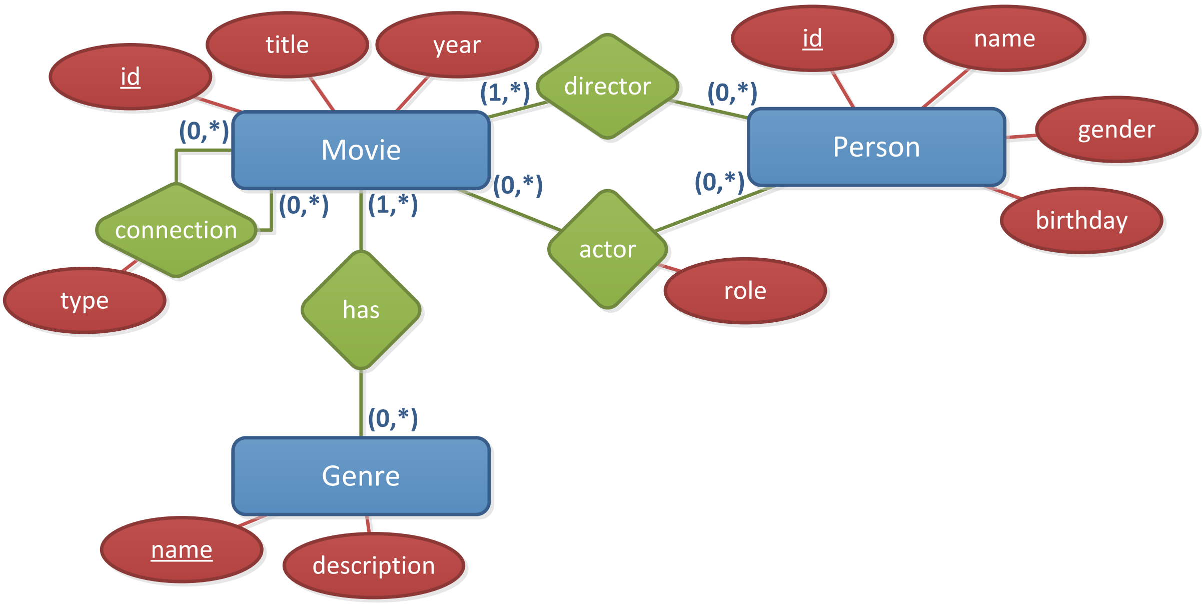
Exercise Sheet 8: SQL

(until Thursday 13.12.2012)

Please note that you need **50%** of all exercise points to receive the “Studienleistung”. Exercises have to be turned in until **Thursday** of each respective week and must be completed in teams of two students each. You may hand in your solutions either on paper **before the lecture** or into the mailbox at the IFIS floor (Informatikzentrum 2nd floor). Please do not forget to write your “Matrikelnummer” and your tutorial group number on your solutions. Your solutions may be in German or English. Please note: To pass the “RDB I Modul” you need both the exercise points and the exam!

Exercise 8.1 (16 points)

Following conceptual schema describes a simple movie database.



The schema describes movie data, including the movie's title and the production year. Also actors and directors are described, by introducing the relationship types "actor" and "director", which connects persons with movies. The actor relationship type also includes the role played by the person in the respective movie. As neither persons nor movies can be uniquely identified by a set of their "natural" attributes, an id was introduced. Persons are further described by their name, gender and birthday. Gender can take the values 'f' and 'm' for female and male and birthday is formatted in little-endian style with dots as separators (e.g. '21.11.1968'). Movies can also be connected to each other. For example one movie can be a "parody" or "sequel" to another movie. In that case the attribute "type" of the relationship type "connection" would yield the value "sequel" or "parody". A movie can also have several genres, which are further described by a description attribute.

With respect to the given conceptual schema the following relation schema was derived:

Movie(id, title, year)
Person(id, name, gender, birthday)
Genre(name, description)
actor(person → Person, movie → Movie, role)
director(person → Person, movie → Movie)
hasGenre(movie → Movie, genre → Genre)
connection(from → Movie, to → Movie, type)

Starting from the above relation schema, please provide SQL statements returning the following results:

a. Names of all persons who played a role in a "parody" of any movie. (1 point)
b. Names of all persons who were born in July. (2 points)
c. Titles of all movies that have been directed by more than one person. (2 points)
d. Id, name and number of movies the person acted in for each person. The result should also include those persons who never played a role in a movie yet. (2 points)
e. Titles of all movies in which "Ted Codd" played a role but "Ray Boyce" did not. (2 points)
f. The names of all persons, who acted and directed in the same movie for at least two times. (3 points)
g. Considering the average number of movies persons played roles in. Return the names of all persons who played roles in more movies than the average. (4 points)

Exercise 8.2 (5 points)

Briefly explain in your own words:

a. What is the difference between the WHERE and the HAVING clause? (1 point)
b. Would SQL have the same expressiveness if the HAVING clause wouldn't exist. (1 point)
c. What is the difference between correlated and uncorrelated subqueries? (1 point)
d. What is the difference between a set and a bag? How can you enforce the DBMS to return a set instead of a bag? (2 points)