Project Documentation

Project Title : SuperStore Sales

Team Members : Abanoub Bassem Mikhael Heshmat & Yousef Ahmed ElAzab Dagher

---

Phase 1 : Cleaning

Data Normalization Overview : The purpose of data normalization is to remove redundancy, improve data integrity, and structure the dataset into multiple related tables following 3rd Normal Form (3NF).

The original Superstore dataset contains repeated customer, order, and product information in every row. To solve this, we split the dataset into four related tables.

1. Orders Table

Purpose: Stores high-level information about each order.

Sample Fields: Order ID *(Primary Key)* , Order Date ,Ship Date ,Ship Mode ,City ,State ,Region , Customer ID

Why this table?

Every order can contain multiple products.
Separating orders avoids repeating order and shipping details on each line item.

2. Customers Table

Purpose: Contains unique information about customers.

Sample Fields: Customer ID *(Primary Key)* **,**Customer Name **,** Segment

Why this table?

Customer information appears in many order records.
We extract it to eliminate duplication and maintain clean customer profiles.

3. Products Table

Purpose: Stores unique information about each product.

Sample Fields: Product ID *(Primary Key)* **,** Product Name **,** Category **,** Sub-Category

Why this table?

Product details repeat across many order lines.
Separating them reduces redundancy and improves consistency.

4. OrderDetails Table

Purpose: Links each order to the products purchased.

Fields: Order ID *(Foreign Key → Orders)* **,** Product ID *(Foreign Key → Products)* **,** Sales

Why this table?

Each order may contain multiple items.
This table stores the transactions at the line-item level.

Relationship Diagram (ERD)

Table	Relationship
Orders ↔ OrderDetails	One-to-Many (1 order has many items)
Customers ↔ Orders	One-to-Many (1 customer places many orders)
Products ↔ OrderDetails	One-to-Many (1 product can appear in many order items)

1.SQl

The following SQL statements create a normalized relational database for the Superstore dataset.
The goal is to remove redundancy and organize the data into separate related tables:

- Customers
- Orders
- Products
- Order Details

This structure follows 3rd Normal Form (3NF).

1. Customers Table

```
CREATE TABLE Customers (
  [Customer ID] NVARCHAR(50),
  [Customer Name] NVARCHAR(50),
  Segment NVARCHAR(50),
  CONSTRAINT cust_pk PRIMARY KEY ([Customer ID]))
```

Purpose:Stores unique customer information.

Explanation:

- Customer ID is the primary key, ensuring each customer appears once.
- Customer Name and Segment describe the customer.
- This table prevents repeating customer details in every order.

2. orders Table

```
CREATE TABLE Orders (
  [Order ID] NVARCHAR(50),
  [Order Date] DATE,
  [Ship Date] DATE,
  [Ship Mode] NVARCHAR(50),
  [Customer ID] NVARCHAR(50),
  [STATE] NVARCHAR(50),
  [CITY] NVARCHAR(50),
  [Region] NVARCHAR(50),
  CONSTRAINT ord_pk PRIMARY KEY ([Order ID]),
  CONSTRAINT ord_fk FOREIGN KEY ([Customer ID])
    REFERENCES Customers([Customer ID]))
```

Purpose:Stores overall details about each order.

Explanation:

- Order ID is the primary key, uniquely identifying each order.
- Order-level attributes such as Order Date, Ship Date, and Ship Mode are stored here.
- Customer ID links each order to the Customers table through a foreign key.
- Address fields (State, City, Region) describe the shipping location.

Why separate this table?

An order can contain multiple products, so storing order information separately avoids duplication.

3. Products Table

```
CREATE TABLE Products (
  [Product ID] NVARCHAR(50),
  [Category] NVARCHAR(50),
  [Sub Category] NVARCHAR(50),
  [Product Name] NVARCHAR(50),
  CONSTRAINT pro_pk PRIMARY KEY ([Product ID]))
```

Purpose:Contains unique product information.
Explanation:

- Product ID is the primary key.
- Category-level attributes (Category, Sub-Category, Product Name) describe each product.
- This table prevents repeating product information across all sales transactions.

4. Order Details Table

```
CREATE TABLE [Order Details] (
  [Order ID] NVARCHAR(50),
  [Product ID] NVARCHAR(50),
  Sales float,
  CONSTRAINT det_pk PRIMARY KEY ([Product ID],[Order ID]),
  CONSTRAINT detor_fk FOREIGN KEY ([order ID])
    REFERENCES orders([order ID]),
  CONSTRAINT detpro_fk FOREIGN KEY ([product ID])
    REFERENCES products([product ID]))
```

Purpose: Stores line-item details — each product inside each order.

Explanation:

- **Composite Primary Key**:([Product ID], [Order ID]) ensures that each product appears only once per order.
- Contains the **Sales** value for each item.
- Two foreign keys:
  - **Order ID** → Orders table
  - **Product ID** → Products table

2.Python

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

df = pd.read_csv("Superstore Sales Dataset.csv")
```

1. Importing Required Libraries

pandas as pd : Used for loading, cleaning, and analyzing structured data. It provides DataFrame : and Series objects, which make data manipulation easy.

numpy as np : A library for numerical operations. It supports mathematical functions, arrays, and efficient computation.

(Even if not used yet, it's commonly imported for future calculation

matplotlib.pyplot as plt : A plotting library used for creating static charts such as line plots, bar charts, histograms, etc.

pd.read_csv() loads a CSV file into a **pandas DataFrame**.

df now contains all rows and columns of the *Superstore Sales Dataset*.

This DataFrame will be used for:

- Data cleaning
- Exploratory Data Analysis (EDA)
- Visualizations
- Statistical summaries

1. Loading the Dataset and Exploring Structure

```
df.info()

df = df[['Customer ID', 'Customer Name','Segment']]

df.isnull().sum()

df.dtypes

df[df.duplicated]

df.duplicated().sum()
```

Explanation

- df.info() shows column names, data types, and memory usage.
- df.isnull().sum() counts missing values in each column.
- df.dtypes displays data types.
- df.duplicated() identifies duplicate rows.
- df.duplicated().sum() counts how many duplicates exist.

This ensures customer data is clean before exporting.

2. Removing Duplicate Customers & Saving Customers Table

```
df = df.copy()

df.drop_duplicates(inplace=True)

df.duplicated().sum()

df.to_csv('Customers_Data.csv', index=False)
```

Explanation

- df.copy() prevents warnings by creating a fresh DataFrame.
- drop_duplicates(inplace=True) removes repeated customer entries.
- The cleaned customer table is exported to Customers_Data.csv.

This forms the Customers table for normalization.

3. Creating the Orders Table

```
df = df[['Customer ID','Order ID', 'Order Date', 'Ship Date', 'Ship Mode','Region','State','City']]

df.head()
```

Explanation

This extracts order-level columns:

- Customer ID (FK)
- Order ID (PK)
- Order & Ship Dates
- Ship Mode
- Region, State, City

df.head() previews the data.

4. Converting Dates Into Datetime Format

```
df = df.copy()

df['Order Date'] = pd.to_datetime(df['Order Date'], format='%d/%m/%Y', errors='coerce')

df['Ship Date'] = pd.to_datetime(df['Ship Date'], format='%d/%m/%Y', errors='coerce')
```

Explanation

- Dates are converted from string format (DD/MM/YYYY) into pandas datetime objects.
- errors='coerce' ensures invalid dates become NaT instead of crashing.
- Exported as Order_Data.csv, becoming the Orders table.

5. Creating the Order Details Table

```
df = df[['Order ID', 'Product ID','Sales']]

df = df.copy()

df.drop_duplicates(inplace=True)
```

Explanation

Extracts only line-item level fields:

- Order ID
- Product ID
- Sales amount

drop_duplicates() ensures no repeated order–product pairs.

This forms the Order Details table.

6. Creating the Products Table

```
df = df[['Product ID', 'Category', 'Sub-Category', 'Product Name']]

df.to_csv('Products_Data.csv', index=False)
```

Explanation

Extracts unique product information:

- Product ID (PK)
- Category & Sub-Category
- Product Name

Saved as Products_Data.csv.

3.Excel

Data Cleaning Process Using Power Query in Excel

1. Dataset Overview

The original dataset was divided into four main tables for better organization and relational analysis:

- Order_Data Contains information about customer orders, including:Order ID Order Date
- Ship Date
- Ship Mode
- Customer ID

2.Customer_Data : Contains customer-related information:

- Customer ID
- Customer Name
- Segment (type of customer, e.g. Consumer, Corporate, Home Office)

3.Product_Data: Contains details about the products:

- Product ID
- Category
- Sub-Category
- Product Name

4.Order_Details: Includes transaction-level details that connect orders and products:

- Order ID
- Product ID
- **Sales**

Steps:

- From the Data tab → Get&Transform Data → From Table/Range.
- Loaded each table into Power Query.
- Renamed queries properly for easy management.

3. Data Cleaning Steps (In Detail)

A. Removing Duplicates

- Removed duplicates in each table based on key identifiers (Customer ID, Order ID, Product ID).
- Order_Details duplicates removed based on Order ID + Product ID.

B. Handling Missing and Null Values

- Checked for null or blank values.
- Removed rows missing key fields (Customer ID, Product ID, Order ID).
- Replaced non-critical missing data (Ship Mode, Segment) with “Not Specified”.

C. Data Type Correction

- Order Date&Ship Date → Date type
- Sales → Decimal number
- IDs → Text
- Names/Categories → Text

D. Removing Extra Spaces and Text Cleaning (TRIM&CLEAN)

- Applied Trim and Clean to remove extra spaces and non-printable characters.
- Used "Capitalize Each Word" for text consistency.

Example: " John Doe " → "John Doe".

E. Logical Consistency Checks

- Ensured Order Date<Ship Date.
- Verified that Customer IDs and Product IDs exist in their respective tables.
- Removed invalid or negative sales values.

F. Column Renaming&Formatting

- Renamed columns to clear consistent format (e.g., Order Date → Order_Date).
- Reordered columns logically.

4. Data Integration and Joining

- Merged queries using Merge Queries in Power Query.
- Joined based on Customer ID, Order ID, and Product ID.
- Created a final merged table "Tables_Join" containing cleaned and valid data.

5. Row Count Summary

- Before Cleaning: 9,800 rows
- After Cleaning: 9,577 rows
- Removed Records: 223 invalid or duplicate rows.

6. Final Validation

- Loaded cleaned tables as Connections.
- Created Pivot Tables for analysis.
- Verified sales totals to ensure no valid data was lost.

Phase 2 : Question & Answers

Structure of Each Question and Answer

1. Questions

- Start by clearly stating what you want to find or analyze.
- Examples:
  - "What are the total sales per region?"
  - "Which product category generates the highest revenue?"
  - "What is the average shipping time for each shipping mode?"

2. Explanation

- Briefly describe why this question is important.
- Mention the business or analytical goal.
- Example formula:
  - "We analyze [metric] across [dimension] to understand [purpose/insight]."

3. How to Answer

Divide this into three perspectives:

A) SQL

- Describe the approach in SQL terms without giving code.
- Template:
  - "Using SQL, we [operation] grouped by [dimension] to calculate [metric]."

B) Python

- Describe the Python workflow without code.
- Template:

  "In Python, we merge relevant tables, aggregate by [dimension], and visualize using a [chart type]."

C) Excel

- Describe the Excel approach without code.
- Template:
  - "In Excel, a Pivot Table with [dimension] as rows and [metric] as values allows quick visualization."

Phase 3 : Visualizations (Dashboards)

Tableau

1. Sales : Dashboard Title Sales Analysis

Purpose:This dashboard provides a comprehensive view of sales performance, trends, and distribution across categories, regions, and cities. It is designed to help management monitor revenue, identify high-performing products and regions, and make informed business decisions.

2. Customers : Dashboard Title Customers Analysis

Purpose: The dashboard provides a comprehensive overview of customer activity, segments, and sales performance. It helps identify top customers, understand customer segmentation, and analyze product categories sold. This insight can guide sales strategy, marketing focus, and inventory planning

3. Products : Dashboard Title Products Analysis

Purpose

The primary purpose of this Tableau dashboard is to provide a visual, summarized view of product performance, sales distribution, and volume trends to inform business decisions related to inventory, marketing, and sales strategy.

4. Orders : Dashboard Title Orders Analysis

Purpose

The primary purpose is to **analyze order volume, trends, and distribution** across various dimensions (time, geography, and product category) to assess sales operation efficiency and identify key areas of demand. It helps stakeholders understand **when, where, and what** products are being ordered most frequently.

5. Ship: Dashboard Title Ship Analysis

Purpose

The primary purpose is to **evaluate shipping performance and its impact on sales revenue** by analyzing delivery timeliness across different shipping modes. It provides insights into logistics performance, profitability, and the reliability of various shipping services.

6. Forecasting

Purpose

The dashboard titled **"Superstore sales"** is a time series visualization focused specifically on **forecasting future sales performance**.