CS2102 Database Systems

Conceptual Database Design

NUS National University of Singapore | School of Computing

Leading The World With Asia's Best

Database Design Process

1. **Requirements Analysis**
   - What does the user want from the database?
2. **Conceptual Database Design**
   - Capture data requirements using a conceptual data model, e.g. Entity-Relationship (ER) data model
   - High level description of the data to be stored in the database, and constraints that hold over the data
3. **Logical Database Design**
   - Convert the conceptual database design into a database schema in the data model supported by DBMS, e.g., relational DBMS
   - Result is a conceptual schema/logical schema

Database Design Process

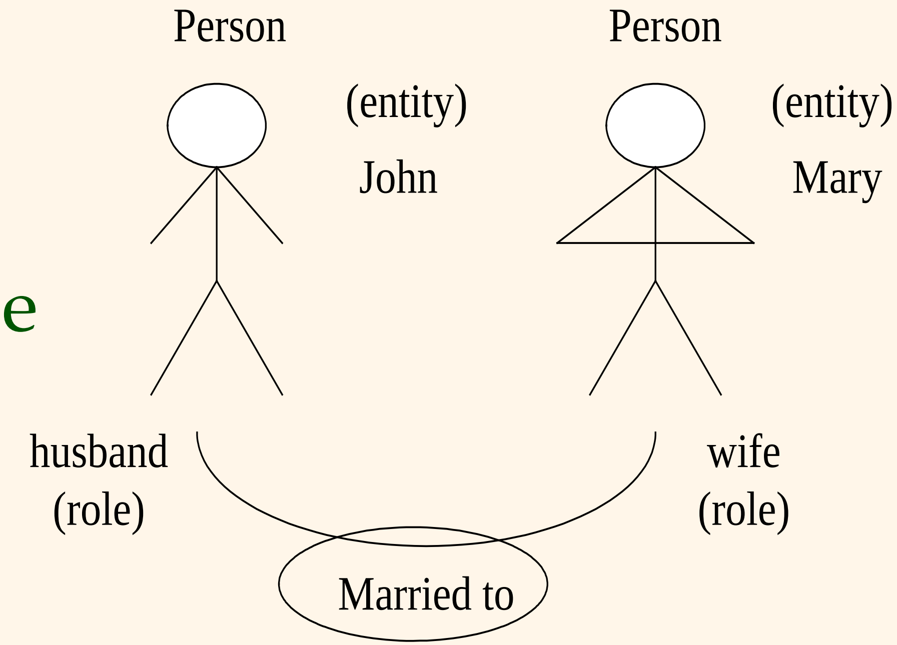
4. **Schema Refinement**
   - Use data constraints to improve the logical schema.
   - Theory of normalizing relations

5. **Physical Database Design**
   - Use performance criteria to design physical schema, e.g. build indexes on some tables or cluster tables

6. **Application and Security Design**
   - Specify access control policies

Entity-Relationship Model

- Proposed by Prof. Peter Chen in 1976
- Most common conceptual data model for capturing data requirements
- Main concepts:
  - entity (or object)
  - relationship
- English correspondence
  - noun → entity
  - verb → relationship

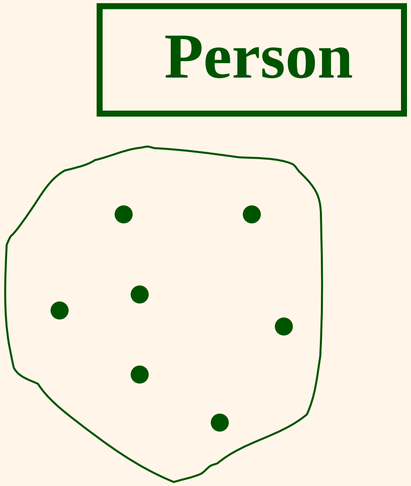


Entities

- An entity is a real world object distinguishable from other objects.
- Example:
  - Ng Hong Kim with IC# 0578936I
  - Account# 563978 of DBS Kent-Ridge Branch
  - Car# SBG 3538P

Entity Types

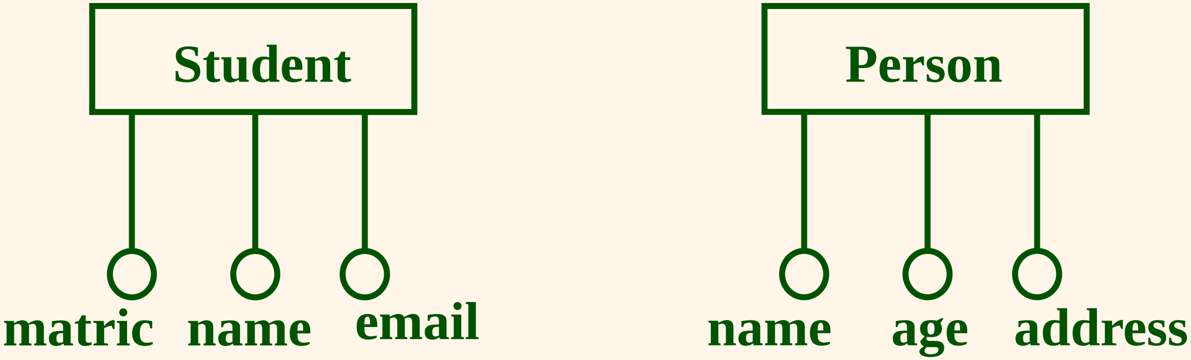
- Entities can be classified into different types
- Each entity type contains a set of entities each satisfying a set of predefined common properties
- Example:
  - Employee
  - Car
  - Courses



Person

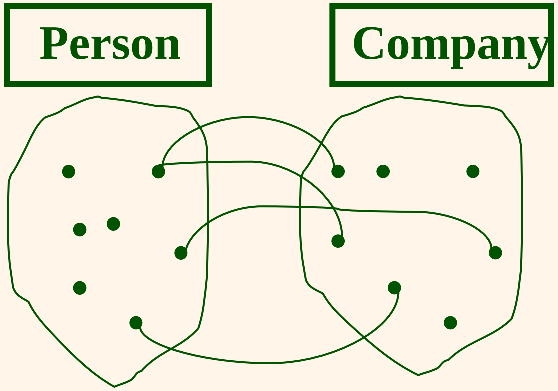
Attributes of Entities

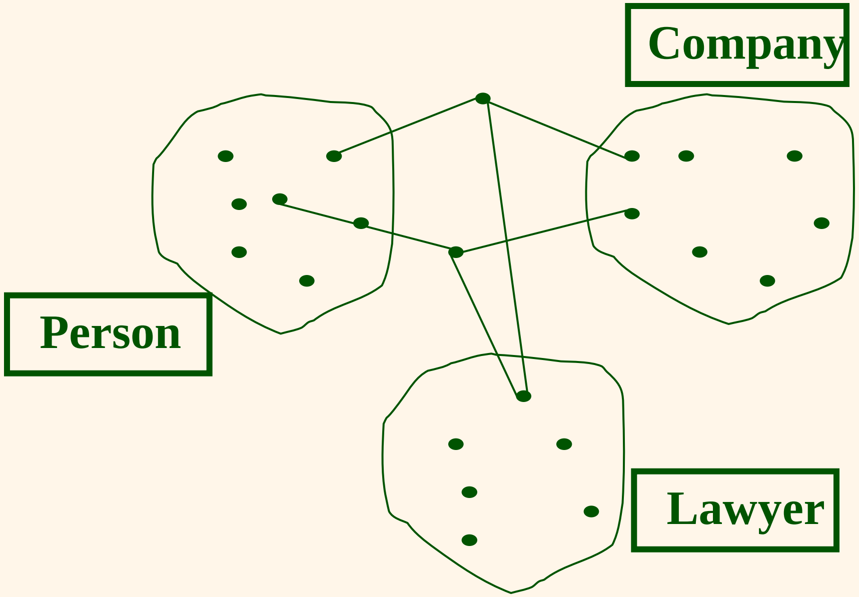
- Attributes are specific information that describe the properties of entities
- All entities in an entity type have the same set of attributes
- Attributes take different values for each entity
- Example:
  - Entity type Student has attributes *matric, name, email*



Relationships

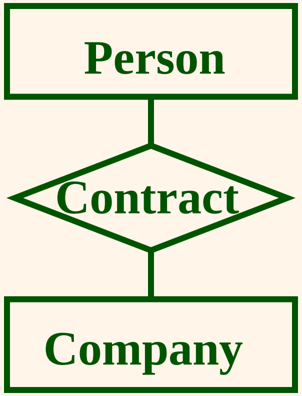
- A relationship is an association among two or more entities
- Example:
  - Bob enrol in MA101 course
  - Customer Alice has a savings account

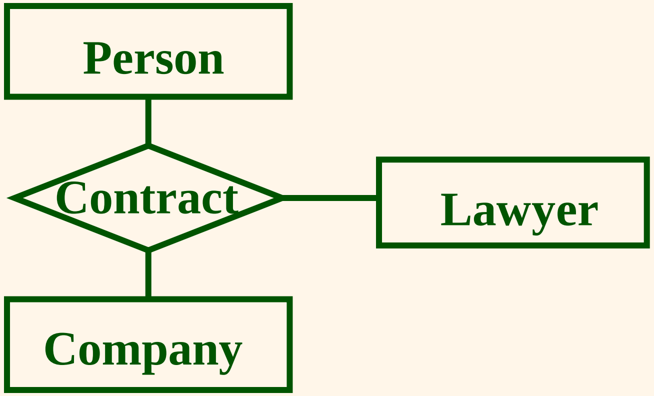




Relationship Set

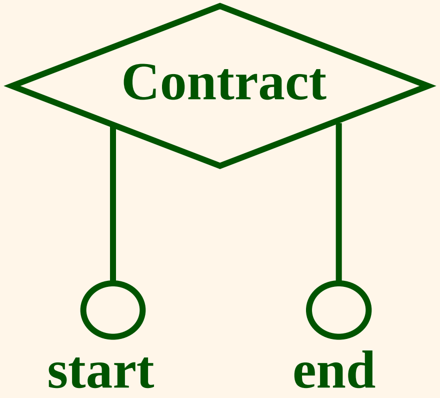
- A relationship set is a set of relationships associating entities from the same entity types
- An *n*-ary relationship set R relates *n* entity types $E_1, E_2, \ldots E_n$
  - Each relationship in R involves entities $e_1 \in E_1$, $e_2 \in E_2$, ..., $e_n \in E_n$
  - *n* is the degree of the relationship set
  - When *n*=2, we have a binary relationship set





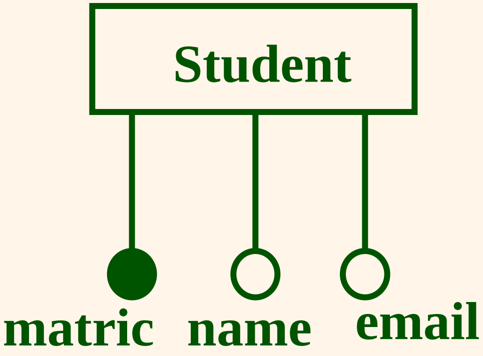
Attributes of Relationships

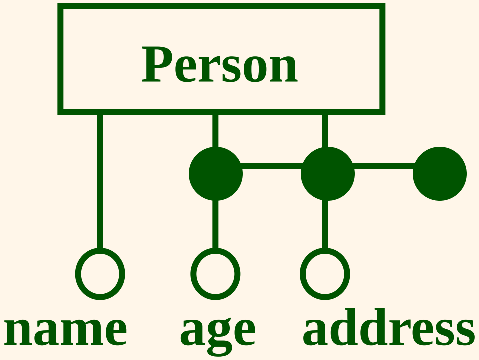
- Relationships have attributes
- All relationships in a relationship set have the same attributes
- Relationships are distinguished by their participating entities



Identifiers of Entity Type

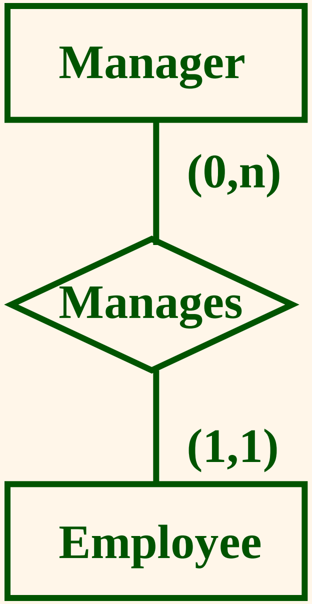
- Each entity type E has a key K, a minimal set of attributes whose values uniquely identify an entity
- A combination of attributes can identify an entity

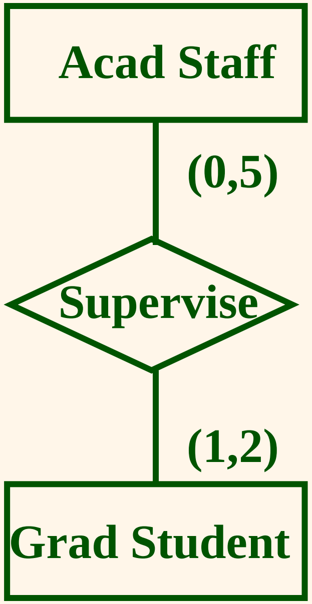




Relationship Cardinality

- The cardinality of the participation in a relationship can be constrained by a minimum and maximum value: (1,1),   (0, n) , (2, 5)





Relationship Cardinality

- Is the participation of an entity type in a relationship set mandatory?

- Optional or partial participation (0, x)
  - Each student can enrol in 0 or more courses

- Mandatory or total participation (1, x)
  - Each student must enrol in at least one course

Relationship Cardinality

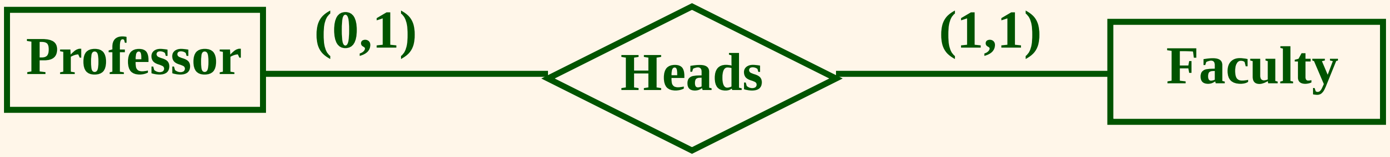
- $(x, 1)$ for all entities involved in a relationship characterizes a one-to-one relationship

- $(x, 1)$ for one entity and (x, N) or (x, y) $y > 1$ for the other entities characterizes a one-to-many relationship

- (x, N) or (x, y) $y > 1$ for all entities characterizes a many-to-many relationship

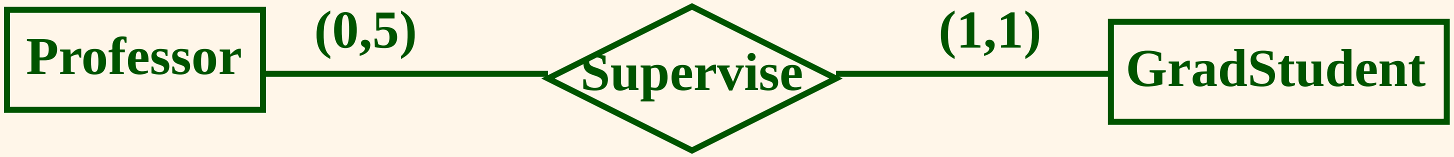
Relationship Cardinalities

- Example of a <u>one-to-one</u> relationship
  - Each faculty is headed by one professor
  - Each professor can head at most one faculty



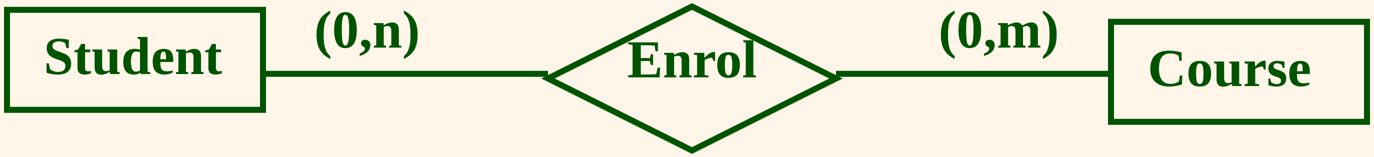
Relationship Cardinality

- Example of a one-to-many relationship
  - Each student can be supervised by at most one professor
  - Each professor can supervise at most 5 students



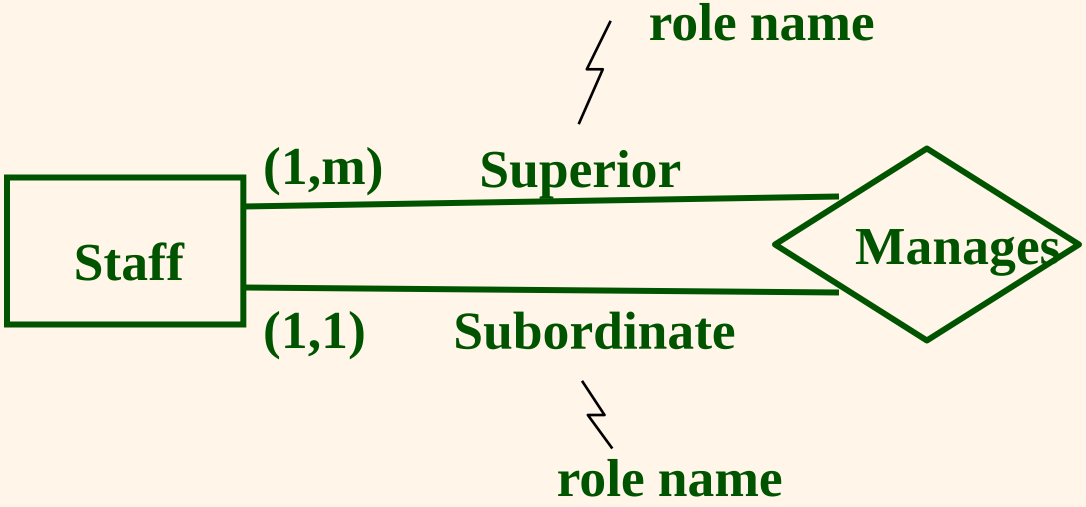
Relationship Cardinality

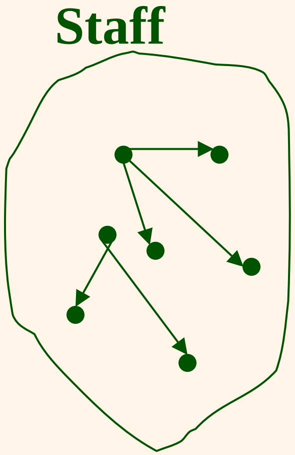
- Example of a <u>many-to-many</u> relationship
  - Each student can enrol in 0 or more courses
  - Each course can be enrolled by 0 or more students



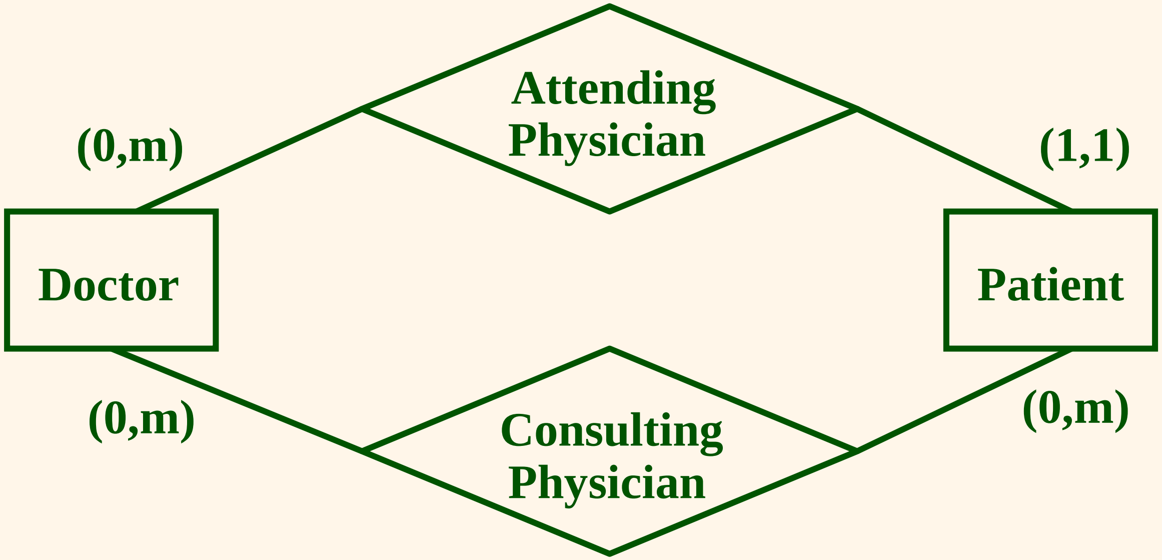
Recursive Relationship Sets and Roles in Relationships

- Relationships can associate entities from the same entity set
- Roles are used



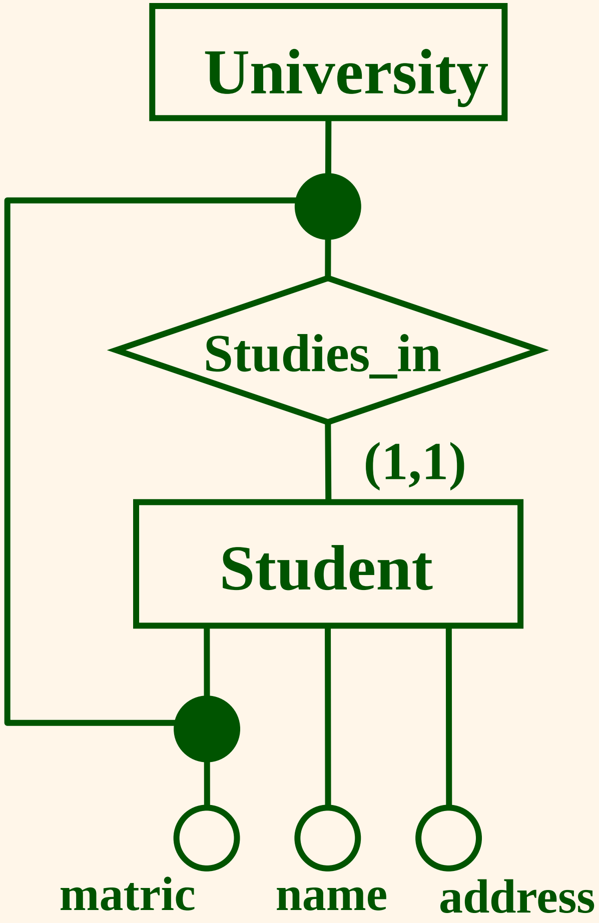


Multiple Relationship Sets between the Same Set of Entity Types



Weak Entities

- Some entities can only be identified within the scope of a relationship with another entity type

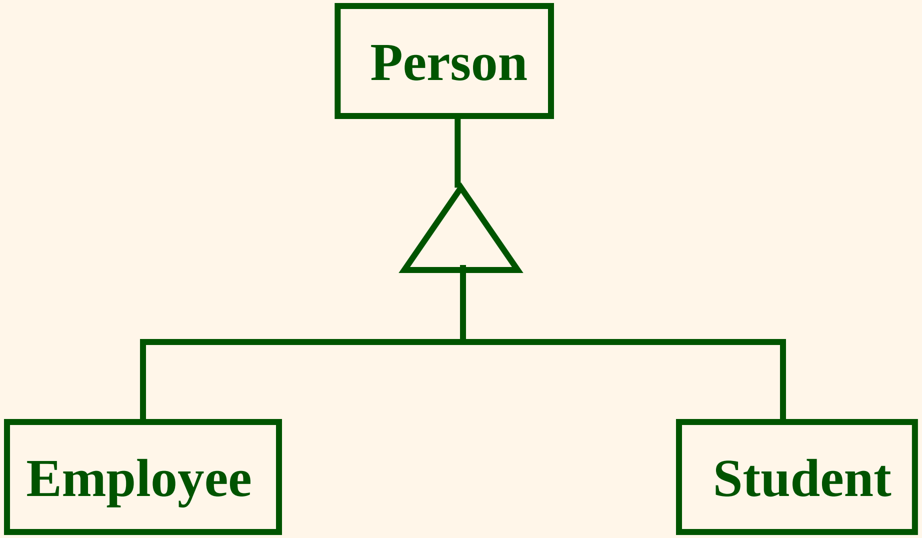


Weak Entities

- The relationship must exist and be unique for each entity in the set
- Weak entities can only be defined for a participation constrained by (1,1) cardinality
- Also called mandatory one-to-many relationships

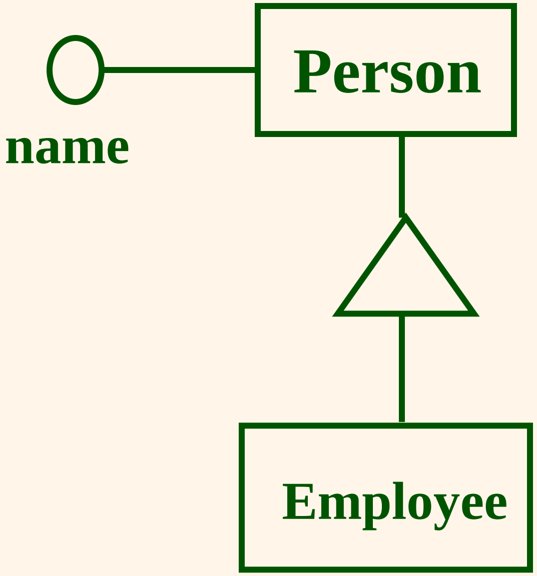
Hierarchies: Subclass and Superclass

- Useful to classify an entity type into subclasses



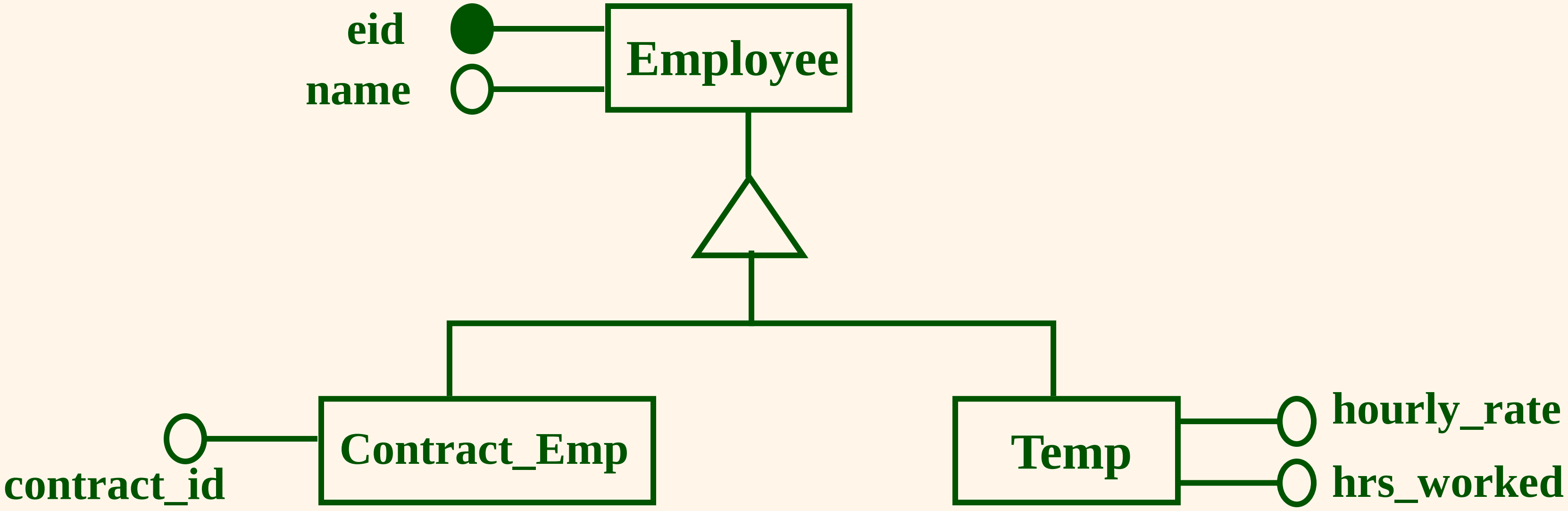
Hierarchies: Inheritance

- Entities in the subclass inherit the attributes of their superclass



Hierarchies: Specialization

- Add descriptive attributes/relationships specific to a subclass



Exercise

- Design a database about movies. The entity types are Movies, Stars and Studios. The Movies enttiy type has fours attributes: title, year (in which movie is made), length and type ("color" or "blackNwhite"). The entity types Stars and Studios have the same two attributes, name and address, each with an obvious meaning. We need to record the stars for each movie, and the studios that own the movie.
- Draw the ER diagram for this database.

Exercise (cont'd)

- Among the kinds of movies in our movie database are cartoons, murder mysteries, adventures, comedies etc. We could define each of these movie types as a subclass of the entity type Movie. A cartoon has, in addition to the attributes and relationships of Movies, an additional relationship called Voices that gives us a set of stars who speak but do not appear in the movie. Movies that are not cartoons do not have such stars. Murder mysteries has an additional attribute weapon.
- Augment the Movie ER diagram with hierarchies

ER Model - Design Decisions

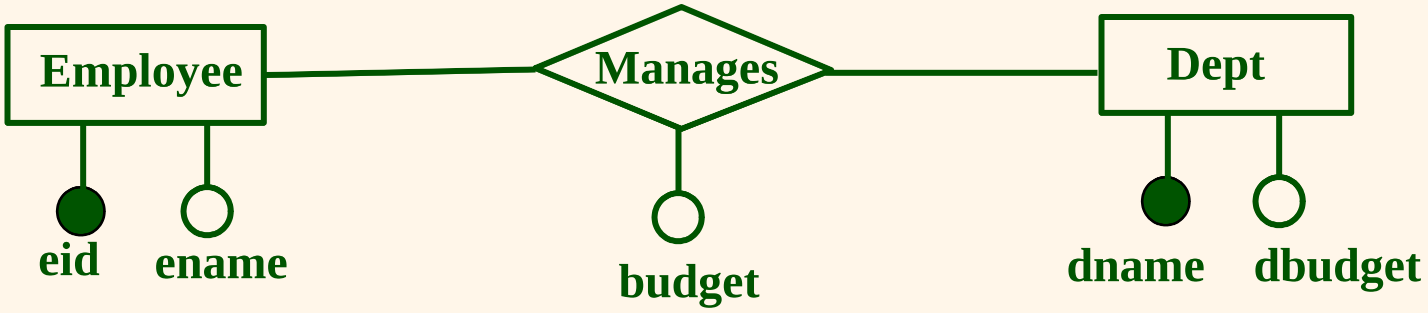
- Should a concept be modeled as an entity type or an attribute?
- Should a concept be modeled as an entity type or a relationship set?
- Should a concept be modeled as multiple binary relationship sets or a single n-ary relationship set?

Entity Type vs. Attribute

- Should *address* be an attribute of Employees or an entity type (connected to Employees by a relationship set)?
- Depends upon how we want to use address information, and the semantics of the data
  - If each employee has several addresses, *address* should be an entity type.
  - If the structure (city, street, etc.) is important, e.g., we want to retrieve employees in a given city, *address* should be modeled as an entity type

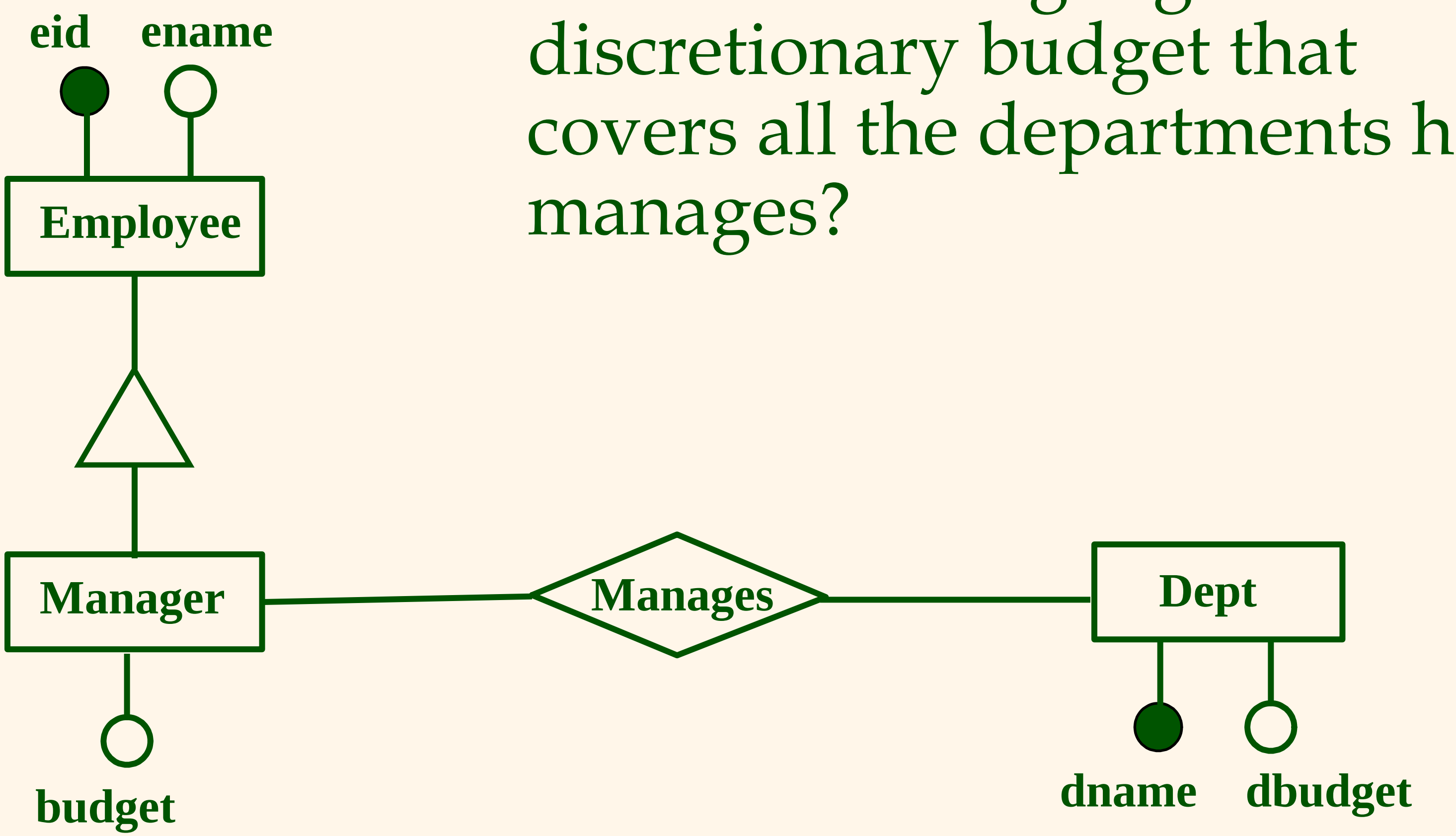
Entity Type vs. Relationship Set

- ER diagram express that a manager gets a separate *discretionary budget* for each department.



Entity Type vs. Relationship Set

- What if a manager gets a discretionary budget that covers all the departments he manages?



Summary

- Conceptual design follows requirements analysis
- ER model popular for conceptual design
  - Constructs are expressive, close to the way people think about their applications.
- Basic concepts: entities, relationships, and attributes
- Additional constructs: weak entities, hierarchies
- ER model is subjective