Rubric:

For Questions Q1-Q3 (worth 1 mark each):

- Approximately correct questions will be deducted 0.5 mark.

For Questions Q4-Q10:

- if the queries do not run they receive a 0 mark. That includes the case of apparently simple syntax errors (there was ample time to test the queries before submission).
- If the query used extra/unnecessary tables/joins or uses of unnecessary ORDER BY/DISTINCT, 0.33 marks will be deducted for Q4-Q7 and 1 mark will be deducted for Q8-Q10.

Appeals:

If you don't agree with the marking and/or have questions about it please contact your TA (not the instructor), the instructor will be contacted by the TAs if there's a need for a "second opinion."

Preamble

In this assignment, you will be provided with two tables and asked to (1) state in English what a given SQL statement does and (2) write SQL statements to answer queries expressed in English. You will need to strictly follow the submission instructions given below.

Questions

The following tables were derived from open datasets publicly available at Kaggle. The first one, NPHDMovies, was derived from Movies on Netflix, Prime Video, Hulu and Disney+, whereas the second one, IMDBMovie, was derived from IMDB data from 2006 to 2016. Refer to those original datasets to fully understand the semantics of the tables' contents. Please note that by "derived," it means that some attributes were removed, some attribute names had to be changed due to syntax, the attributes being used as primary keys changed (which caused duplications to be removed), etc. Marking datasets will be derived from the same sources, but you can use this testing dataset (or derive your own) to test your queries.

One of the entries in the "IMDBMovie" data set has a duplicate entry for the title "The Host." As the titles are considered the primary key here, this will cause a runtime error (UNIQUE constraint failed: IMDBMovie.Title). So, if you are deriving your own dataset, you must make sure that one of the duplicate entries is removed.

```
CREATE TABLE "NPHDMovies" (
"Title" TEXT, -- Title of the movie, PK
"Year" INTEGER, -- Year the movie was released on one of the platforms
"Netflix" INTEGER, -- 1 if the movie is streamed on this platform, 0 otherwise
"Hulu" INTEGER, -- 1 if the movie is streamed on this platform, 0 otherwise
"PrimeVideo" INTEGER, -- 1 if the movie is streamed in this platform, 0 otherwise
"Disney" INTEGER, -- 1 if the movie is streamed in this platform, 0 otherwise
PRIMARY KEY("Title")
);


CREATE TABLE "IMDBMovie" (
"Title" TEXT, -- Title of the movie, PK
"Genre" TEXT, -- A string containing possibly several genres
"Director" TEXT, -- Self-explanatory
"Actors" TEXT, -- A list of actor names
"Rating" REAL, -- Self-explanatory
"Votes" INTEGER, -- Number of votes the movie has received (in IMDB)
"Revenue" REAL, -- Self-explanatory
PRIMARY KEY("Title")
);
```

Questions 1-3 requires short and concise answers (do not explain how the query works, but rather what it does).

Q1 (1 Mark). What does the following query do:

```
SELECT Year, COUNT(*)
FROM IMDBMovie I, NPHDMovies N
WHERE I.title = N.title AND
Revenue is NULL
GROUP BY Year
ORDER BY COUNT(*) DESC
```

Q2 (1 Mark). What does the following query do:

```
SELECT N.year
FROM NPHDMovies N
```

```
WHERE N.Netflix = 1 AND EXISTS (
        SELECT * from IMDBMovie I
        WHERE N.Title = I.title AND
        I.Revenue > 50)
```

Q3 (1 Mark). What does the following query do:

```
SELECT COUNT ( I.Title)
FROM IMDBMovie I, NPHDMovies N
WHERE I.title = N.title AND
I.Votes > (SELECT AVG(Votes)
            FROM IMDBMovie) AND
N.Netflix * N.PrimeVideo = 1
```

Questions 4-10 require you to write SQL expressions that answer the stated queries. (The use of VIEWs is not allowed.)

Q4 (2 Marks). How many of the streamed movies are on at least three platforms?

Q5 (2 Marks). How many movies that are streamed only on Netflix are "drama" movies (note that a movie may belong to several genres)? (Note: SQLite's LIKE operator is case-insensitive.)

Q6 (2 Marks). Find the names of actual directors that have directed at least 2 movies streamed on Disney's platform.

Q7 (2 Marks). Find the titles of non-streamed movies where the director of the movie is among its actors. (You may want to look into SQLite function INSTR().)

Q8 (3 Marks). Considering all streamed movies that are available at least on PrimeVideo, what is the number of movies per year of the genre "drama." Your output should have two columns, one listing all possible years and the other listing the corresponding quantity of drama movies on PrimeVideo in that year, you can output NULL for null quantities.

Q9 (3 Marks). Find, for each platform, the movie that yielded the highest revenue. Your output should have the name of the platform, the title of the movie, and its revenue.

Q10 (3 Marks). What is the percentage of directors that have all their movies with a rating above 8.0? Note that the aggregate function COUNT() returns an INTEGER, you may want to re-cast its output to REAL by using the function CAST().

Submission

Please note that part of the assignment will be marked automatically on the lab machines, using the exact schema above on a larger database. Thus, it is crucial that you follow the instructions below exactly.

Write your answers for each query in a separate file. For questions 1-3 your answers must be saved in a file named 1.txt, 2.txt and 3.txt. For questions 3-10 your answers must be saved in a file named 4.sql, 5.sql, …, 10.sql. The file names are important, deviating from this standard may render your submission non-gradable.

The first line of each file must be as follows:

-- Question X CCID

where X is the number of the question and CCID is your own CCID. For example, the first line of the third file for the user with CCID 'Somename' would be:

-- Question 3 Somename

The rest of each file, starting at the second line, must contain your answer and nothing else.

Include with your submission a README.txt file that has your name, ccid, lab section, and the list of people you collaborated with (as much as it is allowed as per our course's policy) or the single line "I declare that I did not collaborate with anyone in this assignment." Submission without a README.txt file will yield a 10% deduction.

Bundle all your answers into a single tar file named CCID-A2.tgz (where you must substitute CCID with your actual CCID) by executing the following Unix command:

tar -czf CCID-A2.tgz README.txt 1.txt 2.txt 3.txt 4.sql 5.sql 6.sql 7.sql 8.sql 9.sql 10.sql

Finally, submit the file CCID-A2.tgz using this page (email submissions will not be considered).

Note that eClass does not support versioning of submissions, and each new submission replaces your previous one. This makes last-minute submissions somewhat risky. Avoid last-minute submissions as much as you can, and check your submissions after an upload (ideally on the lab machines) to make sure the right content is uploaded. And again, follow the instructions above exactly, any deviations may render your submission non-gradable and therefore worth 0 (zero) marks.