# Project Report: Global Retail Data Integration & Analytics Pipeline

## 1. Executive Summary

This report details the architecture and implementation of a high-performance **ETL (Extract, Transform, Load)** pipeline designed for the processing and analysis of large-scale retail transaction datasets. The solution processes over **1.06 million records**, transforming raw, unstructured CSV data into a refined, audit-logged Data Warehouse on **Oracle XE**. The project concludes with a real-time **Streamlit** executive dashboard for data-driven decision-making.

## 2. Project Architecture Overview

The system follows a professional **Medallion Architecture** (Staging-to-Fact) to ensure data quality and system scalability.

**The architecture consists of four distinct phases:**

1. **Extraction & Sanitization:** Python-based ingestion from the Landing Zone.
2. **High-Speed Ingestion:** Optimized bulk loading into Oracle Staging.
3. **Database Transformation:** PL/SQL-driven business logic application.
4. **Analytical Visualization:** Real-time KPI reporting via Streamlit.

## 3. Detailed Component Analysis

### 3.1 Data Extraction & Orchestration (Python Layer)

- **Orchestrator:** main_etl.py manages the end-to-end workflow.
- **Preprocessing:** Utilizes **Pandas** to handle missing values (NaN), perform type casting, and sanitize string fields to prevent database injection errors (e.g., DPY-3013).
- **Performance Tuning:** Implemented the executemany method for **Bulk Loading**, enabling the ingestion of 1M+ rows in under 60 seconds by reducing network round-trips to the database.

### 3.2 Data Warehousing (Oracle Layer)

- **Staging Layer (STG_RETAIL_DATA):** A transient landing table that mirrors the source file structure, designed for maximum ingestion speed.
- **Production Layer (FCT_RETAIL_SALES):** A strictly typed Fact table containing calculated fields (e.g., Total Sales) and validated records.
- **Data Governance:** An ETL_LOG_AUDIT table tracks metadata for every run, including record counts, timestamps, and job statuses.

### 3.3 Procedural Transformation (PL/SQL Layer)

A specialized stored procedure, SP_TRANSFORM_RETAIL_DATA, handles the heavy lifting inside the database:

- **Business Logic:** Calculates Revenue ($Quantity \times Price$).
- **Data Validation:** Filters out records with null Customer IDs or non-positive quantities.
- **Atomicity:** Ensures that the entire batch succeeds or fails together (ACID compliance).

## 4. Analytical Insights (Frontend Layer)

The project provides an executive dashboard built with **Streamlit** and **Plotly**, connecting directly to Oracle database views.

**Key Dashboard Metrics:**

- **Revenue Analysis:** Interactive bar charts showing the Top 10 countries by sales.
- **Order Distribution:** Pie charts visualizing market penetration.
- **Operational Health:** Direct visibility into the ETL_LOG_AUDIT table to monitor pipeline performance.

## 5. Technical Challenges & Solutions

- **Issue:** Data type mismatch (float vs varchar) during ingestion of 1M+ rows.
- **Solution:** Integrated a transformation layer in Python using .astype(str) and .fillna() to ensure data consistency before reaching the Oracle driver.
- **Issue:** Resource constraints in Oracle XE during high-volume inserts.
- **Solution:** Optimized batch sizes to 50,000 records per transaction, balancing memory usage and insert speed.

## 6. Conclusion

The **Global Retail Data Integration** project successfully demonstrates a scalable, enterprise-grade data pipeline. By combining the flexibility of Python with the robust processing power of Oracle PL/SQL, the system delivers a stable and auditable flow of information from raw files to business-ready insights.