Introduction to Informatics
WiSe 2024/2025

Prof. Dr. Ingo Scholtes
Chair of Informatics XV
University of Würzburg

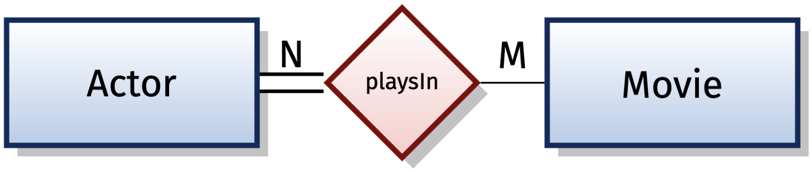
Exercise Sheet 09

Published: January 15, 2025
Due: January 23, 2025

Please upload your solutions to WueCampus as a scanned document (image format or pdf), a PDF document, and/or as a `jupyter` notebook.

1. **Comparing RAID Configurations** You have 6 hard drives, each with a capacity of 8 TB. Your goal is to explore and compare different RAID configurations (RAID 0, RAID 1, RAID 5, and RAID 6) to understand their properties and trade-offs.
   (a) Calculate Usable Storage Capacity For each RAID configuration (0, 1, 5, 6), calculate the **usable storage capacity** in terabytes (TB).
   (b) Explain Fault Tolerance For each RAID level, describe how many disk failures the system can tolerate before data is lost. Compare the fault tolerance of RAID 0, RAID 1, RAID 5, and RAID 6.
   (c) Analyze Storage Efficiency Calculate the **storage efficiency** (percentage of total raw capacity used for data storage) for each RAID level. Discuss the trade-offs between redundancy and usable storage.

2. **ER model**
   (a) In the lecture you saw the followig Many-to-many Relationship between Actor and Movie:



   How would the diagram change if Actors are not necessarily playing in Movies?
   (b) Create an Entity Relationship diagram for a University where you want to store information about the students, teachers, and courses. Think about useful Attributes and Relationships.
   (c) Translate the ER diagram above into a relational schema.

3. **SQL Queries** We are looking at the movie database `movieDB.db` from the lecture.

   When you work with relational databases, data is often spread across multiple tables. `JOIN` is used to combine rows from these tables based on related columns.

   Here:

   - The `Movie` table contains information about movies, including their genre.
   - The `directs` table links movies to directors through `MovieID` and `PersonID`.
   - The `Person` table stores the director names.

By using JOIN, you connect these tables to find the required information. This avoids unnecessary data and ensures that relationships between tables are properly maintained.

The syntax is as follows:

```
SELECT column1, column2, ...
FROM table1
JOIN table2
ON table1.common_column = table2.common_column;
```

(a) Open the movie database movieDB.db from the lecture with DB Browser for SQLite.

(b) Rewrite the following example from the lecture using JOIN statements:

```
SELECT Person.LName FROM
Movie, directs, Person
WHERE Genre = "Phantasy"
AND Movie.MovieID =
  directs.MovieID
AND Person.PersonID =
  directs.PersonID;
```

(c) Add at least 3 more Movies with director and actors to the database. Two of the movies should have the same actor.

(d) Write an SQL Query that returns the names of the movies with this actor.

(e) Try to execute an SQL Query that deletes a Movie from the Movie table. What do you observe?

(f) Set up your own database for the example in task 2.