Project Assignment 2

To understand and be able to write advanced SQL queries that requires the use of subquery, aggregation function etc..

Overview

In this homework you are asked to write **6** SQL queries on the relational **rating** database, which you used before. The database for this assignment consists of the following three tables:

1. Movie (mID, title, year, director)
   English: There is a movie with ID number *mID*, a *title*, a release *year*, and a *director*.
2. Reviewer (rID, name)
   English: The reviewer with ID number *rID* has a certain *name*.
3. Rating (rID, mID, stars, ratingDate)
   English: The reviewer *rID* gave the movie *mID* a number of *stars* rating (1-5) on a certain *ratingDate*.

   `Rating.rID` refers to `Reviewer.rID`, and `Rating.mID` refers to `Movie.mID`.

Your queries will run over a small data set conforming to the schema. Please import the schema and data into your database from the given file *rating.sql* by issuing the command in `mySQL Client(Mariadb)`:

```
MariaDB [(none)]> source c:[\the path rating.sql is in\]rating.sql
e.g.
MariaDB [(none)]> source c:\database\rating.sql
```

- You are to translate the English into **one** SQL query that computes the desired result over all possible databases. All we actually check is that your query gets the right answer on the small sample database. But you **DO NOT** have to turn in the results you get from executing your queries.
- Your queries are auto-graded using Sqlight3. If you use **alternatives**, e.g. MS SQL, please ensure the queries you submit are executable on the Moodle VPL auto-grading system.
- **Eliminate duplicates** tuples from the result.

SQL Query Questions

1. Some reviewers didn't provide a date with their rating. Find the names of all reviewers who have ratings with a NULL value for the date, sort them in ascending order. (save the sql in q1.sql)
2. List all directors (name) and the number of films they directed, in descending order of the number of films they directed, then order by their name in ascending order. Include director whose name is not specified in the database. (in q2.sql)
3. Find the names of all reviewers who have contributed three or more ratings. Sort them in ascending order. (in q3.sql)
4. List movie titles and average ratings, from highest-rated to lowest-rated. If two or more movies have the same average rating, list them in alphabetical order. (in q4.sql)
5. For each director, return their name, the title of the movie(s) they directed that received the highest individual rating, and the value of that rating. Ignore movies whose director is not specified in the database. Sort by the rating from highest to lowest, then by director name.

(in q5.sql, **Note**: Sqlite3 doesn't support keyword `ALL` , consider using `not exists` in the subquery instead.)

6. For each movie, return the title and the 'rating spread', that is, the difference between highest and lowest ratings given to that movie. Sort by rating spread from highest to lowest, then by movie title. (in q6.sql)

Submission and Evaluation

You need to submit in total 6 .sql text files to the Moodle. Each of the files containing only one query statement.

Use the **Edit** tab submit multiple files at one time. Don't forget to click button `evaluate` to get your result. The result can be also seen in the tab `Grade`.