What does 'good' look like?
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.

Ans. `SELECT COLUMN_NAME, DATA_TYPE`
`FROM `scaler-1-444404.buscase1.INFORMATION_SCHEMA.COLUMNS``
`WHERE TABLE_NAME = 'customers';`

## Query results

| | SAVE RESULTS ▼ | OPEN IN ▼ | ↕ |

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH

| 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 | |

Q1.2 Get the time range between which the orders were placed.

solution: `select max(order_purchase_timestamp) as latest,`
`min(order_purchase_timestamp) as oldest`
`from buscase1.orders;`

```
45  select max(order_purchase_timestamp) as latest,
46  min(order_purchase_timestamp) as oldest
47  from buscase1.orders
```
Press Alt+F1 for Accessibility Options.

## Query results

| | SAVE RESULTS ▼ | OPEN IN ▼ | ↕ |

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH

| Row | latest ▼ | oldest ▼ | |
|---|---|---|---|
| 1 | 2018-10-17 17:30:18 UTC | 2016-09-04 21:15:19 UTC | |

Q1.3. Count the Cities & States of customers who ordered during the given period.

Sol: select count(distinct customer_city) as total_city,
        count(distinct customer_state) as total_states
        from buscase1.customers

```
51   select count(distinct customer_city) as total_city,
52          count(distinct customer_state) as total_states
53   from buscase1.customers
54
```
Press Alt+F1 for Accessibility Options.

**Query results**                              SAVE RESULTS ▼     OPEN IN ▼     ↕

JOB INFORMATION     RESULTS     CHART     JSON     EXECUTION DETAILS     EXECUTION GRAPH

| Row | total_city ▼ | total_states ▼ | |
|-----|--------------|----------------|---|
| 1   | 4119         | 27             | |

## 2. In-depth Exploration:

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

Sol: select YEAR,
        no_of_orders,
         no_of_orders-b as trend from
 (select extract(year from order_purchase_timestamp) as YEAR,
     count(order_id) as no_of_orders,
     lag( count( order_id),1)over(order by  count(distinct order_id) asc ) as b
     from buscase1.orders
     group by 1
      order by 1 asc)a

3

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH

| Row | YEAR ▼ | no_of_orders ▼ | trend ▼ |
|---|---|---|---|
| 1 | 2016 | 329 | *null* |
| 2 | 2017 | 45101 | 44772 |
| 3 | 2018 | 54011 | 8910 |

Results per page: 50 ▼    1 – 3 of 3    |<  <  >  >|

SO YES there is a growing trend in the no. of orders placed over the past years.

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

Code:
```
select count(order_id) as NO_OF_ORDERS ,
    format_date('%B', order_purchase_timestamp) as month
    from `buscase1.orders`
    group by 2
    order by 1 desc
```

Yes in august there are maximum sales and in december there is minimum sales .

**Query results**

SAVE RESULTS ▾       OPEN IN ▾       ↕

| JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH |

| Row | NO_OF_ORDERS ▾ | month ▾ | |
|---|---|---|---|
| 1 | 10843 | August | |
| 2 | 10573 | May | |
| 3 | 10318 | July | |
| 4 | 9893 | March | |
| 5 | 9412 | June | |
| 6 | 9343 | April | |
| 7 | 8508 | February | |
| 8 | 8069 | January | |
| 9 | 7544 | November | |
| 10 | 5674 | December | |

Results per page:   50 ▾    1 – 12 of 12    |<   <   >   >|

**Q2.3** 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

```
CODE:  select count(order_id),
 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 TIMEPERIOD
from `buscase1.orders`
group by 2
order by 1 desc
limit 1
```

**Query results**

SAVE RESULTS ▾   OPEN IN ▾   ↕

| JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH |

| Row | f0_ ▾ | TIMEPERIOD ▾ | |
| --- | --- | --- | --- |
| 1 | 38135 | Afternoon | |

Results per page: 50 ▾   1 – 1 of 1   |< < > >|

**Job history**   ↻ REFRESH   ∧

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

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

```
CODE:  select count(order_id) as NO_OF_ORDERS,
         customer_state,
         format_date('%B', order_purchase_timestamp) as MONTH
from buscase1.orders o
left join buscase1.customers c
on o.customer_id=c.customer_id
group by 2,3
order by MONTH asc
```

**Query results**

SAVE RESULTS ▾   OPEN IN ▾   ↕

| JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH |

| Row | NO_OF_ORDERS ▾ | customer_state ▾ | MONTH ▾ | |
| --- | --- | --- | --- | --- |
| 1 | 92 | MT | April | |
| 2 | 3967 | SP | April | |
| 3 | 1172 | RJ | April | |
| 4 | 488 | RS | April | |
| 5 | 318 | BA | April | |
| 6 | 143 | CE | April | |
| 7 | 1061 | MG | April | |
| 8 | 154 | PE | April | |
| 9 | 107 | PA | April | |
| 10 | 50 | PI | April | |

Results per page: 50 ▾   1 – 50 of 322   |< < > >|

— **BRAZIL the country with 4119 cities and 27 states is a country with a great potential of business , considering the business trends between '2016-09-04 21:15:19 UTC'**

and '2018-10-17 17:30:18 UTC' there is a growing trend in the no. of orders placed over the past years with values as follows :

| year | No. of orders |
|------|---------------|
| 2016 | 329 |
| 2017 | 45101 |
| 2018 | 54011 |

Which clearly shows the growing market in the country .

*.  It has been observed that the market activity increases  during the month of august
The boom in business in Brazil during August can be attributed to several key factors, including seasonal trends, economic cycles, and cultural events.example-

A .End of Winter – Seasonal Shopping Changes
B. Strong Agricultural Exports
C.Increased Tourism Activity

*. The month of December shows the least number of orders.

*. People generally place their orders in the afternoon which may be due to the lunch break hours at that time .

Q.3 .2 . How are the customers distributed across all the states?

```
select count(distinct customer_id) as NO_OF_CUSTOMERS,
        customer_state
from buscase1.customers
group by 2
order by 1 desc
```

Query results

| Row | NO_OF_CUSTOMERS | customer_state ▾ |
|-----|-----------------|------------------|
| 1 | 41746 | SP |
| 2 | 12852 | RJ |
| 3 | 11635 | MG |
| 4 | 5466 | RS |
| 5 | 5045 | PR |
| 6 | 3637 | SC |
| 7 | 3380 | BA |
| 8 | 2140 | DF |
| 9 | 2033 | ES |
| 10 | 2020 | GO |

Results per page: 50    1 – 27 of 27

Job history

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

Q4.1 Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

Code:
```
select
    sum(case when extract(Year from order_purchase_timestamp) =2017 then p.payment_value end) as
        totalcost2017,
    sum(case when extract(Year from order_purchase_timestamp) =2018 then p.payment_value end) as totalcost2018,
    round( 100 * (sum(case when extract(Year from order_purchase_timestamp) =2018 then p.payment_value end) - sum(case when extract(Year from order_purchase_timestamp) =2017 then p.payment_value end)  )/sum(case when extract(Year from order_purchase_timestamp) =2017 then p.payment_value end),2)
as increment

    from buscase1.orders o
    left join buscase1.payments p
    on p.order_id=o.order_id
    where extract(Year from order_purchase_timestamp) between 2017 and 2018
and
```

```
        extract(Month from order_purchase_timestamp) between 1 and 8
```

### Query results

JOB INFORMATION · **RESULTS** · CHART · JSON · EXECUTION DETAILS · EXECUTION GRAPH

| Row | totalcost2017 ▾ | totalcost2018 ▾ | increment ▾ | |
|-----|-----------------|-----------------|-------------|--|
| 1 | 3669022.119999… | 8694733.839999… | 136.98 | |

Results per page: 50 ▾    1 – 1 of 1    |< < > >|

**\*. It has been observed that the% increase in the cost of orders from 2017 to 2018 (including months between Jan to Aug) is 136.98% which shows the growing demands of the products or the monopoly of certain companies on certain products .**

Q4.2.Calculate the Total & Average value of order price for each state.

Code:
```
select customer_state ,
    sum(price) as TOTAL_PRICE,
    sum(price)/count(distinct o.order_id) as AVG
from `buscase1.orders` o
left join `buscase1.customers` c
on o.customer_id=c.customer_id
left join `buscase1.order_items` i
on o.order_id=i.order_id
group by 1
order by 1 asc
```

**Query results**

SAVE RESULTS ▼          OPEN IN ▼          ↕

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

| Row | customer_state ▼ | TOTAL_PRICE ▼ | AVG ▼ | |
|---|---|---|---|---|
| 1 | AC | 15982.94999999... | 197.3203703703... | |
| 2 | AL | 80314.81 | 194.4668523002... | |
| 3 | AM | 22356.84000000... | 151.0597297297... | |
| 4 | AP | 13474.29999999... | 198.1514705882... | |
| 5 | BA | 511349.9900000... | 151.2869792899... | |
| 6 | CE | 227254.7099999... | 170.1008308383... | |
| 7 | DF | 302603.9399999... | 141.4037102803... | |
| 8 | ES | 275037.3099999... | 135.2864289227... | |
| 9 | GO | 294591.9499999... | 145.8375990098... | |
| 10 | MA | 119648.2199999... | 160.1716465863... | |

Results per page:  50 ▼     1 – 27 of 27     |< < > >|

**\*. Maximum number of customers are from state SP-41746 and RJ-12852 and minimum from RR-46 AND AP-66 which shows that there is some economic backwardness among the people due to instability and lack of development . Investing in such states needs proper safety arrangements in collaboration with the Government.  SP with 41746 customers  being the maximum customer state is not the highest state with highest priced orders rather AP is the state with highest total value and average value of orders which indicates the economic wellness , education standards and high per capita income of people. Investing in such a state will bring you exceptionally  lucrative results.**

Q4.3.Calculate the Total & Average value of order freight for each state.

```
CODE: select c.customer_state,
      sum(freight_value) as TOTAL_PRICE,
      sum(freight_value)/count(distinct o.order_id) as AVG
from `buscase1.orders` o
left join `buscase1.customers` c
on o.customer_id=c.customer_id
left join `buscase1.order_items` i
on o.order_id=i.order_id
group by 1
```

Query results

SAVE RESULTS ▾     OPEN IN ▾     ↕

JOB INFORMATION     RESULTS     CHART     JSON     EXECUTION DETAILS     EXECUTION GRAPH

| Row | customer_state ▾ | TOTAL_PRICE ▾ | AVG ▾ | |
|---|---|---|---|---|
| 1 | RJ | 305589.3100000... | 23.77756847183... | |
| 2 | RS | 135522.7400000... | 24.79376875228... | |
| 3 | SP | 718723.0699999... | 17.21657332438... | |
| 4 | DF | 50625.49999999... | 23.65677570093... | |
| 5 | PR | 117851.6800000... | 23.36009514370... | |
| 6 | MT | 29715.43000000... | 32.76232635060... | |
| 7 | MA | 31523.77000000... | 42.20049531459... | |
| 8 | AL | 15914.58999999... | 38.53411622276... | |
| 9 | MG | 270853.4600000... | 23.27919724967... | |
| 10 | PE | 59449.65999999... | 35.98647699757... | |

Results per page:   50 ▾    1 – 27 of 27    |<   <   >   >|

Job history                                                     ⟳ REFRESH   ∧

Q.5.Analysis based on sales, freight and delivery time.
Q.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_delivered_customer_date -
order_estimated_delivery_date

CODE: select order_id ,
      date_diff(order_delivered_customer_date , order_purchase_timestamp,day)
as TIME_TO_DELIVER,
      date_diff(order_delivered_customer_date, order_estimated_delivery_date
,day) as diff_estimated_delivery
from buscase1.orders
order by 1

**Query results**

| Row | order_id ▼ | TIME_TO_DELIVER | diff_estimated_delive |
|-----|-----------|-----------------|----------------------|
| 1 | 00010242fe8c5a6d1ba2dd792… | 7 | -8 |
| 2 | 00018f77f2f0320c557190d7a1… | 16 | -2 |
| 3 | 000229ec398224ef6ca0657da… | 7 | -13 |
| 4 | 00024acbcdf0a6daa1e931b03… | 6 | -5 |
| 5 | 00042b26cf59d7ce69dfabb4e… | 25 | -15 |
| 6 | 00048cc3ae777c65dbb7d2a06… | 6 | -14 |
| 7 | 00054e8431b9d7675808bcb8… | 8 | -16 |
| 8 | 000576fe39319847cbb9d288c… | 5 | -15 |
| 9 | 0005a1a1728c9d785b8e2b08b… | 9 | 0 |
| 10 | 0005f50442cb953dcd1d21e1f… | 2 | -18 |

 **Negative values in the third column indicate that the order has been delivered before the expected delivery date and positive values denote the number of days the order is delivered after the expected date.**

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

```sql
select customer_state,
case when RNK <=5 then  'highest'
     when RNK between 23 and 27 then 'LOWEST' end as CATEGORY,
     RNK
  from
(select c.customer_state,
     sum(freight_value)/count(distinct o.order_id) as AVERAGE_FV,
     ROW_NUMBER()over(order by  sum(freight_value)/count(distinct o.order_id )desc)
as RNK
from `buscase1.customers` c
left join buscase1.orders o
on o.customer_id =c.customer_id
left join `buscase1.order_items` i
on o.order_id=i.order_id
group by 1) N
WHERE RNK <= 5 OR RNK BETWEEN 23 AND 27
order by RNK ASC
```

**Query results**

SAVE RESULTS ▼    OPEN IN ▼    ↕

JOB INFORMATION    **RESULTS**    CHART    JSON    EXECUTION DETAILS    EXECUTION GRAPH

| Row | customer_state ▼ | CATEGORY ▼ | RNK ▼ | |
|---|---|---|---|---|
| 1 | RR | highest | 1 | |
| 2 | PB | highest | 2 | |
| 3 | AC | highest | 3 | |
| 4 | RO | highest | 4 | |
| 5 | PI | highest | 5 | |
| 6 | RJ | LOWEST | 23 | |
| 7 | DF | LOWEST | 24 | |
| 8 | PR | LOWEST | 25 | |
| 9 | MG | LOWEST | 26 | |
| 10 | SP | LOWEST | 27 | |

Results per page:    50 ▼    1 – 10 of 10    I<    <    >    >I

**\*. Comparing total freight value and average freight value we conclude that in state of RJ with highest number of total freight value and one of the lowest average values indicates:**

1. **High Volume of Shipments**
2. **Efficient Logistics & Infrastructure**
3. **Shorter Distance Shipments**
4. **Economies of Scale in Logistics**
5. **High Industrial & Trade Activity**

**(RJ)Rio de Janeiro has industries like shipbuilding, steel production, pharmaceuticals, and automobiles. The Port of Rio de Janeiro is a major trade hub.**

**However, challenges include crime and security issues, which sometimes affect business and tourism. Economic inequality is high, with a significant gap between the rich and poor, especially in favelas (slums). Additionally, corruption and political instability, such as the Petrobras scandal, have impacted the economy.**

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

```sql
    WITH RankedStates AS (
  SELECT
      customer_state,
      SUM(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)) /
COUNT(DISTINCT o.order_id) AS TIME_TO_DELIVER,
      ROW_NUMBER() OVER (ORDER BY SUM(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp, DAY)) / COUNT(DISTINCT o.order_id) DESC) AS RNK
  FROM `buscase1.customers` c
  INNER JOIN `buscase1.orders` o ON o.customer_id = c.customer_id
  GROUP BY c.customer_state
)
SELECT
  customer_state,
  RNK,
  CASE
      WHEN RNK <= 5 THEN 'HIGHEST'
      WHEN RNK > (SELECT MAX(RNK) - 5 FROM RankedStates) THEN 'LOWEST'
  END AS CATEGORY
FROM RankedStates
WHERE RNK <= 5 OR RNK > (SELECT MAX(RNK) - 5 FROM RankedStates)
ORDER BY RNK;
```

Query results

SAVE RESULTS ▾    OPEN IN ▾

JOB INFORMATION    RESULTS    CHART    JSON    EXECUTION DETAILS    EXECUTION GRAPH

| Row | customer_state ▾ | RNK ▾ | CATEGORY ▾ |
|-----|------------------|-------|------------|
| 1 | AP | 1 | HIGHEST |
| 2 | RR | 2 | HIGHEST |
| 3 | AM | 3 | HIGHEST |
| 4 | AL | 4 | HIGHEST |
| 5 | PA | 5 | HIGHEST |
| 6 | SC | 23 | LOWEST |
| 7 | DF | 24 | LOWEST |
| 8 | MG | 25 | LOWEST |
| 9 | PR | 26 | LOWEST |
| 10 | SP | 27 | LOWEST |

Results per page:    50 ▾    1 – 10 of 10    |<    <    >    >|

**AP is the state in which orders take maximum time to get delivered and SP is the state where there is a speedy delivery as compared to other states.Amapá (AP) has the longest delivery time due to its remote location, poor road connectivity, and reliance on**

**slow water transport, while São Paulo (SP) enjoys the fastest delivery thanks to its strong logistics network, advanced infrastructure, and high business activity.**

Q.5.4Find 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.

CODE:
```sql
SELECT
    customer_state,
    ROUND(ABS(SUM(DATE_DIFF(order_delivered_customer_date,
order_estimated_delivery_date, DAY))) / COUNT(DISTINCT o.order_id), 2) AS
AVERAGE_DAYS,
    ROW_NUMBER() OVER (ORDER BY SUM(DATE_DIFF(order_delivered_customer_date,
order_estimated_delivery_date, DAY)) / COUNT(DISTINCT o.order_id) ASC) AS RNK
FROM
    buscase1.customers c
LEFT JOIN
    buscase1.orders o ON c.customer_id = o.customer_id
WHERE
    DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date, DAY) < 0
GROUP BY
    customer_state
ORDER BY
    RNK
LIMIT 5;
```

Press Alt+F1 for Accessibility Options.

**Query results**

🖺 SAVE RESULTS ▾    📊 OPEN IN ▾    ↕

| JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH |

| Row | customer_state ▾ | AVERAGE_DAYS ▾ | RNK ▾ | |
|-----|------------------|----------------|-------|---|
| 1 | RR | 23.75 | 1 | |
| 2 | AP | 21.88 | 2 | |
| 3 | AC | 21.54 | 3 | |
| 4 | AM | 20.28 | 4 | |
| 5 | RO | 20.03 | 5 | |

Results per page:  50 ▾    1 – 5 of 5    |< < > >|

Job history    ⟳ REFRESH    ∧

**AP has one of the highest order values, while Roraima RR has one of the lowest. Despite this contrast, both states have the fastest delivery times, indicating an efficient logistics network. This suggests that AP benefits from strong infrastructure supporting high-value orders, while RR's low volume helps maintain quick deliveries. Investing in logistics expansion in AP and stimulating demand in RR could optimize growth opportunities.**

Q.6.Analysis based on the payments:

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

```sql
SELECT
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year_col,
    EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month_col,
    p.payment_type,
    COUNT(DISTINCT o.order_id) AS total_orders
FROM `buscase1.orders` o
LEFT JOIN `buscase1.payments` p
ON o.order_id = p.order_id
```

```
WHERE p.payment_type IS NOT NULL
GROUP BY year_col, month_col, p.payment_type
ORDER BY year_col, month_col;
```

### Query results

SAVE RESULTS ▾     OPEN IN ▾

JOB INFORMATION    **RESULTS**    CHART    JSON    EXECUTION DETAILS    EXECUTION GRAPH

| Row | year_col ▾ | month_col ▾ | payment_type ▾ | total_orders ▾ |
|---|---|---|---|---|
| 1 | 2016 | 9 | credit_card | 3 |
| 2 | 2016 | 10 | credit_card | 253 |
| 3 | 2016 | 10 | UPI | 63 |
| 4 | 2016 | 10 | voucher | 11 |
| 5 | 2016 | 10 | debit_card | 2 |
| 6 | 2016 | 12 | credit_card | 1 |
| 7 | 2017 | 1 | credit_card | 582 |
| 8 | 2017 | 1 | UPI | 197 |
| 9 | 2017 | 1 | voucher | 33 |
| 10 | 2017 | 1 | debit_card | 9 |

Results per page:   50 ▾    1 – 50 of 90    |<   <   >   >|

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

```
SELECT
  payment_installments AS installments,
COUNT(order_id) AS num_orders
FROM `buscase1.payments`
WHERE payment_installments >= 1
GROUP BY payment_installments
ORDER BY num_orders DESC
```

Query results

SAVE RESULTS ▾    OPEN IN ▾    ↕

| JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH |

| Row | installments ▾ | num_orders ▾ | |
| --- | --- | --- | --- |
| 1 | 1 | 52546 | |
| 2 | 2 | 12413 | |
| 3 | 3 | 10461 | |
| 4 | 4 | 7098 | |
| 5 | 10 | 5328 | |
| 6 | 5 | 5239 | |
| 7 | 8 | 4268 | |
| 8 | 6 | 3920 | |
| 9 | 7 | 1626 | |
| 10 | 9 | 644 | |

Results per page: 50 ▾    1 – 23 of 23    |<    <    >    >|

Job history    ↻ REFRESH    ⌃