SQL ETL Pipeline Simulation Report

Abstract

This project simulates an **ETL (Extract, Transform, Load)** process using SQL Server to showcase how raw data can be cleaned, transformed, and loaded into a structured star schema for analytical purposes. Using the Online Retail dataset, the project demonstrates importing data from CSV files into raw tables, applying cleaning and deduplication logic in staging tables, and transforming it into production-ready **dimension and fact tables**. To ensure reliability, an **audit log** is created to monitor data insertions, and triggers are implemented to automate logging. This project illustrates the fundamentals of building scalable and transparent ETL pipelines suitable for real-world data warehousing.

Introduction

In modern businesses, handling large volumes of raw data requires systematic processing pipelines to convert data into actionable insights. ETL processes play a crucial role in this by **extracting data from sources, transforming it into clean structured formats, and loading it into target systems** for reporting and analytics.

The **Online Retail dataset** is used here to simulate such a pipeline within SQL Server. The project follows a **data warehouse design approach (star schema)** by building dimension tables for master data and a fact table for transactional records. It also integrates **data auditing and automation** mechanisms, making the pipeline both robust and reliable.

Tools Used

- **SQL Server**: For database creation, transformations, and schema design.
- **Online Retail CSV dataset**: As the raw data source.
- **SQL Server Management Studio / DB Browser**: For executing SQL queries and database interactions.
- **Windows Environment**: Used for bulk CSV import and file handling.

Steps Involved in Building the Project

1. **Raw Data Import**
    - Created a Raw_OnlineRetail table with all fields as NVARCHAR.
    - Imported CSV file using BULK INSERT.
2. **Staging Area Creation**
    - Designed Staging_OnlineRetail with proper data types (INT, DECIMAL, DATETIME).
    - Converted raw data using TRY_CAST to handle invalid formats.
    - Removed **duplicates** with ROW_NUMBER() and eliminated **rows with nulls** in critical fields.

3. **Data Transformation & Dimension Creation**
    - **DimCustomer** → Stored unique customers with latest country information.
    - **DimProduct** → Stored unique stock codes with most frequent description.
    - **DimDate** → Generated full calendar table (DateKey in YYYYMMDD format) for all invoice dates.
4. **Fact Table Creation**
    - **FactSales** → Captures transactional details (Invoice, CustomerKey, ProductKey, DateKey, Quantity, UnitPrice, TotalAmount).
    - Established foreign key relationships to dimension tables.
5. **Audit Logging**
    - Built an AuditLog table to track number of rows inserted into FactSales.
    - Implemented trigger trg_Audit_FactSales to **auto-log** every new data insertion.
6. **Final Deliverables**
    - Clean **Fact and Dimension tables**.
    - **Audit Log** with ETL tracking.
    - SQL scripts covering the complete ETL pipeline.

Conclusion

The SQL ETL Pipeline successfully demonstrates how data can be transformed from raw CSV files into a structured star schema, ensuring clean, reliable, and consistent datasets. The use of staging tables, audit logs, and triggers illustrates best practices in ETL design.

This project provides a foundational framework for data warehousing and can be extended for advanced analytics, reporting, or integration with BI tools. It highlights the importance of **data quality, transparency, and automation** in building scalable data pipelines.