



HOW TO USE THIS PROJECT IN YOUR PORTFOLIO

Real Data Analyst Project Kit

Python + PostgreSQL + SQL + Power BI

Introduction

This project demonstrates a complete end-to-end data analytics workflow using Python, PostgreSQL, SQL, and Power BI.

The goal of this project is to show how messy raw data can be transformed into meaningful business insights through data cleaning, database management, SQL analysis, and dashboard development.

This project is valuable for portfolios because it demonstrates multiple real-world data analyst skills in one workflow.

Tools Used

- Python
 - pandas
 - PostgreSQL
 - pgAdmin 4
 - SQLAlchemy
 - SQL
 - Power BI
-

Project Workflow

Raw Dataset

↓

Data Cleaning with Python

↓

Cleaned CSV Export

↓

Upload Data into PostgreSQL

↓

Answer Business Questions with SQL

↓

Build Interactive Power BI Dashboard

Skills Demonstrated

This project demonstrates the following data analytics skills:

Python

- Reading CSV files
 - Data cleaning
 - Handling missing values
 - Removing duplicates
 - Standardizing inconsistent data
 - Working with datetime columns
 - Exporting cleaned data
-

SQL & PostgreSQL

- Creating databases and tables
 - Uploading cleaned data into SQL
 - Writing analytical SQL queries
 - Using aggregate functions
 - Grouping and sorting data
-

Power BI

- Connecting Power BI to PostgreSQL
- Creating DAX measures

- Building KPI cards
 - Creating interactive charts
 - Using slicers and filters
 - Designing dashboards
-

Resume Example

Resume Bullet Point Example

Built an end-to-end data analytics project using Python, PostgreSQL, SQL, and Power BI. Cleaned messy e-commerce data, stored data in a PostgreSQL database, analyzed business metrics using SQL queries, and created an interactive Power BI dashboard.

LinkedIn Project Description Example

Created a real-world data analytics project using Python, PostgreSQL, SQL, and Power BI. The project involved cleaning messy e-commerce data, handling missing values and duplicates, loading the cleaned dataset into a PostgreSQL database, answering business questions using SQL, and developing an interactive dashboard in Power BI.

GitHub Portfolio Description Example

This project demonstrates a complete data analytics workflow using Python, SQL, PostgreSQL, and Power BI.

Key project tasks included:

- Data cleaning with pandas
 - Handling missing values and inconsistent data
 - SQL database integration
 - Business-focused SQL analysis
 - Dashboard development with Power BI
-

Interview Talking Points

When discussing this project in interviews, focus on:

- Why data cleaning matters
 - How inconsistent data affects reporting
 - Why SQL databases are important
 - How Power BI connects to SQL databases
 - The difference between COUNT and DISTINCTCOUNT
 - Business value of dashboards
-

What Makes This Project Valuable

This project is valuable because it demonstrates:

- Technical skills
- Real-world workflows
- Business analysis
- Visualization skills
- Problem-solving ability

Instead of isolated tutorials, this project shows the full process from raw data to dashboard reporting.

Final Thoughts

This project can be used as:

- A portfolio project
- A resume project
- A LinkedIn project
- A GitHub project
- An interview discussion project

Understanding how Python, SQL, PostgreSQL, and Power BI work together provides a strong foundation for data analyst roles.

COMMON_ERRORS_AND_FIXES

Common Errors & Fixes

Python + PostgreSQL + Power BI Project

1. PostgreSQL Connection Error

Error Example

could not connect to server

Possible Causes

- PostgreSQL service is not running
 - Wrong password
 - Wrong server name
 - Wrong port number
-

Fix

- Make sure PostgreSQL is running
- Verify username and password
- Use:
 - Server: localhost
 - Port: 5432
- Restart pgAdmin or PostgreSQL service

2. SQLAlchemy Import Error

Error Example

No module named sqlalchemy

Fix

Install SQLAlchemy using pip:

Python example:

```
pip install sqlalchemy
```

3. psycopg2 Error

Error Example

No module named psycopg2

Fix

Install psycopg2:

```
pip install psycopg2-binary
```

4. Power BI Cannot Connect to PostgreSQL

Possible Causes

- **PostgreSQL not installed correctly**
 - **Wrong credentials**
 - **PostgreSQL connector missing**
-

Fix

- Install PostgreSQL
 - Install pgAdmin 4
 - Verify login credentials
 - Download PostgreSQL connector if prompted by Power BI
-

5. Blank Dates in Power BI

Why This Happens

Some dates in the original dataset were invalid or missing.

When converted in Python:

Example:

```
pd.to_datetime(errors='coerce')
```

Invalid dates became blank datetime values.

Fix

Fill missing dates in Python:

Example:

```
df['order_date'] = df['order_date'].fillna(pd.Timestamp('2024-01-01'))
```

6. Duplicate Orders

Problem

Duplicate order IDs may affect totals.

Fix

Use DISTINCTCOUNT instead of COUNT.

Example:

DAX

```
Total Orders = DISTINCTCOUNT(cleaned_orders[order_id])
```

7. Country Names Not Matching

Problem

Different spellings:

- usa
 - USA
 - United States
-

Fix

Standardize values in Python:

```
df['country'] = df['country'].replace({  
    'usa': 'United States',  
    'USA': 'United States'  
})
```

8. CSV File Not Found

Error Example

FileNotFoundError

Fix

Make sure:

- CSV file exists
 - File path is correct
 - Notebook is running in the correct folder
-

9. Power BI Visuals Not Updating

Fix

Refresh the dataset:

Home → Refresh

10. SQL Table Already Exists

Error Example

relation already exists

Fix Options

Option 1:

Drop the table

```
DROP TABLE cleaned_orders;
```

Option 2:

Use append mode in Python

```
if_exists='append'
```

11. Power BI DAX Measure Errors

Common Cause

Using columns instead of measures.

Fix

Always create measures under:

Modeling → New Measure

Final Advice

Errors are normal in real-world analytics projects.

Learning how to troubleshoot problems is one of the most valuable skills for data analysts.

BONUS_SQL_CHEATSHEET

SQL Cheat Sheet for Beginners

Real Data Analyst Project Kit

SELECT

Used to retrieve data.

SQL

```
SELECT *
```

```
FROM cleaned_orders;
```

SELECT Specific Columns

```
SELECT country, product  
FROM cleaned_orders;
```

WHERE

Used to filter rows.

```
SELECT *  
FROM cleaned_orders  
WHERE country = 'United States';
```

ORDER BY

Used to sort results.

```
SELECT *  
FROM cleaned_orders  
ORDER BY price DESC;
```

COUNT

Counts rows.

```
SELECT COUNT(*)  
FROM cleaned_orders;
```

DISTINCT

Returns unique values.

```
SELECT DISTINCT country  
FROM cleaned_orders;
```

SUM

Adds numerical values.

```
SELECT SUM(price)
FROM cleaned_orders;
```

GROUP BY

Groups rows together.

```
SELECT country,
       SUM(price * quantity) AS total_sales
FROM cleaned_orders
GROUP BY country;
```

ORDER BY DESC

Sort highest to lowest.

```
SELECT country,
       SUM(price * quantity) AS total_sales
FROM cleaned_orders
GROUP BY country
ORDER BY total_sales DESC;
```

LIMIT

Limits number of rows returned.

```
SELECT *
FROM cleaned_orders
LIMIT 10;
```

COUNT DISTINCT

Counts unique values.

```
SELECT COUNT(DISTINCT order_id)
FROM cleaned_orders;
```

Monthly Sales Trend

```
SELECT
  DATE_TRUNC('month', order_date) AS month,
  SUM(price * quantity) AS monthly_sales
FROM cleaned_orders
GROUP BY month
ORDER BY month;
```

Top Products Sold

```
SELECT
  product,
  SUM(quantity) AS total_products_sold
FROM cleaned_orders
GROUP BY product
ORDER BY total_products_sold DESC;
```

Total Revenue

```
SELECT
  SUM(price * quantity) AS total_revenue
FROM cleaned_orders;
```

Total Orders by Country

```
SELECT
  country,
  COUNT(*) AS total_orders
FROM cleaned_orders
GROUP BY country
ORDER BY total_orders DESC;
```

Helpful SQL Keywords

Keyword	Purpose
---------	---------

SELECT	Retrieve data
--------	---------------

FROM	Choose table
------	--------------

Keyword	Purpose
---------	---------

WHERE	Filter rows
-------	-------------

GROUP BY	Group data
----------	------------

ORDER BY	Sort data
----------	-----------

SUM	Add values
-----	------------

COUNT	Count rows
-------	------------

DISTINCT	Unique values
----------	---------------

LIMIT	Limit rows
-------	------------

Final Thoughts

SQL is one of the most important skills for data analysts.

Understanding how to query, group, filter, and summarize data is essential for business reporting and analytics.

Thank you for supporting,

