**ReadMe File for METRO Data Warehouse Project**

**1. Project Overview**

This project implements a near-real-time Data Warehouse (DW) prototype for METRO, a leading superstore chain in Pakistan. The DW integrates transactional data with master data using the MESHJOIN algorithm, which allows for real-time data processing. The project uses a star schema to organize data and supports OLAP queries for advanced business intelligence insights, enabling METRO to optimize marketing strategies, manage inventory, and analyze customer behavior.

**2. Datasets Used**

This project uses three datasets that are provided for the implementation:

1. **transactions.csv** — Contains customer transaction data.
2. **products\_data.csv** — Contains product information. This dataset needs preprocessing to fix issues with supplier IDs, where the same supplierID is associated with different suppliers.
3. **customers\_data.csv —** Contains customer information.

**3. Step-by-Step Instructions**

**Step 1: Preprocess the products\_data.csv File**

Before inserting the data into MySQL, we need to preprocess the products\_data.csv file to ensure that each supplierID is unique for each supplier. This is necessary because there were instances in the dataset where multiple suppliers shared the same supplierID.

1. **Install Python Dependencies:**

* Ensure you have pandas installed using:

*pip install pandas*

**2. Run the Preprocessing Script:**

* On visual code or any other python ide run the preprocessing script:

*python preprocess.py*

* The preprocess.py script will clean and preprocess the products\_data.csv file by:
* Assigning unique supplierID for each supplier.
* Converting the productPrice column to a numeric format by removing dollar signs ($).
* This will generate a cleaned version of the file products\_data\_cleaned.csv ready for import into MySQL.

**Step 2: Create Database in MySQL (metro)**

1. Create a MySQL Database and Tables:

* Open MySQL Workbench or any SQL editor, and run and execute the db1.sql script to create a database called metro and three tables: products, customers, and combined in the metro database.
* The products and customers tables will store master data, while the combined table will temporarily store joined data from both transactional and master data during the MESHJOIN process.

1. Insert Data into the products and customers Tables:
   * Insert data into the products and customers tables from their respective cleaned CSV files either through query or data import wizard in mysql workbench.

Step 3: Implement MESHJOIN Algorithm in Java (Eclipse)

1. Connect MySQL with Eclipse:

* In Eclipse IDE, create a Java project and import the MESHJOIN.java file.
* Add the MySQL JDBC driver to the project build path. You can download the MySQL Connector/J driver from the official MySQL website.

1. Configure Database Connection in Java:

* Open the MESHJOIN.java file and update the database connection settings with the correct credentials:

*private static final String DB\_URL = "jdbc:mysql://localhost:3306/metro";*

*private static Connection conn = null;*

1. Run the MESHJOIN Algorithm:

* The MESHJOIN algorithm will:
  + 1. Load the transactional data (transactions.csv) into the Java program.
    2. Load the master data from the metro database (from products and customers tables).
    3. Join the data based on relevant attributes (e.g., ProductID, customer\_id) and form the combined dataset.
    4. Insert the enriched data into the combined table in MySQL.
* Run the MESHJOIN class from Eclipse:

**Step 4: Create New Database in MySQL (starschema)**

1. Create starschema Database:

* Create a new database called starschema to store the star schema.

1. Design and Create the Star Schema:

* Use the starschema.sql script to design and create the star schema consisting of dimension and fact tables, including dim\_product, dim\_customer, dim\_store, dim\_supplier, dim\_date, and fact\_sales.
* Execute starschema.sql to create the tables.

**Step 5: Load Data into the Star Schema**

1. Copy Data from combined Table to starschema:

* Insert the data from the combined table into the relevant tables in the starschema database. This includes populating the fact table (fact\_sales) and dimension tables (dim\_product, dim\_customer, etc.).

**Step 6: Run OLAP Queries**

* Use MySQL Workbench or any SQL client to execute the OLAP queries from the olap.sql file against the starschema database.

4. Final Steps

**Troubleshooting**

* Database Connection Issues:

If you encounter database connection errors, verify your MySQL credentials and ensure the database is running.

* Missing Libraries in Java:

If there are missing libraries (like the MySQL JDBC driver), make sure to download and add them to your project classpath.

* Data Insertion Issues:

If data insertion fails, check if the source data (metro.combined or products\_data.csv) is correctly formatted and that the dimensions contain unique keys.