Brazilian E-Commerce Database

Database Project on **Microsoft SQL Server**

Information Technology Institute - ITI

Eslam Elsagheer
Ahmed Shawky
Hend Abdullah
Abdelrahman Ahmed

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.

Scenario

1. Customer can make one or more order.
2. customer can review order.
3. Order can have one or more item.
4. Order must have a location.
5. Order must have a payment method.
6. Each Product must have Seller.
7. Category might have one or more products.
8. Each item might have a quantity from one product.

Business Problems for Olist Ecommerce Stores:

1. Which part of business doesn't generate Profit ?
2. How to enhance revenue for business in each month ?
3. How effectively do employees generate revenue ?
4. What is customer retention rate ?
5. Will Customers make referrals ?
6. Who are most valuable and costly Customers ?
7. Which Products business should invest more in ?
8. How to enhance Value Delivery Process ?
9. Which Market Segment generate revenue to invest more in ?
10. How can business enhance yearly and monthly growth ?

Database

Tables and summary

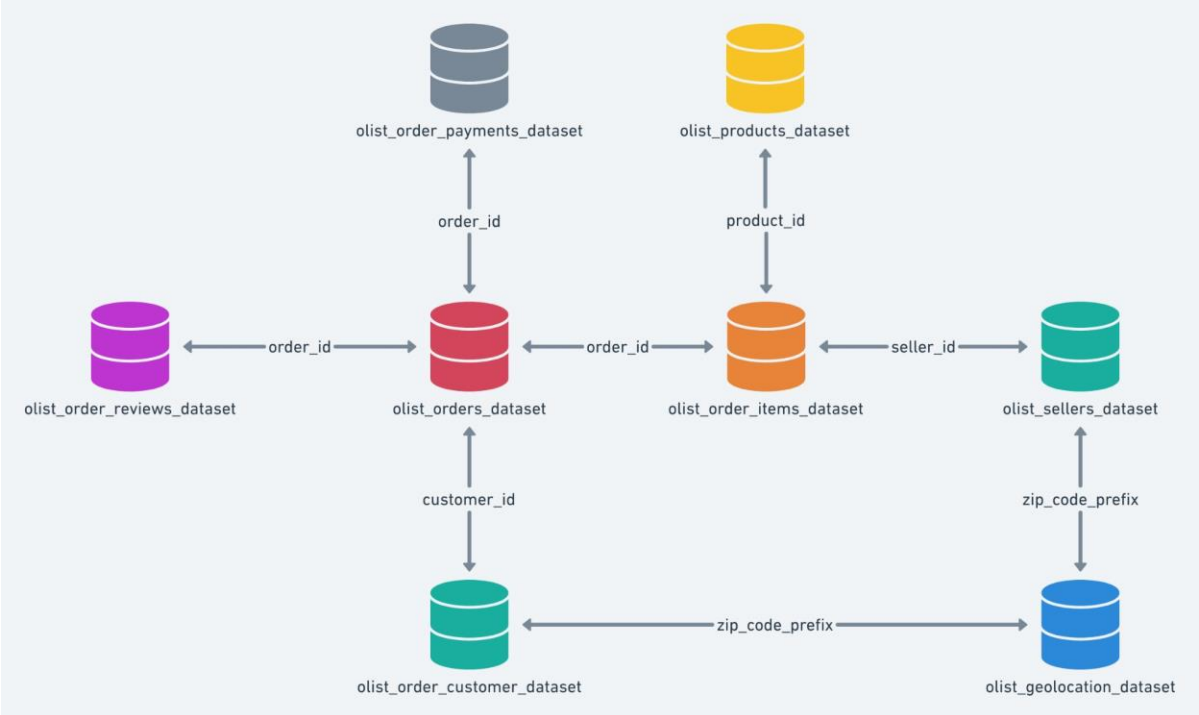
- olist_customers_dataset.csv
- olist_geolocation_dataset.csv
- olist_order_items_dataset.csv
- olist_order_payments_datas...
- olist_order_reviews_dataset...
- olist_orders_dataset.csv
- olist_products_dataset.csv
- olist_sellers_dataset.csv
- product_category_name_tra...

Summary

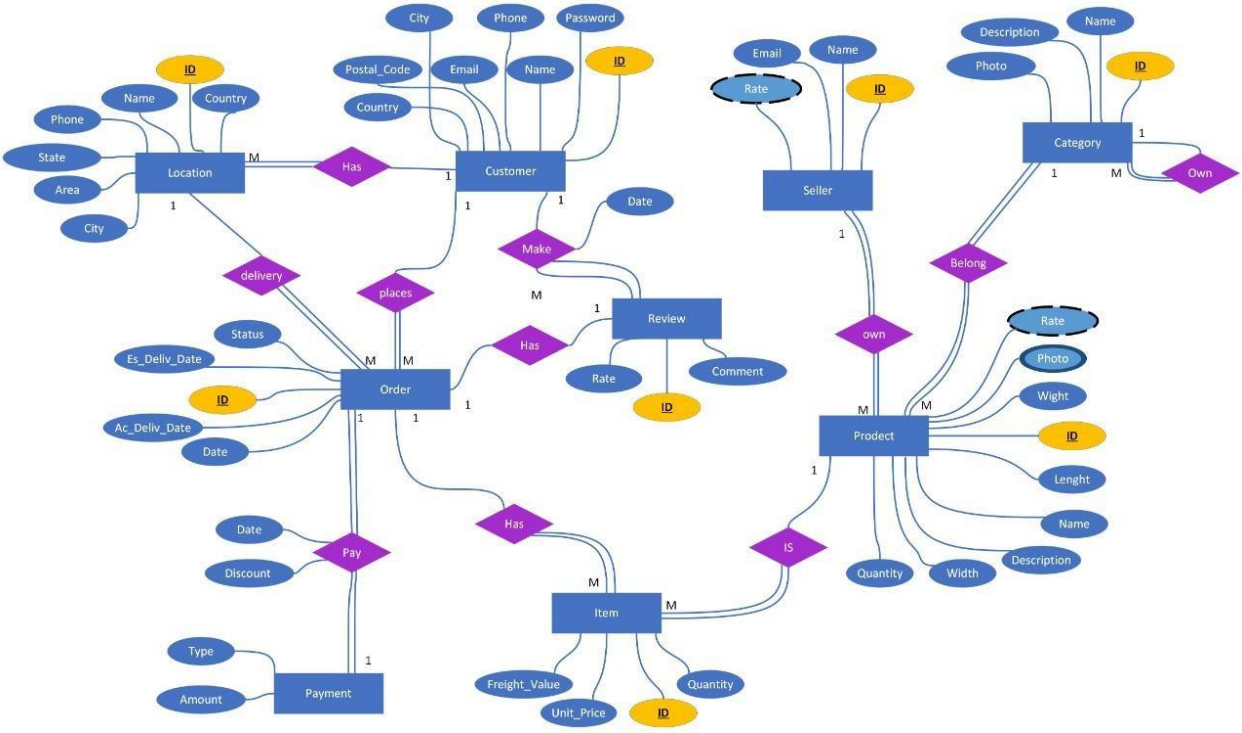
- 9 files
  - .csv 9
- 52 columns
  - String 13
  - Integer 13
  - Uuid 12
  - Other 14

Schema

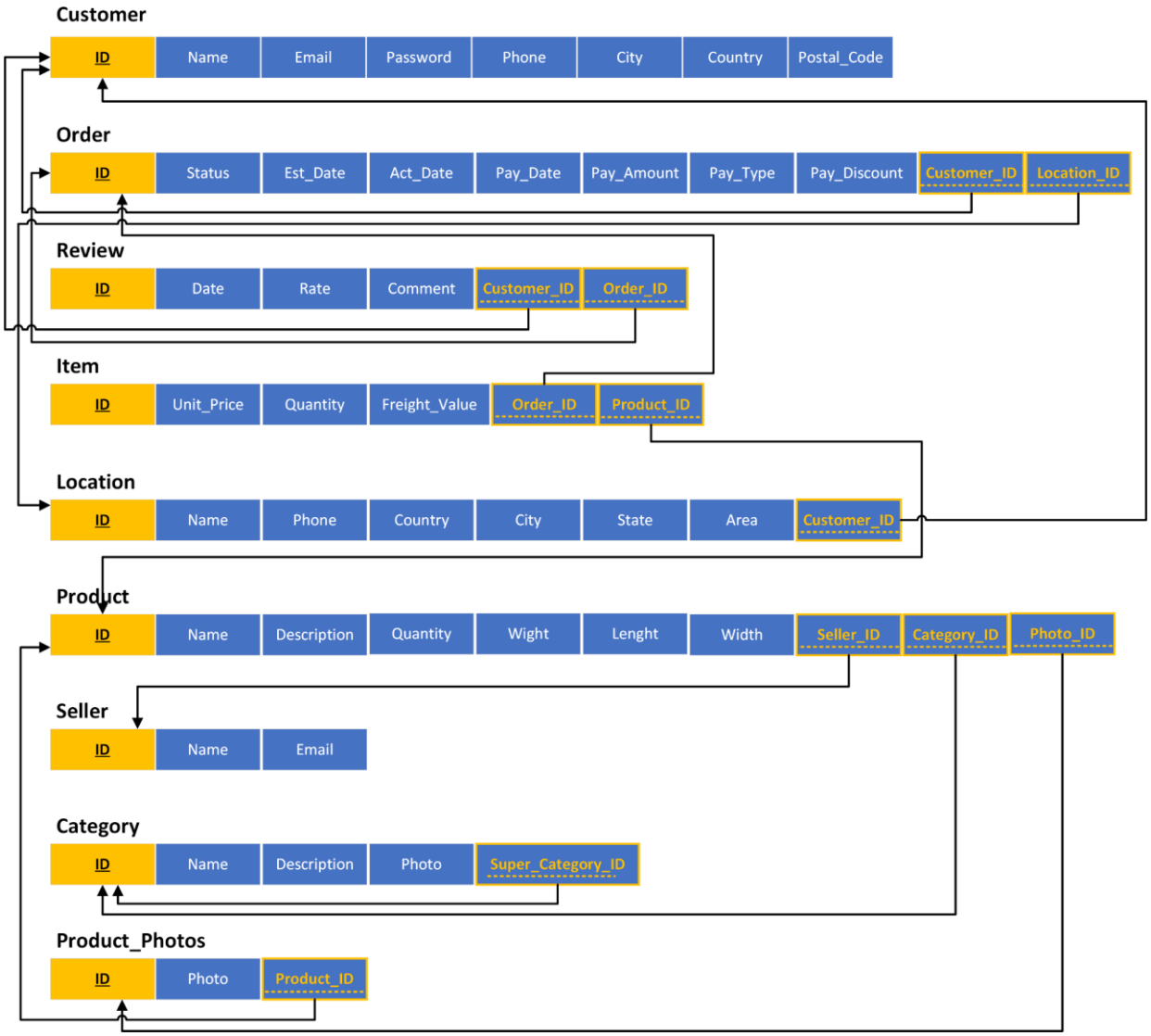
The data is divided into multiple datasets for better understanding and organization. Please refer to the following data schema when working with it:



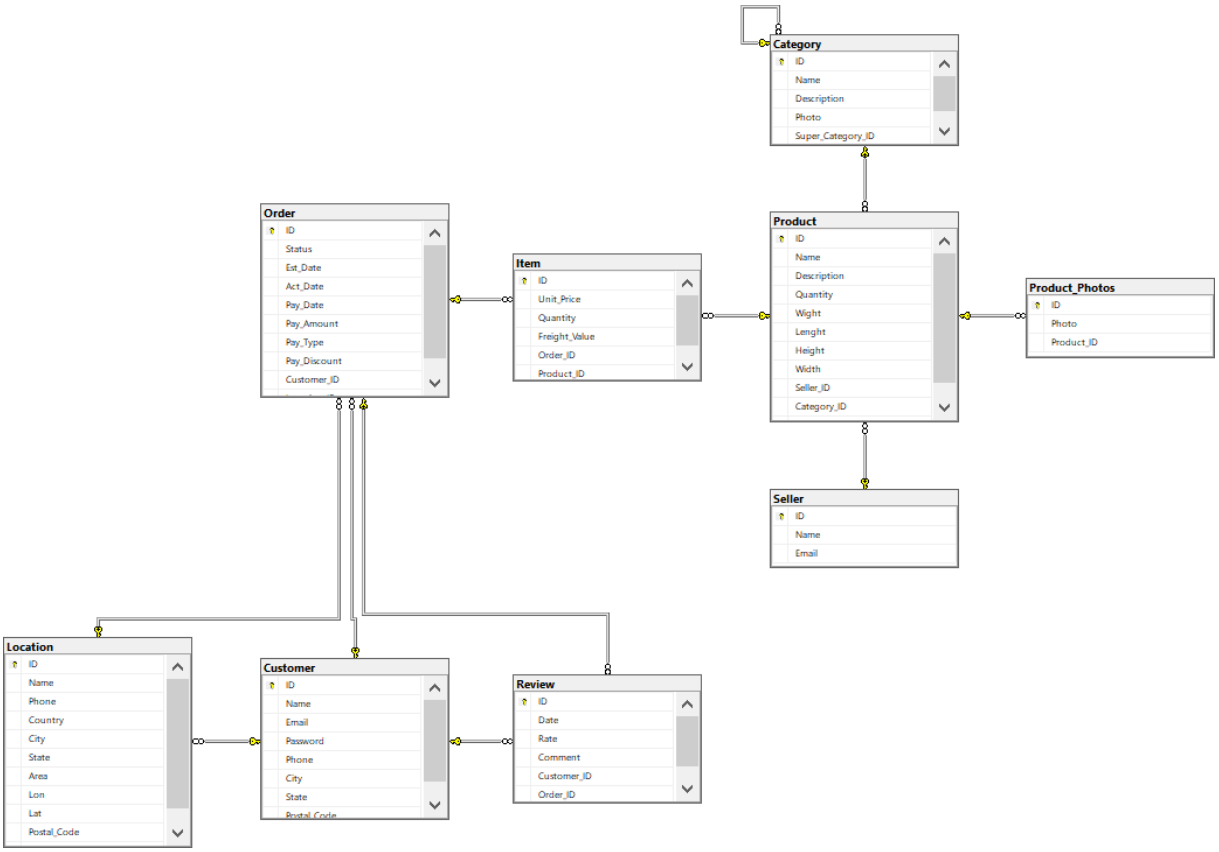
ERD


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
1
1
1
ID
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



Stored Procedure

- **Fill Data, Removing Duplications and Conflicts**

Fill Data, Remove D...Administrator (55))

```
Create Procedure Fill_Location_Data
as
Insert Into dbo.Location
(ID,Name,Phone,City,State,Lat,Long,Postal_Code,Customer_ID)
Select NEWID(),null,null,geolocation_city,geolocation_state,geolocation_lat,geolocation_lng,
geolocation_zip_code_prefix,null
From Dataset.dbo.olist_geolocation_dataset

Exec Fill_Location_Data

Create Procedure Fill_Customer_Data
as
Insert Into dbo.Customers
(ID,Name,Phone,Email,Postal_Code,City,State)
Select customer_id,null,null,null,customer_zip_code_prefix,customer_city,customer_state
From Dataset.dbo.olist_customers_dataset

Exec Fill_Customer_Data

Create Procedure Fill_Category_Data
as
Insert Into dbo.Category
(ID,Name,Description,Photos)
Select NEWID(),product_category_name_english,product_category_name,null
From Dataset.dbo.product_category_name_translation

Exec Fill_Category_Data
```

Fill Data, Remove D...Administrator (55))*

```
/* Remove Duplicate data from customer */

        WITH Remove_Dublicate_customers AS (
            SELECT *, ROW_NUMBER() OVER (PARTITION BY [ID] ORDER BY ID) AS row_num
            FROM [dbo].[Customers]
        )

        DELETE FROM Remove_Dublicate_customers
        WHERE row_num > 1

/*remove conflict data between order and item*/
        WITH Remove_Conflict_Item_Orders
        AS (
            SELECT *
            FROM [dbo].[Item]
            WHERE order_id not in ( SELECT ID FROM [dbo].[Orders]))

        DELETE FROM Remove_Conflict_Item_Orders

/*remove conflict data between order and Customer*/
        WITH Remove_Conflict_Order_Customers
        AS (
        SELECT *
        FROM [dbo].[Orders] AS O
        WHERE O.Customer_ID NOT IN (
                                    SELECT ID
                                    FROM [dbo].[Customers])
        )

        UPDATE Remove_Conflict_Order_Customers SET Customer_ID = null
```

Stored Procedure

- **Crud Operations**

Crud Operations Pr...Administrator (58)) | Fill Data, Remove D...Administrator (55))*

```
Create Procedure Crud_Operations_Category
( @ID varchar(50),@Name varchar(50),@Description varchar(50),@Photos int,@Statment_Type varchar(20) = '' )
as
Begin
If @Statment_Type = 'Select'
    Begin
        Select *
        From dbo.Category
    End

If @Statment_Type = 'Insert'
    Begin
        Insert into dbo.Category
        (ID ,Name ,Description ,Photos)
        Values (@ID ,@Name ,@Description ,@Photos)
    End

If @Statment_Type = 'Update'
    Begin
        Update dbo.Category
        Set Name = @Name ,
            Description = @Description ,
            Photos = @Photos
        Where ID = @ID
    End

Else If @Statment_Type = 'Delete'
    Begin
        Delete From dbo.Category
        Where ID = @ID
    End
End

Exec Crud_Operations_Category '"007FA526-6FB4-46C5-8CC3-784B1BE8E98C"','drinks','bused',5,'Select'
```

Crud Operations Pr...Administrator (58))*

```
Create Procedure Crud_Operations_Item
( @ID varchar(50),@Unit_Price Float,@Quantity int ,@Freight_Value Float,@Order_ID varchar(50),
@Product_ID varchar(50),@Seller_ID varchar(50),@Statment_Type varchar(20) = '' )
as
Begin
If @Statment_Type = 'Select'
    Begin
        Select *
        From dbo.Item
    End

If @Statment_Type = 'Insert'
    Begin
        Insert into dbo.Item
        (ID ,Unit_Price,Quantity,Freight_Value,Order_ID,Product_ID,Seller_ID)
        Values (@ID ,@Unit_Price,@Quantity,@Freight_Value,@Order_ID,@Product_ID,@Seller_ID)
    End

If @Statment_Type = 'Update'
    Begin
        Update dbo.Item
        Set Unit_Price = @Unit_Price ,
            Quantity = @Quantity ,
            Freight_Value = @Freight_Value ,
            Order_ID = @Order_ID ,
            Product_ID = @Product_ID ,
            Seller_ID = @Seller_ID
        Where ID = @ID
    End

Else If @Statment_Type = 'Delete'
    Begin
        Delete From dbo.Item
        Where ID = @ID
    End
End
Exec Crud_Operations_Item
```

Stored Procedure

- Generating Technical Report

Generating Reports....Administrator (55))

```
Create Procedure Top_Customers_By_City_Report
as
begin
Select *
From
(Select Customers.ID, Customers.City , COUNT(Orders.ID) "No. Of Orders" ,
SUM((Item.Unit_Price)*(Item.Quantity)) "Total Revenue" ,
dense_rank() over (partition by Customers.City
Order by SUM((Item.Unit_Price)*(Item.Quantity)) desc , COUNT(Orders.ID) desc ) "Rank"
From Customers , Orders , Item
Where Customers.ID = Orders.Customer_ID And Orders.ID = Item.Order_ID
Group by Customers.ID, Customers.City) T
Where Rank in (1,2,3,4,5)
End

Exec Top_Customers_By_City_Report
```

Generating Reports....Administrator (55))

```
Create Procedure Top_Products_By_City_Report
as
begin
Select *
From
(Select Products.Category, Products.Name , Location.City , COUNT(Orders.ID) "No. Of Orders" ,
SUM((Item.Unit_Price)*(Item.Quantity)) "Total Revenue" ,
dense_rank() over (partition by Location.City
Order by SUM((Item.Unit_Price)*(Item.Quantity)) desc , COUNT(Orders.ID) desc ) "Rank"
From Products , Location , Orders , Item
Where Products.ID = Item.Product_ID And Orders.ID = Item.Order_ID And Orders.Location_ID = Location.ID
Group by Products.Category, Products.Name , Location.City ) T
Where Rank in (1,2,3,4,5)
End

Exec Top_Products_By_City_Report
```