DEBRE BIRHAN UNIVERSITY

COLLEGE OF COMPUTING

DEPARTMENT OF SOFTWARE ENGINEERING

COURSE TITLE: FUNDAMENTAL OF BIG DATA ANALYTICS AND BUSINESS INTELIGENCE

INDIVIDUAL ASSIGNMENT

NAME: ***MUSE MEDALCHO***

ID NO: ***DBUR/3720/13***

SUBMITTED TO: ***DERBEW FELASMAN(MSc)***

SUBMITION DATA: ***2/13/2025***

ETL Pipeline Documentation

This documentation outlines the ETL pipeline for the commerce data table based on the assignment requirements and code provided.

Project Objective

The goal of this project is to build an end-to-end data pipeline that extracts data from e-commerce sources, cleans and transforms it, loads it into a PostgreSQL database, and visualizes insights using Power BI.

Data Extraction

The data extraction step involves reading a large e-commerce dataset in CSV format into a Pandas DataFrame.

Code Snippet for Data Extraction:

```
1. import pandas as pd
2.
3. # Get the data path and store it in data
4. data = pd.read_csv(r"C:\Users\Edu\Downloads\BG\BG\dataset\bg.csv")
5.
6. # View the top and bottom rows
7. data.head()
8. data.tail()
9.
```

Data Transformation

Several transformations were applied to clean and prepare the data:

1. Duplicate Removal: Identified and removed duplicate rows.

```
1.    data.duplicated().sum()  # Check for duplicates
2.    data.drop_duplicates(keep='first', inplace=True)  # Remove duplicates
```

2. Missing Data Handling: Missing values were filled with appropriate default values:

```
1.    data['brand'].fillna('unknown', inplace=True)
2.    data['category_id'].fillna(0, inplace=True)
3.    data['category_code'].fillna('unknown', inplace=True)
4.    data['price'].fillna(0, inplace=True)
5.    data['user_id'].fillna(0, inplace=True)
```

After handling missing data:

```
data.isnull().sum()  # Ensure all missing values are addressed
```

Data Loading

The transformed data was loaded into a PostgreSQL database using SQLAlchemy.

Database Credentials:

Username: postgres

Password: password12

Host: localhost

Port: 5432

Database: postgres

Database Connection and Loading:

```
from sqlalchemy import create_engine

# Create database connection
engine = create_engine(f'postgresql://{username}:{password}@{host}:{port}/{db_name}')

# Load data into the PostgreSQL table
data.to_sql('electronics_Table', engine, if_exists='replace', index=False)

# Close the connection
engine.dispose()

```

Data Visualization

The processed data was visualized using Power BI with the following recommended visualizations:

1. Sales Trends Over Time: Line chart showing sales trends using the `price` and `time` columns.
2. Brand Performance: Bar chart showing total sales per brand using the `brand` column.
3. User Behavior: Analyze user purchase frequency using `user_id` and `session`.

For visualization visit the link below

https://app.powerbi.com/links/caGl5GPQ6V?ctid=1695066a-e388-40d1-8ed5-5d0b28ba9f80&pbi_source=linkShare

Design Choices

- Data Cleaning: Duplicates were removed, and missing values were filled with default values to ensure data quality.
- Database Table: All data was stored in the `electronics_Table` in PostgreSQL.
- Security Considerations: In a real-world scenario, passwords should be handled securely using environment variables or encrypted storage.

Conclusion

The ETL pipeline successfully extracted, transformed, and loaded the data into a PostgreSQL database. Power BI dashboards provided meaningful insights to help analyze sales, user behavior, and category performance.