CS 377: Database Systems

Project #1

Wednesday, February 17, 2016 at 11:59 pm

SUBMISSION: Please submit the project electronically via Blackboard before 11:59 pm. Any submissions afterwards will be marked late and deducted accordingly unless you create a README file that specifies that you are using one of your 2 late assignment exceptions. You should use one file for each portion of the problem and follow these guidelines:

- If you have comments in your SQL query, use C style comments using /* < comment > */
- Any text that is not within the comment brackets will be assumed to be part of your SQL command, and if the file does not execute properly, points will be deducted.
- Each file should be named as "<netID>-project1-<problem>-<part>.sql". For example, the instructor's answer to part b of question 2 would have the file "jho31-project1-2-b.sql".
- At the top of each of your submitted SQL files should be the honor code.

```
/* THIS CODE IS MY OWN WORK.
IT WAS WRITTEN WITHOUT CONSULTING CODE WRITTEN BY OTHER STUDENTS.
_Your_Name_Here_ */
```

1. **Online Bookstore** (50 points): This problem is based in the Online Bookstore exercise from Homework #1. Your task is to design and implement the Relational Model of the Online Bookstore in MySQL. To perform this problem, you need to download and install MySQL from the following URL: `http://dev.mysql.com/downloads/mysql/`. The instructions for downloading and configuring MySQL are described in this video: `https://www.youtube.com/watch?v=iP1wOSsKjW8`.

   (a) (15 points) Design the relational model for the online bookstore using Figure 1 as a reference. If you drew the relational model by hand, make sure to scan it and include it as a pdf or image.

   (b) (15 points) Create the tables (implement the relations from the previous part) using MySQL. Write your commands in a SQL file named "<netID>-project1-1-b.sql" which follows the guidelines listed above.

   (c) (10 points) Insert 3 records in each of the relations of the ER model. Put the commands in a file named "<netID>-project1-1-c.sql" which follows the guidelines listed above.

   (d) (10 points) For each sold book (you can consider it sold if it is in the shopping cart), show the name of the customer, the book tittle, book name, and address of the publisher. Print your answer sorted by customer name.

2. **IMDB Database** (50 points): A copy of the IMDB database (`www.imdb.com`) has been loaded into the MySQL database on cs377spring.mathcs.emory.edu[1]. You can query the database either via the MySQL client command line or a SQL GUI editor (e.g., MySQL Workbench) using the user 'cs377' with the password posted on Piazza. Detailed instructions for accessing the database via the command line will be provided on Piazza. Our IMDB database contains 6 relations which will be used for the second part of the project.

[1]Not all the movies on the actual website will be reflected in this database. But the website will provide a fairly reasonable sanity check for your results.

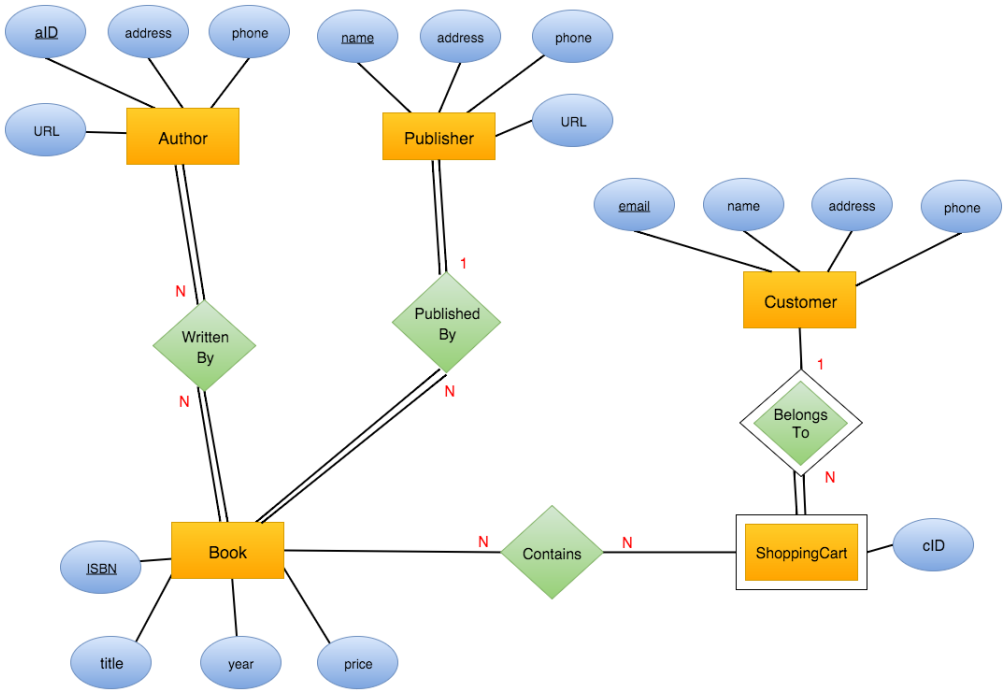

Figure 1: Online Bookstore Entity-Relation Model

- actor(<u>id</u>, fname, lname, gender)
  - id = unique identifier for each actor
  - fname = first name of the actor
  - lname = last name of the actor
  - gender = gender of the actor

  This relation contains information on the actors that can be found in the IMDB database.
- movie(<u>id</u>, name, year)
  - id = unique identifier for each movie
  - name = name of the movie
  - year = year the movie was released

  This relation contains information on the movies that can be found in the IMDB database.
- director(<u>id</u>, fname, lname)
  - id = unique identifier for each director
  - fname = first name of the director
  - lname = last name of the director

  This relation contains information on the directors of the movies that can be found in the IMDB database.
- genre(mid, genre)
  - mid = foreign key referencing movie(id)
  - genre = the genre of the movie

  This relation contains information about the genre (classification) of the movies in IMDB.
- casts(pid, mid, role)

  - pid = foreign key referencing actor(id)
  - mid = foreign key referencing movie(id)
  - role = name of the character in the movie

  This relation contains information about the actors in each movie and the role they play.

- movie_director(did, mid)
  - did = foreign key referencing director(id)
  - mid = foreign key referencing movie(id)

  This relation contains information about the director(s) for each movie in IMDB.

Write SQL queries to answer the following questions. Each query should be in its own file, named using the convention '<netID>-project1-<problem>-<part>.sql", and follow the guidelines set forth above.

(a) What genre(s) does the movie titled "Despicable Me" belong to?

(b) List all the animation movies that were released between 2011 and 2013 ordered by the year (2011 to 2013) and movie title (A to Z).

(c) Name the thriller movie(s) and the associated year of release directed by Steven Soderbergh sorted by the year (most recent first).

(d) List the movie(s) and the respective role(s) that Steve Carell was a part of as an actor.

(e) List the actors sorted by first name (A to Z) that have been directed by Woody Allen? Make sure to remove any duplicate names from the list.