Before seminar and laboratory work

Conceptual database design

Look at chapter 15 and appendix C in Connolly & Begg.

The *conceptual design* is about the process of construction of a model for an organization without taking notice of any physical constraints. The modeling can be done in the steps below:

- Identify the central entities (objects) in the organization. It could be customers, orders, suppliers and so on.

- Identify relationships between the entities. How are they related?

- Identify the attribute of each entity, for example customer No., customer name for the customer entity.

- Decide primary and secondary keys

- Control that you don't have redundancy in your model

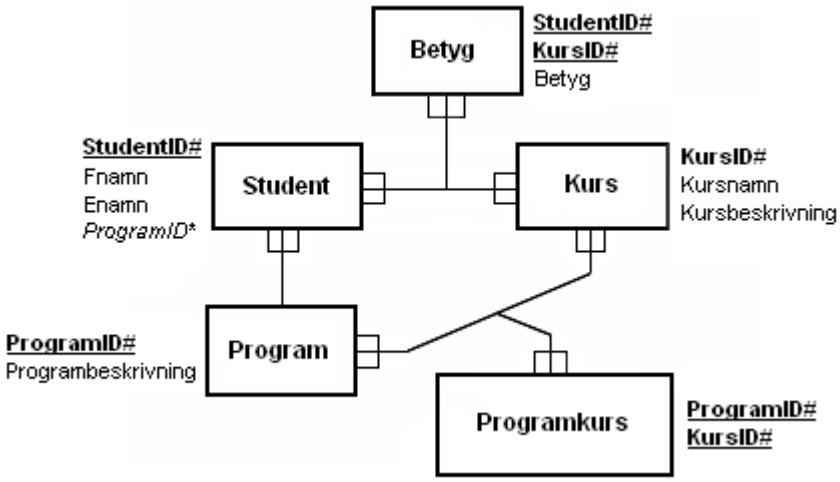
- Validate the conceptual model against the transactions needed

- Verify the conceptual model with the owner of the project (the organization), for example a customer if you are working as a consult.

What is a conceptual model?

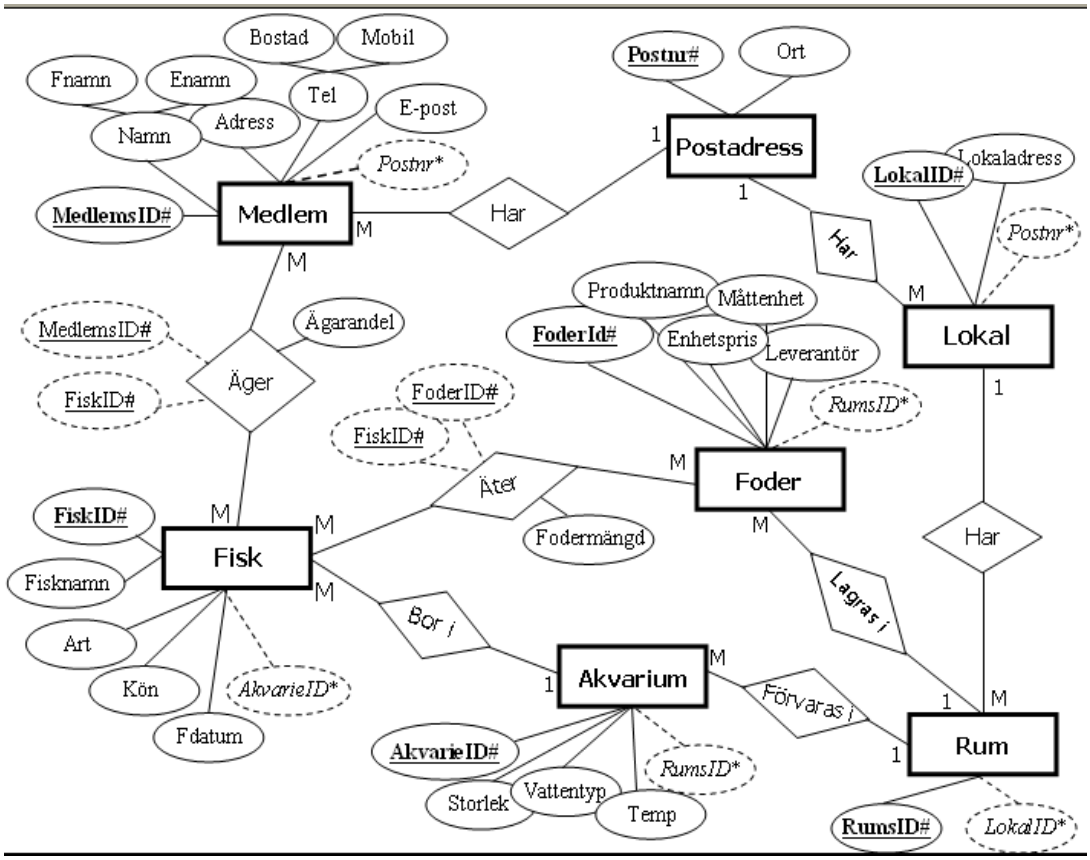
A conceptual model is the result of the modeling; a graphical description that shows entities, attributes and relationships. An example of an entity is (look at the model 1a below) is **Student**. An example of an attribute is **StudentID#,** Fnamn, Enamn and *ProgramID**. **StudentID#** is a primary key and *ProgramID** is a foreign key. In this case shall the personal identification number be used as value for **StudentID#** in the coming database system. It is unique for each student. Notice that it in some circumstances it is not legal to store a personal identification number in a database (for example the Swedish law "personuppgiftslagen", Pul).

Model 1a – an simple conceptual model



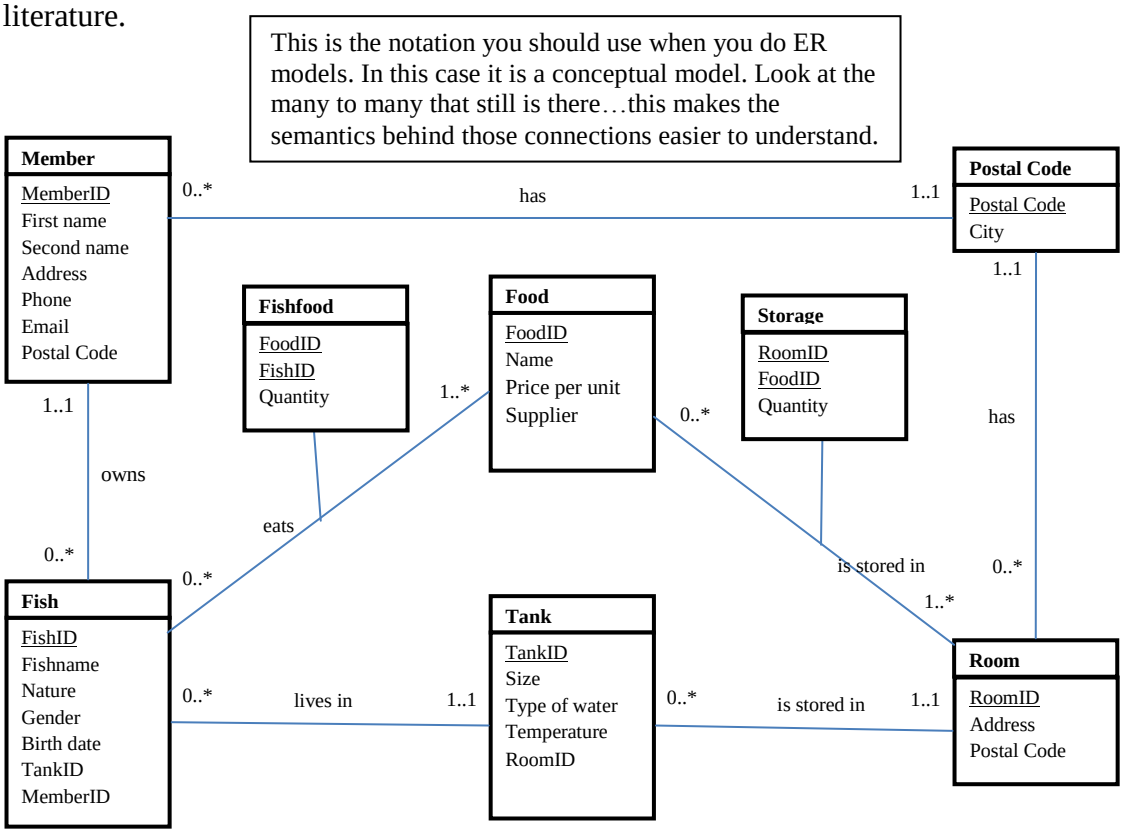
Model 1b – Entity/Relationship model

The notation in the model is not exactly as in Appendix C in Connolly & Begg.



Model 1c – Entity/Relationship model (UML-notation)

Look at chapter 12 in Connolly & Begg. Notation is not exactly as described in the literature.



Modeling rules etc.

How to identify the right entity and it's attributes?

- Analyse the organisation and identify the entities the database need to store data about. If it's a educational organization it could be students, courses, programmes (substantive)
- Examine the attributes which is needed to store data about the entity. For example the entity Student has atributes like personal identification number, name, telephone, adress, ...

How to discover relationships?

- Through associations like: a customer lay orders
- through seeking for structure like: orders contain products

How much details is needed?

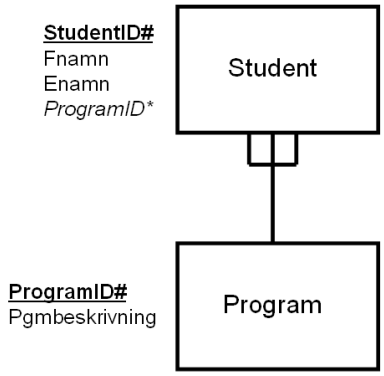
- Enough to describe the organization – the model need to have enough details to create a database normalized to at least 3NF.

Modeling of the 1:M relationship

In a 1:M relationship the entity at the M-side "borrow" the unique attribute from the 1-side (the primary key). The "borrowed" key is called a foreign key and is a referece to the related relation. The relationship can be read like this: *Many students can read a program, for example VIP or ASP, but each student can only*

read one program at the same time (a rule of the organization). In the simple model it is marked as a 1:M relationship through the ”fork” on the M-side.

Modeling rule 1:M relationship: example Student - Program



Relation example: Student - Program

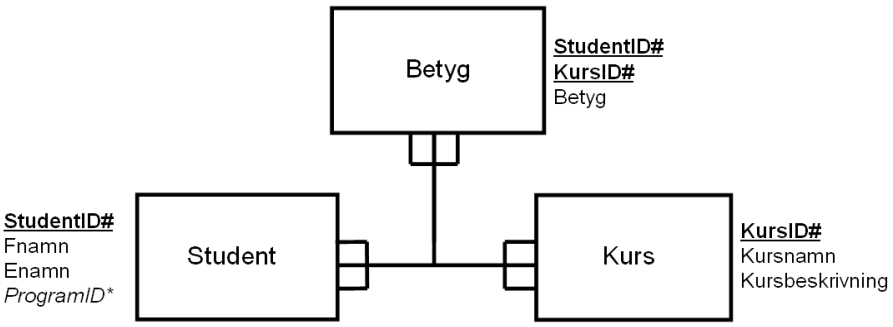
StudentID#	Fnamn	Enamn	*ProgramID**
8201139901	Anna	Ek	VIP07-10
8611221113	Hanna	Ström	IT106-09
8403200278	Bo	Ahlberg	VIP07-10

ProgramID#	Programbeskrivning
VIP07-10	Programmet syftar till.....
IT106-09	Programmet syftar.....
WEB05-08	Programmet.....

Modeling of the M:M relationship

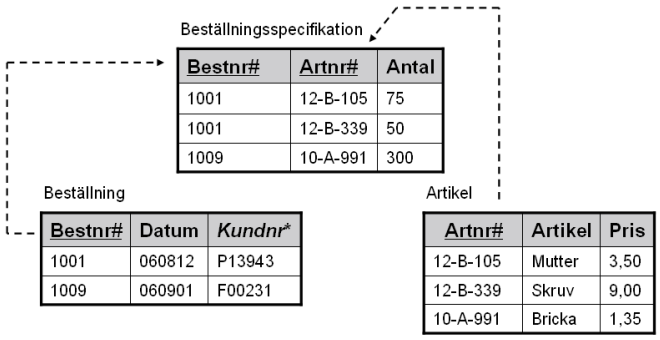
A M:M relationship can be described as two 1:M relationships that need another relation to work, it is a kind of ”relationship table”. The relationship in the example below can be read as: *A student can read and have result in many courses. A course can be read by many students.* To store data about the results we create a new relation (the relationship): Betyg.

Example – relation (relationship) Betyg



The relationship in the example below can be read as: An order can contain many products, and a product can be included in many orders. To solve this we create a new relation (a relationship); Beställningsspecifikation.

Example of tables – the relation (relationship) Beställningsspecifikation



Relationships is not always as easy to find as for example Student, Kurs or Artikel; In some cases you need a lot of work before you find them…