# Notes

You should provide us with an executable .py file, Jupyter Notebook or Google Colab Notebook as well as a requirements.txt file listing the components you are importing in your code.

We strongly recommend you use one of the following versions of Python – 3.7, 3.8, 3.9

All input data required for this exercise is contained in "Fixtures.csv" and this file should not be modified in any way outside of Python.

Database params are in ".env".

**Note for Mac/Linux users:**

Files beginning with a dot, like the above ".env" file, are hidden by default by the OS.

Most editors are configured to show them anyway, but to see the file in Finder, open a new Finder window and press and hold the following: Cmd + Shift + . (period)

# Exercise

Write a single Python module to perform the following:

## Part 1

Load “Fixtures.csv” into a pandas DataFrame.

## Part 2

Create a function that takes the DataFrame from Part 1 as an input and:

* Removes all rows where the fixture contains 'Man City'.
* Updates all cells where 'Brighton' occurs to 'Brighton & Hove Albion'.
* Returns a DataFrame that shows the standings for a given date.

Each team gets 3 points for a win, 1 for a draw and 0 for a loss.

Here’s an idea of what the returned DataFrame should look like:

## <https://www.google.com/search?q=premier+league+table&oq=premier+league+table&aqs=chrome..69i57j0i131i433i512j69i59l2j0i512j69i60l3.3156j1j7&sourceid=chrome&ie=UTF-8>

## Part 3

Using the function you created in Part 2, call the function to return the standings as at 25th December 2020 and export this Data Frame to a CSV file.

## Part 4

Snowflake is a cloud-based relational database, and is the database technology used by Gambling.com Group.

Using the Snowflake Connector, execute a **DELETE** on the STANDINGS table to clear out any existing data and using the function you created in Part 2, write the Standings as at the end of the season to this table in Snowflake.

The table has already been created using the DDL below and is ready for use.

create or replace TABLE DEVELOPER\_SANDBOX.TECH\_TEST.STANDINGS\_13 (

POSITION NUMBER(2,0) COMMENT 'Ranking based on the following order: points, goal difference, goals for.',

CLUB VARCHAR(100) NOT NULL COMMENT 'Name of team.',

MATCHES\_PLAYED NUMBER(2,0) COMMENT 'Total number of matches played.',

WIN NUMBER(2,0) COMMENT 'Total number of matches won.',

DRAW NUMBER(2,0) COMMENT 'Total number of matches drawn.',

LOSS NUMBER(2,0) COMMENT 'Total number of matches lost.',

GOALS\_FOR NUMBER(3,0) COMMENT 'Total number of goals scored.',

GOALS\_AGAINST NUMBER(3,0) COMMENT 'Total number of goals conceded.',

GOAL\_DIFFERENCE NUMBER(3,0) COMMENT 'Total number of goals scored less total number of goals conceded.',

POINTS NUMBER(3,0) COMMENT 'Total number of points accumulated.',

primary key (CLUB)

);

Example of snowflake connector definition:

with snowflake.connector.connect(

account=aaa,

user=bbb,

password=ccc}) as snowflake\_connection:

The following documentation may be useful for working with the Snowflake Connector for Python: <https://docs.snowflake.com/en/user-guide/python-connector.html>

Please ensure you install the official snowflake connector i.e. “snowflake-connector-python”