# Activity Log Pipeline

A production-ready data table for basic analysis (an SHS Company).

This process switches seamlessly from Python to a MySQL Database by utilizing variables and use of SQL simultaneously

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Requirements

All the Python libraries used

- pandas

- numpy

- datetime

- sqlalchemy

- mysql-connector-python

- openpyxl

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Showcase

- ✅ Modular, professional code

- ✅ Seamless switch between Python and SQL

- ✅ One-click execution: python main.py

- ✅ Easy for others (or me) to maintain, extend, debug

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Activity Log Pipeline

A production-ready ETL (Extract-Transform-Load) pipeline designed to automate the ingestion, cleaning, enrichment, and database upload of Activity Log data.

---

## 📚 What This Project Does

1. \*\*Load\*\* contract data from multiple CSV files.

2. \*\*Clean\*\* and sanitize fields (remove empty IDs, standardize formats, correct dates).

3. \*\*Enrich\*\* the dataset (e.g., derive warranty status, customer segmentation).

4. \*\*Save\*\* a local Excel backup of the processed data.

5. \*\*Upload\*\* the clean, enriched data into a MySQL database (`Upya\_Activity\_Log` table).

6. \*\*Delete\*\* any old records before fresh upload to avoid duplicates.

• Data Flow (CSV ➔ Clean ➔ Enrich ➔ MySQL Upload ➔ Excel Output)

---

## 🔥 Pipeline Workflow

- \*\*A\*\* \*\*[Read CSV Files from Directory]\*\*

- \*\*B\*\* \*\*[Filter Unwanted Entries]\*\*

- \*\*C\*\* \*\*[Edit Existing Fields & Clean Dates]\*\*

- \*\*D\*\* \*\*[Create New Fields using Business Logic]\*\*

- \*\*E\*\* \*\*[Connect to MySQL with SQLAlchemy]\*\*

- \*\*F\*\* \*\*[Delete Outdated Records from Target Table]\*\*

- \*\*G\*\* \*\*[Insert Clean Data into MySQL Table]\*\*

- \*\*H\*\* \*\*[Merge with External Tables (e.g., Lost Systems)]\*\*

- \*\*I\*\* \*\*[Perform Final Calculations & Segmentation]\*\*

- \*\*J\*\* \*\*[Rename and Reorder Columns for Consistency]\*\*

- \*\*K\*\* \*\*[Backup Final Output as Excel File (.xlsx)]\*\*

---

## 🧠 Technologies Used

- Python 3.8+

- Pandas (Data manipulation)

- SQLAlchemy (Database connections)

- MySQL Connector (Database driver)

- CSV

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 🚨 Important Notes

- Make sure the Activity\_Log table exists before uploading.

- Ensure all your CSV files match the expected schema (especially column names).

- Backup your database before production runs. Safety first. 🛡️

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## ✨ Future Improvements

- Implement logging instead of print statements.

- Add error handling + retry logic during DB uploads.

- Build a CI/CD pipeline for automated runs.

- Add unit tests for each module (cleaner, loader, etc.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 🧑‍💻 Author

- Olawale | [LinkedIn](https://www.linkedin.com/in/olawale-falodun-b26b61ab/)

- Built with ❤️ for serious operational excellence.