ETL SQL Pipeline Project Report

1. Introduction

The objective of this project is to simulate a simple ETL (Extract, Transform, Load) pipeline entirely using PostgreSQL and pgAdmin4. This demonstrates how raw data can be imported, cleaned, transformed, tracked, and exported using only SQL scripts, triggers, and basic PostgreSQL features.

2. Abstract

In modern data engineering, ETL pipelines automate the flow of raw data from various sources into cleaned, production-ready tables. This project demonstrates a small-scale ETL workflow:
- Load raw CSV files into a staging table.
- Clean the data by removing invalid records and duplicates.
- Transform valid data and insert it into the final production table.
- Maintain an audit log table to track inserted records.
- Automate staging cleanup using a trigger function.
- Export the final, cleaned tables for downstream usage.

3. Tools Used

- Database: PostgreSQL (tested with pgAdmin4)
- Scripting: SQL DDL & DML scripts
- File Format: CSV for input/output

4. Steps Involved

1) Create Tables :

Three main tables were created:
- staging_employees (for raw data)
- employees (cleaned, production data)
- etl_audit_log (logs for tracking loads)

2) Import Raw Data :

Two CSV files (employees_raw.csv and employees_raw_2.csv) are loaded into staging_employees using COPY.

3) Clean & Transform Data :

Invalid rows (with null IDs or names) and duplicates are removed. Remaining valid rows are type-cast and inserted into employees.

4) Track ETL in Audit Log :
After data is loaded, the row count is inserted into etl_audit_log to maintain a basic history of loads.

5) Automated Cleanup :
An AFTER INSERT trigger on the etl_audit_log table runs a function clean_staging() which deletes all rows in staging_employees automatically after each successful load.

6) Export Final Tables :
The cleaned employees table and etl_audit_log are exported to CSV files for external usage or reporting.

Note: Make sure the folder exists: 'D:\ETL_sql_project_file\' for output, Because the PostgreSQL server process (the Postgres service) does not have permission to write/create to that folder.

5. Conclusion

This ETL pipeline simulation shows how SQL alone can implement a basic data workflow without extra tools. While production systems use advanced schedulers or ETL frameworks, this project demonstrates the fundamentals:
- Using staging areas to handle raw data
- Validating and transforming records
- Tracking pipeline activities with audit logs
- Automating repetitive tasks with triggers
- Exporting final, ready-to-use datasets

This project provides a strong foundation for building more complex pipelines with more advanced scheduling and orchestration tools in the future.

Prepared by: Darji Chintankumar Dineshchandra

Date: 2025-07-16