

Project Report: Synthetic Data Generation and PostgreSQL Cloud Deployment

Project Title: DLF

Author: Kumaresan K

Project Duration: October 13, 2025 – October 16, 2025

1. Executive Summary

This report outlines a project focused on building a data pipeline for a system named 'DLF'. The project successfully implemented a workflow for generating realistic, synthetic data using Python's Faker library, storing it in a PostgreSQL database, and deploying the entire system on the Render cloud platform for online accessibility. This established an efficient method for creating large-scale test datasets for development and analytics without using sensitive user information.

2. Project Objective

The primary goal was to create, store, and deploy a data-driven application. The key objectives included:

- Generating realistic sample data for testing and analysis.
- Integrating the generated data into a PostgreSQL database managed with pgAdmin.
- Deploying the database and application to the Render cloud platform for scalable and accessible hosting.

3. Methodology and Implementation

The project was executed in three distinct phases:

- Phase 1: Synthetic Data Generation

To create realistic test data, the Faker library in Python was utilized. This library generated various data points, including names, email addresses, physical addresses, phone numbers, and company details. The psycopg2 library served as the PostgreSQL adapter for Python, enabling a direct connection between the script and the database.

Sample Python code snippet:

```
from faker import Faker
import psycopg2

# Initialize Faker and create sample data
fake = Faker()
data = [(fake.name(), fake.email(), fake.address()) for _ in range(100)]

# Establish database connection
connection = psycopg2.connect(
    host='localhost',
    database='dlf_db',
    user='postgres',
    password='yourpassword'
)
cursor = connection.cursor()

# Execute bulk insert and commit changes
cursor.executemany('INSERT INTO users (name, email, address) VALUES
(%s, %s, %s);', data)
connection.commit()

# Close resources
cursor.close()
connection.close()
```

[Code adapted from source 17-26]

- Phase 2: Database Configuration

Using the pgAdmin management tool, a new PostgreSQL database named dlf_db was created. Tables were structured to accommodate the synthetic data, which was then successfully inserted via the Python script.

- Phase 3: Cloud Deployment

The entire project, including the PostgreSQL database and the Python application, was deployed on Render. Render was chosen for its ability to provide a reliable, scalable, and easily managed hosting environment, simplifying both the initial deployment and long-term maintenance.

4. Results and Conclusion

The project successfully achieved all its objectives. A complete workflow for generating, storing, and deploying a data-driven application was established. The use of Faker proved highly effective for creating sample data, while PostgreSQL and Render provided a robust and scalable backend and hosting solution. The final system performs efficiently, with realistic data readily available for analytics and testing purposes.

5. Technology Stack

Programming Language: Python

Data Generation: Faker Library

Database: PostgreSQL

Database Adapter: psycopg2

Database Management: pgAdmin

Cloud Platform: Render