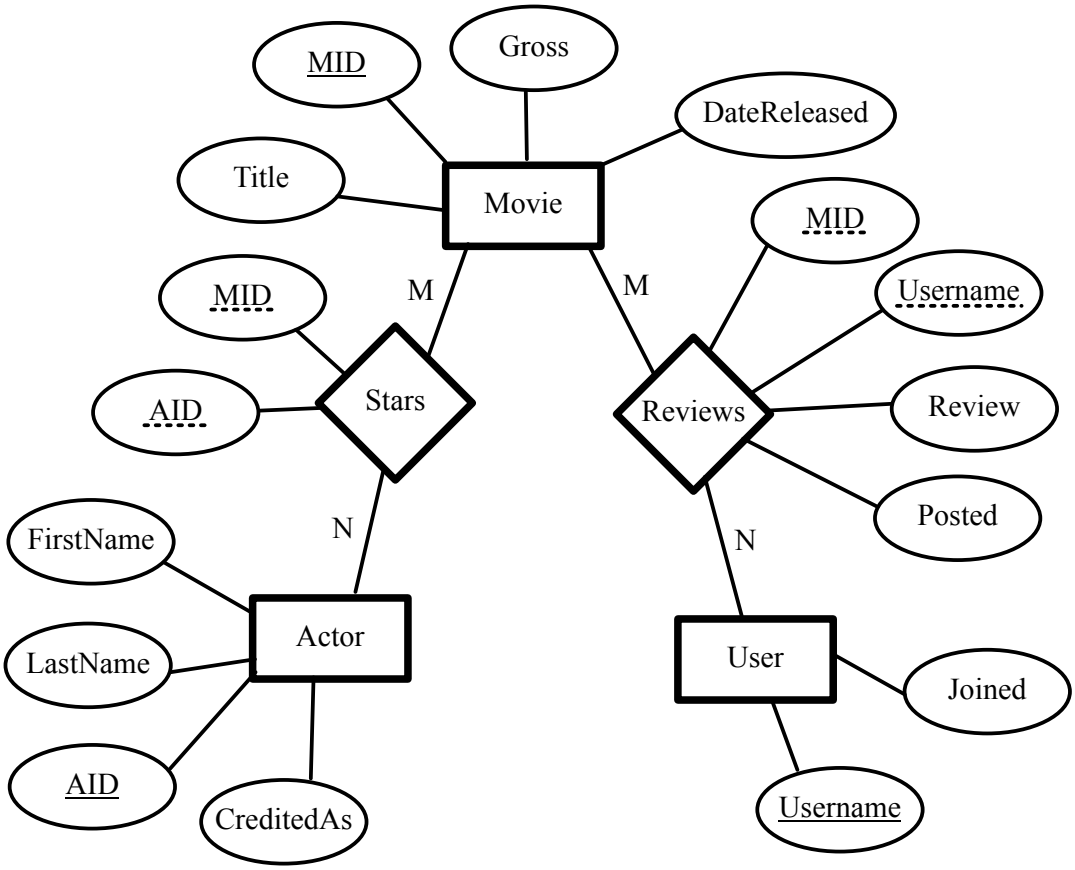
Assignment 04

CS 364 | Fall 2017

For this assignment, you will be developing the SQL statements to set up and populate a database, followed by developing a Java program to access and modify the database. You will be submitting a single `.sql` file and an Eclipse project archive (i.e., `.zip` file). Below is the information for part 1; part 2 will be released later. This assignment uses MySQL and Java. Your assignment (the combination of parts 1 and 2) is due **November 15** at 11 PM.

Context

This assignment deals with the development of a database for a movie review website. Below is an ER diagram that describes the structure of the database.



Additional information is given below for each attribute. You should assume attributes are required unless stated otherwise:

- Movie
  - MID: An auto-generated ID value of type `INT`.
  - Title: A title that is no longer than 127 characters.
  - Gross: A currency value that is no more than $99,999,999.99. Should be capable of holding two decimal places.

  - DateReleased: A month/day/year values of type `DATE`.

- Actor
  - AID: An auto-generated ID values of type `INT`.
  - FirstName: A first name that is no longer than 31 characters.
  - LastName: A last name that is no longer than 31 characters.
  - CreditedAs: An optional attribute for listing an alternate name an actor is credited as. Will never be longer than 31 characters.
- User
  - Username: A username that is no longer than `15` characters.
  - Joined: A `TIMESTAMP` value that records when the user joined the website.
- Reviews
  - Username: A foreign key referencing the username from the `User` table.
  - MID: A foreign key referencing the MID from the `Movie` table.
  - Review: A textual review of the movie that is no longer than 255 characters.
  - Posted: A `TIMESTAMP` value that records when the review was posted.
- Stars
  - MID: A foreign key referencing the MID from the `Movie` table.
  - AID: A foreign key referencing the AID from the `Actor` table.

Part 1: Building the Database

In the MySQL workbench, write the SQL statements to create the `MovieReview` database outlined in the ER diagram; make sure that you set this as your default schema early in the script. Pay careful attention to naming (you should match the naming/capitalization outlined above *exactly*), datatypes, properties (e.g., `PRIMARY KEY`), and foreign key relationships. If you find that you have made a mistake, you might find it easier to drop the entire database/individual tables, but please do not include these drop statements in your final script.

Next, add `INSERT` statements to your script to populate your database with the below tuples. Note that not all tables will be populated. Be sure to comment your code to indicate a separation between database/table creation statements and insert statements.

`Actor`

AID	FirstName	LastName	CreditedAs
1	John	Smith	
2	Emily	Jenson	Em Jenson

`Movie`

MID	Title	Gross	DateReleased
1	The Wild West	6502831.03	2004-10-05

`Stars`

AID	MID
1	1
2	1

`User`

Username	Joined
movieJunkie	2001-08-02 13:45:45

Note that, except for numeric datatypes, all data should be surrounded by quotes for insertion, and the data should be typed in exactly as shown in the above table (e.g., the `Joined` attribute in `User` should be formatted exactly as shown. Also note that there should be specific auto-generated values for some tuples/attributes, which will require ordering your insertion statements in a particular way.

When you are confident in your solution, execute your script and save your file as `db.sql`; all of your statements should be in this file, and you should be sure the ordering is appropriate if the entire script were run. Once you have the database created, navigate to "Users and Privileges" in MySQL. Create a new user called `grader` with the password `gradingpassword`; make sure they have access to your new database.