**SQL ETL Pipeline Simulation – Project Report**

Student Name: M. Neshvitha Vani

Project Duration:2 Weeks

Technology used: MySQL Workbench

GitHub repository: **https://github.com/Neshvitha/SQL\_FinalProject-etl\_pipeline\_simulation**

LinkedIn: https://www.linkedin.com/in/neshvitha-medavarapu-2a1174265

Email: neshvitha18@gmail.com

# Abstract

This project simulates an ETL (Extract, Transform, Load) pipeline using SQL in MySQL Workbench. The objective is to demonstrate how raw CSV data can be imported into staging tables, cleaned, transformed, and moved into production tables with audit tracking and automation using triggers. The final output is exported as cleaned production tables and audit logs.

# Introduction

ETL (Extract, Transform, Load) is a key process in data engineering and analytics. It involves extracting raw data, transforming it into meaningful information, and loading it into a structured environment for reporting or further analysis. This project provides a beginner-friendly simulation of the ETL pipeline using SQL in MySQL Workbench.

# Methodology

The ETL pipeline was implemented in six steps:

## Step 1: Setup Schema

Created a schema named 'etl\_project' and staging tables (stg\_customers, stg\_orders). Staging tables are temporary storage for raw imported data.

## Step 2: Import Raw Data

CSV files (customers.csv, orders.csv) were imported into staging tables using MySQL Workbench Table Data Import Wizard.

## Step 3: Clean and Transform Data

Data cleaning was performed to remove duplicates, handle null values, and standardize inconsistent entries. The cleaned data was inserted into production tables (prod\_customers, prod\_orders).

## Step 4: Audit Logging

An audit\_log table was created to keep track of insert operations. This ensures transparency and accountability of ETL operations.

## Step 5: Triggers for Automation

Triggers were implemented to automatically handle missing values and to log changes after inserts into production tables.

## Step 6: Export Final Tables

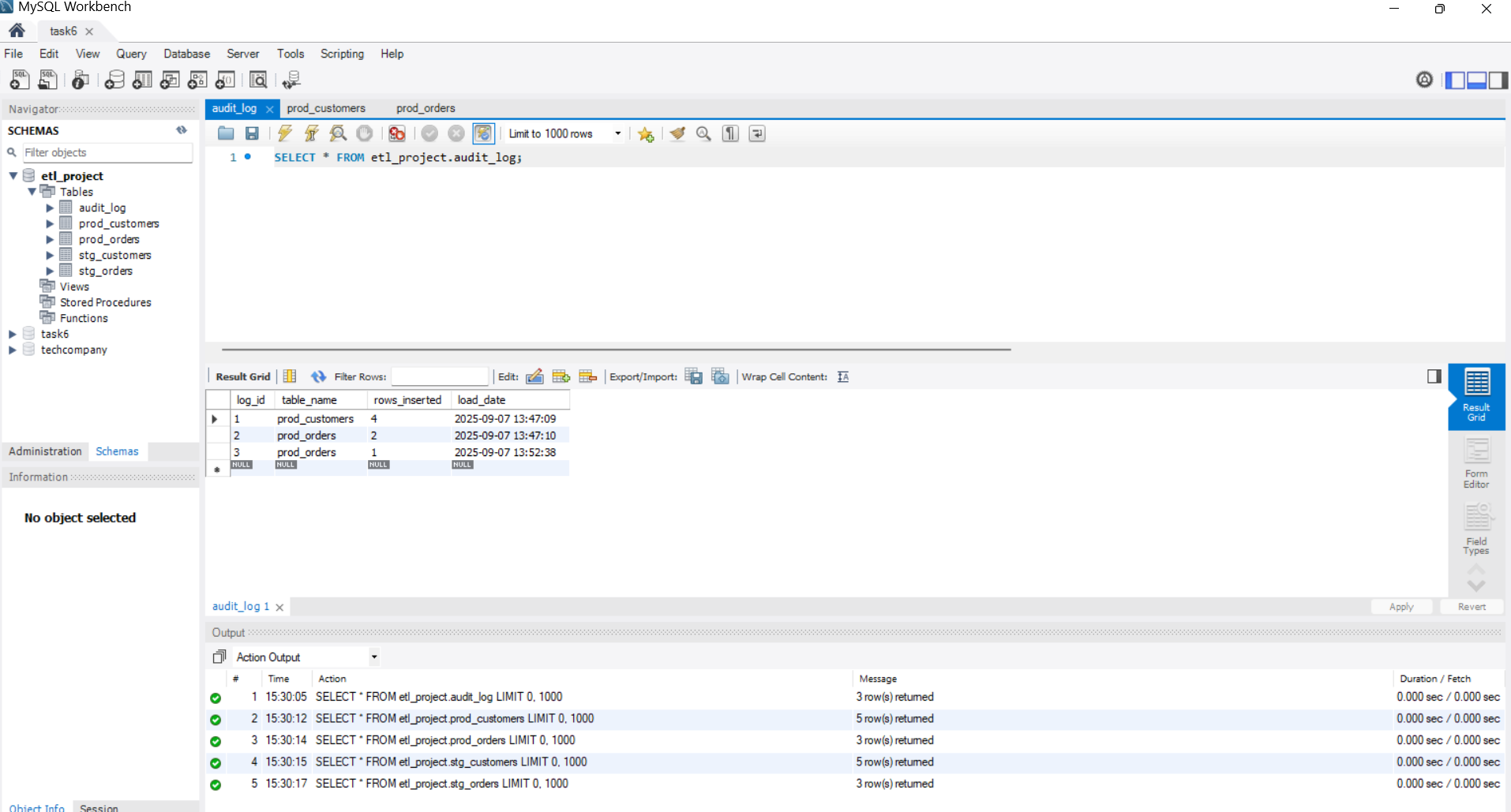
The final cleaned production tables and audit log were exported into CSV files for reporting and submission.

# Results

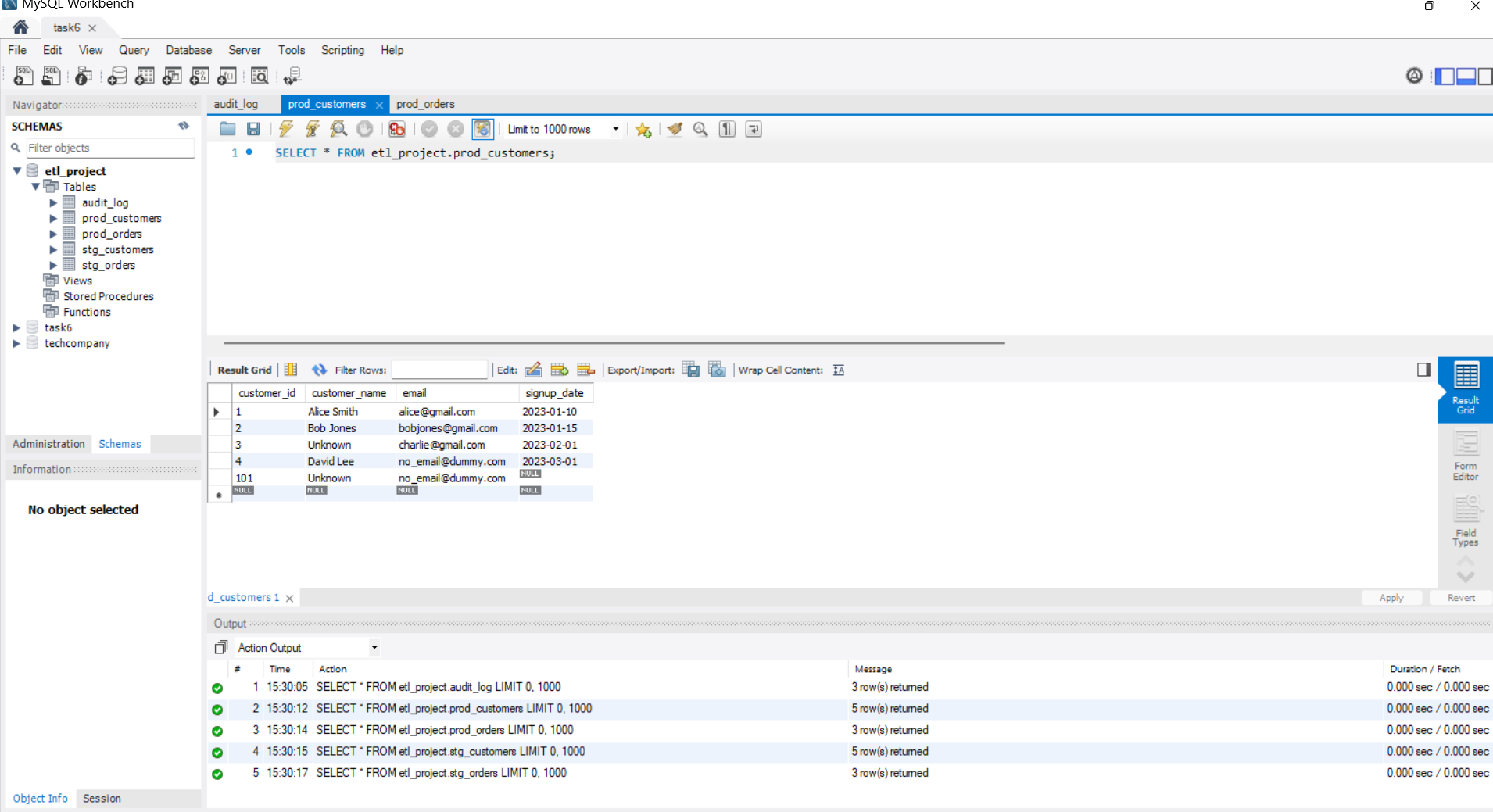
The ETL pipeline successfully cleaned and transformed raw data into production tables. Audit logs recorded the number of rows inserted. Final results include prod\_customers, prod\_orders, and audit\_log tables.

**screenshots of final tables here**

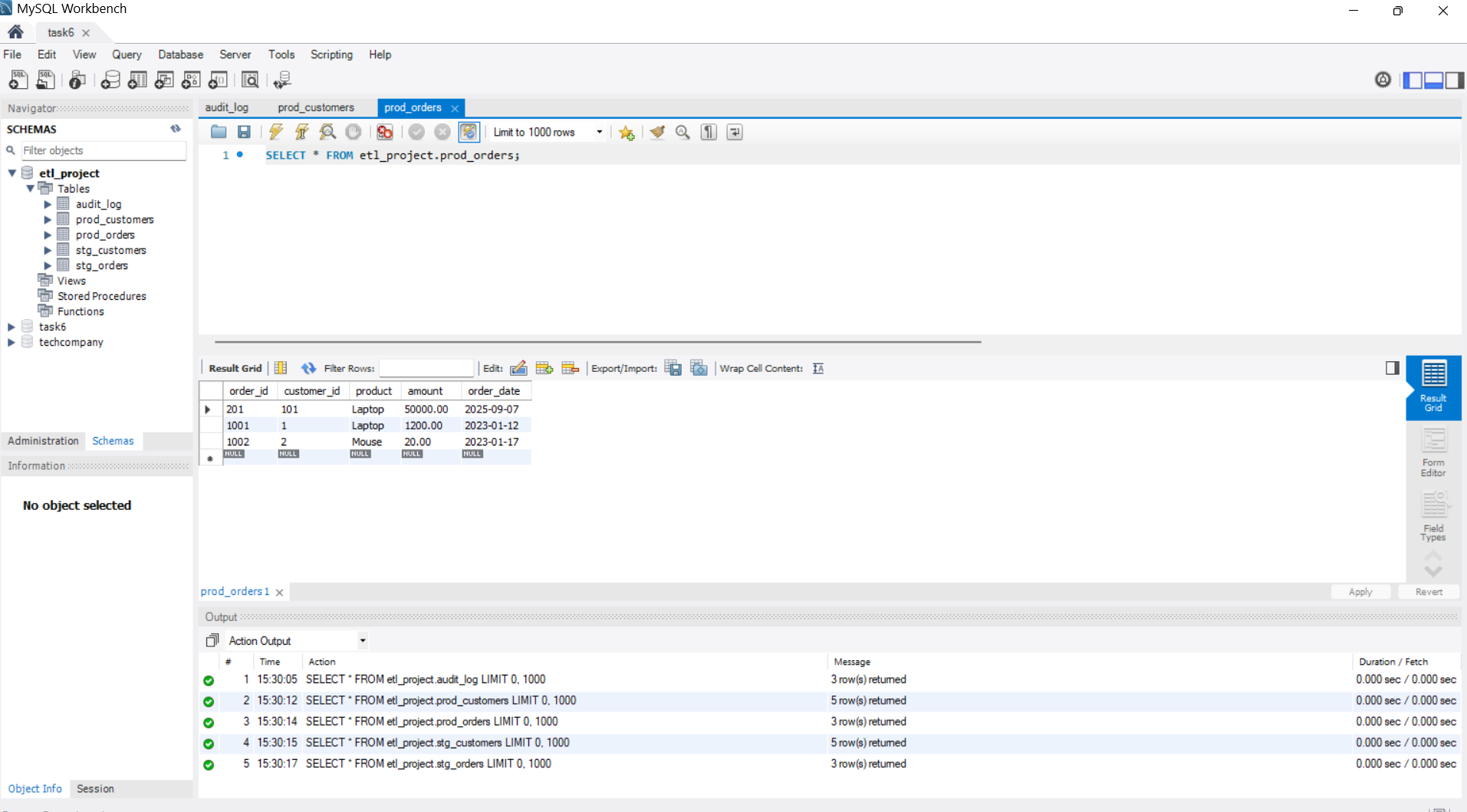
i) audit\_log tables

****

ii) prod\_customers

****

iii) prod\_orders

****

# Conclusion

This project demonstrated how SQL can be used to implement a simple ETL pipeline. The process included importing raw data, cleaning and transforming it, logging ETL operations, and automating tasks with triggers. Through this project, fundamental ETL concepts were practiced in a hands-on environment.

# Learning Outcomes

- Understood the difference between staging and production tables  
- Learned SQL functions for cleaning data (TRIM, COALESCE, NULLIF)  
- Implemented audit logging with SQL  
- Automated tasks with triggers  
- Exported cleaned data for reporting

# References

- MySQL Documentation: https://dev.mysql.com/doc/  
- TutorialsPoint: SQL ETL Basics