School of Computer Science, McGill University

COMP-421 Database Systems, Winter 2022

Written Assignment 1: SQL

Due Date Feb 17, 12:00noon

This is an individual assignment. You are required to work on your own to create the solution.

This assignment will be graded by an automated system. It is very important to read the instructions and follow them exactly. Not following those instructions can result in your assignment receiving 0 points.

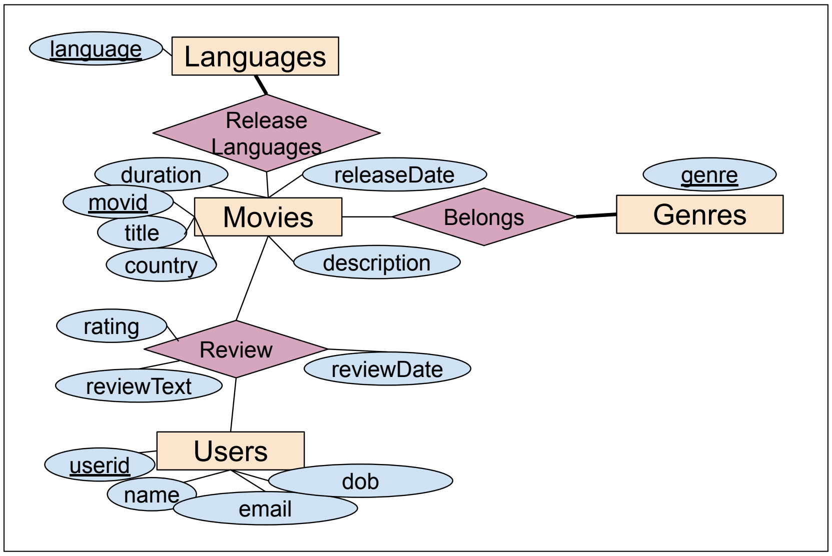
Please pause and go over the included COMP421-A1-SQL-FormatGuidelines.pdf completely once before you continue reading this assignment description. It describes how to setup the sample data set that is provided to you and to ensure that you do not are not going down the wrong path on how to prepare your solutions.

Turn in one tar file in mycourses.

Ex. 1 — SQL (60 Points)

Although the individual questions add up to a total of 75 points, the maximum points that you will get for this assignment is capped at 60 points. Use the extra 15 points as a buffer. **Please remember that any penalty is applied over the capped score and not your overall score**.

The following (part) of ER model and the corresponding relational model are designed for a movie reviews and recommendations system. The system keeps track of very basic information of the movies and users. It also faclitates the users to review movies and and rate them in a scale of 1-10.
For this assignment, you MUST use the PostgreSQL database system that is setup in **winter2022-comp421.cs.mcgill.ca**. Using any other database, including the DB2, will automatically result in a 0 for the assignment. The scripts provided with this assignment is already setup to work with the PostgreSQL database.
A sample schema and few records have been provided as **setup.sh**. Please add more records into this as you may need to test various scenarios. The automated system will use a slightly different data set. So ensure your logic works even for different data sets. **Remember! Just because your query returned the correct results does not mean it is correct. Your query could produce incorrect results with slightly different data**. It is important to review your query and also to make sure that your data set has some "corner" cases (it is not about the number of records you use, but the variety of records that address different scenarios). The table definitions used by the automated system will be the same as the ones in the sample scripts given to you. Use your individual database accounts to work on the assignment.
Please note that for simplicity we are not keeping tracks of languages or movie genres that has no movies associated with them. The relational model given below has been simplified in this aspect.



`movies(movid, title, country, duration, releasedate, description)`
`users(userid, uname, email, dateofbirth)`
`releaselanguages(movid, language)`
movid references movies
`moviegenres(movid, genre)`
movid references movies
`review(userid, movid, reviewdate, rating, reviewtxt)`
userid references users, movid references movies

Important !!

All the sql solutions will be evaluated by an automated system, which compares the output data produced by executing your query on our dataset with the expected output result for the correct query. So it is important that you include the correct column names, in the correct order, perform any ordering on output tuples as asked etc.

Double check your SQL for **typos**, for example if you typed '*BenHur*' instead of '*Ben-Hur*', a query might not return the correct records and you will not get any points.
While the column and table names are not case sensitive, **the data itself can be case sensitive**. So do not write '*BEN-HUR*', where it was required to write '*Ben-Hur*' as this may produce no results or wrong results.
For more details read the attached sql formatting guide. If you have questions about this, post it in Ed.
Remember you will either get 0 or all points for a given SQL question !!

For this assignment you MUST NOT create views or tables in your solution. You can however use the derived table concept as we saw in class in your SQL. You may also use the SQL `WITH` clause if you wish to. Do not use any explicit SQL `CAST` operations to perform numeric data conversions of your own. All your answers should be comprised of only a `SELECT` query. **Output ONLY the attributes in the question, following the exact column positions in the `SELECT` clause as mentioned in the question**. Adding attributes not mentioned can result in a 0 score !

Unless explicitly allowed in the question, your output query should not produce duplicate rows in your output resultset. Use the technique taught in class to eliminate duplicate records from the output when a query can result in such a scenario. Keep in mind that some SQL queries do not produce any duplicate records while others may.
Where an output ordering is asked for, remember to order the output records. The technique for this was also shown in class. Unless stated otherwise, the default ordering should be in the ascending order.

1. (2 Pts) List the movie id and title of all movies released on or after the year 2021. The output should be ordered by the movie id.
2. (2 Pts) Find the user id of all the users who have reviewed the movie (title) `Casablanca`. The output should be ordered by the user id. You may assume that there is only one movie with this title.

3. (6 Pts) List the user id of all the users who gave a favourable review (rating 7 and above) for the 1959 version of the movie (title) `Ben-Hur` but did not review or gave an unfavourable review (rating 4 and below) for the 2016 version of the movie. The output should be ordered by the user id.
4. (3 Pts) List all the movies (title), release date and the ratings for the reviews done by the user whose email is `talkiesdude@movieinfo.com`. The output should be ordered with the highest ratings on top. If two movies have the same ratings, then they should be ordered by their release dates and then by titles.
5. (4 Pts) Give the list of movies (titles) and release dates of movies released in the year 2021 that falls into both the genres `Comedy` and `Sci-Fi` (i.e., a movie should be part of both the genres and not just one). Order the ouput by release date and then by titles.
6. (3 Pts) List the title and release date of movies that were produced in `French` but not in `English`. The output should be ordered by the release date and then by the title.
7. (6 Pts) List the title, release date and language(s) of all the movies that were released in `French` or `Italian` (it may also have releases in other languages but should have been released in at least one of these languages). The output should be ordered by release date and then by title. If a movie is released in both `French` and `Italian` then it should be listed only once in the output as a single output in the form `French,Italian` (specifically in **that alphabetical order** of languages and comma separated, but no spaces between the values). DO NOT output it as **Italian,French** Do not include other languages in the output listing (only the above two). Name this ouput column `languages`.
A sample output format is shown below.

```
 title  | releasedate |   languages
----------------------+-------------+----------------
 Movie1 | 1946-01-16  | French,Italian
 Movie3 | 1967-02-23  | Italian
 Movie2 | 1975-05-10  | French
 Movie4 | 2011-03-13  | French,Italian
```

8. (4 Pts) List the name and email address of all the users who have written a review for a movie that was released ONLY in `French` (they are free to write reviews for any other movies). Order the output by the email.
9. (2 Pts) Find the number of `Comedy` movies that was released in the year 2021. Name the output column `nummovies`.
10. (4 Pts) List the title and release date of all the movies that were released in both (and not just either) `English` and `French` and also had 5 or more reviews. Order the output by release date and then by title. Whether the movies had been released in other languages are irrelevant.
11. (3 Pts) List the title and release date of all movies with less than 2 reviews (make sure to take into account the movies with zero reviews). Order the output by release date and then by title.
12. (4 Pts) List the title, release date and number of reviews of all movies released in 2021. If a movie has no reviews, it should be shown as 0. Name the column with the number of reviews as `numreviews`. Order the output such that the movies with most reviews are on the top. For movies with same number of reviews, order by their release dates and then by titles.
13. (3 Pts) List the title and release date of all movies with the maximum number (count) of reviews. Order the output by release date and then by title.
14. (3 Pts) List the title, release date and average rating of all movies with at the least two reviews. Name the average rating column `avgrating`. Order the output such that the movies with the highest average ratings are at the top. If two movies have the same average rating, then order them further by their release dates and then by titles.
15. (4 Pts) List the title of the movie and the number of reviews for the latest (release date) movie (There could possibly be more than one movie that qualifies). Movies with no reviews yet should not be ignored in the output. Name the number of reviews as `numreviews`. Order the output based on the title of the movie.
16. (5 Pts) This query is a simple attempt of a recommendation system. List the title and release date of movies, their average ratings for `Comedy` movies not reviewed by `cinebuff@movieinfo.com` (email) such that the average rating of each movie is the same or higher than the average rating given by `cinebuff@movieinfo.com` across all `Comedy` movies. Name the average rating column `avgrating`. Order the output such that the movies with a higher average rating is at the top. For movies with same average rating, order them by the release

date and then by the title. You can assume that there will be some Comedy movies reviewed by this user in the database.

17. (5 Pts) List the email of all the other users who have reviewed at least one movie that was reviewed by cinebuff@movieinfo.com (email) and gave a similar review. You can consider two reviews to be similar if their ratings differ only by a maximum value of 1. Order the output by email.
18. (4 Pts) Find the genre that is most liked by cinebuff@movieinfo.com (email). In order to find this, start by computing the average rating this user has given to movies of different genres. Keep in mind that a movie may fall into multiple genres and therefore its rating would contribute to all of those genres. You may also end up with a situation where more than one genre might have the same highest average rating. In such cases, order the output by the genre. You can assume that this user has made some movie reviews.
19. (4 Pts) Find the movies that fell out of popularity. List the title and release date of all the movies that had an average rating of 7 or above before 2019 but has an average rating of 5 or lower starting 2019 (be careful to take into account movies with no reviews in the later period as 0 rating). Order the output by their release date and then by title.
20. (4 Pts) List all the languages in the database and the genre that has most movies for that language. If two genres are equally the most popular for the language, produce two rows in the output (and so forth). Order the output by languages and then by genre.
    A sample format is given below:

    ```
     language |   genre
    ----------+-----------
     French   | Drama
     French   | Sci-Fi
     German   | Drama
     Greek    | Action
     Greek    | Adventure
    ```

Submission

Review the instructions in the attached formatting guide to test your individual solution sqls and then follow the instructions on creating the **.tar.gz** file and submit it. Remember to verify the contents of the tar file to ensure you ran the tar command correctly and it captured the correct contents. You are responsible to verify the integrity of the file you submit to MyCourses. There will be no exceptions.

Please turn in your submission in mycourses under assignment 1. Submissions to wrong folders may not get graded. **You are responsible to download and verify that your submission is correct (not corrupted or incorrect file, etc).** There will be no accommodations if you decided to ignore doing due diligence (you will get a 0).

Resubmissions are allowed (please do not email me saying you do not know how to resubmit files). In such cases, the last submission will be graded. Please ensure that you are submitting a complete set.

General Assumptions

- Unless explicitly mentioned, the ordering asked for a column is implied to be increasing/ascending. Keep in mind that some questions require you to use a combination of ascending and descending on different columns in the output.
- The combination of title and release date of a movie is unique.
- Email is unique to each user id.
- The review tracking system does not record which language release is being reviewed by the user. This is however mostly irrelevant for our simple system and for the questions that are asked in the assignment.
- It is possible that movie may have no reviews and that a user has not written any reviews.
- Every movie is associated with at the least one language and one genre.

- You can assume that the remake of a movie would have the same title (just a different release date). They are not explicitly tracked in the model through any other mechanism.
- You can assume that none of the relations will be empty.

Basic Tips

Since there is more than one way of writing the solution for a question, you may not necessarily end up benefiting from all of these tips.

- Review both the ER and the schema given to you for the database. You will be able to figure out whether some "special cases" are possible or not by reviewing them.
- Explore the use of `COALESCE` or `CASE` operators to work with `NULL`s, especially in the `SELECT` clause of the SQL query.
- `CHAR` and `VARCHAR` data can be concatenated using `||` operator.
- `ABS` function can be used to obtain the absolute value of a column or a numeric expression.
- When you use some SQL aggregation functions (e.g. `AVG`) the result will be a floating point number with lot of decimal places. Do not perform any conversions on it reduce the decimal places or convert them to integer values.
- For some questions, using some variation of an `OUTER JOIN` followed by a `WHERE` clause might make it easier to form the correct query.
- At times it might be easier to write the SQL for the "negative-logic" of certain constraints that are given in the question and then take the negation of that.
- Using derived tables syntax of SQL or the `WITH` clause can simplify the logic you need to write for some of the complex SQL queries.

Important Database Etiquette

Please remember that the server `winter2022-comp421.cs.mcgill.ca` and the databases installed them are meant to be used for the course work of COMP 421. Using these systems for other work (including other course work) is not allowed. You are sharing these resources with other classmates, so be mindful of its proper use.

Please go through the `README` file in MyCourses under "Database" → "Connecting to DB Servers" for comprehensive list of restrictions. Not following them may result in your id being disabled as stated in the course outline.

Questions ?

Please use **Ed** for any clarifications you need (`assignments` → `a1`). Do not email the instructor or TAs as this leads to a lot of duplicate questions and responses (not an efficient system). Such emails will not receive any replies.

Please check the pinned post "A1 general clarifications" in Ed before you post a new question. It might have been already addressed there, in which case we will not address it again.

Questions about general clarifications must be marked public (as other students will also benefit from this and may even have a valid response). TAs and Instructors upon their discretion may toggle any private posts into public mode for the benefit of the student population at large.

There will be specific office hours for the assignment that will be announced closer to the due date.

Extensions and Late submissions

- Remember, your submission is **due on Feb 17th 12:00 noon**. There is no place for excuses.
- A maximum of 1 day of late submission is allowed with a penalty of 20% of the achieved grade per day (rounded up, even for a minute).
- Penalty waivers are granted only for medically documented emergencies and under any circumstances **will not be granted unless requested 24 hours before the due**. I expect you to be better organized and get the job done ahead and leave the last 24 hours only for one last final check.