NHS Digital - Software Developer Technical Task

Introduction

NHS Digital have invested in some tools to capture deployment data. The purpose of collecting the data is to analyse deployment data between projects as a way of optimising our continuous delivery approach.

A sample data file has been prepared which you can find here:

```
https://github.com/NHSDigital/tasks/blob/master/projects.json
```

There is also a description of the data which can be found here:

```
https://github.com/NHSDigital/tasks/blob/master/projects.json.md
```

You are being asked by a delivery manager to prepare some outputs based on the data - so they can start planning to make improvements in future iterations.

The task is to write a tool that can analyse the exported data `projects.json` to produce the data outputs specified below.

1. Day of week deployment frequency

   The delivery managers are concerned that all release activity appears to be concentrated on a certain days during the week, putting more pressure on NHS Digital to manage change on those days. Data is required to see if this is a real issue.

   Generate the following report based on the number of deployments to **Live** (successful or unsuccessful):

   ```
   DayOfWeek,LiveDeployments
   Monday,<?>
   Tuesday,<?>
   Wednesday,<?>
   Thursday,<?>
   ```

```
Friday,<?>
Saturday,<?>
Sunday,<?>
```

This report should to be written to the file `1_deployment_frequency.csv`.

2. Projects with slow releases

   The delivery managers are also concerned that the release time for some projects seems to be faster for some projects than others, and want to see if this is true. Data is required so they can act on it.

   Generate the following based on the average time taken in minutes for a successful release to go from **Integration** to **Live** grouped by **project_group** and ordered by the time taken (longest first).

   ```
   ProjectGroup,AverageTimeToLive
   Alpha,100000
   Beta,10000
   Gamma,1000
   ...
   ```

   This report should be written to the file `2_slow_releases.csv`.

3. Failing releases

   The delivery managers have noticed that time is wasted when releases are deployed to the **Integration** environment, but never to **Live**.

   Prepare a report that identifies project groups with the highest number of releases that have a successful deployment to **Integration** but never have a corresponding successful deployment to **Live**. The output should be ordered by the count (highest first).

   ```
   ProjectGroup,FailedReleases
   Beta,100
   Alpha,20
   Gamma,10
   ```

This report should be written to the file `3_failing_releases.csv`.

Instructions

- The json file can be downloaded (and submitted with the project if required).
- A readme should be provided with instructions on how to run the code, and where to find the output files.
- Don't write your own json parser! Using libraries is a good idea, databases are not a good option for this activity unless you are using an in-memory database like sqlite.
- Submit in the coding language that you are most comfortable with.

Zip up your code and upload to the following URL using your Candidate ID:

https://script.google.com/macros/s/AKfycbz79ptkTEIstNrriGeK5zJ9mwCr1M5rTRGjMGmKVJa4AeEvvS-m/exec