Brazilian E-Commerce Public Dataset by Olist

I was hired by [Olist](https://olist.com/pt-br/) to create a database for their Brazillian E-Commerce public data and conduct preliminary analysis regarding some of the topics they were interested in.

This data primarily contains the orders made at the Olist store and has information on more than 100,000 orders made between 2016 and 2018 in Brazil.

It allows for viewing the data from multiple dimensions, such as product categories, payments, sellers, and reviews provided by the customers. The data is supplemented by the geolocation of the customers. However, the data has been anonymized to protect consumers and sellers.

My goal was to first create a central data warehouse using different datasets provided by the organization and then create data marts for my analytics. The process can be defined using the following:

**Creation of the Operational Layer:**

I first created the database/schema Brazillian\_Olist and used it for my further working. In this database, I created different tables for each datafile I received from the company, defined my columns, and then loaded the data into individual tables.

Before loading, 3 of the csv data files needed some cleaning so I performed a few cleaning operations using Excel. The changes made to each of these data files before loading are as follows:

1. **Customer file:**
   * Converted the zip codes into integers as these were stored as strings.
2. **Geolocation file:** 
   * Deleted all the duplicate zip code values as the file was supposed to give unique zip codes for geographical breakdowns in Brazil. The data was collected from multiple sources, hence contained duplicate values for zip codes with slight variations in latitude and longitude, which signified the different centroid considered by each data provider.
   * some entries in column 'city' included state and country name with commas to separate it, even though the column only needed city name. So, I split the column using Excel, based on the comma in the same column and deleted the city and country name extracted from this column as state was already given in the ‘state’ column.
   * Converted the zip codes into integers in Excel as it was initially stored as a string.
3. **Order\_Reviews file:**
   * Deleted the ‘Review\_Comment’ & ‘Review\_Title’ column since I won't be doing sentiment analysis based on the wordings.

**ANALYTICS:**

Once I loaded the tables into the database, I used the reversed engineering option to pre-populate the tables in the EER diagram. I then created links between the tables as per the data in individual tables.

The tables were then joined to create the Olist\_DW data warehouse. Using this data warehouse, I then created multiple data mart views to answer some of the analytical questions.

**Table Descriptions:**

1. **Customers:** The table contains information on customer id and their location. Customer\_id is the primary key in this table.
2. **Order\_Reviews:** The table contains reviews provided by the customers for different orders and when these were filled. Since none of the columns had unique values, I created an auto-increment integer value as the primary key.
3. **Orders\_Main:** This contains information on the timings of when the orders were made and when the orders were delivered. Order\_id is the primary key in this table.
4. **Order\_Payments:** The table contains information on how the payment was made and the value of the payment made.
5. **Order\_Items:** Contains information on product id, seller id, total price and freight value. Here as well, since there were no columns with unique values, I created an auto-increment integer as the primary key.
6. **Products:** Contains details on the ordered product, such as its dimensions and category. The category here is given in Portuguese and hence this table will be joined with the Categories table to get the English translations of the product categories. The primary key in this table is product\_id.
7. **Categories:** This is a basic table with translations of the product categories into English. The primary key is product\_category\_por in this table.
8. **Sellers:** Contains information on sellers, such as their location. The primary key in this table is the seller\_id.
9. **Geolocation:** The table contains unique zip codes for localities in Brazil along with other location columns. The primary key in this table is zipcodes.

**Relationships:**

1. 1:n between Geolocation and Customers table. Since there could be multiple customers ordering from the same zip code. The tables will be joined using zipcodes.
2. 1:n between the Customers and Orders\_Main table. Since there could be multiple orders placed by the 1 customer. The tables will be joined using customer\_id.
3. 1:n between Orders\_Main and Order\_Reviews table. As a customer could order multiple items in a single order and get multiple review forms for each product. The tables will be joined using order\_id.
4. 1:n between Orders\_Main and Order\_Payments table. Since a customer had the option of paying for an order using a voucher and once of the other commonly used options such as credit card, cash etc. The tables will be joined using order\_id.
5. 1:n between Orders\_Main and Order\_Items table. Since a single order may contain more than 1 item/product. The tables will be joined on order\_id.
6. 1:n between Sellers and Order\_Items table. As a single seller might be fulfilling multiple order items. The tables will be joined on seller\_id.
7. 1:n between Products and Orders\_main table. As a single product might get ordered more than once. The tables will be joined on product\_id.
8. 1:n between Categories and Products table. Since multiple products can fall under a single category. The tables will be joined using Category\_por.

**Database EER Diagram:**

Graphical user interface

Description automatically generated

**Analytical Questions:**

1. **Top 5 categories in terms of sales by year?**
2. **Top 5 sellers by year?**
3. **Top 5 sellers by average review score by year?**
4. **Most famous payment type by year?**

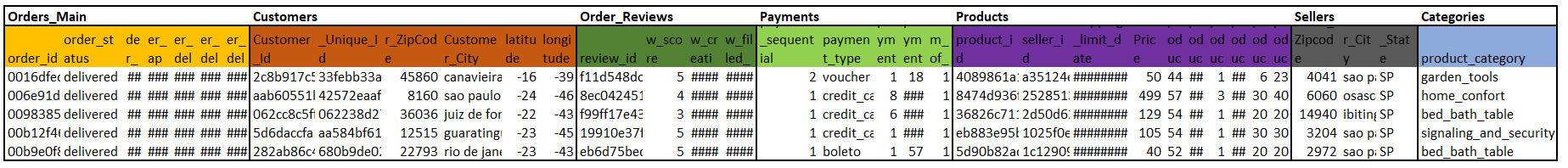
**ANALYTICAL LAYER:**

For the analytical layer, I created the data warehouse and stored it into the table Olist\_DW. The data warehouse contains specific fields from all the above listed tables.

To create the data warehouse, I first created a procedure that would join the tables to create the data warehouse. However, due to the computational limitations of my machine, I had to divide the joining into two steps.

It first joins the tables Orders\_Main, Customers, geolocation, and Order\_Reviews and stores these in a table called Customer\_Order\_Main. Then it joins this Customer\_Order\_Main table with Payments, Order\_Items, sellers, and products to create the final Olist\_DW datawarehouse. Finally, after creating the data warehouse, it deletes the temporary table Customer\_Order\_Main to decrease the load on the server.

A quick snapshot of the data warehouse table is as follows.



**DATA MART:**

For final analytics of the data provided I have created 4 views as data marts, each answering a question provided by the company.

1. **Categories\_by\_year**

This view contains product\_category, order\_purchase\_date which is used to transform and create a new additional column called order\_purchase\_year, order\_id and order\_status for each product enabling the analysts to draw analysis on best performing products by year. The view contains filtered data on the products that have been delivered to the customers.

It also answers the following question.

**Top 5 categories in terms of sales by year?**

I have created a procedure (categories\_by\_year) here that allows the analysts to enter a specific year when calling the procedure and get the top 5 best performing products by counting the instances of each product ordered.

A picture containing table

Description automatically generated

1. **Seller\_by\_year**

This view contains order\_id, seller\_id, order\_purchase\_date which is used to transform and create a new additional column called order\_purchase\_year where the products are delivered to the customers. The view can be used by the analysts to get information on top sellers by year.

It answers the following question in this case.

**Top 5 sellers by year?**

I have created a procedure (sellers\_by\_year) here that allows the analysts to enter a specific year when calling the procedure and get the top 5 best performing sellers in terms of orders delivered by counting the order ids.

Table

Description automatically generated

1. **Sellers\_by\_Review**

This view contains order\_id, seller\_id, year(review\_filled\_date) as review\_filled\_year , review\_score, and review\_id. This contains products without any restriction on delivery status as the review is only generated after the product has been delivered or cancelled.

This view can be used by the analysts to get stats on sellers based on the reviews provided by the costumers.

It answers the following question in this case.

**Top 5 sellers by average review score by year?**

I have created a procedure (sellers\_by\_review) that gives the average review score of each seller and then provides us with the top 5 sellers based on the reviews provided in each year. The analysts can enter a specific year when calling the procedure to get year wise top 5 sellers list by average review score.

Table

Description automatically generated

1. **Payment\_by\_Year**

This view contains order\_id, year(order\_purchase\_date) as payment\_year, and payment\_type. The goal of this view is to allow the analysts to identify the commonly used payment methodologies, in case the company needs to focus on certain types of payment types.

It answers the following question in this case.

**Most famous payment type by year?**

I have created a procedure (payment\_by\_year) that gives the payment methodologies based on the count of each payment methodology in each year. The analysts can enter a specific year when calling the procedure to get year wise top payment types used by the customers.

Graphical user interface, application

Description automatically generated with medium confidence