Brazilian E-Commerce Database

Database Project on **Microsoft SQL Server**

Information Technology Institute - ITI

Mohamed Essam
Gemian Talasoun
Ahmed Essam
Mahmoud Abdelaziz

About dataset

This is a Brazilian e-commerce public dataset of orders made at the Olist Store. The dataset has information on 100k orders from 2016 to 2018 made at multiple marketplaces in Brazil. Its features allow viewing orders from multiple dimensions: from order status, price, payment, and freight performance to customer location, product attributes, and finally reviews written by customers. We also released a geolocation dataset that relates Brazilian zip codes to lat/lng coordinates.

This is real commercial data, it has been anonymized, and references to the companies and partners in the review text have been replaced with the names of Game of Thrones great houses.

Context

This database is about 100,000 Orders with the product, customer, and reviews info.

Using Microsoft SQL Server Management Studio and running queries on it, to get data insights that the decision makers of the companies can use to make their decision based on information extracted from the database.

Attention

1. An order might have multiple items.
2. Each product might be fulfilled by a distinct seller.
3. Customers can review orders.
4. A Category might have multiple products.
5. Each item might have a quantity from one product.
6. Each order must have a payment method.
7. An order must have a location to deliver to it.

Database

Tables and summary

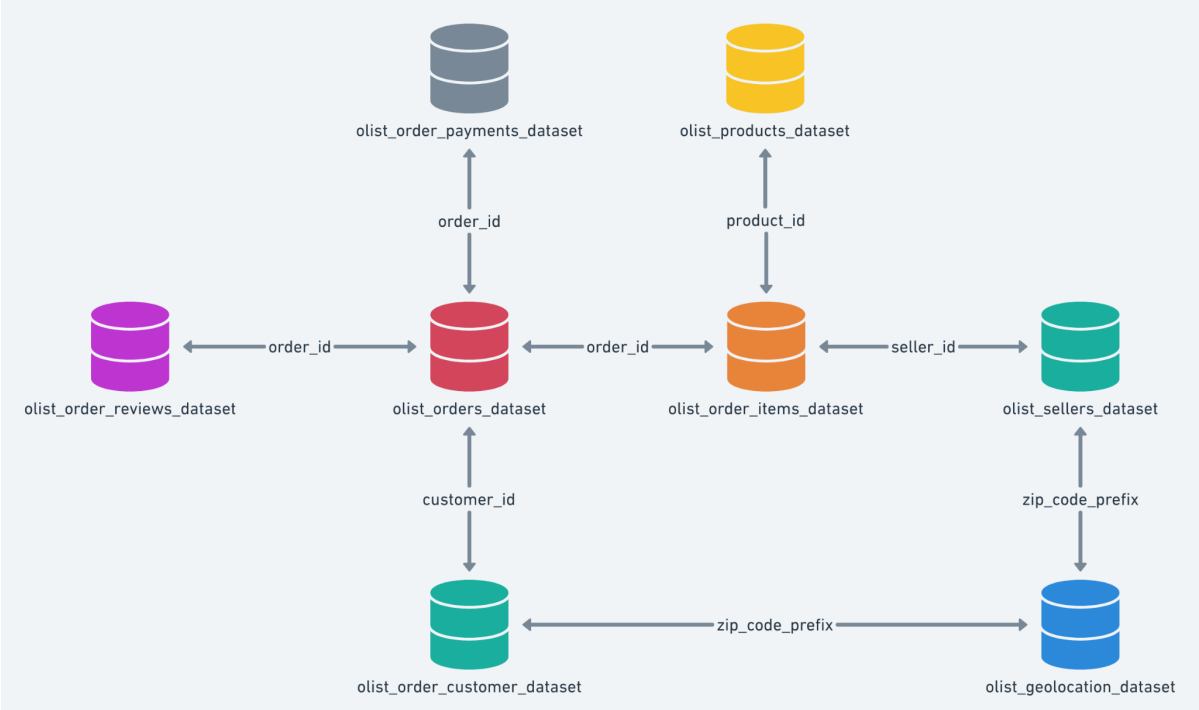
- olist_customers_dataset.csv
- olist_geolocation_dataset.csv
- olist_order_items_dataset.csv
- olist_order_payments_datas...
- olist_order_reviews_dataset...
- olist_orders_dataset.csv
- olist_products_dataset.csv
- olist_sellers_dataset.csv
- product_category_name_tra...

Summary

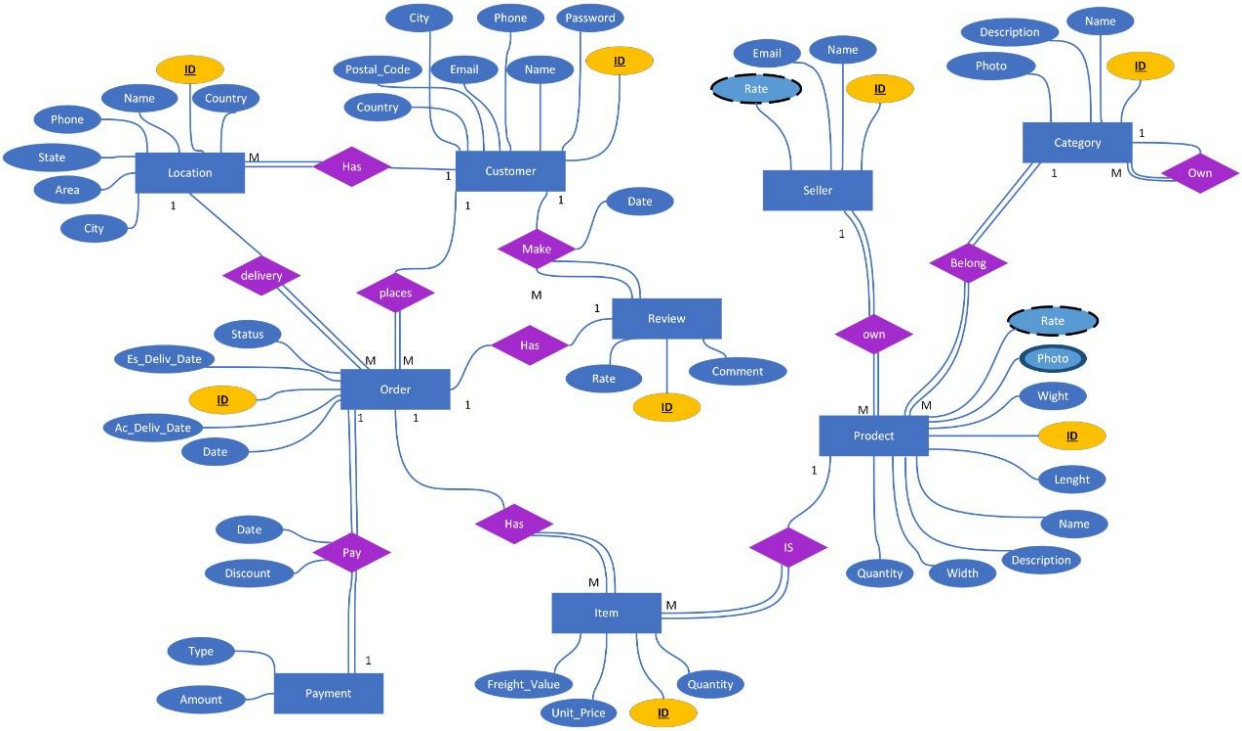
- 9 files
  - .csv 9
- 52 columns
  - String 13
  - Integer 13
  - Uuid 12
  - Other 14

Schema

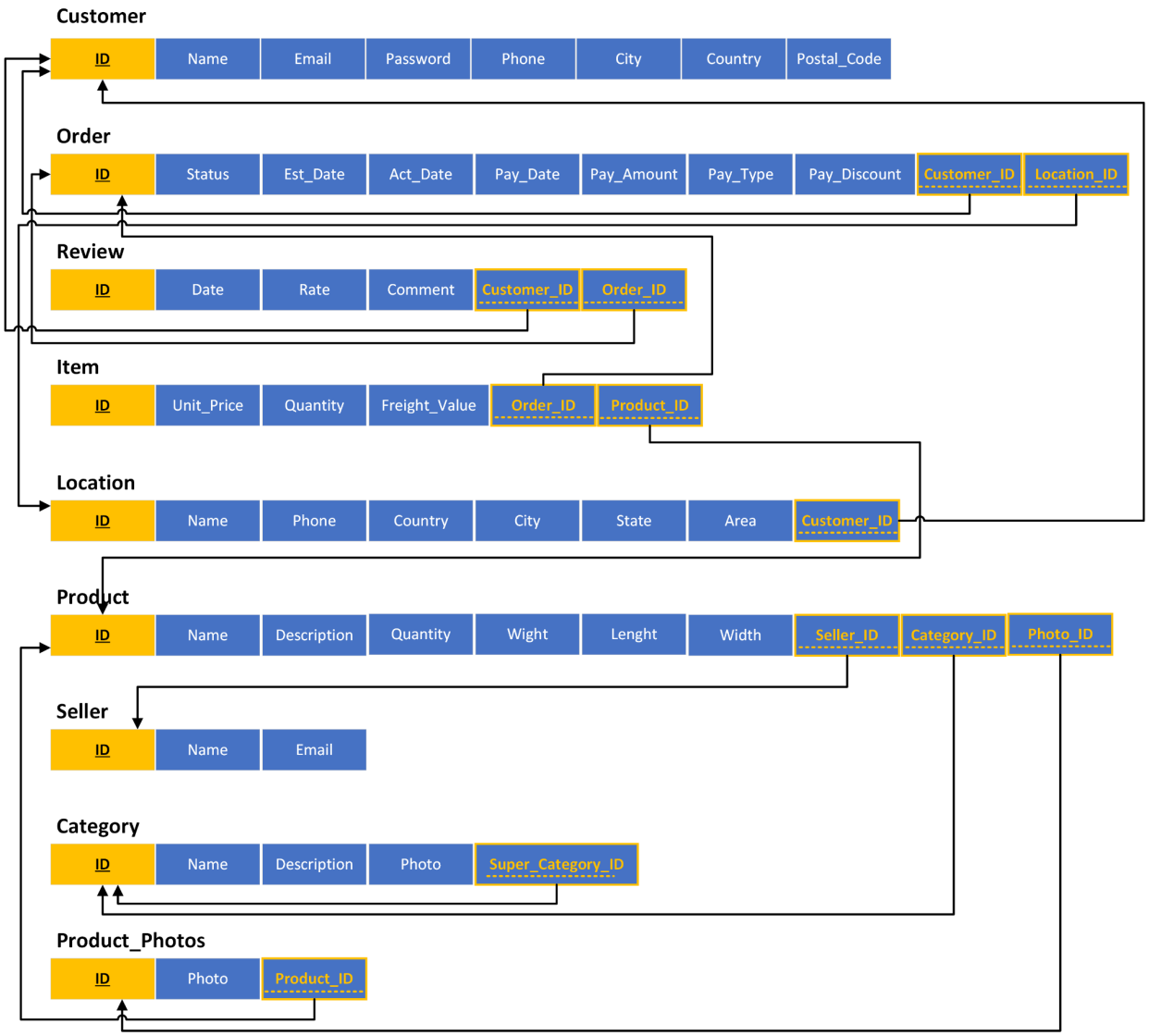
The data is divided into multiple datasets for better understanding and organization. Please refer to the following data schema when working with it:



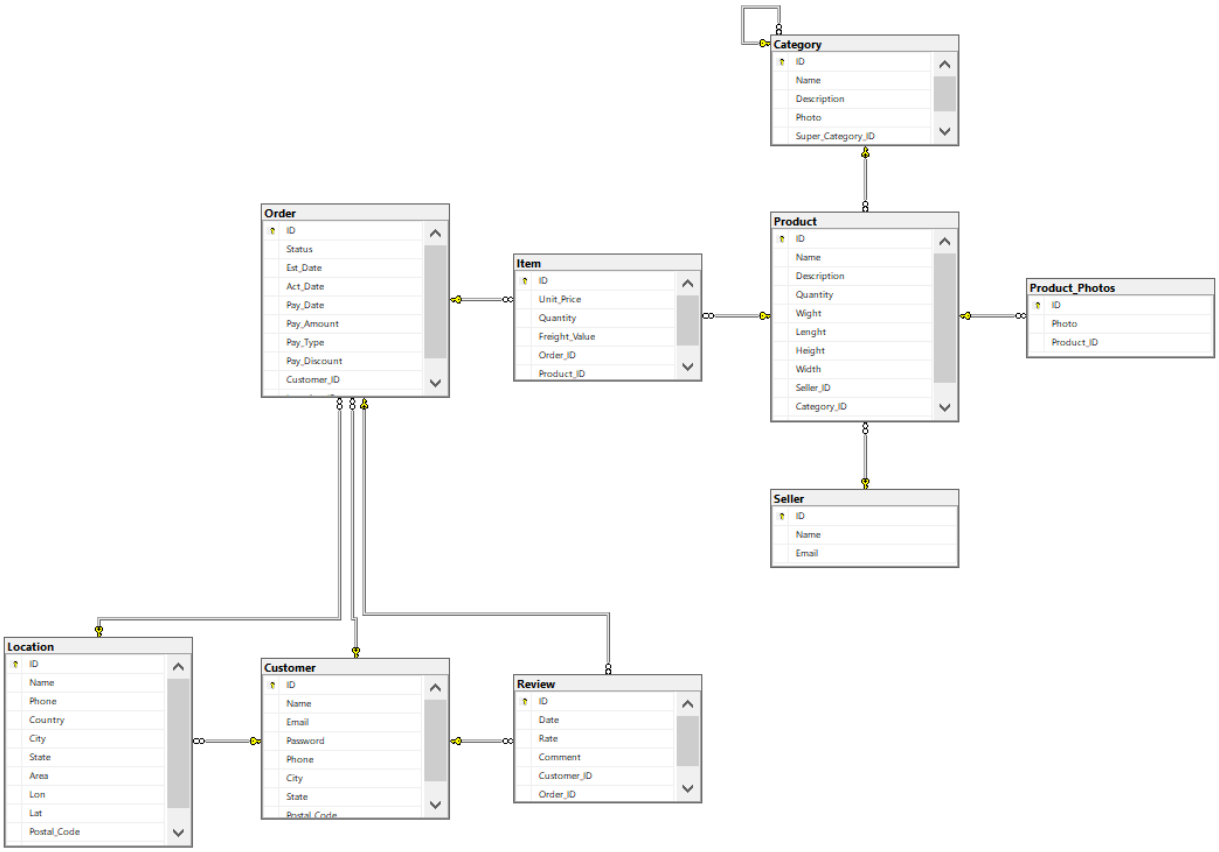
ERD


City
Phone
Password
ID
Postal_Code
Email
Name
ID
Country
ID
Name
Country
Phone
State
Location
M
Has
Customer
Area
City
1
1
1
1
Date
Make
delivery
places
M
1
Review
Status
Es_Deliv_Date
Has
Rate
Comment
ID
M
M
Order
ID
1
1
1
Ac_Deliv_Date
Date
Date
Pay
Discount
Has
M
Item
M
Type
1
Payment
Amount
Freight_Value
Quantity
Unit_Price
ID
Email
Name
Rate
ID
Seller
1
own
M
M
Prodect
1
IS
Quantity
Width
Description
Name
Lenght
ID
Wight
Photo
Rate
Belong
Description
Name
Photo
ID
Category
1
1
M
Own

Mapping


Customer
ID
Name
Email
Password
Phone
City
Country
Postal_Code
Order
ID
Status
Est_Date
Act_Date
Pay_Date
Pay_Amount
Pay_Type
Pay_Discount
Customer_ID
Location_ID
Review
ID
Date
Rate
Comment
Customer_ID
Order_ID
Item
ID
Unit_Price
Quantity
Freight_Value
Order_ID
Product_ID
Location
ID
Name
Phone
Country
City
State
Area
Customer_ID
Product
ID
Name
Description
Quantity
Wight
Lenght
Width
Seller_ID
Category_ID
Photo_ID
Seller
ID
Name
Email
Category
ID
Name
Description
Photo
Super_Category_ID
Product_Photos
ID
Photo
Product_ID

Database Diagram



Insights

- **Data insight**: The top customer cities
- **Business Decision:** Increase advertising in those cities

```
SELECT COUNT(l.Customer_ID) as CountCity , l.City
FROM Location l
GROUP BY l.City
ORDER BY CountCity DESC
```

Results | Messages

CountCity	City
132066	sao paulo
62007	rio de janeiro
27725	belo horizonte

- **Data insight**: People who have purchased over 5000 in a given year (2018)
- **Business Decision:** Predicting the future purchased customers by previous years

```
Go;
CREATE FUNCTION Purchased_Over (@money INT, @year NVARCHAR(4))
RETURNS TABLE AS
RETURN
    SELECT O.Customer_ID, SUM(O.Pay_Amount) AS [Purchased]
    FROM [dbo].[Order] AS O
    WHERE O.Customer_ID IS NOT NULL AND YEAR(O.Act_Date) = @year
    GROUP BY O.Customer_ID
    HAVING SUM(O.Pay_Amount) > @money

Go;
SELECT * FROM Purchased_Over(5000, '2018')
```

Results | Messages

Customer_ID	Purchased
f48d464a0baaea338cb25f816991ab1f	6922.2099609375
ec5b2ba62e574342386871631fafd3fc	7274.8798828125

- **Data insight**: The types and number of payment methods used
- **Business Decision:** Facilitating the payment process for the most used methods

```
47      GO;
48      CREATE PROC Payment_Type
49      AS (
50          SELECT O.Pay_Type, Count(Pay_Type) AS [Number_Used]
51          FROM [dbo].[Order] AS O
52          GROUP BY O.Pay_Type
53      )
54
55      EXEC Payment_Type
56
```

Results | Messages

Pay_Type	Number_Used
credit_card	75291
not_defined	3
debit_card	1528
boleto	19784
voucher	2834

- **Data insight**: The number of orders in each status
- **Business Decision:** Analyzing and improving the conversion between the phases

```
61      GO;
62      CREATE VIEW Orders_in_Status
63      AS (
64          SELECT O.[Status], COUNT(O.[Status]) AS Number_of_Orders
65          FROM [dbo].[Order] AS O
66          GROUP BY O.[Status]
67      )
68
69      Go;
70      SELECT * FROM Orders_in_Status
71
```

Results | Messages

Status	Number_of_Orders
approved	2
delivered	96477
created	5
invoiced	314
processing	301
unavailable	609

- **Data insight**: Top 10 sellers in a given year in terms of the number of products sold
- **Business Decision:** Approaching them with special offers

```
    Go;
    CREATE PROC Top_sellers @year NVARCHAR(4)
    AS (
        SELECT ID, [Number_of_Products_Sold]
        FROM (
            SELECT *, ROW_NUMBER() OVER (ORDER BY [Number_of_Products_Sold] DESC) AS [rank_num]
            FROM (
                SELECT S.ID, Count(S.ID) AS [Number_of_Products_Sold]
                FROM [dbo].[Seller] AS S
                INNER JOIN [dbo].[Product] AS P
                ON S.ID = P.Seller_ID
                INNER JOIN [dbo].[Item] AS I
                ON P.ID = I.Product_ID
                INNER JOIN [dbo].[Order] AS O
                ON O.ID = I.Order_ID
                WHERE year(o.Act_Date) = @year
                GROUP BY S.ID) AS A) AS B
        WHERE [rank_num] <= 10
    )

    Go;
    EXEC Top_sellers '2018'
```

Results | Messages

ID	Number_of_Products_Sold
955fee9216a65b617aa5c0531780ce60	1279
6560211a19b47992c3666cc44a7e94c0	1113
1025f0e2d44d7041d6cf58b6550e0bfa	1035
1f50f920176fa81dab994f9023523100	978

- **Data insight**: The number of delivered orders every year
- **Business Decision:** Analyzing and improving the growth rate in each year

```
    Go;
    CREATE VIEW Orders_Year
    AS (
        SELECT YEAR(O.Act_Date) AS [Year], COUNT(Act_Date) AS [Number_of_Orders]
        FROM [dbo].[Order] AS O
        WHERE O.Status = 'delivered' AND Act_Date IS NOT NULL
        GROUP BY YEAR(O.Act_Date)
    )

    Go;
    SELECT * FROM Orders_Year
```

Results | Messages

Year	Number_of_Orders
2016	266
2017	40930
2018	55273

- **Data insight**: The number of delivered orders in every state in 2018
- **Business Decision:** Analyzing and improving the delivered rate in each state

```
121     Go;
122     CREATE FUNCTION Deliverd_Orders_State (@year NVARCHAR(4), @status NVARCHAR(20))
123     RETURNS TABLE AS
124     RETURN (
125         SELECT L.State, COUNT(L.State) AS Number_OF_Orders
126         FROM [dbo].[Order] AS O
127         INNER JOIN [dbo].[Location] AS L
128         ON L.ID = O.Location_ID
129         WHERE O.Status = @status AND YEAR(O.Act_Date) = @year
130         GROUP BY L.State
131     )
132
133     Go;
134     SELECT *
135     FROM Deliverd_Orders_State('2018', 'delivered')
136     ORDER BY Number_OF_Orders DESC
```

Results | Messages

State	Number_OF_Orders
SP	6657
MG	1603
RJ	1546
RS	951
PR	919

- **Data insight**: Reviews rate number
- **Business Decision:** Approaching the customers of top rates with offers, and working on solve the issues of the customers whose rates are low

```
143     Go;
144     CREATE VIEW Orders_Rate
145     AS (
146         SELECT R.Rate, COUNT(R.Rate) As [Number_of_Rates]
147         FROM [dbo].[Review] AS R
148         RIGHT OUTER JOIN [dbo].[Order] AS O
149         ON O.ID = R.Order_ID
150         WHERE R.Rate IS NOT NULL
151         GROUP BY R.Rate
152     )
153
154     Go;
155     SELECT * FROM Orders_Rate
156
```

Results | Messages

Rate	Number_of_Rates
1	11281
2	3114
3	8097
4	19007
5	56910

- **Data insight**: Bestselling Products in every state
- **Business Decision:** Improving the quality of the products and increasing the market of these products in these states

```sql
WITH Num_Product_Every_State
AS (
    SELECT I.Product_ID, L.[State], COUNT (I.Product_ID) AS Number_Of_Products
    FROM [dbo].[Product] AS P
    INNER JOIN [dbo].[Item] AS I
    ON P.ID = I.Product_ID
    INNER JOIN [dbo].[Order] AS O
    ON O.ID = I.Order_ID
    INNER JOIN [dbo].[Location] AS L
    ON L.ID = O.Location_ID
    GROUP BY I.Product_ID, L.[State]
)

SELECT Product_ID AS [Best_Product], [State], Number_Of_Products
FROM (
    SELECT * , ROW_NUMBER() OVER (PARTITION BY [State] ORDER BY Number_Of_Products DESC) AS [rank_num]
    FROM Num_Product_Every_State
) AS A
WHERE [rank_num] = 1
ORDER BY Number_Of_Products DESC
```

Results | Messages

Best_Product	State	Number_Of_Products
aca2eb7d00ea1a7b8ebd4e68314663af	SP	74
d1c427060a0f73f6b889a5c7c61f2ac4	RJ	27
3dd2a17168ec895c781a9191c1e95ad7	MG	24

- **Data insight**: The number of orders every month in a specific year in order
- **Business Decision:** Analyzing and improving the growth rate in each month in comparison with the previous month

```sql
Go;
ALTER PROC Orders_in_Months @year NVARCHAR(4)
AS
    SELECT FORMAT(O.Act_Date,'MMMM') AS [Month], COUNT(O.ID) AS [Number_of_Orders]
    FROM [dbo].[Order] AS O
    WHERE O.[Status] = 'delivered' AND YEAR(O.Act_Date) = @year
    GROUP BY FORMAT(O.Act_Date,'MMMM')
    ORDER BY MONTH(FORMAT(O.Act_Date,'MMMM') + '1,1')


EXEC Orders_in_Months '2017'
```

Results | Messages

Month	Number_of_Orders
January	283
February	1351
March	2382
April	1849
May	3751
June	3223
July	3455
August	4302
September	3965
October	4494
November	4670
December	7205

- **Data insight**: Bestselling products every month in a given year
- **Business Decision:** Increasing the market of these products

```
Go;
CREATE FUNCTION Num_Product_Every_Month(@year NVARCHAR(4))
RETURNS TABLE AS
RETURN (
    SELECT Product_ID AS [Best_Product], [Month], Number_Of_Products
    FROM (
        SELECT * , ROW_NUMBER() OVER (PARTITION BY [Month] ORDER BY Number_Of_Products DESC) AS [rank_num]
        FROM (
            SELECT I.Product_ID, FORMAT(O.Act_Date,'MMMM') AS [Month], COUNT (I.Product_ID) AS Number_Of_Products
            FROM [dbo].[Product] AS P
            INNER JOIN [dbo].[Item] AS I
            ON P.ID = I.Product_ID
            INNER JOIN [dbo].[Order] AS O
            ON O.ID = I.Order_ID
            INNER JOIN [dbo].[Location] AS L
            ON L.ID = O.Location_ID
            WHERE YEAR(O.Act_Date) = '2017'
            GROUP BY I.Product_ID, FORMAT(O.Act_Date,'MMMM')
        ) AS B
    ) AS A
    WHERE [rank_num] = 1
)
Go;

SELECT *
FROM Num_Product_Every_Month('2017')
ORDER BY [Month]
```

Results | Messages

Best_Product	Month	Number_Of_Products
7e0dc102074f8285580c9777f79c90cf	April	8
f1c7f353075ce59d8a6f3cf58f419c9c	August	13
53759a2ecddad2bb87a079a1f1519f73	December	30

- **Data insight**: The orders which quantity is one
- **Business Decision:** Approaching the customers with offers to increase the quantity

```
CREATE PROCEDURE Quantity
as (
select [Order_ID]
from [dbo].[Item]
where [Quantity] = 1)
go
EXEC Quantity
```

- **Data insight**: Comparison between 2017 and 2018 in each month
- **Business Decision:** to know if decrease or increase from previous year

```
205     SELECT [Month], [2017],[2018]
206     FROM (
207         SELECT FORMAT(O.Act_Date,'MMMM') AS [Month], FORMAT(Act_Date,'yyyy') AS [Year], O.ID
208         FROM [dbo].[Order] AS O
209         WHERE O.[Status] = 'delivered' AND YEAR(O.Act_Date) IN ('2017', '2018')
210         GROUP BY FORMAT(O.Act_Date,'yyyy'), FORMAT(O.Act_Date,'MMMM'), O.ID
211     ) AS T1
212     PIVOT (
213         COUNT(ID)
214         FOR  [Year]
215         IN ([2017], [2018])
216     ) AS T2
217     ORDER BY MONTH([Month] + '1,1')
```

108 %

Results | Messages

	Month	2017	2018
1	January	283	6597
2	February	1351	5850
3	March	2382	6824
4	April	1849	7850
5	May	3751	7111
6	June	3223	6829
7	July	3455	5839
8	August	4302	8314
9	September	3965	56
10	October	4494	3
11	November	4670	0
12	December	7205	0

- **Data insight**: Number of Customers who did not make any purchases.
- **Business Decision:** to make for him some offers

```
221     Go;
222     CREATE VIEW Customers_not_purchases
223     AS (
224         SELECT Count(C.ID) AS [Number of Customers]
225         FROM [dbo].[Customer] AS C
226         WHERE C.ID  NOT IN (
227             SELECT O.Customer_ID FROM [dbo].[Order] AS O)
228     )
229     GO;
230
231     SELECT * FROM Customers_not_purchases
232
```

108 %

Results | Messages

	Number of Customers
1	0