

**DEBRE BERHAN UNIVERSITY**

**COLLAGE OF COMPUTING**

**DEPARTMENT OF SOFTWARE ENGINEERING**

**BIG DATA PROJECT DOCUMENT**

NAME=AMANUEL BEKELE   
ID =DBUR/1227/13

SUBMITION DATE FEB 13/2025  
 SUBMITED TO: DERBEW Felasman(MSc)

Table content:

Contents

[**Data Pipeline Documentation** 2](#_Toc190339499)

[Introduction 2](#_Toc190339500)

[**1. Data Source Identification & Understanding** 2](#_Toc190339501)

[**Key Aspects Considered:** 2](#_Toc190339502)

[**2. Data Extraction** 2](#_Toc190339503)

[**Steps:** 2](#_Toc190339504)

[3. Data Transformation 3](#_Toc190339505)

[Key Transformation Steps: 3](#_Toc190339506)

[4.Database Schema Table Definition 4](#_Toc190339507)

[4. Data Loading into PostgreSQL 4](#_Toc190339508)

[Steps to Load Data: 4](#_Toc190339509)

[5. Data Visualization and Insights in Power BI 5](#_Toc190339510)

[Connecting PostgreSQL to Power BI: 5](#_Toc190339511)

[Visualizations Created: 5](#_Toc190339512)

[Business Insights: 7](#_Toc190339513)

[7. Conclusion 7](#_Toc190339514)

**Data Pipeline Documentation**

## Introduction

This document provides a comprehensive guide to the data pipeline process, covering data extraction, transformation, loading into PostgreSQL, and visualization in Power BI. The dataset used is the **bankdataset** from Kaggle, containing over 1 million rows of transactional data. The objective is to process this data efficiently and derive meaningful insights through visualization.

**1. Data Source Identification & Understanding**

The dataset used for this project is the **bankdataset** obtained from Kaggle. It consists of over 1 million rows of transactional data related to banking activities. Understanding the data source is crucial to ensure data integrity and applicability in further processes.

**Key Aspects Considered:**

* The origin of the dataset (Kaggle)
* The structure of the dataset (columns, data types, and their significance)
* Understanding missing values, duplicates, and outliers
* Identifying data relationships for effective transformation and visualization

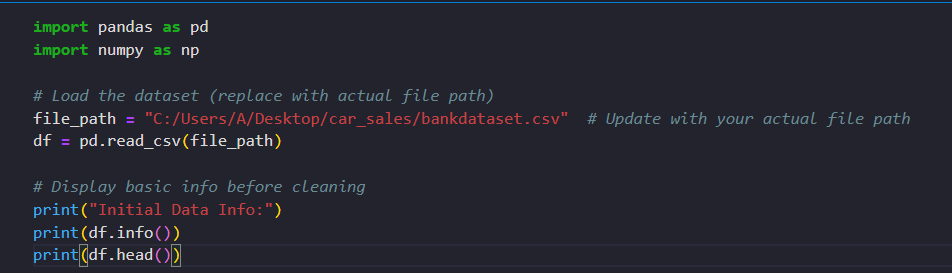
**2. Data Extraction**

Data extraction involves importing the dataset from a CSV file into a pandas DataFrame in Python.

**Steps:**

1. Install necessary libraries: pandas and numpy
2. Load the dataset from the specified file path
3. Display the first few records to verify successful extraction

**Code Implementation:**



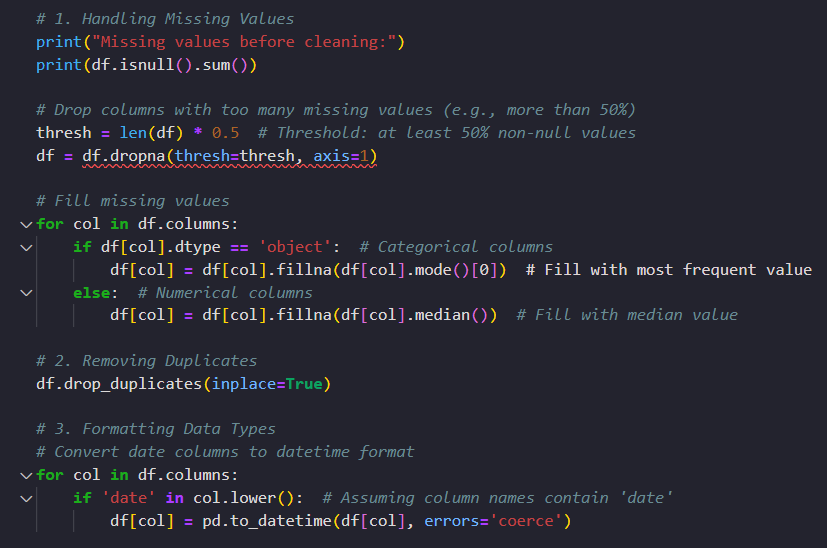
## 3. Data Transformation

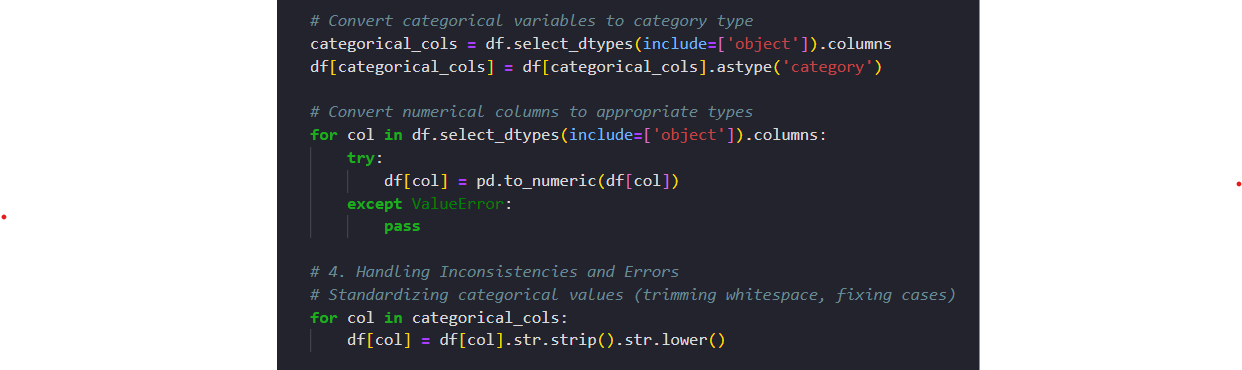
Data transformation ensures data quality before loading it into PostgreSQL. The process involves:

### Key Transformation Steps:

* Handling missing values by imputing or dropping them
* Removing duplicate records
* Formatting data types (e.g., converting date columns)
* Addressing inconsistencies in categorical variables
* Removing outliers using statistical methods like IQR

#### Code Implementation:





## 4.Database Schema Table Definition

Below is the detailed schema definition of the transactions table

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **Column Name** | **Data Type** | **Constraints** | **Description** | | id | SERIAL | PRIMARY KEY | Unique identifier for each transaction | | transaction\_date | DATE | NOT NULL | Date of the transaction | | domain | VARCHAR(50) | NOT NULL | Category of the transaction (e.g., Retail, Medical) | | location | VARCHAR(100) | NOT NULL | Location where the transaction occurred | | value | NUMERIC | NOT NULL | Transaction value amount | | transaction\_count | INT | NOT NULL | Number of transactions for the given entry | |

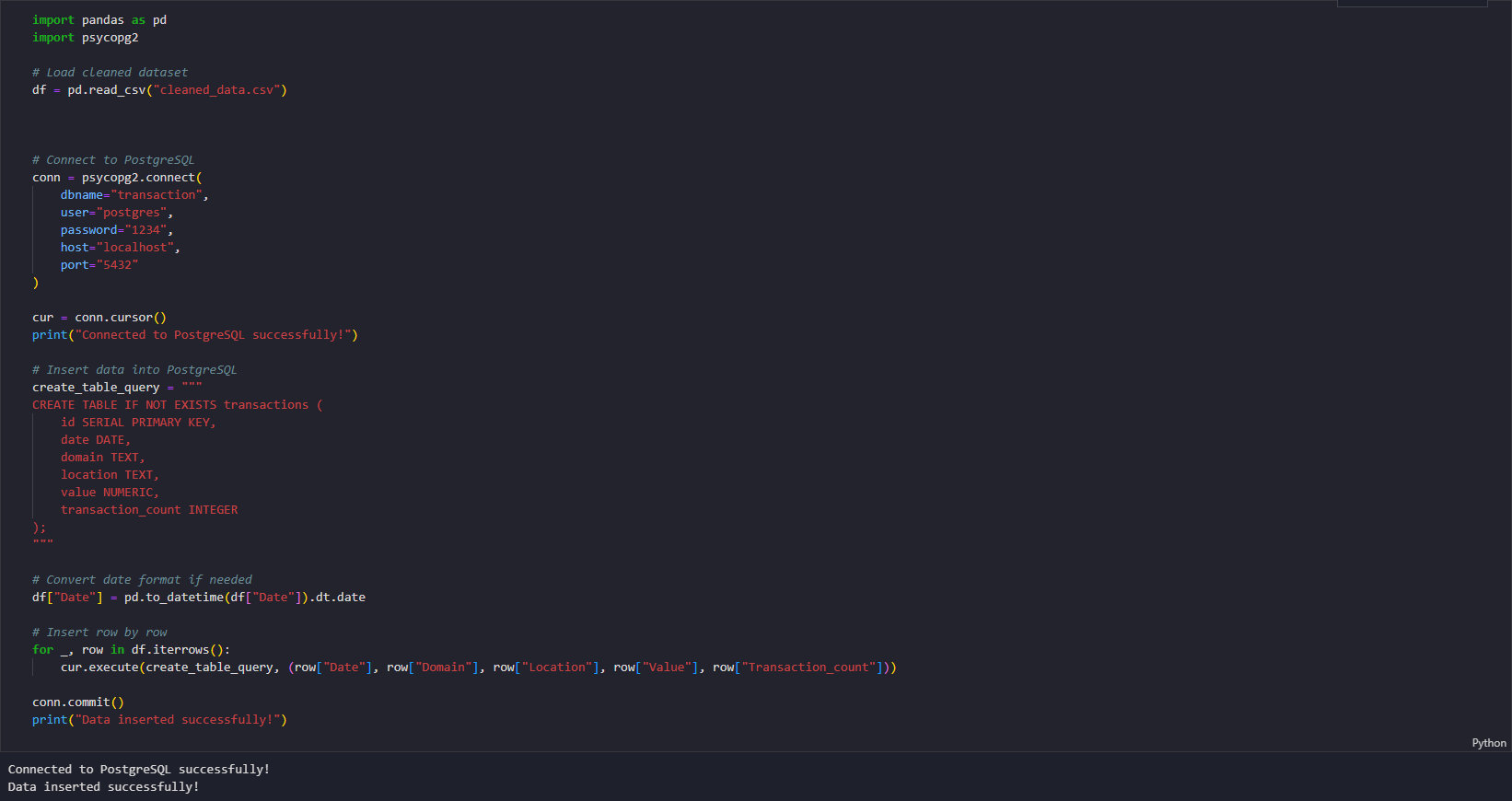
## 4. Data Loading into PostgreSQL

After transforming the dataset, we load it into a PostgreSQL database for further analysis.

### Steps to Load Data:

1. Install and configure PostgreSQL.
2. Establish a connection using psycopg2.
3. Create a table in PostgreSQL.
4. Insert cleaned data into the database.

#### Code Implementation:



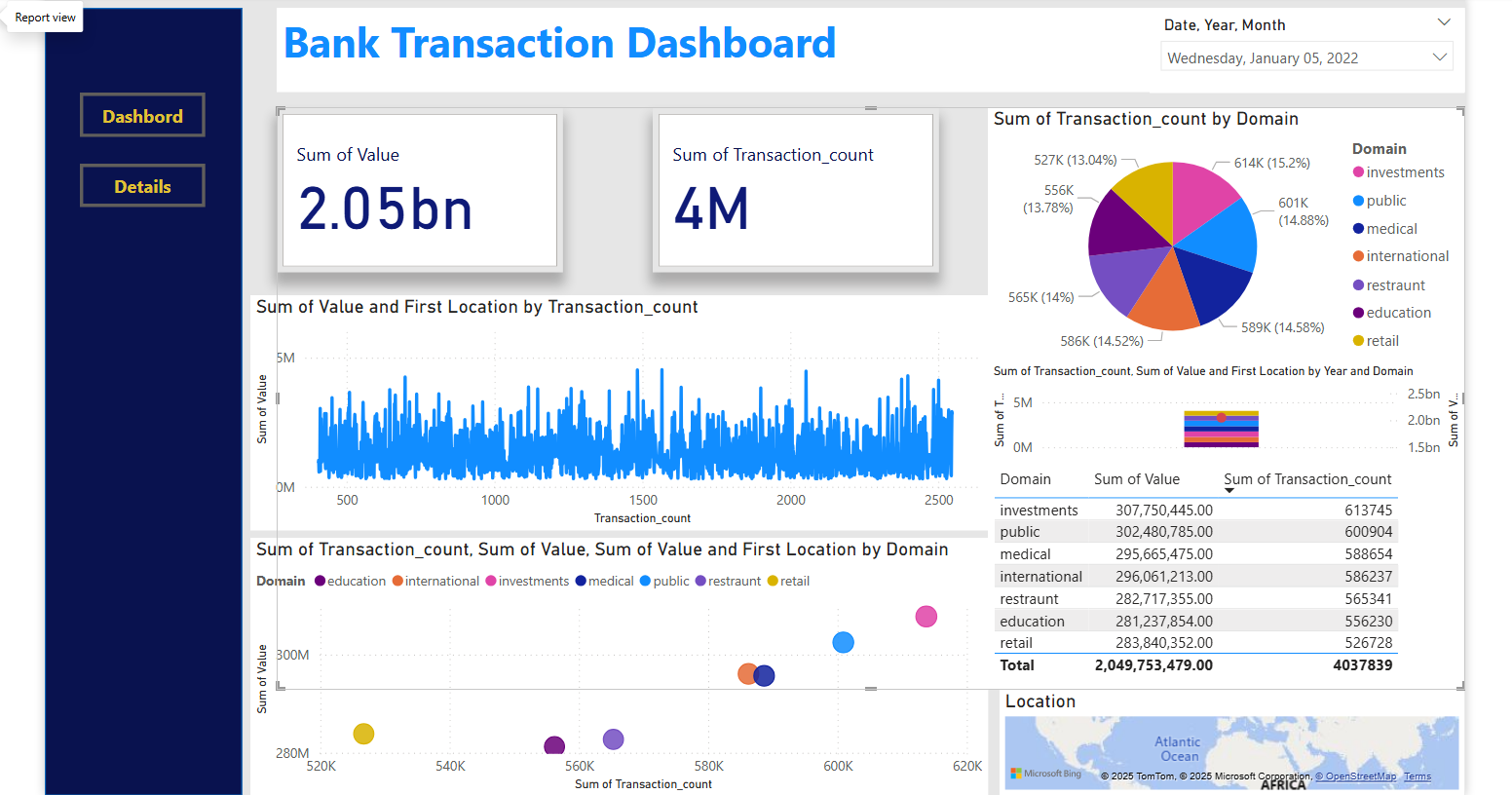
## 5. Data Visualization and Insights in Power BI

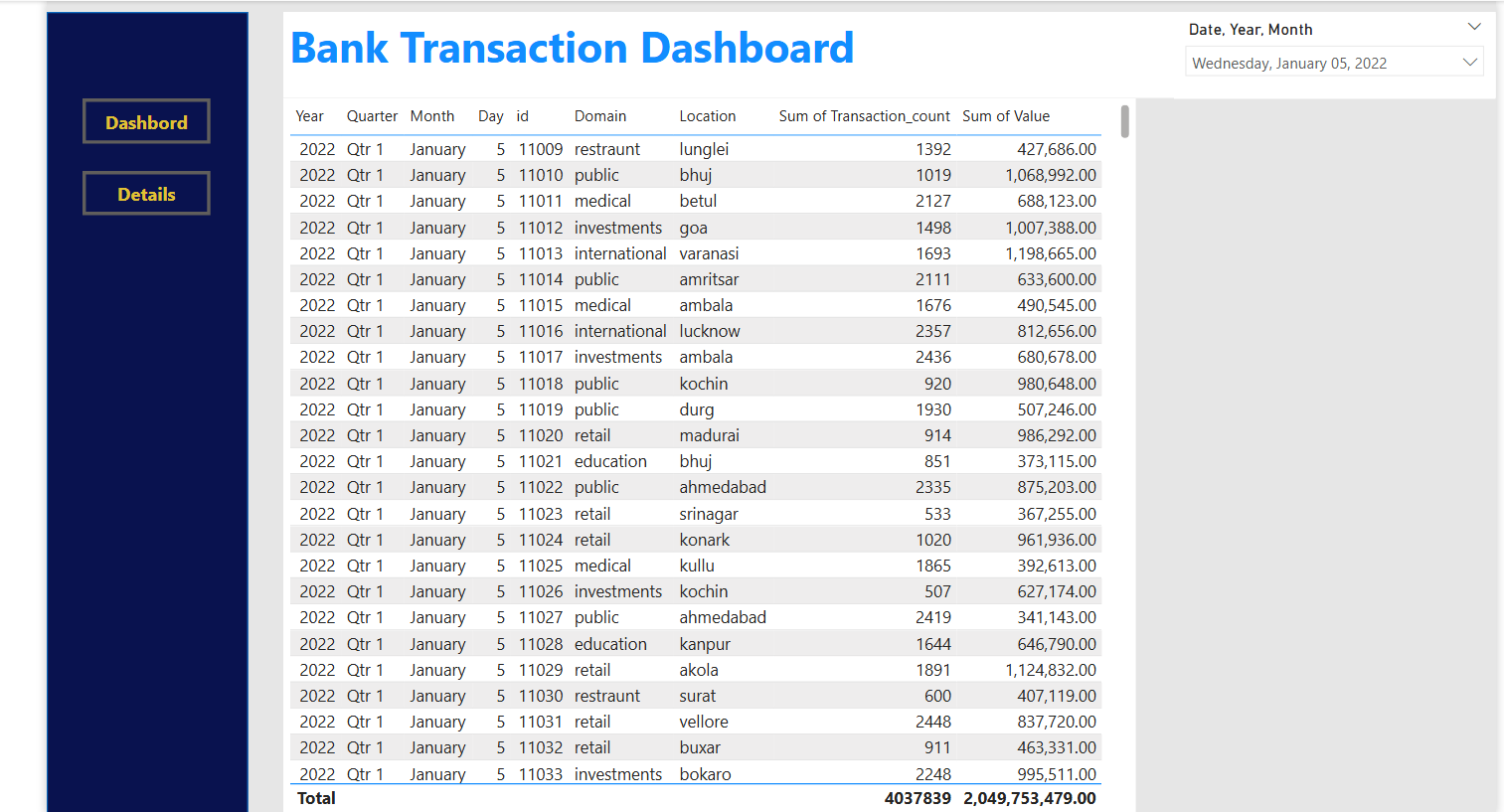
### Connecting PostgreSQL to Power BI:

* Open Power BI and navigate to **Home > Get Data > PostgreSQL Database**.
* Enter database credentials and load the dataset.
* Perform data transformations using Power Query if necessary.

### Visualizations Created:

* **Sales Trends:** Line charts to visualize transaction trends over time.
* **Customer Segmentation:** Pie charts to show the distribution of transaction categories.
* **Geographical Insights:** Map visualizations displaying transaction volume by location.
* **Correlation Analysis:** Scatter plots analyzing relationships between transaction value and count.





### Business Insights:

* Identification of peak transaction periods.
* High-value transaction domains and locations.
* Patterns in transaction counts across different regions and business sectors.

## 7. Conclusion

This documentation outlines the end-to-end data pipeline process, from identifying and understanding the dataset to extracting, transforming, loading, and visualizing it for business insights. The integration of PostgreSQL and Power BI enables efficient data management and analysis, ensuring high-quality decision-making based on transactional trends and patterns. By leveraging proper data cleaning techniques and structured database storage, businesses can gain valuable insights into financial activities, optimize operations, and enhance customer engagement. Future improvements can include automation of the data pipeline and real-time reporting for even more efficient analysis and decision-making.