

Hand-in 1: Database design

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. You would like to make a design that incorporates the above to the extent possible, while of course avoiding bad database design. In the initial design, database performance should *not* be considered.

Note: IMDB stores lots of information. You cannot include all types of data in your model, hence you have to choose which information to include and what 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).

To be handed in

1. A front page with your group number, and names of group members who contributed.
2. A model of your database in the E-R notation of RG, or alternatively in the notation supported by MySQL Workbench or Microsoft Visio. 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.
3. A corresponding schema for your movie database, written in SQL. This should be what you get by following the method for translating an E-R model to a relational one. 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.
4. 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.

Course goals covered by this hand-in

After the course the students should be able to:

- Define a database design by E-R modeling, using the concepts entity, attribute, key, cardinality, and relationship.
- Suggest a database design according to the relational model, and present it as an SQL schema, using the concepts key, type, and constraint.

The first hand-in should be handed in by each group using the LearnIT system no later than

Friday September 21, 23.59 CET.

It suffices that one group member uploads each hand-in as a single PDF file with a file name that includes the group number (to be announced on LearnIT). Later, the submitting student should communicate any feedback received through LearnIT to the other group members. Parts of the hand-in can be written by hand and scanned, but it has to be handed in electronically.