Report

IT 300

Business Intelligence and Database Management Systems

Business Intelligence Mini-Project
Online Store in India

Authors:
Roua Othmani

Submitted to:
Dr. Manel Abdelkader

Contents

1 Introduction

This business intelligence project is focused on analyzing order details data of an online shop located in India.
The project is designed to cater to a range of objectives, including understanding the performance of different products, identifying key sales and revenue drivers, and discerning customer preferences and behaviors.
Our goals are centered on unraveling patterns within units ordered per product and category, profit margins, monthly sales trends, regional revenue distribution, preferred payment modes, and customer retention rates.

The project's deliverables include actionable recommendations to enhance profitability, strategic insights for market expansion, and a deeper understanding of customer dynamics.

2 Implementation

2.1 Data Gathering

We extracted the Orders and Customers dataset in India from Kaggle. This is a link to the dataset.

2.2 Data Preparation

We used Python to prepare the data for the warehouse by manipulating it and setting it up. As a first step, the data was converted into one format (CSV) from various formats (CSV, JSON).
To make data more easily manipulatable and manageable, we restructured it. Initially, we merged the two CSV files into one dataset named order_details. Then, we used Pandas to perform the transformations needed to match our defined data warehouse server (cleaning, column renaming, data type conversion, deriving new columns from existing ones, and data aggregation), all the transformations are provided in the ETL.ipynb file.

Below is the code for some data transformation examples:

```
#handling duplicates
has_duplicates = order_details.duplicated(subset='Order ID').any()
if has_duplicates:
    print("There are duplicate rows based on the 'Order ID' column.")
else:
    print("There are no duplicate rows based on the 'Order ID' column.")
order_details = order_details.drop_duplicates(subset='Order ID')

#rename the columns
order_details.rename(columns={'Amount': 'TotalPrice'}, inplace=True)

#Date type conversion
order_details['Order Date'] = pd.to_datetime(order_details['Order Date'] , format='mixed')

#Customer status column extraction
order_details['CustomerStatus'] = order_details.duplicated(subset='CustomerName',
keep='first').map({True: 'Repeat', False: 'New'})

#Profitability aggregation
order_details['ProfitMargin'] = (order_details['Profit'] / order_details['TotalPrice']) *
100

product_profit_margin=order_details.groupby('Product')['ProfitMargin'].mean()
.reset_index().sort_values(by='ProfitMargin', ascending=False)
```

2.3 Data Storage

2.3.1 Storage

As a data warehouse system, we are using PostgreSQL, we used SQLAlchemy to map tables in our code, and psycopg2, a Python library that facilitates the seamless integration between the two.

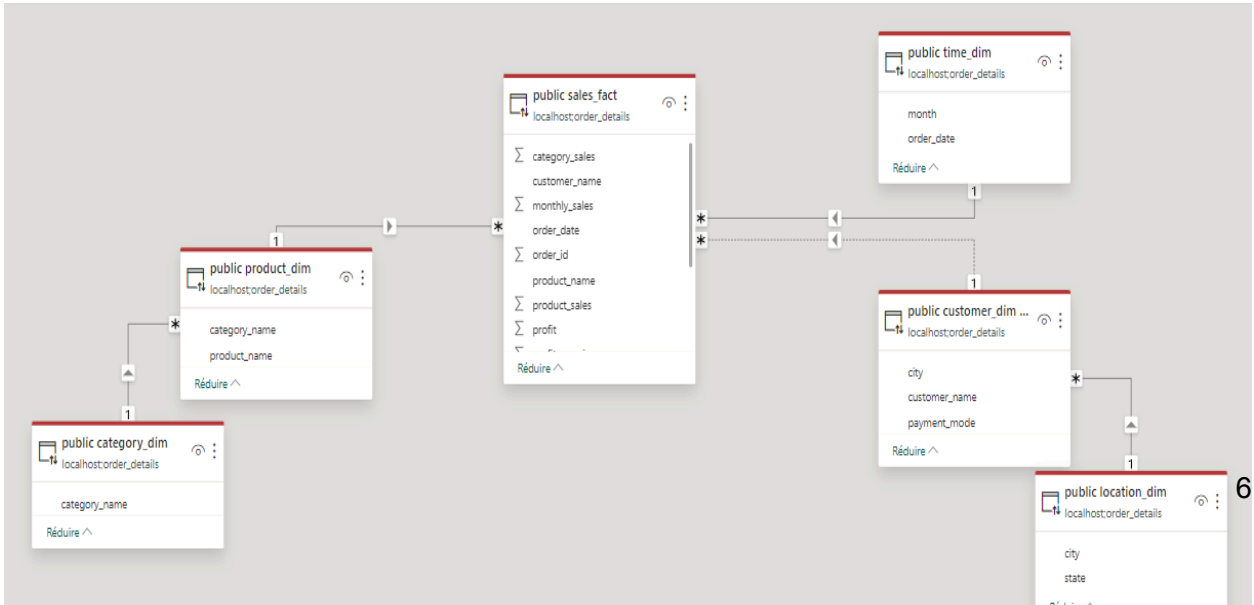
We have one main table named order_details and the tables we extracted to look for meaningful insights like sorted_category_sales, sorted_product_sales, sorted_sales_per_city, monthly_sales, product_profit_marin. You can find the detailed code for data loading in Loading to DWH.ipynb.

Below, is the code for the Facts and Dimensions tables creation and loading and

```
from sqlalchemy import create_engine, Table, Column, Integer, String, Date, Numeric,
ForeignKey
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import relationship
from datetime import datetime

engine = create_engine('postgresql://postgres:roua13653200@localhost:5432/order_details')

# Create a base class for declarative class definitions
Base = declarative_base()
class CategoryDim(Base):
    __tablename__ = 'category_dim'
    category_name= Column(String, primary_key=True)

class ProductDim(Base):
    __tablename__ = 'product_dim'
    product_name= Column(String, primary_key=True)
    category_name = Column(String, ForeignKey('category_dim.category_name'))
    category = relationship('CategoryDim')

class LocationDim(Base):
    __tablename__ = 'location_dim'
    city = Column(String, primary_key=True)
    state = Column(String)

class CustomerDim(Base):
    __tablename__ = 'customer_dim'
    customer_name = Column(String, primary_key=True)
    city=Column(String, ForeignKey('location_dim.city'))
    location = relationship('LocationDim')
    payment_mode = Column(String)
```

```
class TimeDim(Base):
    __tablename__ = 'time_dim'
    order_date = Column(Date, primary_key=True)
    month = Column(Integer)


class SalesFact(Base):
 __tablename__ = 'sales_fact'
    order_id = Column(Integer, primary_key=True)
    product_name = Column(String, ForeignKey('product_dim.product_name'))
    order_date = Column(Date, ForeignKey('time_dim.order_date'))
    customer_name = Column(String, ForeignKey('customer_dim.customer_name'))

    units_ordered = Column(Integer)
    total_price = Column(Numeric(10, 2))
    profit_margin = Column(Numeric(10, 2))
    profit=Column(Numeric(10, 2))
    monthly_sales = Column(Numeric(10, 2))
    product_sales = Column(Numeric(10, 2))
    category_sales = Column(Numeric(10, 2))
    sales_per_city = Column(Numeric(10, 2))

    product = relationship('ProductDim')
    location = relationship('LocationDim')
    customer = relationship('CustomerDim')

     Base.metadata.create_all(engine)
```

Snowflake Schema



Customers: represents the customers' name, city, and preferred payment mode.
Products: represents the product name and the category name.
Category: represents the name of the category.
Location: represents the customer's state and city.
Time: represents the order date.
Sales: represents the category sales, monthly sales, product sales, profit, profit margin, sales per city, total price, and units ordered.

2.3.2 Fact

The fact of this dataset is the sales of the online store and it includes:

- Category sales
- Monthly sales
- Product sales
- Profit
- Profit margin
- Sales per city
- Total price
- Units ordered

2.3.3 Dimensions

Dimensions included :

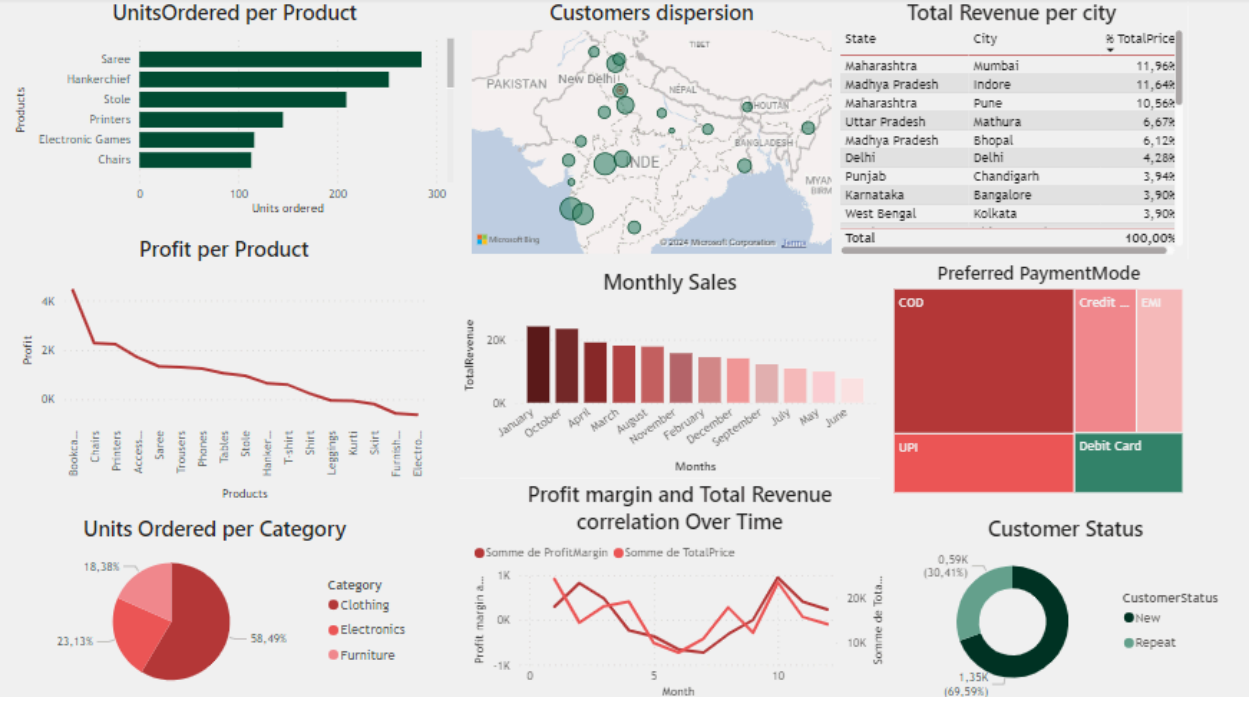
- Customer
    - ➔ Dimensions derived from the customer dimension
    - Location

- Products
    - ➔ Dimensions derived from the product dimension

- Category
- Time

We opted for a snowflake schema instead of a star schema because of the many-to-many relationships between the dimensions.

2.4 Data Visualization



We visualized our data in a [dashboard] to extract the following insights and derive conclusions:

- ❖ Units ordered per product:

- The 3 most sold products are the saree, handkerchief, and stole.
- The least sold products are kurtis, tables, and trousers.

- ❖ Units ordered per category:

- The most sold products are from the clothing category with a percentage of nearly 60%.

- ❖ Profit per product:

- Bookcases, chairs, and printers are the most profitable products.

- ❖ Monthly sales:

- The months of January and October are the ones with the highest sales of the year 2018.

- ❖ Total revenue per city:

- Mumbai, Indore, and Pune are the cities that generate the highest sales.
- Nearly all of the customers are from India. However, we can detect that there is interest in some other countries like Russia, Vietnam, Bangladesh,..

- ❖ Preferred payment mode:

- Most customers prefer to pay using the Cash On Delivery mode.

- ❖ Customer status:

- 30% of customers are returning clients.

- ❖ Profit and total revenue correlation over time:

- From January to May: Negative correlation between the profit margin and total revenues.
  - ➔ The business was facing some challenges in managing costs and maintaining profitability, and corrective measures were needed to improve profitability.
- From June to December: Positive correlation between the profit margin and the total revenues.
  - ➔ This indicates that an effective cost management strategy was implemented.
  - ➔ The business efficiently converted sales into profits, and its financial performance was strong.

3. Conclusion:

The study indicates that the highest-selling products do not necessarily translate into the most profitable ones. To enhance profitability, it is recommended to explore strategies such as increasing the selling price of popular items and implementing targeted marketing for high-profit products.

In response to international demand, they should consider an expansion strategy.

Furthermore, the observation that a significant portion of the customer base consists of returning clients suggests the successful implementation of an effective customer retention strategy. This underscores the importance of maintaining and building upon existing customer relationships for sustained business success.