# Data Engineer Assessment

## Purpose

The purpose of this exercise is to write a little python and SQL code to solve a problem our data engineer may be tasked with during his or her first weeks on the job. In this scenario we have data sources we want to merge together; however, we do not have a mapping from one table to the other which makes the mapping a non-trivial task.

All of the coding work should be done in python using any packages you see fit that are available through pip. Please include any command line tools and commands used during this exercise. Example, if you use home brew to install PostgreSQL please include that in the code you submit.

## Test cases

During the exercise create a total of 2 test cases for the portions of the exercise you think is the most important to test.

## Build a local PostgreSQL database with Python

Download PostgreSQL to your local computer along with python version 3.7.

## Download data files from google cloud storage

There is a public google cloud storage bucket named **data\_engineer\_assessment.** In this folder there are two data files and the document you are currently reading. Write a python script to download the files. They are named **snowflake\_table.txt** and **sqlserver\_table.txt**.

## Process Data Files from google cloud storage

Determine which character each data file is delimited by. Write a python class or function to parse the data files and to make two tables in your local PostgreSQL database. Also, write a python class or function to create each table with your recommended data types.

The **snowflake\_table.txt** filedoes not have a unique key in the file to identify each *unique* property therefore we should make a python class or function to create a unique property id for each property using a combination of the columns available in the data set when it is loaded into the database.

## Create a mapping between the two data tables

Now that the two tables are loaded into the local database, we would like to create a python function that can map the two tables together using a combination of the variables in each table. The apt name in each table is not consistent since the data is from two different sources. Also, some of the names may have misspellings and other data issues therefore you may have to be creative when making the mapping between the two tables.

After making the function that maps the two tables together build final table that contains the unique ids for each table so that it could be used later to merge the tables together.

Make a python class or function that creates a metric that you can give with each property mapping that indicates the likelihood that the properties you matched up are correct.

Make a histogram of the results of your metric to give an idea of how well you think the matching works.

## Further recommendations

We can possibly pull more information from both data sources. If this was possible what data points do you think would help improve the matching?