SQL ETL Pipeline Simulation — Project Overview

1. Introduction

The SQL ETL Pipeline Simulation project demonstrates how raw data can be systematically collected, cleaned,transformed, and stored into a final production-ready database table. ETL (Extract, Transform, Load) is a coreconcept in data engineering and analytics.This project shows how an ETL workflow can be built entirely using SQL, using staging tables, cleaned tables, a production table, and audit logging. It is ideal for beginners who want tounderstand data pipeline concepts without using external ETL tools.

2. Abstract

This project simulates a complete ETL pipeline using SQL and a CSV dataset.Raw data is imported into a staging area, cleaned and validated in a transformation layer, and finally loaded into a production table for reporting and analytics.An audit log table tracks ETL operations, improving transparency. The pipeline also includes an optional SQL trigger to automate cleanup of staging data after loading.

The ETL pipeline ensures:

- Clean, consistent data
- Accurate transformations
- Proper historical tracking
- Automation of repetitive tasks

This project helps understand how real-world data pipelines are structured inside organizations.

3. Tools Used

Software

MySQL – Database engine

DB Browser for SQLite / DBeaver – GUI tool to run SQL scripts and import CSV

Techniques Used

- SQL DDL (CREATE TABLE)
- SQL DML (INSERT, SELECT)
- Data Cleaning (COALESCE, TRIM, DATE conversion)
- Creating derived metrics (total_amount)
- Audit logging
- SQL Triggers

4. Steps Involved in Building the Project

Step 1: Extract — Import Raw Data
A dataset (CSV file) is imported into a staging table (staging_sales).
No cleaning is performed at this step.
This simulates the landing zone of the ETL workflow.

Step 2: Transform — Clean and Process the Data

Data is cleaned inside SQL using:

- TRIM() → remove extra spaces
- COALESCE() → replace missing values
- date() → convert date formats
- Filtering out empty or invalid records
- Cleaned results are stored in cleaned_sales.

Step 3: Load — Load Data into Production Table

A new production table (production_sales) is created with:

- ➢ Cleaned values
- ➢ Proper data types
- ➢ Additional calculated field:
  total_amount = quantity * price
- ➢ Data is inserted from cleaned table to production.

Step 4: Create ETL Audit Log

An audit table (etl_audit_log) records:

- ETL process name
- Rows inserted, Run timestamp
- This ensures ETL transparency and traceability.

Step 5: Automation with SQL Trigger

A trigger automatically deletes data from staging after production insertion:

- ➢ Prevents duplication
- ➢ Keeps staging clean
- ➢ Simulates real automated pipelines

5. Conclusion

This SQL ETL Pipeline Simulation project replicates the architecture of real-world data engineering workflows using only SQL. It demonstrates how raw data flows through different layers—staging, transformation, and production—while ensuring quality, traceability, and automation.

The project builds strong understanding of:

ETL concepts

SQL data cleaning

Data pipeline architecture

Audit logging and triggers