**Report for Pharmaceutical Data Project**

1. **Introduction**

The purpose of this report is to document the data cleaning procedures applied to the pharmaceutical dataset to ensure data integrity, consistency, and readiness for analysis. This report provides a record of all steps taken to clean and structure the data, making it reproducible and transparent.

2. **Overview of Dataset**

Dataset Name: Pharm Data

The dataset contains information about pharmaceutical sales transactions, including details on products, distributors, customers, and sales representatives.

3**. Data Cleaning Procedures**

Each step in the cleaning process was necessary to address inconsistencies, remove irrelevant information, and standardize data types. This section outlines each procedure.

3.1 **Ensuring Consistent Data Types**

Objective: Standardize data types across the dataset to ensure compatibility and avoid data type errors in SQL queries.

Process: Each column was assessed for data type consistency.

Result: Columns were confirmed to have consistent data types, reducing errors during analysis.

3.2 **Addressing Special Characters and Formatting**

Objective: Remove or replace special characters that could cause inconsistencies or errors.

Process: Special characters such as ‘�’ and ‘?’ in the City column were removed:

sql

UPDATE "Pharm Data"

SET City = REPLACE(City, '�', '')

WHERE City LIKE '%�%';

UPDATE "Pharm Data"

SET City = REPLACE(City, '?', '')

WHERE City LIKE '%?%';

Result: Special characters were removed, ensuring readability and consistency.

4. **Database Table Setup: Fact and Dimension Tables**

The dataset was structured into fact and dimension tables to support efficient analysis and create a well-organized database schema suitable for an enterprise data warehouse.

4.1 **Fact and Dimension Table Overview**

Fact Table: FactSales

The FactSales table is the primary table that contains the measurable sales transactions. Each row represents a unique sale, with details on quantities, pricing, and total sales.

Columns:

SalesID (Primary Key)

DistributorID (Foreign Key)

CustomerID (Foreign Key)

ProductID (Foreign Key)

DateID (Foreign Key)

SalesRepID (Foreign Key)

Quantity

Price

Sales (calculated as Quantity Price)

Dimension Tables:

1. Distributor:

Stores information about distributors, including city and country details.

Columns include DistributorID, Distributor, Country, and City.

2. Customer:

Contains details about customers, such as name and channels through which they make purchases.

Columns include CustomerID, CustomerName, Channel, and SubChannel.

3. Product:

Stores product details, such as name and classification.

Columns include ProductID, ProductName, and ProductClass.

4. Date:

Contains calendar details to support datebased analysis.

Columns include DateID, Year, Month, and Day.

5. SalesRep:

Stores information about sales representatives, such as name, team, and manager.

Columns include SalesRepID, SalesRepName, Manager, and SalesTeam.

4.2 **Table Creation SQL Statements**

To create these tables, SQL commands were executed in SQLite DB Browser to enforce data integrity through primary and foreign key constraints. Here’s an outline of the SQL commands used:

1. Create FactSales Table:

sql

CREATE TABLE FactSales (

SalesID INTEGER PRIMARY KEY AUTOINCREMENT,

DistributorID INTEGER,

CustomerID INTEGER,

ProductID INTEGER,

DateID INTEGER,

SalesRepID INTEGER,

Quantity INTEGER,

Price REAL,

Sales REAL,

FOREIGN KEY (DistributorID) REFERENCES Distributor(DistributorID),

FOREIGN KEY (CustomerID) REFERENCES DimCustomer(CustomerID),

FOREIGN KEY (ProductID) REFERENCES Product(ProductID),

FOREIGN KEY (DateID) REFERENCES DimDate(DateID),

FOREIGN KEY (SalesRepID) REFERENCES DimSalesRep(SalesRepID)

);

2. Dimension Tables (example for Distributor):

sql

CREATE TABLE Distributor (

DistributorID INTEGER PRIMARY KEY AUTOINCREMENT,

Distributor TEXT,

Country TEXT,

City TEXT

);

Repeat similar CREATE TABLE commands for Customer, Product, Date, and SalesRep, ensuring each primary key is unique and correctly linked to FactSales.

4.3 **Data Insertion and Key Relationships**

Data from the original Pharm Data table was inserted into the fact and dimension tables through SQL joins to populate these tables accurately.

Foreign Key Setup: Ensured integrity between tables by linking foreign keys in FactSales to the primary keys of each dimension table.

Example Insertion Query:

sql

INSERT INTO FactSales (DistributorID, CustomerID, ProductID, DateID, SalesRepID, Quantity, Price, Sales)

SELECT

d.DistributorID,

c.CustomerID,

p.ProductID,

dt.DateID,

sr.SalesRepID,

pd.Quantity,

pd.Price,

pd.Sales

FROM "Pharm Data" pd

JOIN Distributor d ON pd.Distributor = d.Distributor AND pd.Country = d.Country AND pd.City = d.City

JOIN Customer c ON pd."Customer Name" = c.CustomerName AND pd.Channel = c.Channel AND pd."Subchannel" = c.SubChannel

JOIN Product p ON pd."Product Name" = p.ProductName AND pd."Product Class" = p.ProductClass

JOIN Date dt ON pd.Month = dt.Month AND pd.Year = dt.Year

JOIN SalesRep sr ON pd."Name of Sales Rep" = sr.SalesRepName AND pd.Manager = sr.Manager AND pd."Sales Team" = sr.SalesTeam;

5. **Summary**

The initial data cleaning process ensured a high level of data quality by standardizing data types, and eliminating special characters. This clean dataset is now ready for further analysis, and the documented procedures ensure that future modifications can replicate these results if needed.

Fact and Dimension Tables Created: Each table was created to structure the data in a relational format, supporting easy querying and analysis.

Relationships Established: Foreign key relationships between FactSales and dimension tables enforce data integrity and support star schema design.