**INTERVIEW MINI-PROJECT: EMPLOYEE DATA MART**

**Task #3 - Plan an execution**

Describe how you would execute and schedule the whole data loading process from the previous two tasks.  
Detail what steps you would follow from the start, when you pull the data, until the end, having the processed and transformed data in the final set of tables. Describe how each script would be executed and in what order.

If you know any tools to do some of the work with, you can name them, but it is not necessary to describe the way the tool works, just a general idea of the concepts you are using.

You can also use pseudo code if you want to, to get some structure to the data loading scripts.

Task # 3 Answer:

Please create a folder inside “InterviewProject\_<YourName>” with the name “task3” including a .docx document with the plan’s detailed steps

**TECHNOLOGY:** AIRFLOW

**PIPELINE STEPS:**

1. **getFiles**: BashOperator which finds all files with “.xlsx” extension and outputs it to a .txt file.
2. **loadData**: PythonOperator which reads the .txt output file from the previous task. Loops through the files, reads each file into a DataFrame, applies formatting and then uploads it to SQL SERVER.
3. **normalizeEmployee**: MsSqlOperator that normalizes the employee\_roster\_data table. It’s important to run this script first because it creates the users table (PK user\_id must be created first before assigning FK).
4. **normalizeSkills and normalizeHours**: MsSqlOperators that normalize both skills and hours tables. Adds FKs constraints referencing the user\_id from the users table.
5. **exportTable**: PythonOperator that exports tables in database to .xlsx files in output folder.
6. **convertRateUSD**: PythonOperator that query MSSQL DB to obtain data from the employee\_roster\_data table. Save it into a DataFrame. Requests through fixer.io API and modifies the salary converting it into USD.