The Entity Relationship Model

1 Why a data model?

- A model: an abstract representation of something existing in the real world
- Models help make complex things understandable
- In databases:
  - DDL is too low level
  - not easily understandable by most users
- We have seen:
  - Relational data model
  - Relations between data → stored in tables
  - based on the concept of mathematical relations

2 Entity-Relationship Model

- Relational data model
  - Sometimes still too low level for big companies
  - we need a model of communication that is non technical and free of ambiguities
- Entity relationship model - a graphical description of the DB
- A **top down** approach to database design
  - Start with a set of requirements
  - identify the types of ″things″ that you need to represent data about
  - Identify the attributes of those ″things″
- Objective of the ER model
  - To help understanding of the nature and relationships among the data
  - To help deriving the tables in the relational data model
- Most important before building the ER model:
  - proper understanding of the problem domain (the scenario)
  - lack of understanding → wrong ER model → wrong DB design
- Basic concepts:
  - the important data objects (entities)
  - the important properties of the entities (attributes)
  - the associations between the entities (relationships)
- Furthermore - constraints on the entities, relationships and attributes
- Several notations for representing the ER model:
  - Crow’s foot notation
  - UML notation (unified modelling language)

3 Main components of an ER model

- An entity
  - any real thing that we recognise as a separate concern within the database
  - important enough to be represented separately
  - A rectangle represents an entity type
- Relationship
  - A named association between two entity types
  - Shown by a labelled line connecting two entities

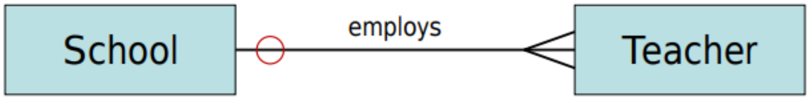
4 Cardinality

- The cardinality of a relationship - the number of entity occurrences that are related to a single occurence of an associated entity type through this relationship
- The cardinality of a relationship can be
  - one to one
  - one to many
  - many to many

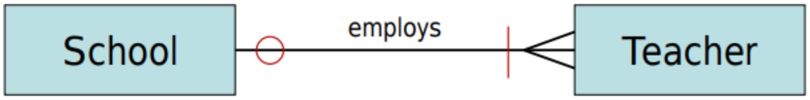
Many to one is not needed, as it is just the opposite way round as one to many.

5 Optionality and participation

- Optionality is where a relationship could or could not exist, and is denoted by a circle in the ER diagram



- if an entity participates optionally in a relationship:
  - it has partial participation
  - otherwise it has total participation
- In the ER diagram total participation is denoted by a vertical bar



6 Attributes

- The attributes of an entity
  - The set of all common characteristics that are shared by all entity occurrences of an entity type
- Diagrammatic representation of attributes have primary keys underlined
  - for every attribute one **labeled ellipse** attached to the entity rectangle
  - all attributes in the lower part of the entity rectangle

6.1 Determining the attributes

What are the appropriate attributes

- We first look for the natural characteristics of an entity
- an attribute can be associated with a value domain
- Single value attribute - holds only one value
- Multivalued attribute - holds many values
- Derived attribute - derivable from the value of a related attribute

7 Step by step construction of an ER model

1. Identify entities from the scenario/real world
2. Find relationships
3. Draw rough ER diagram
4. Fill in the relationships
5. Define primary keys and resolve many to many relationships
6. Identify attributes and map them to the entities
7. Draw full ER diagram showing keys, attributes, relationships
8. Check it reflects the real world system