Project Databases: Assignment 1

For this first assignment, you need the information provided in the Data modeling (ERD) classes, and the Data modeling reader (which you can find on Moodle).

Assignment 1, part 1: ERD

First we design the database using an Entity Relationship Diagram (ERD). To come to a correct database design, it is important to first analyze the Someren case, and identify all required entities, relationships, and attributes. The Someren case can be found in the pdf file "**Project description – Someren.pdf**".

After all entities, relationships and attributes have been identified, draw the ERD with the appropriate symbols (entity: rectangle, relationship: diamond, attribute: ellipse). Use also an **attribute list** for all attributes.

As a final step in designing the ERD, add the cardinality to each relationship:

- add to each relationship the **Functionality** (1, n, m);
  (Functionality: the maximum number of occurrences for one entity in relation to another)
  Use phrases to avoid confusion (like: *"A certain … can occur <u>once</u> in relation to a …"*)

- add to each relationship the **Totality** (T, O);
  (Totality: the minimum number of occurrences for one entity in relation to another)
  Use phrases to avoid confusion (like: *"for <u>every</u> … this occurs in relation to …"*)

The ERD / database can be further supplemented in the coming weeks but for now this description is a sufficient starting point.

Assignment 1, part 2: Relational model

After designing the ERD, it should be finalized as a relational model than can be directly translated to a database.

Use the 2 conditions and 5 conversion rules below to convert the ERD into a relation database model (see the Data modeling reader for some examples).

<u>Condition 1</u>
Give each entity type and relationship type a unique name (except is_a).

<u>Condition 2</u>
All entity types have a primary key.

Conversion rule 1

An **n-to-m relationship** from an ERD is represented in a relational model by means of three tables: two for the entity types and one for the relationship type. In the table created by the relationship, reference keys are included to both entity types. These reference keys together form the primary key to the new table.

Conversion rule 2

In the case of an **n-to-1 or a 1-to-n relationship** from an ERD, the relationship type is not converted into a separate table. The 1-sided entity type is converted into a table that, in addition to the attribute types of the relationship, also contains a reference key to the n-side entity type. The other entity type remains unchanged.

Conversion rule 3

In the case of a **1-to-1 relationship** from an ERD, the relationship type is not converted into a separate table. One of the entity types is converted into a table that contains all attribute types of the relationship as well as a reference key to the other entity type. This reference key in the table is subject to a uniqueness restriction. The other entity type remains unchanged.

Conversion rule 4

When a table with a reference key participates in a total relationship, it is mandatory to fill the reference key in that table (not nullable).

Conversion rule 5

When a table without a reference key participates in a relationship totally, all key values must appear at least once as the value of the reference key in the other table.