Introduction to Database Design, F2016
IT University of Copenhagen
Carsten Schürmann

Project 1

This project assignment is a group assignment.

Wednesday, 28 September 2016, 23:55

Rules:

- Submit your groups work as a single PDF file with a file name that includes the groups name.
- Upload your work on `learnit.itu.dk`.
- Late hand-ins will not be considered.
- The grade of all project assignments will also be averaged and determine the grade of assignment 4, which also counts with 15% towards your grade.

In this project, which consists of three parts, you are going to work with a movie database. The first step is to make a suitable data model, documented by an E-R diagram and implemented as an SQL schema definition. Later on you will be working with actual data about movies. An anonymous student suggested the following relation schemas for some of the information:

```
Person(id,name,gender,birthdate,deathdate,height)
Movie(id,title,year,color,country,language,imdbRank)
Contract(personId,movieId,isActor,isDirector)
```

To extend this to a more comprehensive data model, use The Internet Movie Database (IMDB), www.imdb.com, for inspiration. The database should store information about actors/actresses, directors, writers, movies, genres, awards, ratings, etc. Your design should incorporate the above to the extent possible. In your initial design, there is no need to worry about database performance or other things. Note: IMDB stores lots of information. You cannot include all types of data in your model, hence you have to decide which information to include and which to exclude. Try to choose the most interesting data, both in terms of the application area and in terms of the model. E.g. it is not necessary to have 10 attributes describing aspect ratios and other technical details of a movie. Feel free to include additional information not found in IMDB, if you like. The final relational model should not include more than about 10-15 tables and 5- 10 tables can be enough, depending on your choice of tables and what kind of data you choose to store in your database. You should aim to create a design that is in normal form (BCNF).

Question 1: (30 points) Devise a model of your database in the E-R notation of RG. Remember to include all attributes, cardinality constraints, and keys. You probably have to make assumptions about the data. Include a description of the assumptions and choices you have made.

Question 2: (20 points) Give a corresponding schema for your movie database (see RG, page 59) This should be what you get by following the method for translating an E-R model to a relational one.

Question 3: (20 points) Translate the relational schema into the SQL/DDL syntax, You should make sure that your schema is accepted by the MySQL DBMS. (Install MySQL on your own machine using the links on the course home page, or create an account at mysql.itu.dk.) The schema must include

- Names of relations,
- column names and types,
- primary keys and possibly other candidate keys, and
- foreign keys constraints
- all other constraints that you deem necessary to guarantee the continuous integrity of your database.

Question 4: (30 points) Give an analysis of keys and functional dependencies in all your relations. This should be detailed enough that it would convince another student who knows about normalization that your design is in normal form. For example, you could address the top-10 reasons why someone might doubt that your design is in normal form. If you end up with a design that is not in normal form, go back and modify your E-R model to correct this.