Kasey Suszko
Paylocity Technical Challenge 02/03/2022

SQL REPORT
Format : MS SQL
Assumption : An SQL report means informal word document with SQL queries
If a more formalized format was expected or a .sql file, please contact me and I would be more than happy to prepare one
Important Info : I do not have MS SQL on my personal computer and due to time constraints I could not create an accurate testing environment, so I might have small syntax errors,

1. Write a SQL report that gives us the average salary for all employees

```
--AVERAGE SALARY FOR ALL EMPLOYEES
SELECT AVG(SALARY) AS AVG_SAL_EMP FROM EMPLOYEE;
```

2. Update previous query to give us the average salary for each company

```
--AVERAGE SALARY FOR EACH COMPANY
SELECT AVG(EMPLOYEE.SALARY) AS AVG_SAL_COMP, COMPANY.NAME,
COMPANY.COMPANYID
FROM EMPLOYEE
INNER JOIN COMPANY
ON EMPLOYEE.COMPANYID = COMPANY.COMPANYID
GROUP BY COMPANY.COMPANYID;
```

3. Let's assume your previous query was long running how would you go about debugging it and finding the root causes of its sluggishness.

Debug Techniques:

1. If working with a large amount of data, use "SELECT TOP 20 (required columns)" to make sure the issue is not an issue with access to the data in SQL (sometimes I have found when I have a temp wifi outage, my query will run indefinitely once I'm reconnected to wifi)
2. Run the query by just joining (not using an aggregate or group by) to make sure the join condition is not the cause. Potentially change the join condition if other options
3. For the above query, I could add in a WHERE statement potentially to reduce the data being joined (dependent on more details about the data, such as some companies not being needed).
4. Find other ways I can filter data as it will help to improve performance.
5. Execute with debugger or utilize stored procedures for various tests examining the data to use in the future

Python Overview
Script : kaseySuszkoDE.py

If more formal documentation or explanation is expected, please contact me and I will gladly generate it.

Usage:
Please enter the file name in the input loop. To exit the code after file input, enter anything but "RUN",, and to re-run it, enter "RUN". The default file input is currently set to the sample file as if it were in the same directory as the python code.

Requirements.txt: (I don't believe any of these are needed other than the python version, but I am sharing because this is generated from the environment I worked in)
python = 3.8
numpy==1.22.1
pandas==1.4.0
python-dateutil==2.8.2
pytz==2021.3
six==1.16.0

Overview:
For my script, it runs in a while loop where you input the data file name and can choose to re-run the script or stop it at the end. I performed various tests while developing, and some of the informal and formal ones I used are at the bottom of my code. Assumptions I made and improvements I would like to add to the code if it were going to go to production are included in comments at the top of the python script. It is composed of the functions below:

read_file(file) = read the input file and return a list of json objects

actions(dict_list) = utilizes lists of the json objects in each table to perform the actions by iterating through the objects. Includes counts of each action to use for data validation. In the future, more error handling could be added in.

table_lists(raw_data) = utilizes the raw data to create lists per table

output(table_lists, raw_data) = runs actions() and prints in a pretty format