Simpelt E-R diagram

Hvordan ser de tilsvarende relationer ud?
What Should an Entity Type Be?

**SHOULD BE:**
- An object that will have many instances in the database
- An object that will be composed of multiple attributes
- An object that we are trying to model

**SHOULD NOT BE:**
- A user of the database system
- An output of the database system (e.g. a report)
Examples

- The actor "Bruce Willis" should not be an entity type - there is only one instance.
- "Middle-aged actors" is not an entity type - could be a query
- Year is not an entity because it does not have multiple attributes (that we want to model, anyway)
- The title of a film is not an entity type, unless we want to associate more info with each title (like relationships among movies with that title)
Sample E-R Diagram (Figure 3-1)
Figure 3-10a Relationship type and instances - Relationship type (Completes)

Employee_ID Employee_Name Birth_Date Course_ID Course_Title Topic

EMPLOYEE Completes COURSE

Figure 3-10b Relationship type and instances - Relationship instances

Employee Course
Chen C++
Melton Java
Ritchie COBOL
Celko Visual Basic
Gosling Perl

Chapter 3 © 2005 by Prentice Hall 6
Figure 3-12b  Examples of relationships of different degrees - Binary relationships

- EMPLOYEE
  - Is_assigned
  - PARKING PLACE
  - One-to-one

- PRODUCT LINE
  - Contains
  - PRODUCT
  - One-to-many

- STUDENT
  - Registers_for
  - COURSE
  - Many-to-many
Basic relationship with only maximum cardinalities – Figure 3-16a

Default minimum is 0, default maximum is 1.

Mandatory minimum cardinalities – Figure 3-17a
Here, the date completed attribute pertains specifically to the employee’s completion of a course...it is an attribute of the relationship.

**NOTE:** Only one value for each relationship instance.
Problemsession (5-10 min)

- Vi vil designe en database til en lille butik med information om kunderne, ordrer og kredit:
  - Navne og adresser
  - Telefonnumre
  - Kundetyper (privat eller forretning)
  - Varer
  - Aktuelle ordrer
  - Kundenumre
  - Maximum kredit
  - Aktuel kredit

Tegn et muligt E-R diagram for databasen.
Figure 3-7 – A **composite** attribute

An attribute broken into component parts:

- **Street_Address**
- **City**
- **State**
- **Postal_Code**
Figure 3-12a  Examples of relationships of different degrees - Unary relationships
Figure 3-13a – A unary relationship with an attribute. This has a many-to-many relationship

Representing a bill-of-materials structure
Figure 3-12c -- A ternary relationship (with attributes)
Figure 3-13b: Representing a bill-of-materials structure - Two ITEM bill-of-materials structure instances

Mountain Bike
MX300

- Handle Bars
  HX100
  Qty: 1

- Transmission System
  TX100
  Qty: 1

- Wheels
  WX240
  Qty: 2

Tandem Bike
TR425

- Handle Bars
  HT200
  Qty: 2

- Transmission System
  TX101

- Wheels
  WX340
  Qty: 2

- Wheel Trim
  WT100
  Qty: 2
Entities can be related to one another in more than one way
Identifiers

- **Candidate Key** - An attribute (or combination of attributes) that uniquely identifies individual instances of an entity type.

- **Identifier (or “Key”)** – one particular candidate key that was chosen to uniquely identify entity instances.
Identifier design criteria

- Should not change in value
- Should not be null
- No “intelligent identifiers” (e.g. containing locations or people that might change)
- Substitute new, simple keys for long, composite keys
Figure 3-9a – Simple key attribute

The key is underlined

Chapter 3

© 2005 by Prentice Hall
The key is composed of two subparts

Figure 3-9b – Composite key attribute
Figure 3-8 – Entity with a multivalued attribute (Skill) and derived attribute (Years_Employed)

What’s wrong with this?

- **Multivalued**: an employee can have more than one skill
- **Derived**: from date employed and current date

- **Employee_Name**
- **Employee_ID**
- **Address**
- **Skill**
- **Years_Employed**
- **Date_Employed**
Figure 3-19 – An attribute that is both multivalued and composite

This is an example of time-stamping
Problem: Dependent_name not unique (not even together with Date_of_Birth)
Strong vs. Weak Entity Types, and Identifying Relationships

- **Strong entity type**
  - exist independently of other types of entities
  - has its own unique identifier
  - represented with single-line rectangle

- **Weak entity type**
  - dependent on a strong entity... cannot exist on its own
  - does **not** have a unique identifier
  - represented with double-line rectangle

- **Identifying relationship**
  - links strong entity type to weak entity type
  - represented with double line diamond
Discussion of weak entities

- Always possible to add "artificial" identifier to avoid them.
- However, sometimes more natural to form a composite key involving a foreign key given by the identifying relationship.
- Saves a bit of space too...
Associative Entities

- It’s an **entity type** – it has attributes, identifier.
- AND it’s a **relationship** – it links entities together.
- Should be seen as a way of visualizing the above, **but**: Behaves in all ways just like an entity type.

**Figure 3-11b** An associative entity - An associative entity (CERTIFICATE)