Apple Retail Sales Analysis

Project Overview

This project focuses on the analysis of Apple retail sales data, encompassing over 250K rows of data. The primary goal is to preprocess and clean the data to prepare it for advanced SQL and Python-based analysis. The project emphasizes building a robust data model, cleaning the dataset, and ensuring it is ready for further analysis.

Team Members

Data Wizard Team

- **Omar Essam**: Team Leader (responsible for data preprocessing, SQL querying, and project management).
- **Omar Ahmed**: Data Analyst (responsible for database management, optimization, and advanced SQL techniques).
- **Mohamed Magdy**: Data Analyst (responsible for interpreting insights and business recommendations).
- **Heba Khaled**: Data Analyst (responsible for creating dashboards and visualizations).

Data Overview

Introduction to the Dataset

The dataset consists of five main tables, each containing specific information about Apple retail sales, products, stores, and warranty claims. Below is a simplified explanation of each table and its purpose:

1. Stores Table

This table contains information about Apple retail stores worldwide. It includes:

- **store_id**: A unique identifier for each store.
- **store_name**: The name of the store.
- **city**: The city where the store is located.
- **country**: The country of the store.

This table helps us understand where sales are happening geographically.

2. Category Table

This table defines the different product categories available in Apple stores. It includes:

- **category_id**: A unique identifier for each category.
- **category_name**: The name of the category (e.g., iPhone, MacBook, iPad).

This table helps us categorize products and analyze sales by product type.

3. Products Table

This table contains detailed information about each product sold in Apple stores. It includes:

- **product_id**: A unique identifier for each product.
- **product_name**: The name of the product (e.g., iPhone 13, MacBook Air).
- **category_id**: A reference to the category table to identify the product type.
- **launch_date**: The date when the product was launched.
- **price**: The price of the product.

This table helps us analyze product performance and pricing trends.

4. Sales Table

This table records all sales transactions. It includes:

- **sale_id**: A unique identifier for each sale.
- **sale_date**: The date of the sale.
- **store_id**: A reference to the stores table to identify where the sale occurred.
- **product_id**: A reference to the products table to identify the product sold.
- **quantity**: The number of units sold in the transaction.

This table is the core of our analysis, as it allows us to track sales performance over time and across stores.

5. Warranty Table

This table records warranty claims made by customers. It includes:

- **claim_id**: A unique identifier for each warranty claim.
- **claim_date**: The date the claim was made.
- **sale_id**: A reference to the sales table to identify the product and sale associated with the claim.
- **repair_status**: The status of the warranty claim (e.g., Paid Repaired, Warranty Void).

This table helps us analyze warranty trends and customer satisfaction.

Tasks

Data Preprocessing

- **Build a Data Model**:
  - Design an Entity-Relationship Diagram (ERD) to represent the relationships between the tables (stores, category, products, sales, and warranty).
  - Ensure the data model is normalized to avoid redundancy and improve efficiency.
- **Clean and Preprocess the Data**:
  - Handle missing values, duplicates, and inconsistencies in the dataset.
  - Convert data types where necessary (e.g., dates, numeric values).
  - Ensure data integrity by validating relationships between tables (e.g., foreign keys).
  - Perform data transformations such as aggregations, filtering, and feature engineering.

Tools

- **SQL**: For querying, joining, and aggregating data from the database.
- **Python (pandas, Matplotlib)**: For data cleaning, preprocessing, and initial visualization.

Deliverables

Cleaned Dataset Ready for Analysis

- A fully cleaned and preprocessed dataset stored in a database or CSV files.
- The dataset should be free of missing values, duplicates, and inconsistencies.
- Ensure that all relationships between tables are maintained and validated.

Data Preprocessing Notebook

- A Jupyter Notebook or Python script documenting the data preprocessing steps.
- The notebook should include:
  - **Data Loading**: Code to load the raw data from the database or CSV files.
  - **Data Cleaning**: Steps to handle missing values, duplicates, and inconsistencies.
  - **Data Transformation**: Code for aggregations, filtering, and feature engineering.
  - **Data Validation**: Checks to ensure data integrity and consistency.
  - **Visualizations**: Initial visualizations using Matplotlib to explore data trends and anomalies.
  - **Exporting Clean Data**: Code to export the cleaned dataset for further analysis.