Institut für Informationssysteme
Technische Universität Braunschweig

Technische Universität Braunschweig
**Institut für Informationssysteme**
http://www.ifis.cs.tu-bs.de

Silke Eckstein, Benjamin Köhncke, and Joachim Selke

# **SQL Lab: Assignment 4** (due to 25./26.01.2011)

## General Information

This week, you will "repair" some shortcomings of the IMDB data model we used in the last assignment and combine the repaired data with additional information. The following schemas should be used:

- IMDBRAW
  - Includes the IMDB tables you already used in the last assignment.
- IMDBEXTRA
  - *Mapping*
    - imdb_id: new internal used ID
    - title_id: ID used in IMDBRAW schema
  - *Review*
    - Includes user opinions for movies.
    - The texts are stored as CLOBS.
- MOVIELENS
  - *Mapping*
    - movie_id: ID used for MOVIELENS movies
    - imdb_id: Corresponding ID in IMDB
  - *Rating*
    - Rating for MOVIELENS movies

## Task

All of the following tasks should be performed for the **movies included in MOVIELENS that were released between 2000 and 2005**. All other tables should only include data related to these movies (e.g. Persons or Reviews). Your current storage limit is 1.5 GB – if you receive a storage limit warning you imported the wrong data!

### *Create Tables*

Use SQL CREATE TABLE statements to re-create tables which can store the respective data inside your own DB2 schema (i.e. the one matching your login name, e.g. DBLAB73). **Include all statements** you used in your result sheet. Consider the following requirements:

- Create a new table in your own schema (DBLABXX) and insert all IMDB movies that are included in the MOVIELENS movies table. Use only the movies created between 2000 and 2005. You have to use new IDs for the IMDB movies which can be found in the respective mapping table in the IMDBEXTRA schema. Attributes that should be included are: id, type, title, year, numeral, suspended.

- All **persons** involved in movie-making (actors, actresses, directors; all other participants can be ignored) should be unified into a single person table
    - If known, also the gender has to be stored. Please note that the data provided by IMDB contains some errors that have to be fixed. Some persons are included in the *actors* table as well as in the *actresses* table.
- A table '*participation*' is needed to store the relation between persons and movies.
    - It should be specified how a person was involved (director or cast) in a movie.
- Also the respective data from the '*ratings*' table has to be transferred. Do not forget to update the IDs.
- Furthermore, the MOVIELENS *rating* table and the IMDBEXTRA *review* table have to be transferred and updated (new IDs).
- Each created table must define a **primary key.**
- Every time information in multiple tables is related, define **foreign keys.**
- Add simple consistency constrains to your tables (i.e. UNIQUE, NOT NULL, simple CHECK conditions, …) wherever appropriate.
- **Do not import any data which is not related to any MOVIELENS movie from 2000 to 2005!**
- Finally you should end up with the following tables: movie, person, participation, imdb_rating, movielens_rating and review.

### *Query Data*

Create SQL queries for the following questions. Use your own schema! **Include** the used **statements** and the **answer** in your result sheet.

a) Create a table including all average values of all MOVIELENS ratings per movie.
b) Compare the ratings of IMDB and MOVIELENS and print the top 100 movies with the biggest rating differences regarding both schemas. Please note, the rating tables use different scales! Therefore, you need to find a way to make the ratings comparable. Furthermore, consider only movies that have 'enough' ratings.
c) Count the number of movies where the director also participates as an actor.
d) How many reviews for the movie 'Lord of the Rings: Return of the King' include the words Frodo and Sauron.
e) Which Lord of the Ring movie has the highest MOVIELENS rating?
f) How many IMDB votes are available for movies with Brad Pitt and what is the average score?

**Email** all used queries or results to your tutor. In addition, also **print** your queries and answers on paper and bring them to the respective lab meeting.