
Lecture 5: The ER Model

Dragan Mirkovic
Department of Computer Science
University of Houston

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Announcements

- Quiz #3 9/25/2004
 - Ch. 5 (SQL)
- Questions?
- Today:
 - E/R Model and conceptual database design
 - Chapter 2 in Molina, Ullman & Widom

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Introduction

- How to design a database?
 1. Analyze the information (data) that needs to be stored in database
 - Conceptual model
 2. Design the database schema
 - Describes the structure of the database
 3. Commit the design to a desired form and implement in DBMS
- Entity-Relationship Data Model is very useful in the initial database design



D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Conceptual Data Models

- Conceptual data models provide languages to describe conceptual schemas.
- Conceptual schemas are used to describe the classes of objects that occur in an application area, their properties, their relationships, and the constraints that hold with respect to those classes of objects.
- Center on “**what** kind of objects a database contains” and not on “**how** these objects are stored” (→ Internal Schema) and not on “**how** these objects are represented / displayed to a person that accesses the database” (→ External Schema).

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

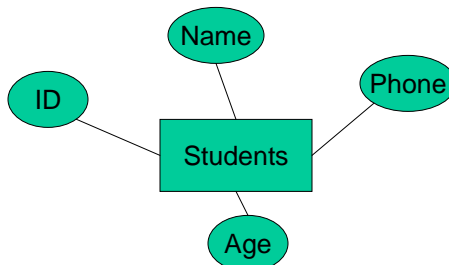
Applications

- Conceptual Data Models are useful
 - As a database design tool formalizing the information requirements of the end users
 - As a documentation tool for databases (to help programmers, especially those that have to update the database)
 - As a data model of a database management system (only very few experimental systems exist)
 - As a tool to describe domain properties
 - As a tool of system analysis

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004


Elements of the E/R model

- E/R Model is the most common model for abstract representation of the database structure
- Graphic representation of the database structure
 - E/R diagram
- Element types
 - Entity sets
 - Attributes
 - Relationships



D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

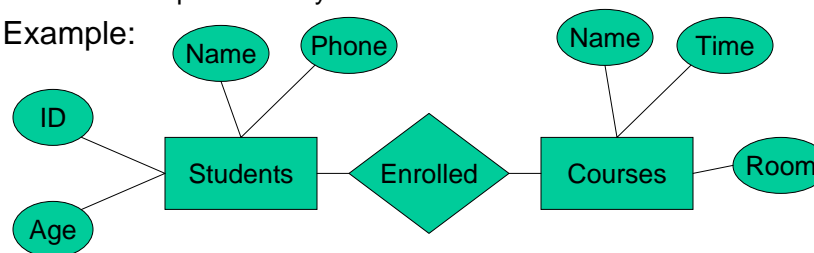
Entity Sets, Attributes, ...

- Entity:
 - Abstract object
- Entity set
 - Class of abstract objects, a collection of similar entities
 - Example
 - Students, Courses, ...
 - Set of all students = entity set
 - In diagrams
 - Entity set = **Rectangle**
- Attributes
 - Properties of the entities
 - Example
 - ID, Name, Age, ...
 - In diagrams
 - Attribute = **Oval**
 - We will assume that attributes are atomic values (string, integer, real,...)
- Relationships
 - Connect two or more entity sets
 - In diagrams represented by diamonds 

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

E/R Diagrams

- Graph representing entity sets, attributes and relationships
- Edges connect
 - Entity sets and its attributes
 - Relationships and entity sets
- Example:



D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Relationship Set

- The entry of the relationship set is an ordered pair (or tuple)
- Relationship set = table
- One column for each of the connected entity sets
- One row for each list of entities, one from each set
- Example:
 - Enrolled

Students	Courses
Sally	COSC2410
Sally	COSC3480
Joe	COSC2410
...	

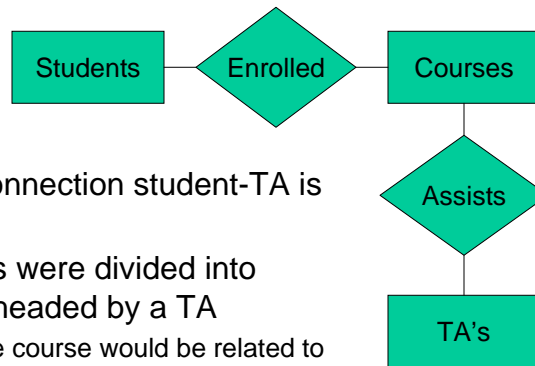
D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Multiway Relationships

- Usually binary relationships (connecting two E.S.) suffice
 - However, there are some cases where three or more E.S. must be connected by one relationship
 - Example: relationship among students, courses, TA's.

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Solution?

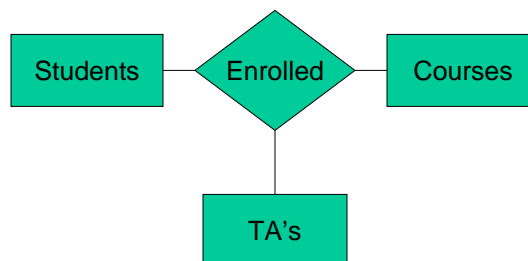


- Works only if connection student-TA is only via course
- What if students were divided into sections, each headed by a TA
 - A student in the course would be related to only one of the TA's for that course
- Need a 3-way relationship

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

3-Way Solution

- Enrolled is a 3-way relationship between Students, Courses and TA's
 - Student enrolls in a course section assisted by a particular TA



D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Multiplicity of binary relationships

- In general any member of one of the entities to any number of members of the other entity set
- Common types:
 - Many-to-one
 - One-to-one
 - Many-to-many
- Representation of Many-to-one
 - Arrow pointing to one
 - Rounded arrow = exactly one

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Many-Many Relationships

- Think of a relationship between two entity sets, such as *Enroll* between *Students* and *Courses*, or *Sells* between *Bars* and *Beers*.
- In a *many-many* relationship, an entity of either set can be connected to many entities of the other set.
 - E.g., a bar sells many beers; a beer is sold by many bars.

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Many-One Relationships

- Some binary relationships are *many - one* from one entity set to another.
- Each entity of the first set is connected to at most one entity of the second set.
- But an entity of the second set can be connected to zero, one, or many entities of the first set.

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Example

- *Favorite*, from *Drinkers* to *Beers* is many-one.
- A drinker has at most one favorite beer.
- But a beer can be the favorite of any number of drinkers, including zero.

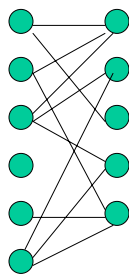
D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

One-One Relationships

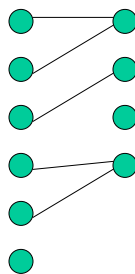
- In a one-one relationship, each entity of either entity set is related to at most one entity of the other set.
- Example: Relationship *Best-seller* between entity sets *Manfs* (manufacturer) and *Beers*.
 - A beer cannot be made by more than one manufacturer, and no manufacturer can have more than one best-seller (assume no ties).

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

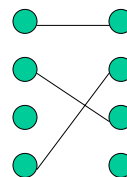
In Pictures:



many-many



many-one

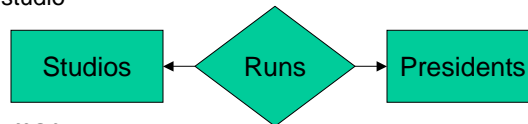


one-one

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Examples

- One-to-one:
 - Studio can have only one president and a president can run only one studio



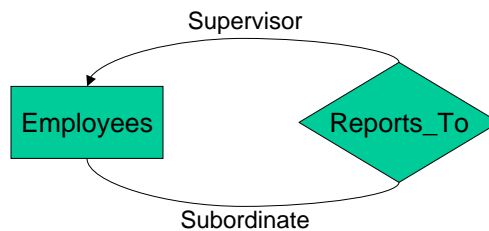
- One-to-many:
 - Each department can have only one manager but one employee can be associated with many departments as a manager.



D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Roles in Relationships

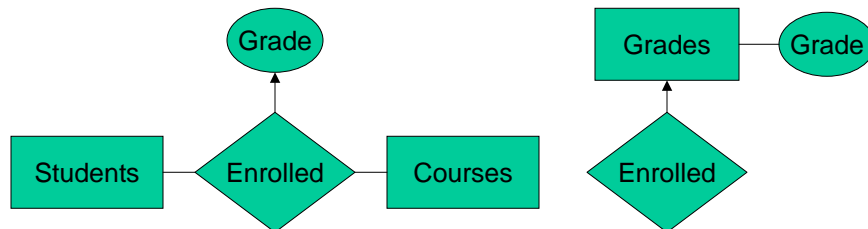
- One entity set may appear two or more times in a single relationship
- One line for each different role
- The edges are labeled by different roles
- Example:



D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Attributes on Relationships

- In some cases attributes are associated with relationships
- Example:
 - The attribute 'Grade' is associated with both the student and the course – hence it is the attribute of their relationship.
 - It can be avoided by introducing a new entity 'Grades'



D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Converting Multiway Relationships to Binary

- Some data models (ODL) are restricted to binary relationships
- Any multiway relationship can be converted into a collection of binary many-to-one relationships
 - Create new entity set to represent rows of a relationship set

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

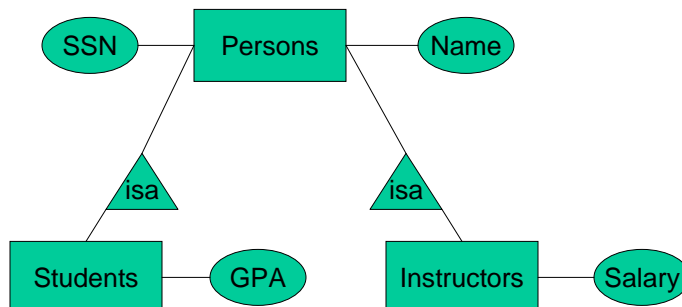
Subclasses in the E/R Model

- One entity set can contain several subclasses
 - Entities that have special properties not associated with all members of the set
 - Similar (but different) to classes in OO
 - An entity has a representation in all the subclasses to which it logically belongs
 - Its properties are the **union** of the properties of these classes
 - OO: An object (entity) belongs to exactly one class
 - It inherits properties of its superclasses
- Example:
 - Both Students and Instructors are Persons (have SSN, Name, Phone, Address)
 - In the university database they have a number of different attributes associated only with one of them

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Examples

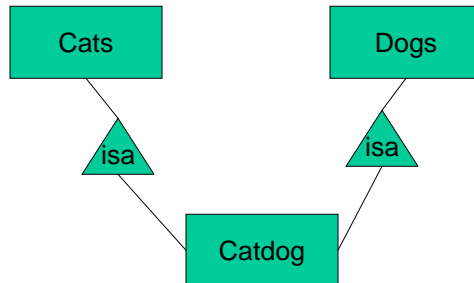
- Notation:
 - *isa* triangles indicate the subclass relation



D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Multiple Inheritance

- Theoretically, an E.S. could be a subclass of several other entity sets



D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Design Principles

- Correctness
- Faithfulness
 - Attributes should reflect reality
- Avoiding redundancy
 - Specifying all properties only once
 - Example:
 - Studio – Owns – Movies. Avoid attribute studio with Movies.
- Simplicity
 - Avoid introducing more elements than necessary

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

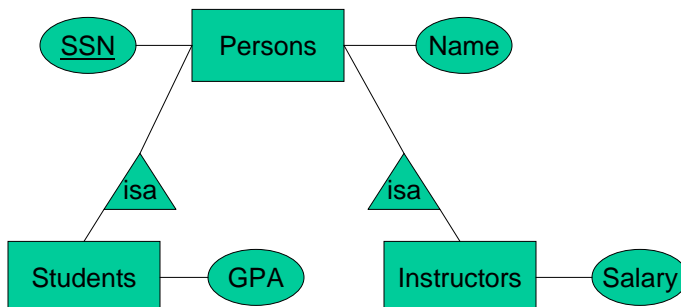
Constraints in E/R

- Keys in E/R:
 - Set of attributes whose value can belong to at most one entity
 - In E/R model, every E.S. must have a key
 - It could have more than one key, but one set of attributes is the “designated” key
 - In E/R diagrams, the keys are indicated by underlying all attributes of the designated key.

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Example

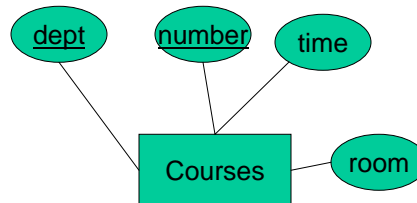
- SSN is key for Persons
 - It is also key for Students and Instructors
 - In general, key at root is key for all.



D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Example

- Multiattribute key:
 - dept + number is key for Courses
 - Possibly, time + room also forms a key, but we have not designated it as such



D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Weak Entity Sets

- Sometimes an E.S.'s key comes not (completely) from its own attributes, but from the keys of one or more E.S.'s to which the first is linked by a many-to-one relationship
 - Called a *weak* E.S.
 - Represented by putting double rectangle around the weak E.S. and double diamond around each relationship to which the weak E.S. is linked to an E.S. that provides a part of its key
 - Use of many-to-one relationship (includes 1-1) is essential
 - With many-to-many, we wouldn't know which entity provided the key value

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Causes of Weak E. S.

- Hierarchical classification which is not a subclass-type hierarchy
- Examples:
 - IP addresses, Login names, file names (path + file), species names, ...
- Entity sets introduced to eliminate a multiway relationship
 - Insufficient attributes to make a key
 - All connecting E. S. are weak.

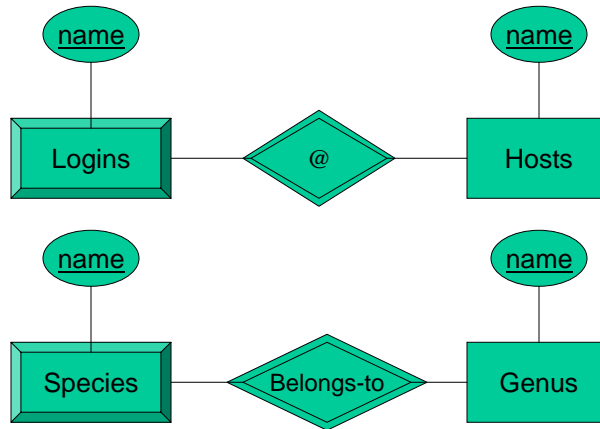
D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Examples

- Logins (Email Addresses)
 - Login name = user name + host name, e.g. `mirkovic@istra.cs.uh.edu`
 - A “login” entity corresponds to a user name on a particular host
 - Passwd table doesn’t record the host, just the user name
 - Key for a login = user name at the host (unique for that host only) + the IP address of the host (unique globally)
- A species is designated by its genus and species names
 - Ex. Humans: *Homo sapiens*
 - (*Homo* = genus name, *sapiens* = species name)

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Examples: Diagrams



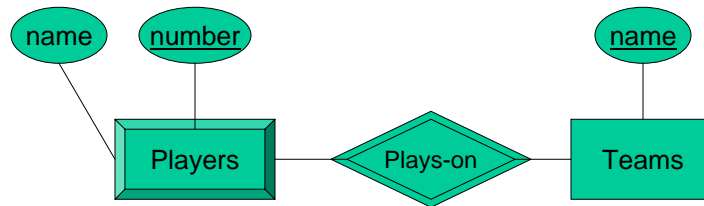
D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Example

- name is almost a key for football players, but there might be two with the same name.
- number is certainly not a key, since players on two teams could have the same number.
- But number, together with the Team related to the player by Plays-on should be unique.

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

In E/R Diagrams



- Double diamond for *supporting* many-one relationship.
- Double rectangle for the weak entity set.

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Don't Overuse Weak Entity Sets

- Beginning database designers often doubt that anything could be a key by itself.
 - They make all entity sets weak, supported by all other entity sets to which they are linked.
- In reality, we usually create unique ID's for entity sets.
 - Examples include social-security numbers, automobile VIN's etc.

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

When Do We Need Weak Entity Sets?

- The usual reason is that there is no global authority capable of creating unique ID's.
- Example: it is unlikely that there could be an agreement to assign unique player numbers across all football teams in the world.

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Identifying Keys and Relationships for Entity Types

- Each entity type that is not subtype of another entity type needs:
 - Case1: Normal Entity Type (single rectangle)
 - A single attribute (straight line) or
 - A set of attributes
 - Case2: Weak Entity Type (double rectangle)
 - A set of relationships (double diamond) or
 - A set of relationships or a single attribute (dotted line) or
 - A set of relationships and a set of attributes (dotted line)that uniquely identifies the instances of the entity type
- Remark: min-max cardinalities for weak entity types for their participation in identifying relationships have to be (1,1)!!

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Valid E/R Diagrams

- An E/R diagram is valid if and only if:
 - It is syntactically correct (e.g. specifies all key constraints,...)
 - It specifies the entity types, relationship types, attribute types, and subtype relationships necessary to satisfy all information requirements.
 - It does not specify any invalid constraints.

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004

Homework

- Read Ch 2. In “Database Systems. The complete Book”, by H. Garcia-Molina at all.
 - You may download Chapter 2 from the Book website.
- Problems:
 - Discussed in class

D. Mirkovic, COSC 3480: Design of File and Database Systems, Fall 2004