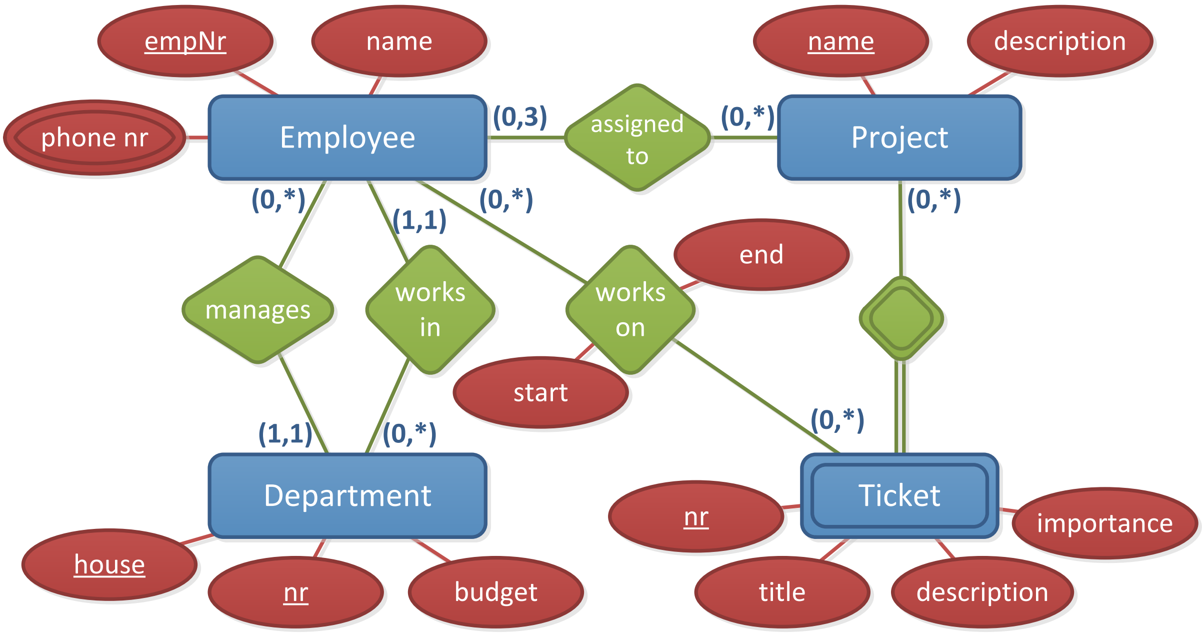
Exercise Sheet 9: SQL2

(until Thursday 20.12.2012)

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 9.1 (14 points)

Following conceptual schema describes a simple company database.



The company is divided into several departments. Each department is uniquely defined by combination of a letter and a number, where the letter describes the house (e.g. Department C-15 for house 'C', nr 15). Each department is furthermore managed by an employee. Each employee is assigned to a department and to several projects to work on, where the number of projects an employee works on must not exceed 3. In order to manage the tasks belonging to a project more efficiently employees are allowed to create tickets for the project. Each ticket has a number, which is only unique within the scope of the project. It also has a title and description as well as an importance level which can take the values 'minor', 'medium' and 'major'. At the time an employee starts to work on the ticket, the start date will be noted and as soon as the task described in the ticket is done, the end time is noted. Consequently the ticked is assumed to be closed.

Based on the given conceptual schema, please provide SQL statements to create the according tables described in the schema. The created tables should regard as much constraints as possible. Additionally list all constraints that are not regarded by the created tables.

Exercise 9.2 (6 points)

Considering a subset of the relation schema used last weeks:

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

Given are following SQL statements to create the tables described by the relation schema above as well as some data.

```
CREATE TABLE movie(
    id INTEGER NOT NULL PRIMARY KEY,
    title VARCHAR(255) NOT NULL,
    year INTEGER
)

CREATE TABLE person(
    id INTEGER NOT NULL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    gender CHAR(1) CHECK (gender IN ('f', 'm')),
    birthday CHAR(10)
)

CREATE TABLE actor(
    person INTEGER NOT NULL REFERENCES person,
    movie INTEGER NOT NULL REFERENCES movie,
    role VARCHAR(255) NOT NULL,
    PRIMARY KEY (person, movie)
)

CREATE TABLE director(
    person INTEGER NOT NULL REFERENCES person,
    movie INTEGER NOT NULL REFERENCES movie,
    PRIMARY KEY(person, movie)
)

CREATE TABLE connection(
    from_movie INTEGER NOT NULL REFERENCES movie ON DELETE CASCADE,
    to_movie INTEGER NOT NULL REFERENCES movie ON DELETE CASCADE,
    connection_type VARCHAR(255)
        CHECK (connection_type IN ('parody','sequel')),
    PRIMARY KEY(from_movie, to_movie)
)
```

person	id	name	gender	birthday
	11	Joe	m	15.02.1987
	37	Ann	f	08.09.1990
	51	Bob	m	22.01.1985
	85	Jill	f	08.11.1992
	100	Clyde	m	19.10.1984

director	person	movie
	51	2
	100	5
	51	5
	85	9
	85	11

movie	id	title	year
	2	Adventures with RDB	1998
	5	The mighty Oracle	1972
	9	The legacy of Codd	1998
	11	The legacy of Codd 2	1999

connection	from	to	type
	2	5	parody
	11	9	sequel

actor	person	movie	role
	51	2	student
	100	2	forest ranger
	51	5	janitor
	85	5	secretary
	100	9	RA hacker
	11	9	old man
	100	11	RA hacker
	11	11	old man
	37	11	Ted Codd

Based on the given statements and data, explain the consequences of the following operations:

a. `INSERT INTO connection VALUES (2, 5, 'sequel')`

b. `DELETE FROM actor WHERE role = 'forest ranger'`

c. `DELETE FROM movie WHERE title = 'Adventures with RDB'`

d. `INSERT INTO actor VALUES (6, 85, 'important looking man')`

e. `DROP TABLE person`