

### Problem Statement

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

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

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

Query : `SELECT column_name, data_type FROM `retailer-businesscase.retail.INFORMATION_SCHEMA.COLUMNS` where table_name = 'customers'`

Output:

| Query results   |                          |           |      |
|-----------------|--------------------------|-----------|------|
| JOB INFORMATION |                          | RESULTS   | JSON |
| Row             | column_name              | data_type |      |
| 1               | customer_id              | STRING    |      |
| 2               | customer_unique_id       | STRING    |      |
| 3               | customer_zip_code_prefix | INT64     |      |
| 4               | customer_city            | STRING    |      |
| 5               | customer_state           | STRING    |      |

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

Query : `SELECT min(order_purchase_timestamp), max(order_purchase_timestamp) FROM `retailer-businesscase.retail.orders``

Output:

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

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

Query : `SELECT count(distinct customer_city) customer_city, count(distinct customer_state) customer_state  
FROM `retailer-businesscase.retail.customers``

Output:

| Query results   |               |                |      |
|-----------------|---------------|----------------|------|
| JOB INFORMATION |               | RESULTS        | JSON |
| Row             | customer_city | customer_state |      |
| 1               | 4119          | 27             |      |

## 2. In-depth Exploration:

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

Yes, there is growing trend

Query :`SELECT EXTRACT(YEAR from order_purchase_timestamp) year, count(order_id) order_count  
FROM `retailer-businesscase.retail.orders`  
group by EXTRACT(YEAR from order_purchase_timestamp)`

Output:

| Query results   |      |             |      |
|-----------------|------|-------------|------|
| JOB INFORMATION |      | RESULTS     | JSON |
| Row             | year | order_count |      |
| 1               | 2017 | 45101       |      |
| 2               | 2018 | 54011       |      |
| 3               | 2016 | 329         |      |

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

Query : `SELECT month, count(order_purchase_timestamp) monthly_order_count from (  
SELECT order_purchase_timestamp, EXTRACT(YEAR FROM order_purchase_timestamp) year,  
EXTRACT(MONTH FROM order_purchase_timestamp) month  
FROM `retailer-businesscase.retail.orders`) year_mon_wise_orders  
GROUP BY month`

ORDER BY month

Output:

| Row | month | monthly_order_count |
|-----|-------|---------------------|
| 1   | 1     | 8069                |
| 2   | 2     | 8508                |
| 3   | 3     | 9893                |
| 4   | 4     | 9343                |
| 5   | 5     | 10573               |
| 6   | 6     | 9412                |
| 7   | 7     | 10318               |
| 8   | 8     | 10843               |
| 9   | 9     | 4305                |
| 10  | 10    | 4959                |
| 11  | 11    | 7511                |

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 hrs : Afternoon
- 19-23 hrs : Night

Query: 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  
"Mornings"

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 timeofday,

count(order\_purchase\_timestamp) order\_count

FROM `retailer-businesscase.retail.orders`

GROUP BY timeofday

Output:

## Query results

| JOB INFORMATION |             | RESULTS       | JSON | EXECUTION |
|-----------------|-------------|---------------|------|-----------|
| Row             | timeofday ▼ | order_count ▼ |      |           |
| 1               | Mornings    | 27733         |      |           |
| 2               | Dawn        | 5242          |      |           |
| 3               | Afternoon   | 38135         |      |           |
| 4               | Night       | 28331         |      |           |

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

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

Query : `SELECT c.customer_state,  
EXTRACT(MONTH from order_purchase_timestamp) month, count(order_id) orders_count  
FROM `retailer-businesscase.retail.orders` o  
JOIN `retailer-businesscase.retail.customers` c  
ON o.customer_id=c.customer_id  
GROUP BY c.customer_state, month`

Output:

| Query results   |                  |         |                | <a href="#">SAVE RESULTS</a> ▼ | <a href="#">CHART</a>         |
|-----------------|------------------|---------|----------------|--------------------------------|-------------------------------|
| JOB INFORMATION |                  | RESULTS | JSON           | EXECUTION DETAILS              | CHART <a href="#">PREVIEW</a> |
| Row             | customer_state ▼ | month ▼ | orders_count ▼ |                                |                               |
| 1               | RN               | 1       | 51             |                                |                               |
| 2               | RN               | 12      | 30             |                                |                               |
| 3               | RN               | 5       | 39             |                                |                               |
| 4               | CE               | 2       | 101            |                                |                               |
| 5               | CE               | 3       | 126            |                                |                               |
| 6               | CE               | 5       | 136            |                                |                               |
| 7               | CE               | 4       | 143            |                                |                               |
| 8               | RS               | 3       | 569            |                                |                               |
| 9               | RS               | 6       | 526            |                                |                               |
| 10              | SC               | 8       | 365            |                                |                               |

Results per page: 50 ▼ 1 – 50 of 322

[PERSONAL HISTORY](#) [PROJECT HISTORY](#)


2. How are the customers distributed across all the states?


Query : `SELECT customer_state, count(customer_id) no_of_customers  
FROM `retailer-businesscase.retail.customers``

GROUP BY customer\_state

Output:

Query results

 SAVE RESULTS



JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

CHART

PREVIEW

EXEC

| Row | customer_state | no_of_customers |
|-----|----------------|-----------------|
| 1   | RN             | 485             |
| 2   | CE             | 1336            |
| 3   | RS             | 5466            |
| 4   | SC             | 3637            |
| 5   | SP             | 41746           |
| 6   | MG             | 11635           |
| 7   | BA             | 3380            |
| 8   | RJ             | 12852           |
| 9   | GO             | 2020            |
| 10  | MA             | 747             |

Results per page:

50

1 – 27 of 27

PERSONAL HISTORY

PROJECT HISTORY

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

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.

Query:

```
with ORDERS_HISTORY as (
select order_id, EXTRACT(MONTH from order_purchase_timestamp) month, EXTRACT(YEAR
from order_purchase_timestamp) year
from retailer-businesscase.retail.orders
),
FILTERING_YEAR_2017_ORDERS as (
select month month_2017, year year_2017, sum(payment_value) payment_value_2017
from ORDERS_HISTORY oh
left join retailer-businesscase.retail.payments p on p.order_id = oh.order_id
where month between 1 and 8 and year = 2017
group by year, month),
FILTERING_YEAR_2018_ORDERS as (
select month month_2018, year year_2018, sum(payment_value) payment_value_2018
from ORDERS_HISTORY oh left join retailer-businesscase.retail.payments p on p.order_id =
oh.order_id
where month between 1 and 8 and year = 2018
group by year, month
)
```

```
select year_2017, year_2018, month_2017 month, ((payment_value_2018-
payment_value_2017)/payment_value_2017)*100
pct_increase_in_cost_of_orders
from FILTERING_YEAR_2017_ORDERS payments_2017 join FILTERING_YEAR_2018_ORDERS
payments_2018
on payments_2017.month_2017 = payments_2018.month_2018
order by month_2017
```

Output:

Query results

 SAV


| JOB INFORMATION |           | RESULTS   | JSON  | EXECUTION DETAILS |                      | CHART |
|-----------------|-----------|-----------|-------|-------------------|----------------------|-------|
| Row             | year_2017 | year_2018 | month |                   | pct_increase_in_cost |       |
| 4               | 2017      | 2018      | 4     |                   | 177.8407701149...    |       |
| 5               | 2017      | 2018      | 5     |                   | 94.62734375677...    |       |
| 6               | 2017      | 2018      | 6     |                   | 100.2596912456...    |       |
| 7               | 2017      | 2018      | 7     |                   | 80.04245463390...    |       |
| 8               | 2017      | 2018      | 8     |                   | 51.60600520477...    |       |

2. Calculate the Total & Average value of order price for each state

Query:

```
SELECT c.customer_state, sum(price) total_order_price, avg(price) avg_order_price FROM
`retailer-businesscase.retail.customers` c
join `retailer-businesscase.retail.orders` o on c.customer_id = o.customer_id
join `retailer-businesscase.retail.order_items` oi on oi.order_id = o.order_id
group by c.customer_state
order by c.customer_state
```

Output:


| Query results   |                  |                     |                   |                   |  SAV |
|-----------------|------------------|---------------------|-------------------|-------------------|---|
| JOB INFORMATION |                  | RESULTS             | JSON              | EXECUTION DETAILS | CHART   |
| Row             | customer_state ▼ | total_order_price ▼ | avg_order_price ▼ |                   |   |
| 1               | AC               | 15982.94999999...   | 173.7277173913... |                   |   |
| 2               | AL               | 80314.81000000...   | 180.8892117117... |                   |   |
| 3               | AM               | 22356.84000000...   | 135.4959999999... |                   |   |
| 4               | AP               | 13474.29999999...   | 164.3207317073... |                   |   |
| 5               | BA               | 511349.9900000...   | 134.6012082126... |                   |   |
| 6               | CE               | 227254.7099999...   | 153.7582611637... |                   |   |
| 7               | DF               | 302603.9399999...   | 125.7705486284... |                   |   |
| 8               | ES               | 275037.3099999...   | 121.9137012411... |                   |   |
| 9               | GO               | 294591.9499999...   | 126.2717316759... |                   |   |
| 10              | MA               | 119648.2199999...   | 145.2041504854... |                   |   |

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

Query :

```
SELECT customer_state, sum(freight_value) total_order_freight_val, avg(freight_value)
avg_order_freight_val FROM `retailer-businesscase.retail.customers` c
join `retailer-businesscase.retail.orders` o on c.customer_id = o.customer_id
join `retailer-businesscase.retail.order_items` oi on oi.order_id = o.order_id
group by customer_state
order by customer_state
```

Output :

| Query results   |                  |                       |                     |                   |  |
|-----------------|------------------|-----------------------|---------------------|-------------------|---|
| JOB INFORMATION |                  | RESULTS               | JSON                | EXECUTION DETAILS | CH.   |
| Row             | customer_state ▼ | total_order_freight_v | avg_order_freight_v |                   |   |
| 1               | AC               | 3686.749999999...     | 40.07336956521...   |                   |   |
| 2               | AL               | 15914.589999999...    | 35.84367117117...   |                   |   |
| 3               | AM               | 5478.889999999...     | 33.20539393939...   |                   |   |
| 4               | AP               | 2788.500000000...     | 34.00609756097...   |                   |   |
| 5               | BA               | 100156.6799999...     | 26.36395893656...   |                   |   |
| 6               | CE               | 48351.589999999...    | 32.71420162381...   |                   |   |
| 7               | DF               | 50625.499999999...    | 21.04135494596...   |                   |   |
| 8               | ES               | 49764.599999999...    | 22.05877659574...   |                   |   |
| 9               | GO               | 53114.979999999...    | 22.76681525932...   |                   |   |
| 10              | MA               | 31523.770000000...    | 38.25700242718...   |                   |   |

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

- 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

Query :

```
SELECT order_id, DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,
DAY) time_to_deliver,
DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY)
diff_estimated_delivery
FROM `retailer-businesscase.retail.orders`
where order_status = 'delivered'
```

Output:



| Query results     |                               |                   |                           |                   |  |
|-------------------|-------------------------------|-------------------|---------------------------|-------------------|---|
| JOB INFORMATION   |                               | RESULTS           | JSON                      | EXECUTION DETAILS | CH  |
| Row               | order_id ▼                    | time_to_deliver ▼ | diff_estimated_delivery ▼ |                   |   |
| 1                 | 635c894d068ac37e6e03dc54e...  | 30                | 1                         |                   |   |
| 2                 | 3b97562c3aee8bdedcb5c2e45...  | 32                | 0                         |                   |   |
| 3                 | 68f47f50f04c4cb6774570cfde... | 29                | 1                         |                   |   |
| 4                 | 276e9ec344d3bf029ff83a161c... | 43                | -4                        |                   |   |
| 5                 | 54e1a3c2b97fb0809da548a59...  | 40                | -4                        |                   |   |
| 6                 | fd04fa4105ee8045f6a0139ca5... | 37                | -1                        |                   |   |
| 7                 | 302bb8109d097a9fc6e9cefc5...  | 33                | -5                        |                   |   |
| 8                 | 66057d37308e787052a32828...   | 38                | -6                        |                   |   |
| 9                 | 19135c945c554eebfd7576c73...  | 36                | -2                        |                   |   |
| 10                | 4493e45e7ca1084efcd38ddeb...  | 34                | 0                         |                   |   |
| Results per page: |                               |                   |                           |                   |   |

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

Query:

**Top 5 states with highest average freight value**

```
SELECT customer_state, avg(freight_value) avg_freight_val
FROM `retailer-businesscase.retail.order_items` oi
join `retailer-businesscase.retail.orders` o on oi.order_id=o.order_id
join `retailer-businesscase.retail.customers` c on c.customer_id = o.customer_id
group by customer_state
order by avg_freight_val desc
limit 5
```

Output:

| Query results   |                  |                   |                |
|-----------------|------------------|-------------------|----------------|
| JOB INFORMATION |                  | RESULTS           | JSON EXECUTION |
| Row             | customer_state ▼ | avg_freight_val ▼ |                |
| 1               | RR               | 42.98442307692... |                |
| 2               | PB               | 42.72380398671... |                |
| 3               | RO               | 41.06971223021... |                |
| 4               | AC               | 40.07336956521... |                |
| 5               | PI               | 39.14797047970... |                |

### Top 5 states with lowest average freight value

```

SELECT customer_state, avg(freight_value) avg_freight_val
FROM `retailer-businesscase.retail.order_items` oi
join `retailer-businesscase.retail.orders` o on oi.order_id=o.order_id
join `retailer-businesscase.retail.customers` c on c.customer_id = o.customer_id
group by customer_state
order by avg_freight_val
limit 5

```

Output:

| Query results   |                  |                   |                  |
|-----------------|------------------|-------------------|------------------|
| JOB INFORMATION |                  | RESULTS           | JSON EXECUTION D |
| Row             | customer_state ▼ | avg_freight_val ▼ |                  |
| 1               | SP               | 15.14727539041... |                  |
| 2               | PR               | 20.53165156794... |                  |
| 3               | MG               | 20.63016680630... |                  |
| 4               | RJ               | 20.96092393168... |                  |
| 5               | DF               | 21.04135494596... |                  |

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

### Top 5 states with highest average delivery time

```

SELECT customer_state, avg(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp, DAY)) avg_time_to_deliver
FROM `retailer-businesscase.retail.orders` o
JOIN `retailer-businesscase.retail.customers` c
ON o.customer_id = c.customer_id

```

```

where order_status = 'delivered'
group by customer_state
order by avg_time_to_deliver desc
limit 5

```

Output:

| Query results   |                  |                     |      |
|-----------------|------------------|---------------------|------|
| JOB INFORMATION |                  | RESULTS             | JSON |
| Row             | customer_state ▼ | avg_time_to_deliver |      |
| 1               | RR               | 28.97560975609...   |      |
| 2               | AP               | 26.73134328358...   |      |
| 3               | AM               | 25.98620689655...   |      |
| 4               | AL               | 24.04030226700...   |      |
| 5               | PA               | 23.31606765327...   |      |

**Top 5 states with lowest average delivery time**

Query:

```

SELECT customer_state, avg(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp, DAY)) avg_time_to_deliver
FROM `retailer-businesscase.retail.orders` o
JOIN `retailer-businesscase.retail.customers` c
ON o.customer_id = c.customer_id
where order_status = 'delivered'
group by customer_state
order by avg_time_to_deliver
limit 5

```

Output:

| Query results   |                  |                     |           |
|-----------------|------------------|---------------------|-----------|
| JOB INFORMATION |                  | RESULTS             | EXECUTION |
| Row             | customer_state ▼ | avg_time_to_deliver |           |
| 1               | SP               | 8.298093544722...   |           |
| 2               | PR               | 11.52671135486...   |           |
| 3               | MG               | 11.54218777523...   |           |
| 4               | DF               | 12.50913461538...   |           |
| 5               | SC               | 14.47518330513...   |           |

- 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.

Query:

```
SELECT customer_state,
avg(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY))
avg_delivery_between_estimated_actual
FROM `retailer-businesscase.retail.orders` o
JOIN `retailer-businesscase.retail.customers` c
ON o.customer_id = c.customer_id
where order_status = 'delivered'
group by customer_state
order by avg_delivery_between_estimated_actual
limit 5
```

Output:

| Query results   |                |                      |      |
|-----------------|----------------|----------------------|------|
| JOB INFORMATION |                | RESULTS              | JSON |
| Row             | customer_state | avg_delivery_between |      |
| 1               | AL             | 7.947103274559...    |      |
| 2               | MA             | 8.768479776847...    |      |
| 3               | SE             | 9.173134328358...    |      |
| 4               | ES             | 9.618546365914...    |      |
| 5               | BA             | 9.934889434889...    |      |

#### 6. Analysis based on the payments:

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

Query:

```
SELECT EXTRACT(YEAR from order_purchase_timestamp) year,
EXTRACT(MONTH from order_purchase_timestamp) month, payment_type,
count(o.order_id) no_of_orders
FROM `retailer-businesscase.retail.orders` o
join `retailer-businesscase.retail.payments` p
on p.order_id = o.order_id
group by payment_type, year, month
order by year, month
```

Output:

## Query results

[SAVE RESULT](#)

| JOB INFORMATION |        | RESULTS | JSON           | EXECUTION DETAILS | CHART | PREVIEW |
|-----------------|--------|---------|----------------|-------------------|-------|---------|
| Row             | year ▼ | month ▼ | payment_type ▼ | no_of_orders ▼    |       |         |
| 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                |       |         |
| 8               | 2017   | 1       | UPI            | 197               |       |         |
| 9               | 2017   | 1       | credit_card    | 583               |       |         |
| 10              | 2017   | 1       | debit_card     | 9                 |       |         |

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

Query:

```
SELECT payment_installments, count(order_id) no_of_orders FROM `retailer-businesscase.retail.payments`
group by payment_installments
```

Output:

## Query results

| JOB INFORMATION |                     | RESULTS        | JSON |
|-----------------|---------------------|----------------|------|
| 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            |      |

#### Actionable Insights and Recommendations

1. More number of orders were placed in May, July and August.
2. State with code SP has placed more number of orders.
3. More number of orders has been placed using credit cards
4. In some of states, the freight value has costed upto 1lakh. We can try to reduce this cost
5. Some of the orders are getting delivered late than the estimated delivery
6. Cost of orders has been increased over the years