IFN509 Assignment 1

Assignment Overview

Relates to learning outcomes: 1 and 2

Weight: 25%

Group or Individual: You may work on this assignment in pairs (two students). Note that only a single submission is required from each pair, however make sure both the report and the MySQL file feature the names and student numbers of both students

Due date: Friday April 17th 9:00AM

Scenario

You have been given three datasets in *CSV* format containing information about movies, reviews of these movies, and information about the reviewers respectively. The information has been exported from the database that runs the **Rotten Tomatoes** website. **For this assignment it is your role to create an sql file that creates a database, imports this information into it, and runs the required queries correctly.**

The 3 *CSV* files contain headers as follows:

Movies(movieID, Title, Year)
Reviewers(reviewerID, name, yearJoined, trustRating)
Ratings(reviewerID, movieID, rating, comments)

The following pages outline a number of queries for you to write that address future requirements of the database. You will demonstrate that these queries work on the sample dataset, using the specific values provided in *italic* where required.

There are also some additional theory questions for you to consider.

Scenario

You are to submit the following files through Blackboard:

1. A **Report** with your answers to questions 1, 6 and 7
2. A **MySQL File** with all the required queries (including creating the database and importing the information), which will be tested. It should follow the indications given before question 2.

Question 1: Relational Schema (6 Marks)

- Provide a relational schema for the database you will create.
- Identify the primary keys (1 mark), the foreign keys (1 mark), and the constraints (1 mark).
- Identify the data types (1 mark) and write a query to create the database with MySQL (1 mark) and one to import the sample dataset you were provided with (1 mark)

Preparing the SQL File for Testing

In order to run on your marker's computer, your SQL file should have the following commands at the beginning. Note:

- You need to replace "IDXXXXXX" by your student ID.
- You need to include your own table definition instead of "*YOUR CODE".*
- You need to include more CREATE and LOAD command for other tables you may need to create.
- You should NOT modify the location provided for the csv file ("./")
- You will include in comments (lines starting with "--") the questions before each query.

```
DROP DATABASE IF EXISTS moviesdb_IDXXXXXX;
CREATE DATABASE moviesdb_IDXXXXXX;
USE moviesdb_IDXXXXXX;

CREATE TABLE Movies
(
        YOUR CODE
);
... Create the other tables here

LOAD DATA INFILE './movies_movies.csv' INTO moviesdb_IDXXXXXX. Movies
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n' On Windows OSs change this to \r\n
IGNORE 1 ROWS;

... Import the other files here

-- Question 2: Simple SQL queries
-- 2.a Write a query to list the names of the reviewers
SELECT YOUR CODE

...
```

Question 2: Simple SQL queries (5 marks total, 1 each)

2.a Write a query to list the names of the reviewers

2.b Write a query to list the movies released after a given year: *1980*

2.c Write a query to list the movies in alphabetical order of their title

2.d Write a query to give the average rating of reviewer with a given ID: *541*

2.e Write a query to list the average ratings of each reviewer

Question 3: Complex SQL queries (5 marks total, 1 each)

3.a Write a query to know the average rating of a movie given its title: *Dumbo*

3.b Write a query to list the average rating of all the movies for each year before a given year: *1980* (and write NIL if there are no ratings for that movie)

3.c Write a query to list the movies that have at least 1 numerical rating in common with a given movie: *Dumbo*

3.d Write a query to give the average year of movies that received at least once a given rating: *5*

3.e Write a query to give the name of the favorite movie of the reviewer that gives the highest average rating.

Question 4: SQL Functions (3 marks total, 1 each)

4.a Write a query to count the number of movies in the database.

4.b Write a query to list the names of the reviewers who have been active for longer than a given duration: *10 years*

4.c Write a query to get the difference between the normal average and the weighted average of a given movie: *Ratatouille*. The weighted average is obtained by considering the trustRating of a reviewer as the weight of their review score for the movie. You should also include the average and weighted average in the output.

Question 5: Regular Expressions (2 marks total, 1 each)

5.a Write a query with a regular expression to list the names of movies that start with a certain word: *Pirates*

5.b Write a query with a regular expression to list the comments of ratings of at least 4 that don't feature any of the following words: great, like, nice, fantastic.

Question 6: Scaling up the Database (2 marks)

Discuss what changes you would make to the set up of the database if you were expecting 10,000 movies and 100,000 reviews listed. (1 mark)

Discuss what further changes you may implement if the database was expecting 10 million movies and over 3 billion reviews. (1 mark)

Question 7: Adding Information about the Movies (2 Marks)

Discuss what changes you may require if you wanted to add more information about the movies: long descriptions, list of actors, categories, and if you wanted to mainly run queries to find movies titles (for search or recommendation for example).