Excel (Multiple Sheets) to PostgreSQL using Python & Pandas

1. Project Overview

This project demonstrates a real-world ETL-style data ingestion process where a multi-sheet Excel file is automatically loaded into PostgreSQL using Python and Pandas.

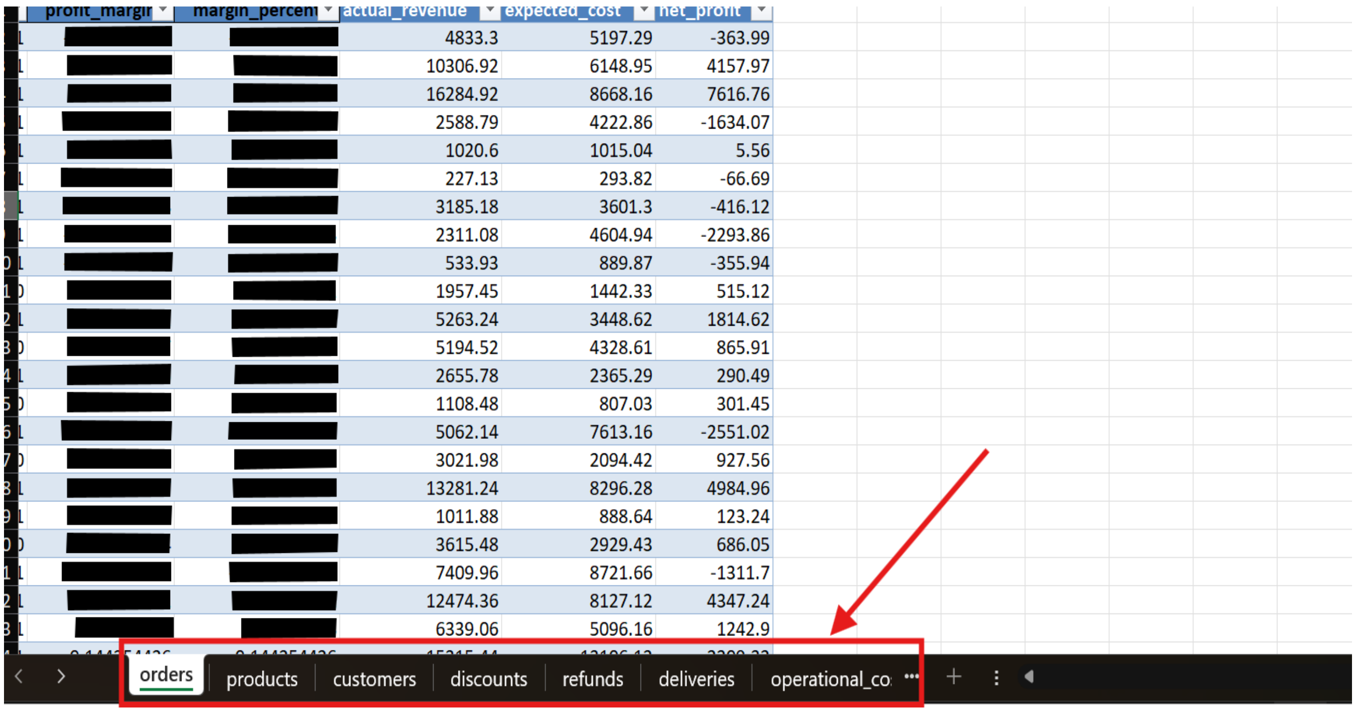
2. Tools & Technologies Used

• Python
• Pandas
• PostgreSQL
• SQLAlchemy
• psycopg2

3. Problem Statement

PostgreSQL does not directly support Excel files. Handling multiple sheets manually using CSV files is inefficient and not scalable.

[Excel file with multiple sheets]



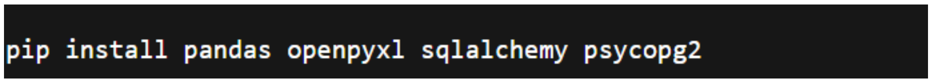
profit_margir	margin_percen	actual_revenue	expected_cost	net_profit
[redacted]	[redacted]	4833.3	5197.29	-363.99
[redacted]	[redacted]	10306.92	6148.95	4157.97
[redacted]	[redacted]	16284.92	8668.16	7616.76
[redacted]	[redacted]	2588.79	4222.86	-1634.07
[redacted]	[redacted]	1020.6	1015.04	5.56
[redacted]	[redacted]	227.13	293.82	-66.69
[redacted]	[redacted]	3185.18	3601.3	-416.12
[redacted]	[redacted]	2311.08	4604.94	-2293.86
[redacted]	[redacted]	533.93	889.87	-355.94
[redacted]	[redacted]	1957.45	1442.33	515.12
[redacted]	[redacted]	5263.24	3448.62	1814.62
[redacted]	[redacted]	5194.52	4328.61	865.91
[redacted]	[redacted]	2655.78	2365.29	290.49
[redacted]	[redacted]	1108.48	807.03	301.45
[redacted]	[redacted]	5062.14	7613.16	-2551.02
[redacted]	[redacted]	3021.98	2094.42	927.56
[redacted]	[redacted]	13281.24	8296.28	4984.96
[redacted]	[redacted]	1011.88	888.64	123.24
[redacted]	[redacted]	3615.48	2929.43	686.05
[redacted]	[redacted]	7409.96	8721.66	-1311.7
[redacted]	[redacted]	12474.36	8127.12	4347.24
[redacted]	[redacted]	6339.06	5096.16	1242.9

4. Solution Approach (Step-by-Step)

Step 1: Install Required Libraries

Libraries such as pandas, openpyxl, sqlalchemy, and psycopg2 were installed for Excel handling and PostgreSQL connectivity.

[Installing required libraries]

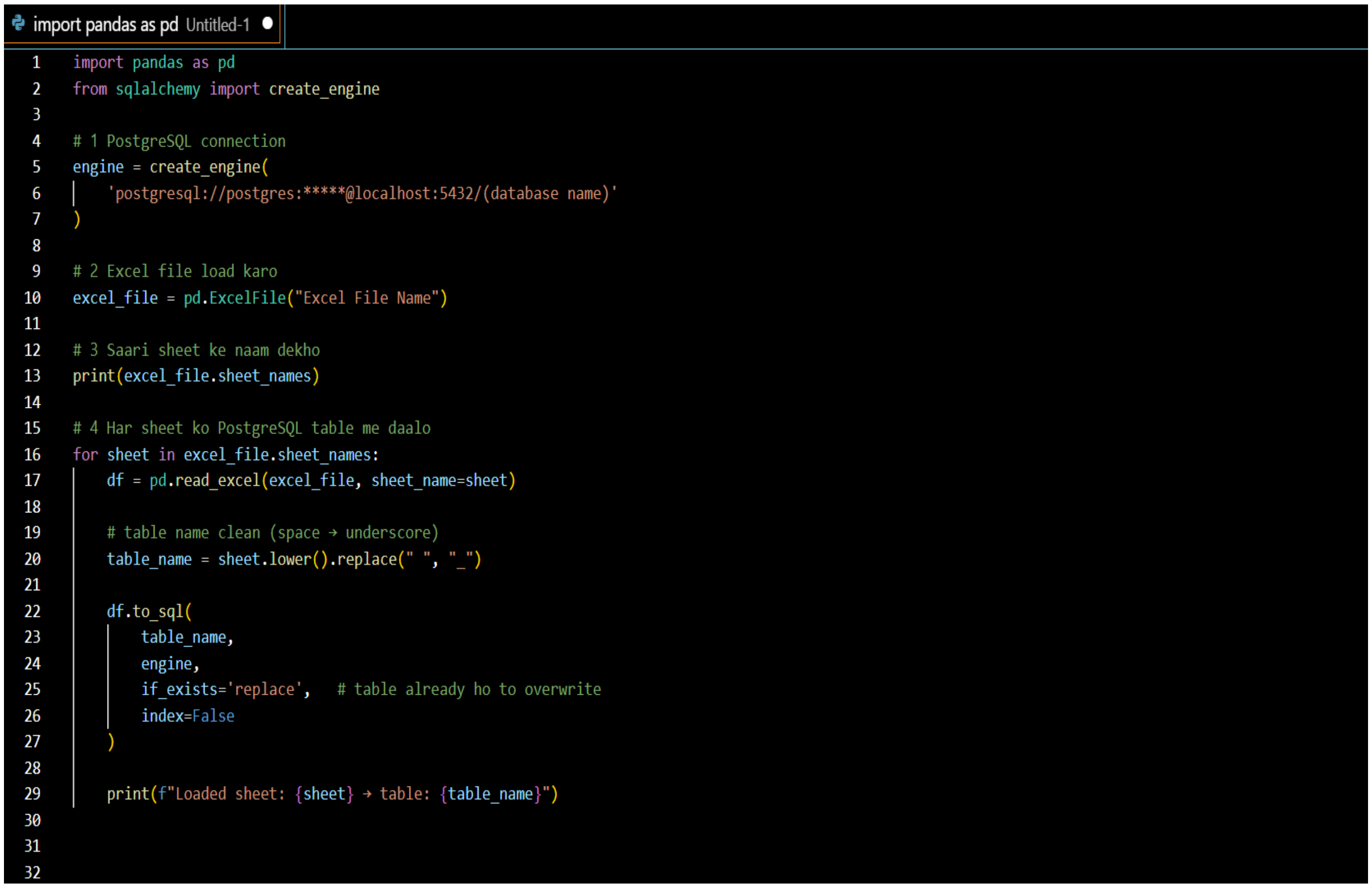


```
pip install pandas openpyxl sqlalchemy psycopg2
```

Step 2: Create PostgreSQL Connection

SQLAlchemy was used to create a connection string to connect Python with PostgreSQL securely.

[PostgreSQL connection string (password hidden)]

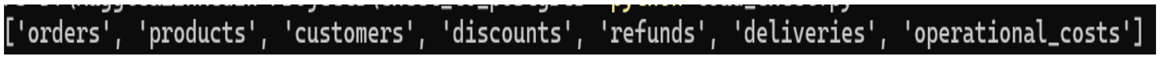


```python
import pandas as pd
from sqlalchemy import create_engine

# 1 PostgreSQL connection
engine = create_engine(
    'postgresql://postgres:*****@localhost:5432/(database name)'
)

# 2 Excel file load karo
excel_file = pd.ExcelFile("Excel File Name")

# 3 Saari sheet ke naam dekho
print(excel_file.sheet_names)

# 4 Har sheet ko PostgreSQL table me daalo
for sheet in excel_file.sheet_names:
    df = pd.read_excel(excel_file, sheet_name=sheet)

    # table name clean (space → underscore)
    table_name = sheet.lower().replace(" ", "_")

    df.to_sql(
        table_name,
        engine,
        if_exists='replace',   # table already ho to overwrite
        index=False
    )

    print(f"Loaded sheet: {sheet} → table: {table_name}")
```

Step 3: Read Excel File

The ExcelFile() function was used to detect all sheet names present in the Excel file.

[Excel sheet names output]

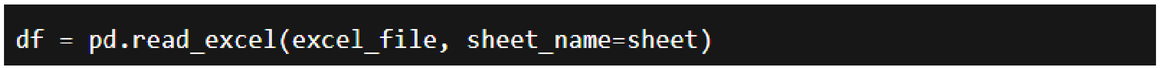


```
['orders', 'products', 'customers', 'discounts', 'refunds', 'deliveries', 'operational_costs']
```

Step 4: Load Data into PostgreSQL

Each sheet was loaded into a Pandas DataFrame using read_excel() and then written to PostgreSQL tables using to_sql().

[Python code using read_excel() and to_sql()]



```python
df = pd.read_excel(excel_file, sheet_name=sheet)
```

5. Errors Faced & Common Issues (Important)

During execution, a PostgreSQL authentication error occurred due to incorrect credentials. This was resolved by verifying the correct username and password using pgAdmin and psql

Error 1: Password Authentication Failed

Cause: Incorrect PostgreSQL username or password.
Solution: Verified credentials using pgAdmin and corrected the connection string.

Error 2: ModuleNotFoundError

Cause: Required Python libraries were not installed.
Solution: Installed missing libraries using pip.

Error 3: Excel File Not Found

Cause: Python script and Excel file were not in the same directory.
Solution: Ensured both files were placed in the same project folder.

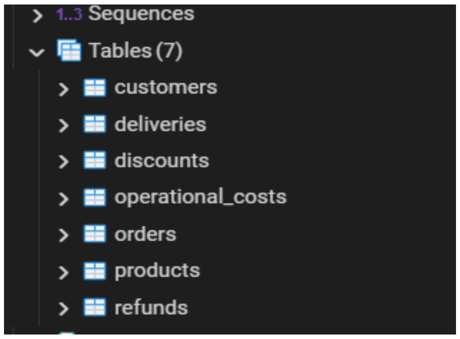
Error 4: Permission Denied or Connection Refused

Cause: PostgreSQL service not running or incorrect port.
Solution: Checked PostgreSQL service status and verified port 5432.

6. Final Output

All Excel sheets were successfully loaded into PostgreSQL as individual tables.

[Tables visible in pgAdmin]



```
> 1..3 Sequences
v Tables (7)
    > customers
    > deliveries
    > discounts
    > operational_costs
    > orders
    > products
    > refunds
```

7. Key Learnings

Practical experience with real-world ETL workflows
- Error handling and debugging database connections
- Automating data ingestion using Python

8. Conclusion

This project reflects a realistic data ingestion scenario commonly faced by data analysts and data engineers in production environments.