Introduction to Database Design

UVic C SC 370

Dr. Daniel M. German

Department of Computer Science

University of Victoria

May 5, 2004 Version: 1.1.1

Overview

✤ What are the steps in designing a database?

✤ What is the entity-relationship (ER) model?

✤ How does UML related to the ER model?

Chapter 2 of textbook

ER Model

✤ The **Entity-Relationship data model** allows us to describe the data involved in a real-world system in terms of objects and their relationships

✤ It is widely used in database design

Database Design

Database design can be divided in six major steps:

✤ Requirements analysis

✤ Conceptual Database design (mostly done using the ER model)

✤ Logical Database design

✤ Schema refinement

✤ Physical Database Design

✤ Application and Security Design

ER diagram

ER Diagram:

- an approximate description of the data,
- constructed through a subjective evaluation of the information,
- that was collected through the requirements analysis phase.

Entities

- ✤ **Entity**: is an object in the real world that is distinguishable from other objects
- ✤ **Entity Set**: collection of similar objects
- ✤ An entity is described using a set of **attributes**
- ✤ Each attribute has a **domain** of possible values.
- ✤ For each entity set, we should select a **key**
- ✤ A **key** is a minimal set of attributes that uniquely identify an entity in a set
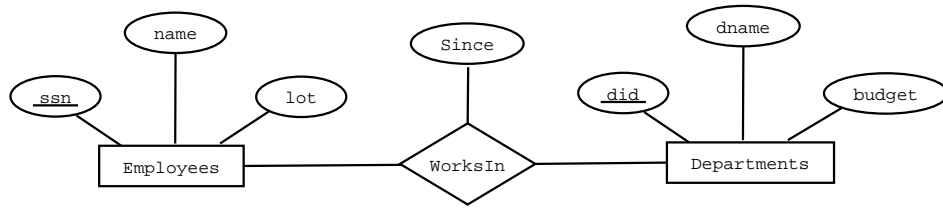
Example of an Entity
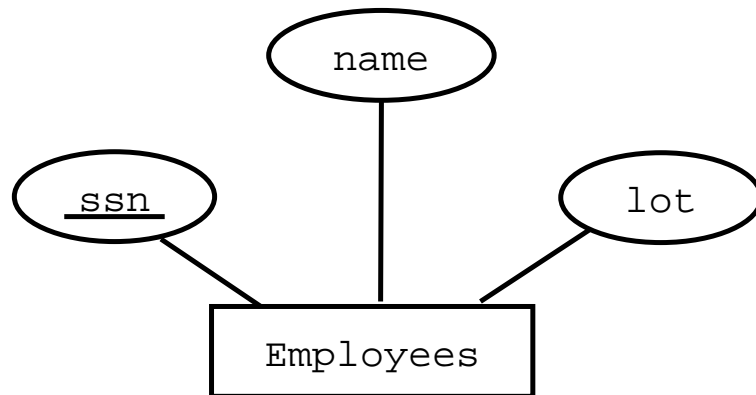
Relationships

- ✤ A **relationship** is an association among two or more entities
- ✤ A set of similar relationships is called a **relationship set**:

$$\{(e_1, ..., e_n)|e_1 \in E_1, ..., e_n \in E_n\}$$

- ✤ Each n-tuple denotes a relationship involving n entities $(e_1, ..., e_n)$ where e_i is in the entity set E_i
- ✤ A relationship can also include its own attributes (called **descriptive attributes**
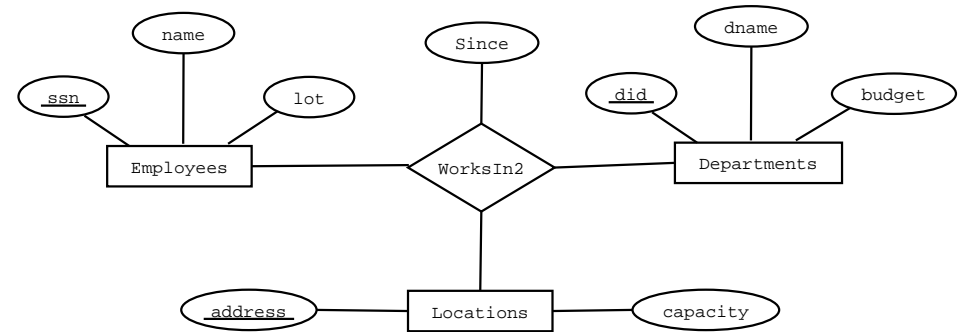- ✤ A relationship must be uniquely identified by its participating entities

Example of a Relationship

Example of a Ternary Relationship

Role Indicators
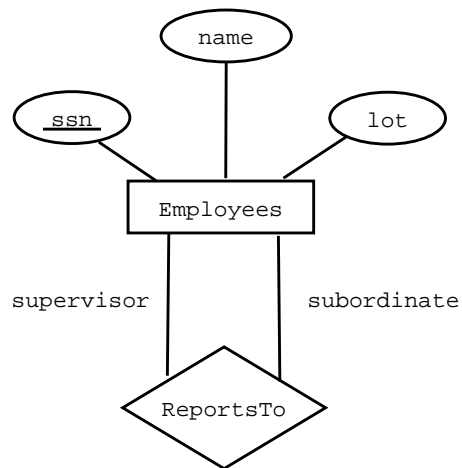
Key Constraints
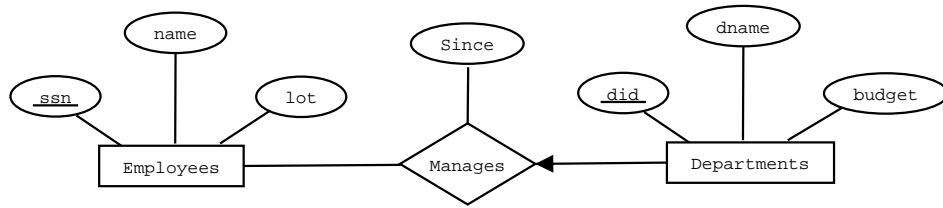
✛ **One-to-many**: an entity is related to many other entities, but each of these entities can only be related to one entity

✛ **Many-to-many**: an entity is related to many other entities, and vice-versa

Key Constraint on Manages

Key Constraint on a Ternary Rel.
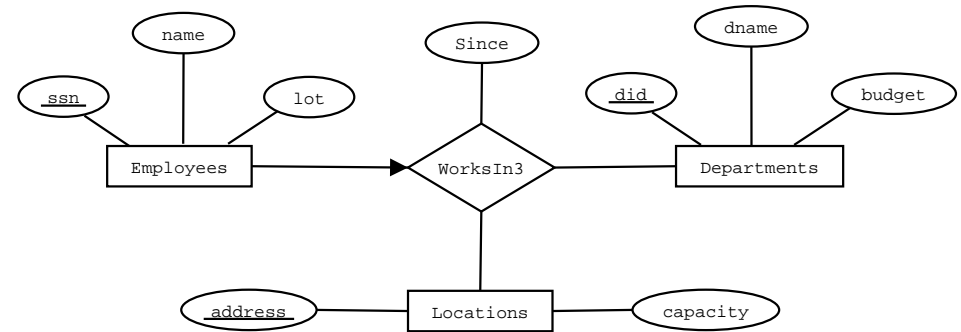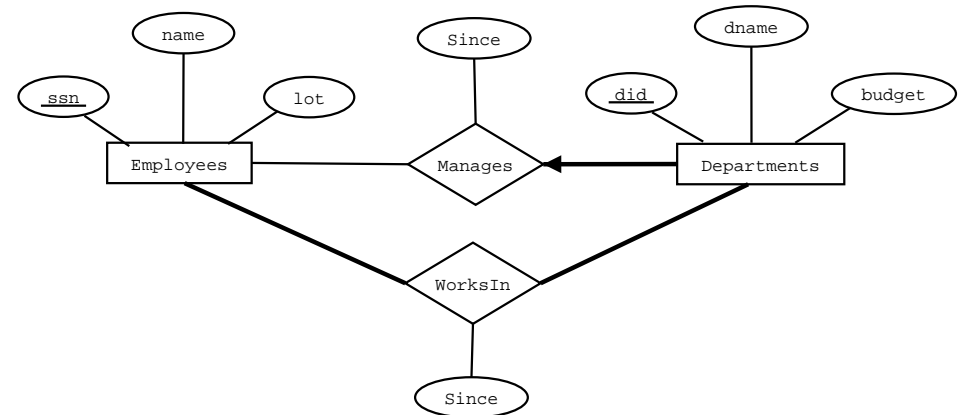
Participation Constraints

There are two types of participation constraints for an entity in a relationship:

- ✤ **Total**: Every instance of the entity is present in the relationship (represent it by a thick line)
- ✤ **Partial**: Not every instance of the entity is present in the relationship represented

Total participation

Weak Entities

✤ Sometimes the attributes associated with an entity set do not include a key

✤ A **weak entity** can be identified uniquely only by considering some of its attributes in conjunction with the primary key of another entity (called **identifying owner** entity)

✤ The following restrictions must hold:

◆ The owner entity set is a one-to-many to the weak entity (**identifying relationship set**)
◆ The weak entity set should have total participation in the identifying relationship set.

✤ Represented by drawing the relation and the weak entity in thick lines

A Weak Entity Set
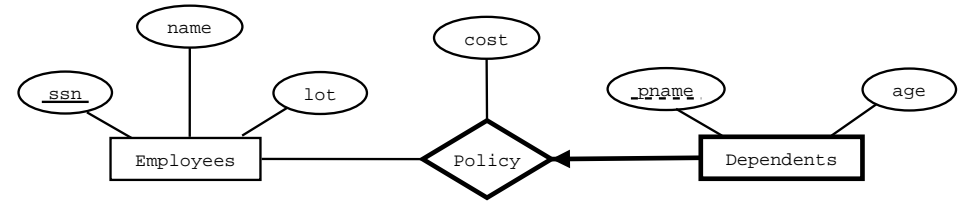
Class Hierarchies

✤ Sometimes we need to define entities as "derivations" of others (**ISA**)

✤ That is, the attributes of an entity are those of another entity (its parent) plus other ones

✤ A class hierarchy can be seen in two different ways:

◆ Specialization: identify subsets of an entity that share some distinguishing characteristics
◆ Generalization: An entity is created that includes several characteristics common to different entity sets.

A Class Hierarchy
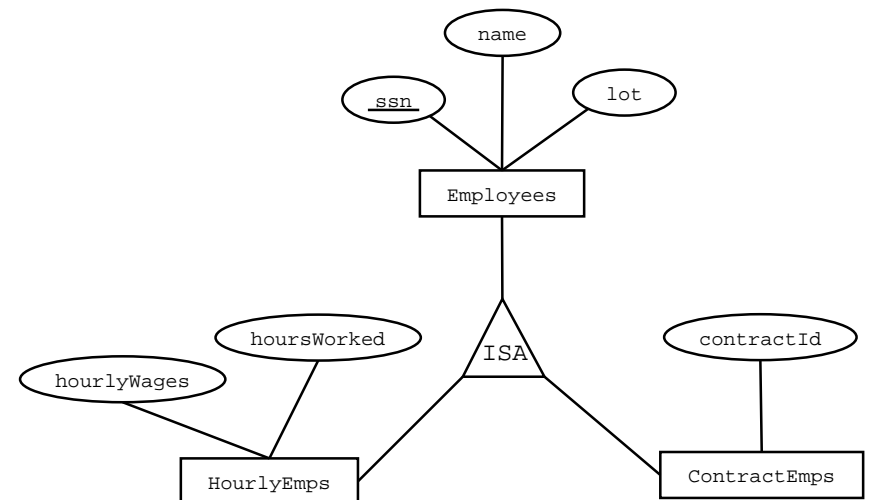
Class hierarchies...

- ✢ Why do we subclass?
  - ◆ We might want to include attributes that only make sense for the subclass (specialization)
  - ◆ We might want to identify a set of entities that participate in a given relation (generalization)

Aggregation

- ✢ Sometimes a relationship needs to relate one relationship with a collection of entities or other relationships

- ✢ Aggregation allows us to indicate that a relationship set participates in another relationship set.

- ✢ Illustrated by drawing a dashed box around the set of related entities and relationships.

Aggregation...
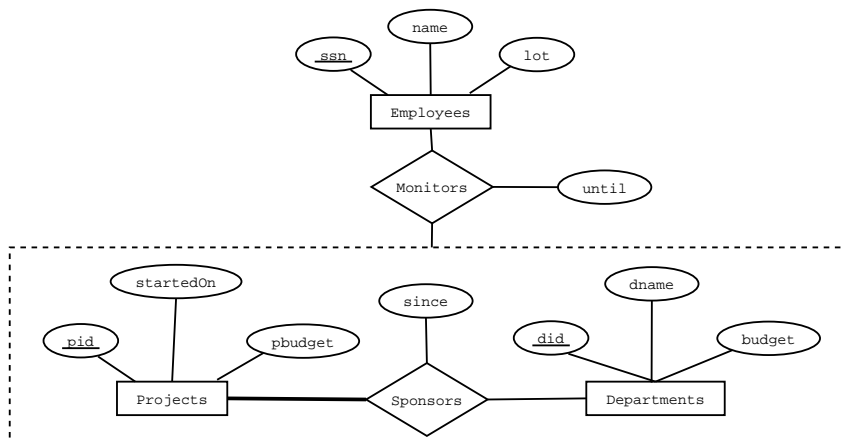
Conceptual Design with the ER Model

Developing an ER diagram presents several choices:

- ✢ Should a concept be modelled as an entity or an attribute?
- ✢ Should a concept be modelled as an entity or a relationship?
- ✢ What are the relationship sets and their participating entity sets?
- ✢ Should we use binary or ternary relationships?
- ✢ Should we use aggregation?

Entity vs. Attribute

✥ It is not always clear what should be an attribute of an entity and what should be moved to a new entity set

✥ In general, an attribute should not be an entity unless:
  ◆ We need to record the same attribute(s) for more than one entity
  ◆ We want to capture the structure of this "attribute" in our ER-diagram

Entity vs. Attribute...
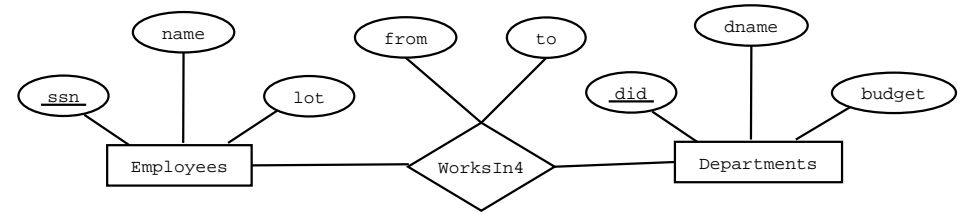
Entity vs. Attribute...
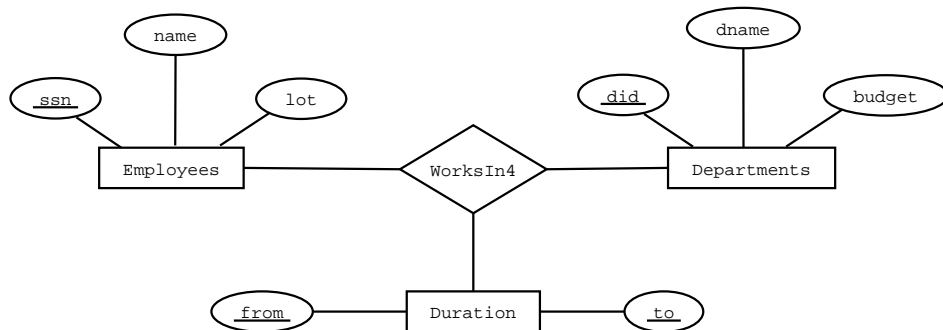
Entity vs. Relationship

✥ The imprecise nature of ER modelling makes it difficult to recognize when to define an attribute as part of an entity or as part of a relationship

✥ The only solution (at this point) is to apply common sense: is the attribute part of the relation, or is it part of the entity?

✥ In general, a mistake in this stage will lead to wasted storage

✥ We will fix this in the future (normalization)

# Entity vs. Relationship...
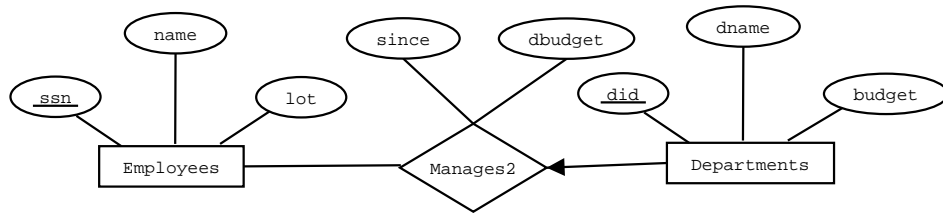
Binary vs. Ternary Relationships

✢ In cases where we can use either a binary or ternary relationship,

✢ the decision is usually determined by **any restrictions** (integrity constraints) on the relationship that we are trying to model and

✢ if we can or cannot do it with a ternary relationship

Binary vs. Ternary...
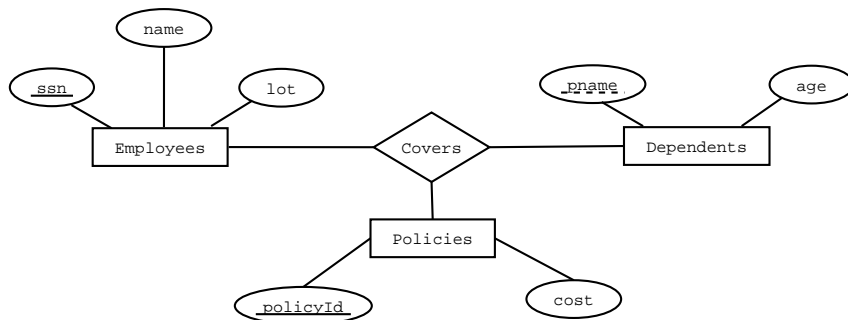
- A policy cannot be owned jointly by two or more employees

- Every policy must be owned by some employee

- Dependents is a weak entity (uniquely identified by *policyid* and *pname*)

Binary vs. Ternary...
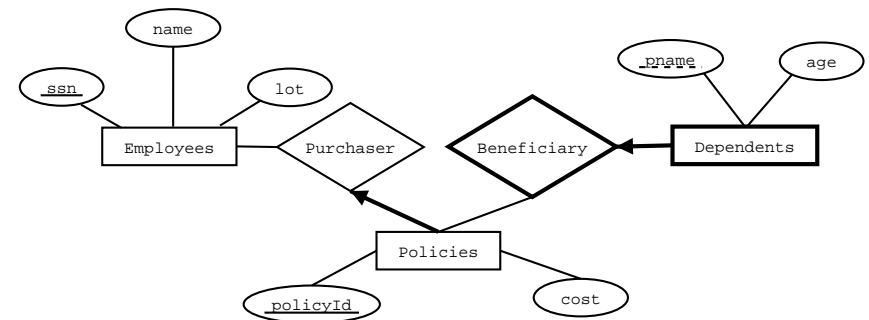
Aggregation vs. Ternary Relationships

✤ Again, the choice depends on integrity constraints

Aggregation vs. Ternary...