PythonApp to Access an SQL and a Neo4j Database

# Approach

This project is made of four key files:

* main - this is the file to call to run the programme
* menu - manages the logic of the menu and calling the functions to make each option operate correctly
* sql\_appdbproj - contains all the functions to interrogate and update the sql database
* neo4j\_functions - contains all the functions to interrogate and update the neo4j database actorsMarried

# Innovation

Throughout where an invalid input is entered a prompt is shown. The prompt has greater information to try and encourage a valid input.

## Menu 2

The get\_birth\_month function takes in only the first 3 letters of a text input. Text inputs are transformed to lower case and then the input is checked against the dictionary object before this is used in getting the data from the database.

## Menu 3

First the code checks the ID is not already in use to prevent accidental overwriting of a record. Then the date format entered is checked to ensure it is in the correct format to add to the database. The input handling for the gender allows some flexibility so users can enter the single first letter or the full word for the gender. Allowing the user to input the single first letter, makes data entry faster. Finally, the country id is checked against the database. If an invalid ID is entered the list of countries with their country IDs is shown to help the user make a valid input.

When the details are all added correctly. The code updates the database and then queries the added actor by the ID entered to prove they have been successfully added.

## Menu 4

If the code finds a spouse in the neo4j database a query is run on the sql database to bring back their name for a more meaningful return to the user. Then the code checks if there is already a valid marriage between them, preventing an unnecessary update to the database. If there is not an existing marriage between them, each actor is checked individually for a marriage to anyone else. If no marriage is found the marriage is added to the neo4j database.

## Menu 5

Each of the actor IDs are checked against the sql to ensure they exist. The user is prompted to add a new user ID until a valid record is found.

## Menu 7

A new menu item has been added to handle adding a new studio, this pairs with menu 6 view all studios. The menu asks for the user to input the name of a studio, if this matches a name already in the database an error is returned. If the name is new, it is saved to the studio cache. This cache will be saved when the programme is exited and at this point the next available StudioID is assigned.

## Menu x

If there are any new studios in the cache they are added to the database at this point. Then the code sequentially checks database connections and ensures they are closed.