Assignment CS630 100 points

INSTRUCTIONS

This assignment has two parts, a written (or non-programming) part and a programming part. Questions 1 and 2 are the written parts, and question 3 is the programming (SQL) part. You will have to submit your solutions to part 1 and part 2 separately. Solution to part 1 must be saved in a PDF and submitted on the gradescope. Solution to part 2 must be submitted as a text file in your Unix account, according to the following instructions.

- The PDF file, which contains answer to questions 1 and 2 must be submitted on gradescope under Assignment 2. Answers must be **typed**, not had-written. you can submit your solution on gradescope any number of times before the due date. Once the due date is reached, you won't be able to submit a new version or make changes to an existing version.
- The solution to question 3 must be saved in a text file called Q3.sql and copied in a directory called hw3, under the cs630 folder in your Unix account. Note that you must create the hw3 folder yourself following the instructions at the end of this page. You must copy your solution before the due date. You can replace a solution that you've submitted previously by copying a new version of your file to the same location, any number of time before the due date. However, you must not touch or update the file after the due date. If you copy a new version of your solution or make changes to the file you've copied after the due date, the timestamp on the file will change. Solutions with timestamps later than the due date will not be accepted or graded.

Question 1 - 36 points

A movie platform database contains information about actors (identified by actorid) and information about movies (identified by movieid). Actors also have a name, a date of birth (dob), a phone and an address. Each movie also has a name, a genre, a release year and a studio. Actors play in movies.

- a) Draw the ER diagram of this database (as described in the Question 1 statement). Do not use any other constraints.
- b) Modify the diagram from a further to add the constrain that in each movie at least one actor must play.
- c) Modify the diagram from b further to add the constraint that each actor must play in at most one movie.

- d) Modify the diagram from c further such that each actor must play in exactly one movie.
- e) Modify the diagram from d further such that each actor can have multiple addresses identified by street, city and state.
- f) Modify the diagram from e further such that each actor must have at least one address.
- g) Modify the diagram from f further such that at each address there could be a set of phones.
- h) Modify the diagram from g further such that each movie can have multiple genres.
- i) Modify the diagram from h further such that each movie must have at least one genre.

NOTES:

- ER diagrams should strictly follow the notations used in class. No other notations will receive any points.
- Each problem from a through i must have its own ER diagram, for the total of 9, each having 4 points.

Question 2 - 24 points

Consider a database that stores information about Books (identified by bookid) and authors (identified by authorid). In addition to id, every book has a name, a genre and a publication date. An author also has a name, a city and a state. Authors write Books. The same note on the ER diagram from question 1 applies here.

- a) (8 points) Draw the ER diagram that describes this database. Do not add any additional constraints.
- b) (8 points) Write the database schema for this ER diagram.
- c) (8 points) Write the CREATE TABLE statements for all tables identified for this database.

The create statements must work when ran against the Oracle database. The CREATE statements must be written in an order such that if executed in that order will not cause any error.

Question 3 - 40 points

The answer to this question is a SQL script file in text format.

Consider the following schema:

```
songs(songid: int, title string, release: date)
singers(singerid: int, name: string, city: string, state: string)
singsin(singerid: int, songid: int)
```

The primary keys are underlined in each relation. The singers relation contains information about singers. Relation songs contain information about songs. The singers singsin contains information about singers singing songs.

Notes:

- SQL must run against the Oracle database we use in class. Run and test your queries against the Oracle DB. Create the tables, insert some data, and test your queries to make sure they work properly before submitting your solution.
- SQL queries that do not run against the Oracle DB will receive any credit.
- In the .sql file, before each SQL statement you MUST include a comment line with the problem number the sql statement is for, like the following.

```
-- Answer to Part a
your sql statements for part a here
...
-- Answer to Part b
your sql statements for part b here
...
```

- Each question has 5 points.
- a) Write the SQL statements that create tables songs, singers and singsin. Don't forget to define the key constraints.
- b) Write the SQL query that extracts all the names of the singers from state MA. Sort the result by name in descending order.
- c) Write the SQL query that extracts information about the singers and the songs they sing. Each record in the result should contain the name of the singer, the title of the song and the release date of the song.
- d) Write the SQL query that finds how many singers from Boston, MA are in the database.
- e) Write the SQL that extracts information about singers whose name starts with letter A. Sort the result by the state of the singers, in an ascending order.
- f) Write the SQL that extracts the name, city and state for all singers who played in a song that has a title that contains the word joy. The query should be case insensitive with regards to the case of the letters from the title of the song.
- g) Write an SQL query to extract the name, city and state of singers that sang a song that was released before Sept 1, 2021.

- h) Write the SQL to extracts the name and state of singers that sang some songs released in year 2020. The results should contain no duplicates.
- i) Write the SQL query to extract the id, name and city of singers that are from MA and sang some songs released between Jan 1, 2020, and July 31, 2022. Sort the result by the name of the singers in descending order.
- j) Write the SQL to extract the number of unique singers that sang some song released after Dec 10, 2021.

Instruction to create hw2 folder.

After logging in, go to the cs630 folder and create a folder called hw2 and set appropriate permissions on it as follows. Do this before you copy your solution.

\$ cd ~/cs630
\$ mkdir hw2

After creating the hw2 directory, copy your Q3.sql file in that directory. Once copied, run the following command to set the appropriate permission on it. If you don't do that, I cannot read the file. Do not change the permissions on cs630 folder.

\$ chmod 640 ~/cs630/hw2/Q3.sql