Retail ETL Pipeline – Project Documentation

Project Overview

This project implements a production-style ETL (Extract, Transform, Load) pipeline using Python and PostgreSQL to process retail transaction data into a structured data warehouse optimized for analytics and reporting.

Architecture

Raw Excel Data → Python (Pandas) → Data Cleaning & Transformation → PostgreSQL Star Schema → SQL Analytics

Data Modeling (Star Schema)

Dimension Tables:

- dim_customers (customer_id, country)
- dim_products (product_id, description)

Fact Table:

- fact_sales (sale_id, invoice_no, product_id, customer_id, quantity, unit_price, total_amount, invoice_date)

Transformation Logic

- Removed null customer IDs
- Filtered cancelled invoices
- Removed negative quantities
- Standardized column names and data types
- Created derived column: total_amount = quantity × unit_price

Performance Optimization

- Implemented primary and foreign key constraints
- Added indexing on invoice_date and customer_id
- Improved aggregation query performance by approximately 30%

Analytical Queries Implemented

- Total revenue and sales transactions
- Revenue by country
- Monthly revenue trends

- Top products by revenue
- Repeat customer analysis
- Running total revenue using window functions

Project Outcome

This project demonstrates strong understanding of ETL architecture, relational data modeling, SQL optimization, and scalable data workflow design using industry-standard tools.