

*The overall goal in the whole mini-project is to create and work with a movie database in five incremental steps. The first step is to identify the information that is going to be represented as well as special requirements. Then, the information is modeled in an ER diagram and mapped to a relational schema. The database schema is then refined and normalized, so that it can be instantiated in PostgreSQL. After filling database with data, it will be queried and optimized.*

## Refinement, normalization, and SQL-DDL

In the previous selfstudy you have created a relational schema. In this selfstudy, the main task is to extend it and improve it by identifying functional dependencies and applying normalization.

*The normalized relational schema will be the basis for the following selfstudy.*

Assume you have a client who wants to use your database and is interested in the following pieces of information. You kindly agreed to check your database schema and adapt it if necessary<sup>1</sup>. Thus, your database should be able to store at least

- information about movies: title, language, year, director, writer, cast, ratings, genre, connections<sup>2</sup> to other movies, awards
- information about actors: first name, last name, birthday, year of death, movies that he/she acted in, awards
- information about directors: first name, last name, birthday, year of death, directed movies, awards
- information about writers: first name, last name, birthday, year of death, movies (screen-plays) written, awards
- information about users (persons contributing ratings): user name, ratings about movies

### Steps

- Adapt your design according to the feedback on the previous report and extend it if necessary according to the above requirements.
- Identify functional dependencies in your data.
- If necessary, transform your schema into 3NF and formally show that the result is.
- Determine for each obtained relational schema the highest normal form it still supports, i.e., check if your relations are also in BCNF.
- List the SQL statements that create all your tables properly and ensure the functional dependencies that you have identified.

*Hint: Consider UNIQUE NOT NULL constraints to enforce candidate key characteristics.*

<sup>1</sup>If you have used the list of questions attached as preview in the previous self study, all this information should already be in your schema

<sup>2</sup>A connection is a reference to another movie (sequel, parody, etc.).

Of course, if you identify flaws in your original design: revise your ER diagram and iterate.

## Report

- Extend your previous report with a new chapter – each report you hand in should hence contain the complete “history” of what you have done in earlier self studies.
- Current (revised) ERD and derived relational schema
- A list of functional dependencies
- A list of normalized relations (based on the ones derived from the ERD) along with
  - their definitions (name, attribute names and types, keys)
  - information on how and why you derived them from the original relations (i.e., note down the steps of the algorithms that you applied)
  - information on the highest normal form that a relation fulfills along with an explanation of why you come to this conclusion
- A list of SQL statements that can be used to create your tables - make sure your statements include information about keys.
- Reflections on the initial design from self studies 1 and 2. Questions you should consider are:
  - What are the differences to your previous design?
  - Why did these differences occur?
  - What advantages does the new design have over the old one?
  - ...

## Course goals covered by this self study

- Create and evaluate a database schema that adheres to normal forms (logical design)
- Improving the quality of an existing database design
- Using SQL to create tables

**Selfstudy: 18.03.2014**

The report must be handed in via Moodle no later than  
**24.03.2014, 23:55 CET**