Univ.-Prof. Dr.-Ing. Matthias Boehm
Graz University of Technology
Computer Science and Biomedical Engineering
Institute of Interactive Systems and Data Science
BMK endowed chair for Data Management

1 Data Management WS20: Exercise 01 – Data Modeling

Published: October 12, 2020 (last update: Oct 19, simplified last sentence Task 1.1)
Deadline: November 03, 2020, 11.59pm CET

This exercise on data modeling aims to provide practical experience in Entity-Relationship (ER) modeling, ER-relational mapping, and relational normalization. The expected result is a zip archive named `DBExercise01_<student_ID>.zip`, containing the partial results of the individual sub-tasks, submitted in TeachCenter.

1.1 ER Modeling (12/25 points)

Create an ER diagram in Modified Chen (MC) notation—including entity types, relationship types, attributes, cardinalities, and keys (create surrogate keys if natural unique identifiers are missing)—for managing the movies dataset[1]. It is up to you if you use existing tools for data modeling or draw this by hand. There are multiple correct ways of modeling this discourse, however the diagram should capture the following information:

- A *movie* has a unique ID, an original title, an English title (translation), a release date, a runtime in minutes, and an optional homepage. Each movie had a certain budget (production cost) and revenue (income yielded so far). Based on the budget, movies are classified as low-, medium- and high-budget productions.

- A movie can be produced in arbitrarily many *countries*, each being described by an ISO 3166-1 alpha-2 country code, and name. Movies are further described by a list of *genres*, the *language* of the original title, and a list of spoken *languages*. Every language has an ISO 639-1 code, and an endonym (i.e., internal) name.

- The cast of a movie is composed of one or multiple *actors*, each playing one or multiple roles (characters) in the movie. A single actor might belong to the cast of one or many movies, has a unique ID, a non-unique name, a gender (1 female, 2 male, 3 non-binary), and lives in exactly one *country*.

- Every movie might receive an unlimited number of user *ratings*. A single rating of a movie is characterized by a score (scale 0.5-5), a date, and an anonymous user ID. A single rating refers to exactly one movie.

Partial Result: ERDiagram.pdf

---

[1] `https://github.com/tugraz-isds/datasets/tree/master/movies`

1.2 Mapping ER Diagrams into the Relational Model (10/25 points)

Create a relational schema for the ER diagram designed in Task 1.1 and bring it in third normal form, assuming functional dependencies from country and language codes to their names. This schema should include the relations and typed attributes, as well as all primary and foreign keys. It is up to you if you provide either a SQL script (CreateSchema.sql) with `CREATE TABLE` statements, or provide a text schema (Schema.txt) in the following text notation (please, use this notation with unchanged parentheses and delimiters, and limit yourself to common data types because this sub-task is subject to automatic grading):

\<Table\>(\<Primary key\>:\<type\>(PK), \<Attribute\>:\<type\>, ..., \<Foreign key\>(FK))

Partial Result: Schema.txt, or CreateSchema.sql

1.3 Relational Normalization (3/25 points)

Briefly explain why the relational schema from Task 1.2 is indeed in the third normal form. The resulting description (text file NFExplain.txt) should explain the conditions for the first, second, and third normal forms, as well as why the provided schema meets these conditions.

Partial Result: NFExplain.txt