ITEC649 2018 Python Assignment

REPORT

Introduction:

The assignment involved writing Python code to extract information about jobs, people and companies from data files (CSVs and HTMLs) and load them into a consistent SQL database which is sqlite3. It is an example of an Extract-Transform-Load (ETL) task.

There were totally three tables which are:
1. People – Extracted data from people.csv file
2. Company – Extracted data from company.csv file
3. Positions – Extracted data from index.html file (50 Positions)

Implementation:

The normalized database has to be generated itec649.db using the above-mentioned data and given table structures. The database.py is a readily available file which was used to create database tables and structures. When database.by is run, itec649.db is generated. Main.py file contains a set of functions and a main function which calls the functions to read the data from CSV files and HTML file. There are separate functions which insert the values of the extracted data into the database created. Finally, the database is used to generate a CSV files with only required set of data fields and data. The extracted CSV file is named “output.csv”.

Real-time bigger data problems and Solutions:

When trying to implement the same for bigger data, there are some hiccups like Non-uniformity in data, Extra data fields, Unavailability of some data, huge CSV files.

These issues might arise given the quantity of data, but minor tweaks and efficient code writing can reduce these issues and make the system error-free. The code should be able to handle errors successfully. By using a default value for non-uniform and unavailable data fields. The extra data fields should be discarded, since it only adds up to the number of errors. Python parsing using BS4 will require some strong processing power. Scraping requires a lot of memory, so using large servers instead of local computers might improve the speed. Sometimes the implementations are also to be carefully considered to produce high performance. The result will be the same if implemented in different methods, but the time and the processing power might vary which will greatly affect the efficiency of the system.

For Example, to improve performance, one might use different implementation methods like using “regex” instead of “lxml” or using “lxml” which is far better than “bs4”. These are the solutions for the potential problems that might arise while using real-time data.