ETL Processing Reflection

What struck me about the creation of the ETL processor was the parts of the project I'd anticipated would be more difficult or simple were not what I'd expected. For one, the topic that we had chosen to do the project on was on Anti-Money Laundering efforts. It was difficult and in some cases downright impossible for me to find any open source datasets. In hindsight, this is to be expected due to the sensitive nature of financial transactions and the sensitive information these sorts of datasets contain. For me personally, it was the most tedious portion of the code to ensure that all possible file formats were accounted for. This was the portion in which my partner and I had to manually input the types of file formats (CSV, JSON, SQL etc). The main drawbacks to this process would be the need to take into account the various types of file formats and the exceptions caused if there is a file that is none of the above.

However, after doing the most manual portion of the process, the rest of the ETL process was quite routine with the usual "if, elif, else" statements doing the rest of the work to convert the process once it's known what the file format is.

An ETL processor would be a good general use tool for a variety of different data projects for first-pass data cleaning. Most datasets, even within the same organization or even within the same project, are not all created in the exact same way and are not all loaded into the same format. With every individual set of data, one could always just look through and manually convert the data and begin cleaning, but that's a time consuming process. The ETL processor is a means to automate one aspect of the data cleaning workflow and to allow flexibility in the ways data can be retrieved and stored. If I were to create a potential extension project on this same set of data, I would potentially look into ways of making the stored data more secure due to the sensitive nature of financial data and transactions.