Mercado Libre DataSec Technical Challenge

Instructions

- Create a separate file for each question.
- Commit all the solutions to a single public GitHub repository.
- Solve all the questions using python 3. Specify in the comments the exact version of python you used.

1. Minesweeper Number of Neighbouring Mines

Create a function that takes a list representation of a Minesweeper board, and returns another board where the value of each cell is the amount of its neighbouring mines.

Examples

The input may look like this:

```
[
 [0, 1, 0, 0],
 [0, 0, 1, 0],
 [0, 1, 0, 1],
 [1, 1, 0, 0]
]
```

The `0` represents an **empty space** . The `1` represents a **mine**.

You will have to replace each **mine** with a `9` and each **empty space** with the number of adjacent mines, the output will look like this:

```
[
 [1, 9, 2, 1],
 [2, 3, 9, 2],
 [3, 9, 4, 9],
 [9, 9, 3, 1]
]
```

Notes

- Since in the output the numbers `0-8` are used to determine the amount of adjacent mines, the number `9` will be used to identify the mines instead.
- You can use the wikipedia page explaining how Minesweeper works

2. REST API: Best TV Shows in Genre

Use the HTTP GET method to retrieve information about recent television shows. Query https://jsonmock.hackerrank.com/api/tvseries to find all the shows in a genre. The query result is paginated. To access additional pages, append ?page={num} to the URL where num is the page number.

The response is a JSON object with the following 5 fields:

```
   "name": "Game of Thrones",
   "runtime_of_series": "(2011-2019)",
   "certificate": "A",
   "runtime_of_episodes": "57 min",
   "genre": "Action, Adventure, Drama",
   "imdb_rating": 9.3,
    "overview": "Nine noble families fight for control over the lands of Westeros, while an ancient enemy returns after being dormant for millennia.",
   "no_of_votes": 1773458,
   "id": 1
```

In data, each tv series has the following schema:

- *name*: (String)
- *runtime_of_series*: years with a new season (String)
- *certificate*: rating (String)

- *runtime_of_episodes*: average length per episode in minutes (String).
- *genre*: genre (String)
- *imdb_rating*: average viewer rating (Number)
- *overview*: short description (String)
- *no_of_votes*: how many votes were cast at imdb (Number)
- *id*: unique id (Number)

Given a genre, find the series with the highest imdb_rating. If there is a tie, return the alphabetically lower name.

Function Description

Complete the function *bestInGenre* as follows:

bestInGenre has the following parameter(s):
string genre: the genre to search

Return

string: the highest-rated show in the genre, with the lowest name alphabetically if there is a tie

Sample Input

```
Action
```

Sample Output

```
Game of Thrones
```

Explanation

The 4 highest-rated shows in the 'Action' genre are shown:

- 'Game of Thrones', 9.3
- 'Avatar: The Last Airbender', 9.2
- 'Hagane no renkinjutsushi', 9.1

- 'Shingeki no kyojin', 8.9

3. SQL: Advertising System Failures Report

As part of HackerAd's advertising system analytics, a team needs a list of customers who have a maximum number of failure events (status = "failure") in their campaigns.

For all customers with more than 3 events with status = 'failure', report the customer name and their number of failures.

The result should be in the following format: customer, failures.

- customer is a candidate's full name, the first_name and last_name separated by a single space.
- The order of the output is not important.

Schema

There are 3 tables:

customers		
name	**type**	**description**
id	SMALLINT	Customer ID
first_name	VARCHAR(64)	Customer first name
last_name	VARCHAR(64)	Customer last name

campaigns

name	type	description
id	SMALLINT	Campaign ID
customer_id	SMALLINT	Customer ID
name	VARCHAR(64)	Campaign name

events		
name	**type**	**description**
dt	VARCHAR(19)	Event timestamp
campaign_id	SMALLINT	Campaign ID
status	VARCHAR(64)	Event status

customers		
id	**first_name**	**last_name**
1	Whitney	Ferrero

2	Dickie	Romera

Sample Data

Replicate the following data in a local MySQL instance

campaigns		
id	customer_id	name
1	1	Upton Group
2	1	Roob, Hudson and Rippin
3	1	McCullough, Rempel and Larson
4	1	Lang and Sons
5	2	Ruecker, Hand and Haley

events		
dt	campaign_id	status
2021-12-02 13:52:00	1	failure

2021-12-02 08:17:48	2	failure
2021-12-02 08:18:17	2	failure
2021-12-01 11:55:32	3	failure
2021-12-01 06:53:16	4	failure
2021-12-02 04:51:09	4	failure
2021-12-01 06:34:04	5	failure
2021-12-02 03:21:18	5	failure
2021-12-01 03:18:24	5	failure
2021-12-02 15:32:37	1	success
2021-12-01 04:23:20	1	success
2021-12-02 06:53:24	1	success

2021-12-02 08:01:02	2	success
2021-12-01 15:57:19	2	success
2021-12-02 16:14:34	3	success
2021-12-02 21:56:38	3	success
2021-12-01 05:54:43	4	success
2021-12-02 17:56:45	4	success
2021-12-02 11:56:50	4	success
2021-12-02 06:08:20	5	success

the expected output is:

customer	**failures** ▼
Whitney Ferrero	6