           | 43.0880434... | RR                |                |
| 3               | 19.2820512...    | 19.0805860...           | 41.3305494... | RO                |                |
| 4               | 20.3296703...    | 20.0109890...           | 40.0479120... | AC                |                |
| 5               | 18.9311663...    | 10.6826003...           | 39.1150860... | PI                |                |

6.

```

select avg(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,
Day)) AS time_to_delivery,

```

```

avg(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,Day)) AS diff_estimated_delivery,

```

```

avg(I.freight_value) AS Freight_Value,C.customer_state

```

```

from `Target.orders` AS O

```

```

join `Target.order_items` AS I

```

```

on O.order_id=I.order_id

```

```

join `Target.customers` AS C

```

```

on O.customer_id=C.customer_id

```

```

where order_status="delivered"

```

```

group by C.customer_state

```

```

order by time_to_delivery DESC

```

```

LIMIT 5;

```

```

3 avg(I.freight_value) AS Freight_Value,C.customer_state
4 from `Target.orders` AS O
5 join `Target.order_items` AS I
6 on O.order_id=I.order_id
7 join `Target.customers` AS C
8 on O.customer_id=C.customer_id
9 where order_status="delivered"
10 group by C.customer_state
11 order by time_to_delivery DESC
12 LIMIT 5;

```

## Query results

| JOB INFORMATION |                  | RESULTS                 | JSON          | EXECUTION DETAILS | EXECUTION TIME |
|-----------------|------------------|-------------------------|---------------|-------------------|----------------|
| Row             | time_to_delivery | diff_estimated_delivery | Freight_Value | customer_state    |                |
| 1               | 27.8260869...    | 17.4347826...           | 43.0880434... | RR                |                |
| 2               | 27.7530864...    | 17.4444444...           | 34.1604938... | AP                |                |
| 3               | 25.9631901...    | 18.9754601...           | 33.3106134... | AM                |                |
| 4               | 23.9929742...    | 7.97658079...           | 35.8706557... | AL                |                |
| 5               | 23.3017077...    | 13.3747628...           | 35.6290132... | PA                |                |



```

7.
select avg(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,
Day)) AS time_to_delivery,

avg(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,Day)) AS diff_estimated_delivery,
avg(I.freight_value) AS Freight_Value,C.customer_state
from `Target.orders` AS O
join `Target.order_items` AS I
on O.order_id=I.order_id
join `Target.customers` AS C
on O.customer_id=C.customer_id
where order_status="delivered"
group by C.customer_state
order by diff_estimated_delivery DESC
LIMIT 5;

```

```

3 avg(I.freight_value) AS Freight_Value,C.customer_state
4 from `Target.orders` AS O
5 join `Target.order_items` AS I
6 on O.order_id=I.order_id
7 join `Target.customers` AS C
8 on O.customer_id=C.customer_id
9 where order_status="delivered"
10 group by C.customer_state
11 order by diff_estimated_delivery DESC
12 LIMIT 5;

```






### Query results

| JOB INFORMATION |                  | RESULTS                 | JSON          | EXECUTION DETAILS |
|-----------------|------------------|-------------------------|---------------|-------------------|
| Row             | time_to_delivery | diff_estimated_delivery | Freight_Value | customer_state    |
| 1               | 20.3296703...    | 20.0109890...           | 40.0479120... | AC                |
| 2               | 19.2820512...    | 19.0805860...           | 41.3305494... | RO                |
| 3               | 25.9631901...    | 18.9754601...           | 33.3106134... | AM                |
| 4               | 27.7530864...    | 17.4444444...           | 34.1604938... | AP                |
| 5               | 27.8260869...    | 17.4347826...           | 43.0880434... | RR                |

6. 1. Month-by-month count of orders for various payment types:

```
select extract(year from O.order_purchase_timestamp) AS year,
extract(month from O.order_purchase_timestamp) AS month,P.payment_type,count
(P.order_id)
```

```
from `Target.payments` AS P
join `Target.orders` AS O
on P.order_id=O.order_id
group by P.payment_type,year,month
order by year,month;
```

| <div> <div> RUN</div> <div> SAVE</div> <div> SHARE</div> <div> SCHEDULE</div> <div> MORE</div> </div> |                                                               |  |  |  |  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|--|--|--|--|
| 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | select extract(year from O.order_purchase_timestamp) AS year, |  |  |  |  |
| 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | extract(month from O.order_purchase_timestamp) AS month,      |  |  |  |  |
| 3                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | P.payment_type,count(P.order_id)                              |  |  |  |  |
| 4                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | from `Target.payments` AS P                                   |  |  |  |  |
| 5                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | join `Target.orders` AS O                                     |  |  |  |  |
| 6                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | on P.order_id=O.order_id                                      |  |  |  |  |

| Query results   |      |         |              |      |                   |
|-----------------|------|---------|--------------|------|-------------------|
| JOB INFORMATION |      | RESULTS |              | JSON | EXECUTION DETAILS |
| Row             | year | month   | payment_type | fo_  |                   |
| 1               | 2016 | 9       | credit_card  |      | 3                 |
| 2               | 2016 | 10      | credit_card  |      | 254               |
| 3               | 2016 | 10      | voucher      |      | 23                |
| 4               | 2016 | 10      | debit_card   |      | 2                 |
| 5               | 2016 | 10      | UPI          |      | 63                |
| 6               | 2016 | 12      | credit_card  |      | 1                 |
| 7               | 2017 | 1       | voucher      |      | 61                |

2. Count of orders based on the number of installments

```
select count(order_id) AS Orders,payment_installments
from `Target.payments`
group by payment_installments;
```

|   |                                                       |  |  |  |  |
|---|-------------------------------------------------------|--|--|--|--|
| 1 | select count(order_id) AS Orders,payment_installments |  |  |  |  |
| 2 | from `Target.payments`                                |  |  |  |  |
| 3 | group by payment_installments;                        |  |  |  |  |

| Query results   |        |                      |             |
|-----------------|--------|----------------------|-------------|
| JOB INFORMATION |        | RESULTS              | JSON        |
| Row             | Orders | payment_installments | EXECUTION D |
| 1               | 2      | 0                    |             |
| 2               | 52546  | 1                    |             |
| 3               | 12413  | 2                    |             |
| 4               | 10461  | 3                    |             |
| 5               | 7098   | 4                    |             |
| 6               | 5239   | 5                    |             |
| 7               | 3920   | 6                    |             |
| 8               | 1626   | 7                    |             |
| 9               | 4268   | 8                    |             |