Introduction to Data Science

Homework 1

60 points (+7 bonus points)

What to turn in on Canvas:

Write your name at the beginning of the file.

The answers are expected to have the following items: the SQL queries, the results of running these queries in MySQL database provided, the Relational Algebra expressions, and the query plans/trees.

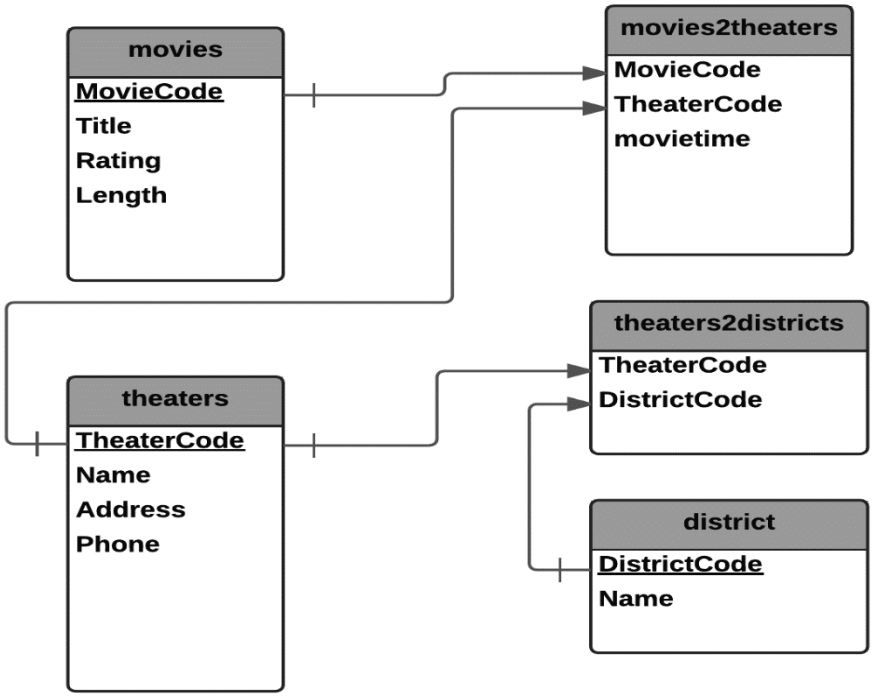
Database connection instructions:

1. You will receive an email with your password. The **username is your FSU ID**. The password is the same for both SSH and MySQL console. Emails will be sent to your myFSU email address. If you have not received the mail, check the spam folder; if the spam folder is empty, let me know about it.
2. Use ssh to connect to bl2.cs.fsu.edu. You can do it either from the internal network (any machine in cs.fsu.edu domain) or outside (e.g. from shell.cs.fsu.edu – note, you may have a different user name on shell.cs.fsu.edu, assigned to you by the department)
3. Run this command in the terminal to open the **MySQL console**: *mysql –u <your_username> –p (hit Enter) <your_password>. Replace the text in red (i.e. <your_username> and <your_password>) with our username (FSU ID) and the password that you received.*
4. Run the following command in the MySQL console to open the database:

   ***use film**;*
5. Start exploring the database (e.g. *show tables;*) and run some queries (e.g. *select * from movies;*)
6. For questions 16 and 17 you will need to use your personal database named after your FSU ID (you can switch to it by using the command - *use <dbname>;*). You can find this database by executing the following command in the MySQL console: *show databases;*

Homework Description

Consider a database with the following schema:



1. The *movies* table stores the movies played in theaters throughout the city.
2. The *districts* table stores the city districts where the movie theaters are located.
3. The *theaters* table stores movie theaters.
4. The *movies2theaters* table stores the movies and the corresponding theaters that play that movie.
5. *The theaters2districts* table lists the theaters and their districts; note that some theaters belong to several districts.

Tables like *movies2theaters* and *theaters2districts* are sometimes called "junction" tables. They store foreign keys from main tables and connect them in this way. Junction tables are not usually meant to store any additional data/objects.

Before you start answering the questions, it is helpful to understand the schema and browse the data. You could start by issuing a few trivial *select* * queries from tables in the schema (e.g. *SELECT * FROM movies;*).

You can use your home folder as temporary storage for your queries and result listings.

Good luck!

1. [2 points] Write a SQL query that lists only the length of all movies with a ‘G' rating.
2. [2 points] Write a Relational Algebra expression that performs the same task.

3. [3 points] Write a SQL query that lists only the phone number and district name of Boeing Imax Theater.
4. [3 points] Write a Relational Algebra expression that performs the same task.

5. [3 points] Write a SQL query that lists the title, length, and movietime of movies that are played in the Cineplex Odeon Uptown theater.
6. [3 points] Write a Relational Algebra expression that performs the same task.
7. [4 points] Compose the query plan for the Relational Algebra expression from question 6.

8. [4 points] Write a SQL query that lists the districts that have at least two theaters playing either 'Chicken Run' or any ‘PG’-rated movie.

9. [3 points] Write a SQL query that lists the average length of movies played at Landmark Neptune Theatre.
10. [4 points] Write a SQL query that shows how many times the Landmark Metro Cinemas theater has played a movie that is shorter than 2 hours.
11. [7 points] Write a SQL query that lists the name and address of all theaters showing less than five different movies.
12. [7 points] Write a SQL query that lists theater names, phone numbers, and the number of different movies that are shown in these theaters sorted by this number (of different movies) in descending order.

13. [7 points] Write a SQL query that lists all theaters in Capitol Hill district that play the longest R-rated movie and output their names (in ascending order) and telephone numbers.

14. [4 points] Write the relational algebra expression for the following statement: *Addresses and phone numbers of theaters that play 'Frozen'*

movie.

15. [4 points] Compose the query plan for the query from question 14.

Switch to your personal database *(listed in the result of the query* ***"show databases;"****) to answer the following questions:*

Bonus questions

16. [3 points] Create a table called 'query16' with the result of the query from question 5. Add a new boolean-valued column called 'want to see' to the new table.
17. [4 points] Copy the table 'query16' into a new table, 'query17'. Modify the column named 'want to see' of the newly created table so that it is true for the movies that contain the word 'The' in their title and false otherwise.