**EECS 495**
**Introduction to Database Systems**
**Fall 2018**
**Instructor: Mas-ud Hussain**
**Project No. 2**
**Due: Sunday, November 04, 2018**

# The Assignment

Consider the following schema (similar to the one in the previous project, with only one column added to the MADE_MONEY table):

```
APPEARED_IN (STAR, MOVIE)
MADE_MONEY (MOVIE, HOW_MUCH, DAY_OPENED, CATEGORY)
MARRIED (COUPLE_NUM, DAY)
DIVORCED (COUPLE_NUM, DAY)
IN_COUPLE (STAR, COUPLE_NUM)
LOG_DATA (MOVIE, CATEGORY)
```

Write SQL triggers for the following equally weighted questions (20 points per trigger, +10 for extra credit). For any credit, you must run your SQL statement in MySQL, and then copy and paste the query into a text file using an editor like Notepad (or, MS word, or whatever you like) and turn that in Canvas. Also, you will have to perform some INSERT statements in this project to test the effects of the triggers (and constraints), and include the results in your finally submitted file. We will re-run all the queries and grade you based on the output.

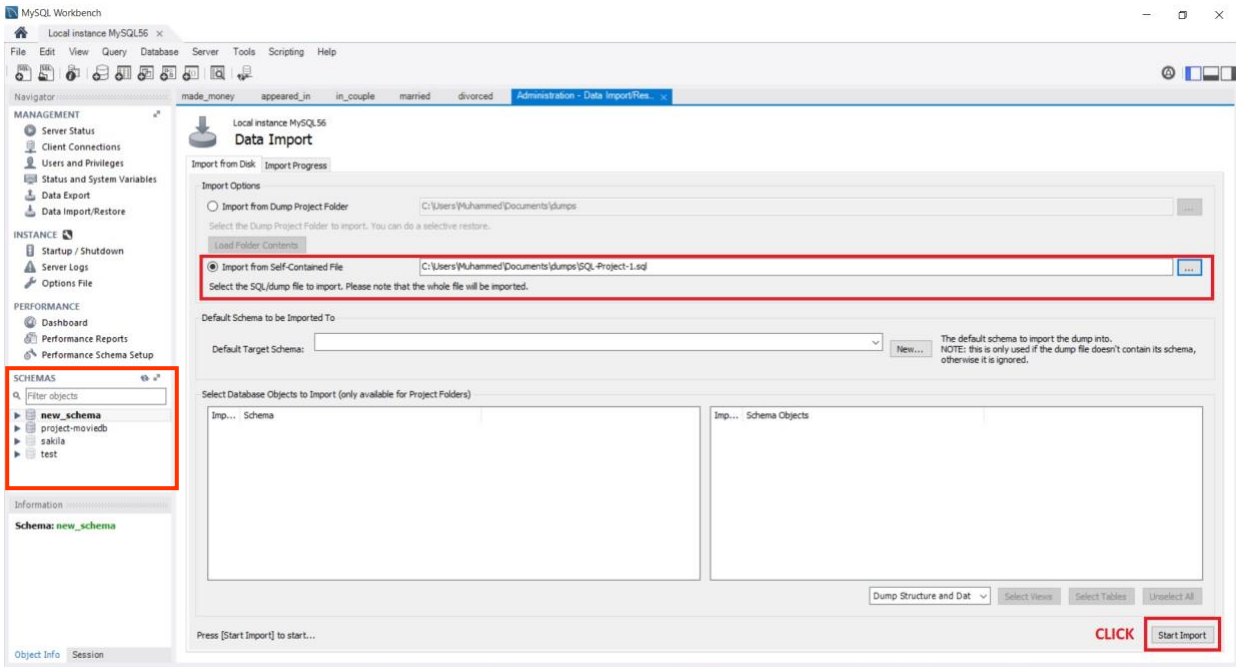
For full credit, your query should work over any database with the above schema, and not just for the one that you test it on.

# Assumptions

- Couple is identified by COUPLE_NUM.
- There is only one couple in any movie.
- If a couple remarried each other, it is a different (new) couple from the old one.

# Loading Schema & Data

Download the attached file with the assignment (*SQL-Project-2.sql*). You should be able to import the file within your MySQL environment. To do so, you have to first launch the MySQL Workbench. Go to server -> data import. Select the options “Import from Self-Contained File” and select the *SQL-Project-1.sql* file from your local directory. Then click “start import” button. The screenshot is given below:




When the import is successfully done, restart the MySQL Workbench. You will find a new schema (*project2-moviedb*) in the schema section of your MySQL Workbench (highlighted in the above figure). Use *project2-moviedb* schema for all the SQL query testing for this assignment.

# Questions

1. Enforce the constraint that categories of a movie must be either “Romantic”, “Comedy”, “Drama”, or “Action”. Suppose, the default value for the Category field is “Action”. If a non-allowed value is inserted/updated, the category for that tuple must be changed to the default value.
2. Enforce the following condition: A star can only be a part of a “Comedy” movie, only if he/she has performed in at least one “Romantic”, “Comedy”, or “Drama” movie previously. Upon insertion of a tuple violating this (e.g., a Comedy movie associated with a star who has previously done only “Action” movies), the category of the movie must be updated to “Drama”.
3. Enforce the constraint: A star cannot be married to multiple stars simultaneously.
4. Enforce that, a movie must make at least $1,000 in the box office, and cannot make more than 3 billion ($3,000,000,000) in the box office. Also, if a movie category is “Action”, then it should make at least $10,000, and if category is “Comedy”, it cannot make more than $1,000,000,000.
5. Using a trigger, ensure that the divorce date of a couple is at least the same or after their marriage date. If this is violated, set the divorce date to be the same as the marriage date.

**Extra Credit (+10)**

6. We want to keep a log file containing data (movie & category) from rows that have been inserted into "MADE_MONEY" table into the given "LOG_DATA" table. Use a trigger to accomplish this goal.

## Notes/Tips

Here are a couple of more notes/tips:

- NOTE: MySQL does not support CHECK CONSTRAINTS (at least up to MySQL versions in 2017), so you will have to use only triggers (before/after) to enforce constraints as well (Question 3, 4). For more references (and solutions), Google "*MySQL CHECK CONSTRAINTS using Triggers*".
- As discussed in class, triggers are computational mechanism that enable databases to have a reactive type of behavior, in accordance with the ECA (on Event, if Condition, then Action) paradigm.
- Start working on your problems (esp. triggers) "bottom-up" – e.g., look at the condition and write a separate SQL query that will implement it (just to test its correctness) via "New Query" button at the top-left of the screen.
- For detailed syntax for triggers in MySQL, look at the following links: **https://dev.mysql.com/doc/refman/8.0/en/trigger-syntax.html.**

## Collaboration

Lastly, and most importantly, a note on what is allowed and what is not allowed in terms of collaboration. It is OK to speak in generalities about how one might approach these problems with others. In other words, you can bounce ideas off of one another. However, it is NOT OK TO DISCUSS SQL ANSWERS & STATEMENTS AT ALL, as it relates to these problems, with anyone else in class. It is NOT OK to look at anyone else's answers, or to specifically discuss answers to the problems with anyone else. In short, two sets of answers to the given problems that are turned in with the same SQL statements will be viewed with extreme suspicion (and will be penalized accordingly if cheating is proved).

## Submission Guidelines

The assignment is due by the midnight on the day mentioned on the front page. Please submit the assignment electronically via Canvas, but be sure to send the files in a format that we can read (.txt, .doc, .pdf, etc.). There are three kinds of submissions for this assignment:

(1) Submit the syntax for the trigger(s).

(2) Submit your observations (success/error/any change of values, etc.) as to what happens in the database when you perform the following statements (after triggers are created):

a) Insert a new movie, with values ("IRON MAN", 1000000, 2008-05-02, "ACTON") in MADE_MONEY table.
b) Update the CATEGORY of the movie "Fight Club" to "Horror in MADE_MONEY table.
c) Insert a new tuple in APPEARED_IN table, with values ("Matt Damon", "Bruce Almighty").
d) Insert a new tuple in MARRIED, with values (1, 2015-06-26).
e) Insert two new tuples in MADE_MONEY, having values ("Most Welcome", 8000, 2012-07-07, "Action") and ("Speed", 9000, 2010-03-28, "Comedy").
f) Insert a new tuple in MADE_MONEY, having values ("Hangover", 1500000000, 2011-03-05, "Comedy").
g) Insert a new tuple in DIVORCED, with values (6, 2004-01-01).

(3) Submit the final state of the LOG_DATA table after performing the above statements. (*If trying for the extra credit*)

You also can (and may be should) optionally upload another text file (apart from the answers) where you can specify any additional assumptions you have made, or describe your method briefly in case it is not producing the desired result.

# Late Policy

1 day is 10% off
2 days is 20% off
We will not grade any submission done after that, i.e., Wednesday, November 06, 2018.