

InfSci 2511

Information System Analysis and Design

Lecture 2

9/11/2000

Reference Books on Reserve

The following reference books will be available on reserve in the Information Science Library:

- **Database System Concepts** (3rd edition)
 - Silberscharz, Korth, Sudarshan. McGraw Hill, 1999.
- **Database Design for Smarties:
using UML for data modeling**
 - Robert J. Muller. Morgan Kaufmann, 1999.
- **The Unified Modeling Language User Guide**
 - Booch, Rumbaugh, Jacobson. Addison Wesley, 1999.

...from our previous lecture

1. Types of Information Systems
2. The roles of the Analyst and the Designer
- 3. System development life cycle (SDLC)**
4. Systems and Organizations
5. Project planning
- 6. Term projects**

System Development Life Cycle

- Identify: problems or opportunities
- **Determine: the requirements**
- **Analyze: the system needs**
- Design: a solution – the intended system
- Develop: the system
- Test: functional correctness and performance
- Implement: deploy the system
- Maintenance/Evaluation: fixes/enhancements

Term Project

- In groups up to 3 persons; OK to do by yourself, but no special consideration for credit.
- Need one page **proposal**: **goal** and **scope**
- Final Report about 15 to 30 pages.
- Presentation in class, with discussion.
- Small projects: from analysis to implementation
- Large projects: analysis and design

InfSci 2511 **Information System Analysis and Design** Fundamentals of Entity-Relationship Modeling

Reading Assignment

Reference Book

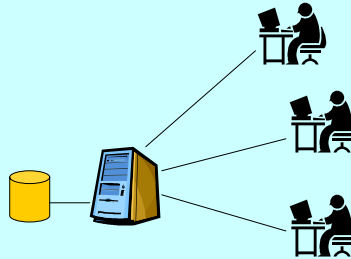
- Silberschatz, Korth, and Sudarshan
Database System Concepts.
Chapter 2 (Sections 2.1 – 2.6)
...on the fundamentals of data modeling.

Topics

- **Data Modeling in the context of ISAD**
- Basic Concepts: Entities and Attributes
- Relationships and Roles
- Design Issues
- Structural Constraints
- Uniqueness and Key Attributes
- Weak Entity Set
- Assignment 1

Our view of Information System

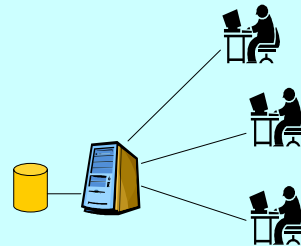
- People using software programs to access and interrogate data.



- Many people using many software programs, but *one* set of data.

Our view of Information System

- While the software programs provide the functions which the users need, the information in use must have a stable structure.



- The system designer must discover a structure to integrate the different user/functional views.
- Such an information structure is called the **conceptual schema**.

...for ISAD: the development cycle

- Identify: problems or opportunities
- **Determine: the requirements**
- **Analyze: the system needs**
- Design: a solution – the intended system
- Develop: the system
- Test: functional correctness and performance
- Implement: deploy the system
- Maintenance/Evaluation: fixes/enhancements

...for ISAD: the development cycle

For analysis, we focus on:

- **Determine: the requirements**
- **Analyze: the system needs**
 - first for the DATA we are interested in;
 - then for the software programs (applications) to provide the FUNCTIONS we need.

Data Modeling in ISAD

To determine the DATA requirements...

- To discover the “inherent” information structure
- To integrate the different views of different users
- To design a “unified” view – hopefully stable
- To formulate a model of the information we will use in the system – that is, a conceptual schema, independent of the specific software or platform we may use.

Entity-Relationship Modeling

To design/formulate a conceptual schema:

- There were many different techniques; the one commonly applied today for relational databases is Entity-Relationship Modeling (ER modeling).
- The ER modeling was first introduced in 1976 by Peter Chen as a logical design methodology to analyze data requirements.
- The ER modeling method has since been extended in many different ways.

Entity-Relationship Model

Briefly introduced...

- **Entity** – thing
- **Relationship** – nature of association between things (entities)
- **Entity-Relationship Modeling** – a way to model the real world, as things, and how the things are associated together.
- **Entity-Relationship Diagrams** – we present our models graphically...

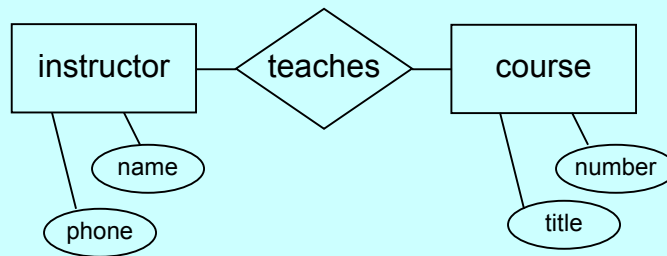
Entity-Relationship Model

- **Entity** – a *specific* thing, physical or conceptual. Example: the sun, the number one, the color red, the man Adam.
- **Attribute** – to abstract an aspect of an entity.
- **Relationship** – the way *specific* things are associated together by the nature of the relationship. Example: Adam and Eve as husband and wife. The related entities together constitute a **relationship instance**.

Entity-Relationship Diagram

Example... (ISAD for the university registrar)

- There are these entities: courses and instructors.
- Each instructor has a name and a phone number.
- Each course has course number and title.
- Instructors teach courses...



Topics

- Data Modeling in the context of ISAD
- **Basic Concepts: Entities and Attributes**
- Relationships and Roles
- Design Issues
- Structural Constraints
- Uniqueness and Key Attributes
- Weak Entity Set
- Assignment 1

Basic Concept: **Entity**

- An entity is a *specific* thing in the world, physical or conceptual.
 - **physical**: we can touch, occupying space
 - **conceptual**: cannot touch, but can still be *identified*
- The keyword here is: **specific**, so that it can be identified, physical or conceptual.
- We use the model of an entity to understand how business is done in an information system, whether or not it may involve a computerized engine.

Entity Type and Entity Set

An entity is a specific thing,
but we also need to refer to entities in groups...

- An **entity type** is a proposition (logical statement) about the entity – which may be true or false, depending on the specific entity.
- An **entity set** is a collection of entities for which the entity type statement is true – instances of the entity type

Mathematically...

Entity Set = { entity e : $P(e)$ };

Entity Type = the propositional statement $P(e)$.

$P(e)$ usually refers to a collection of attributes and the set of values accepted for the attributes.

Examples...

Entity Type: **teacher of DIST**

Entity Set: { Hirtle, Karimi, Spring, Wu, ... }

Entity Type: **car in Pittsburgh**

Entity Set: { Honda Acura, Ford Mustang, ... } ?

Entity Set: { truck, van, sedan, hatchback, convertible, ... } ?

Entity Set: { ... } ?

Entity Type: **car made and model**

Entity Set: { Honda Legend, Ford Taurus, ... }

Entity Type: **color**

Entity Type: { Red, Green, Blue, Yellow, ... }

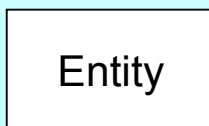
Entity Type: **book**

Entity Set: { SAD by KK, DSC by SKS, UML by CL, ... } ?

Entity Set: { ... } ?

Entity Set – in ER Model

- An entity type *defines* an entity set.
- Graphically, we use a **rectangle** to represent the entity set – a set of entities.



... think of it as a deck of cards, each representing an entity of that type; each identifies a specific entity in the set.

Examples... Entity Set in ER Model

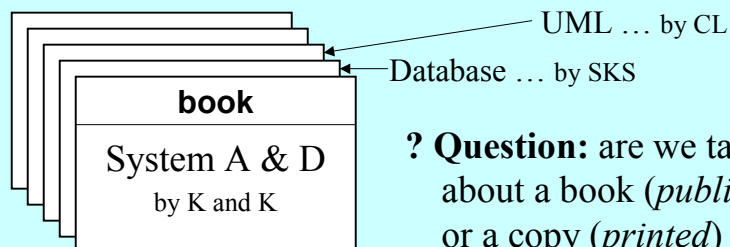
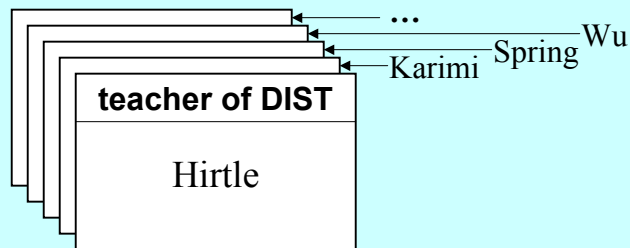
teacher
in DIST

The set of teachers in DIST.
Each entity in the set is a *specific* teacher.

BOOK

The set of books. Each
entity in the set is a *specific*
book – do we mean a copy?

Examples... Entity Set in ER Model



? Question: are we talking
about a book (*published*)
or a copy (*printed*) ?

Basic concept: **Attributes and Values**

How do we represent an entity in our model?

- with the appropriate **values** to a set of **attributes** about the entity... one value for each attribute.
- A teacher is an entity.
- We model each teacher with two attributes:
 - name:
 - phone:
- Every specific teacher is modeled by the specific values to the attributes name and phone.
 - name: *M. Spring*
 - phone: *x9429*

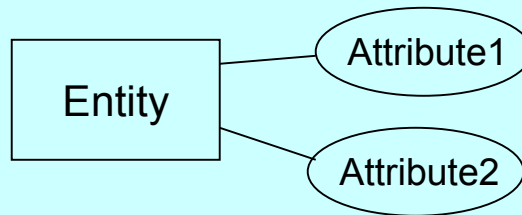
Basic concept: **Attributes and Values**

Note that... (difference between attribute and value)

- An attribute is like a property or characteristic.
(attributes of a course: title, course number, ...)
- Each attribute is identified by a name.
(Entity Set: **course**. Attributes: **title**, **course number**.)
- Entities of the same Entity Set have the same attributes.
(Each course should have a title and a course number.)
- Each entity may have different values for the attributes.
- For a *specific* course (ie, entity in the Entity Set **course**):
(the attribute is **title**; the value is “**Information Systems**”.)
(the attribute is **course number**; the value is **2510**.)

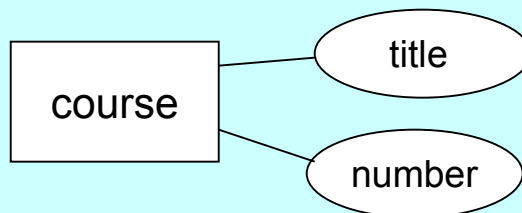
Entity Set with Attributes

- An entity type describes an entity set.
- Graphically, we use a **rectangle** to represent the entity set, and use an **oval** to represent each attribute we are interested in...



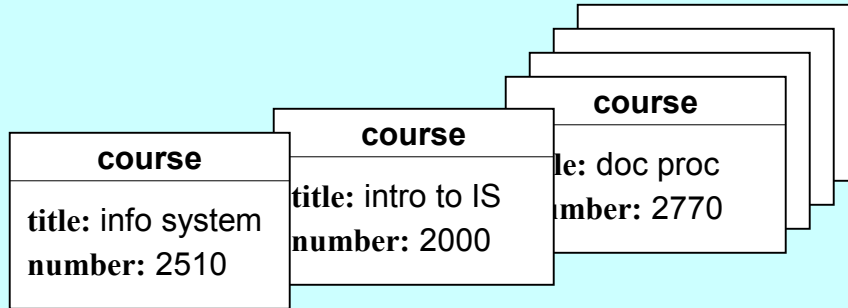
Entity Set with Attributes - Example

- Entity Type – **course**, describes an entity set.
- Every course (entity in the entity set **course**) has two attributes – **title** and **number**.



- Each specific course has specific values for each of the two attributes. The **title** is a text string; the **number** is a 4 digit decimal number.

Example... Entity Set with Attributes



... think of each entity set as a deck of cards.

Each card is labeled with entity type, and the **same attributes**, being of the same type.

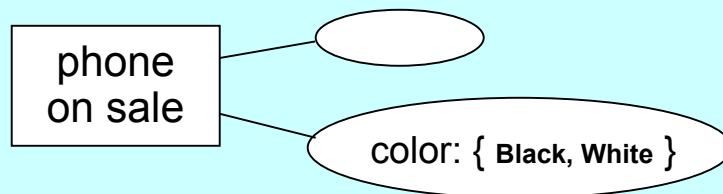
Each card has its own **specific values** for each of the attributes, representing a specific entity.

Attributes and Value Sets

- An attribute is a property or characteristic of an entity. For example, the **title** of a course, the **name** of an instructor.
- For an entity, an attribute takes a *specific* value. For example, this specific course has the **title** “*Information System Analysis and Design.*”
- An attribute should have a **data type**, which indicates the **set of values** it can take. For example, the **title** of a course is a *text string*.

Attributes and Value Sets

- An attribute may be very limited in terms of the values it can take. For example, color of phones on sale can only be black, or white. Then we say that value set is { **Black, White** }.



Entity and Attribute, formally

- For an entity set E , we are interested in the attributes A_1, A_2, \dots, A_n of its entities $e \in E$.
- Each attribute A_i is a function
$$A_i: E \rightarrow D_i$$
 where D_i is the value set for A_i .
- Every entity $e \in E$, is represented by the values corresponding to its attributes A_1, A_2, \dots, A_n such that
$$(d_1, d_2, \dots, d_n) \in D_1 \times D_2 \times \dots \times D_n$$
- The entity type is a statement $P(e)$ about e , that is the attribute values; the entity set is therefore
$$E = \{e \text{ as } (d_1, d_2, \dots, d_n) \in D_1 \times D_2 \times \dots \times D_n : P(e)\}$$

Topics

- Data Modeling in the context of ISAD
- Basic Concepts: Entities and Attributes
- **Relationships and Roles**
- Design Issues
- Structural Constraints
- Uniqueness and Key Attributes
- Weak Entity Set
- Assignment 1

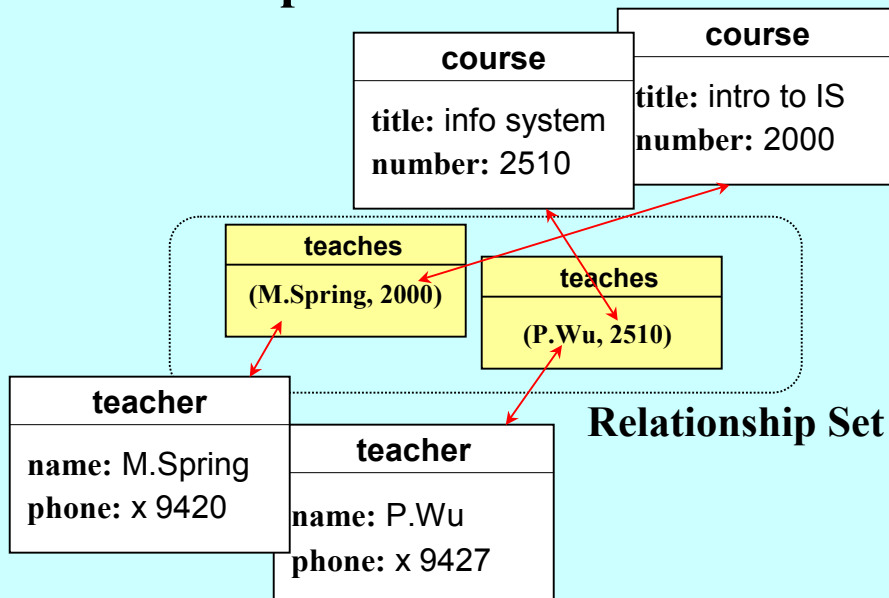
Relationship and Roles

- Entities may be related one to another, each entity takes a **role** in the **relationship**.
Example: a course, a teacher, and a student may be related – the student is taking the course taught by the teacher.
- The number of **roles** (therefore entities) involved is called the **degree** of the relationship.
- In our ER modeling, we will focus on:
 - relationships of a fixed degree;
 - relationships of degree 2, binary relationships.
- Refer to Section 2.2.3 of SKS for justification.

Relationship Type, Relationship Set

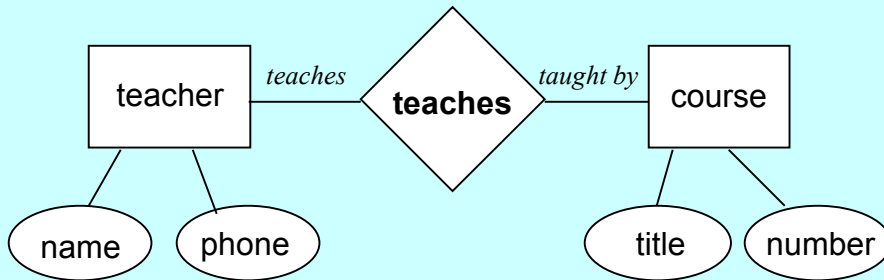
- A **relationship type** is a generic description of the nature of a relationship between the entity types.
- An **instance** of the relationship type an **ordered** tuple of the related entities.
- A **relationship set** is the **collection of all such instances** of the relationship type.
- **Example:** “an instructor teaches a course” is the generic description. (Wu, 2510), (Spring, 2000) are two **instances** of the **relationship type**.
- **Relationship Set** = {(Wu,2510), (Spring,2000), ...}.

Relationship Instances...

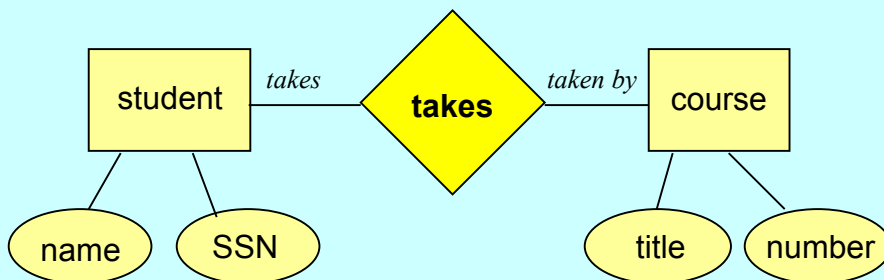


Relationship Set in ER Model

- A **relationship set** is represented as a *diamond* in the ER diagram, connecting entity sets.
- We *may* label the roles along the connecting lines.

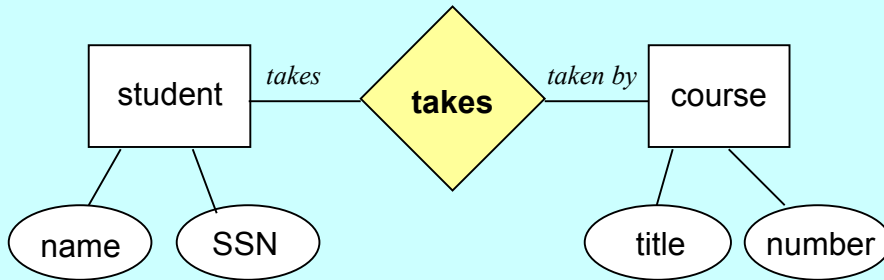


Relationship Set, Relationship Instances



- If there are 42 students, and there is only one course offered in the program, and all the students take that course, how many **relationship instances** are there in the **relationship set "takes"**?

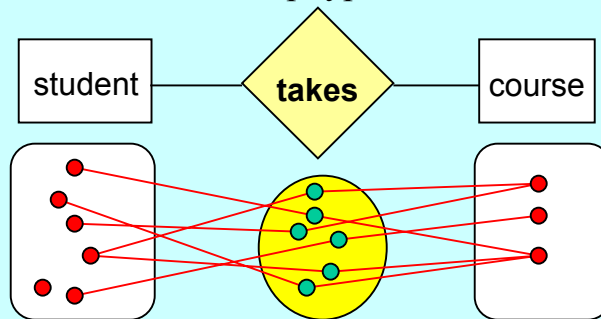
Relationship Set, Relationship Instances



- There are 42 students, and there two courses offered. 30 students take one course, and 25 students take the other course. How many **relationship instances** are there in the **relationship set** “takes”?

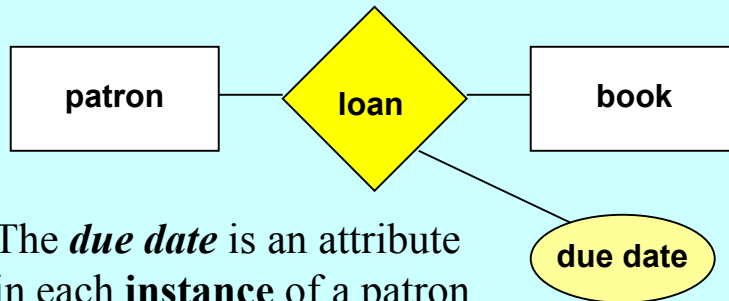
Relationship Type, Relationship Set

- A **relationship type** is a generic description of the nature of a relationship between the entity types.
- A **relationship instance** brings together the *specific* entities which are related by the relationship.
- A **relationship set** is the collection of all the relationship instances of the relationship type.



Relationship and Attributes

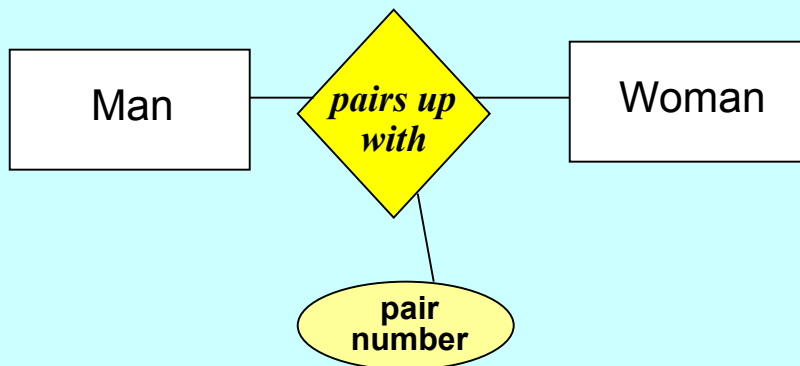
- A Relationship Type can have attributes, too.
- Identifies a property or characteristic of each instance of the relationship type...



- The *due date* is an attribute in each **instance** of a patron taking out a book on loan.

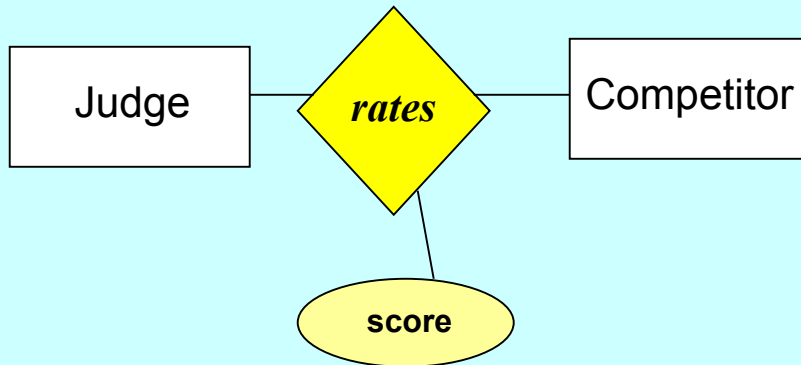
Relationship and Attributes

Men and Women members of a skating club are going to pair up in a competition....



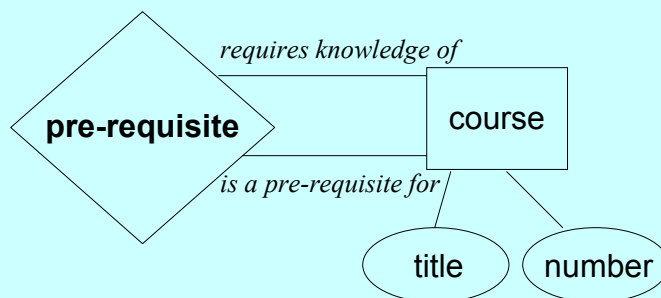
Relationship and Attributes

Each competitor will compete and get rated by a panel of judges, each giving a score...



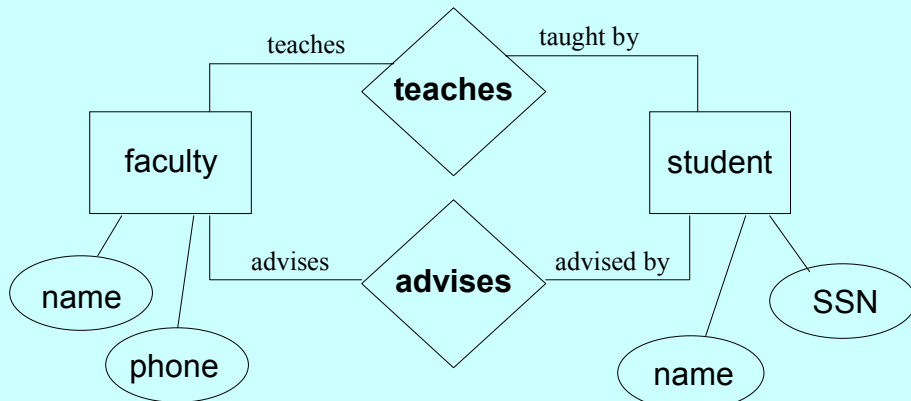
Relationship and Roles

- The same entity set can be related to the same relationship set; note the *different* roles.



Relationship and Roles

- The **same two entity sets** can be related to each other in two *different* relationship sets.



Topics

- Data Modeling in the context of ISAD
- Basic Concepts: Entities and Attributes
- Relationships and Roles
- **Design Issues**
- Structural Constraints
- Uniqueness and Key Attributes
- Weak Entity Set
- Assignment 1

Design Issues: what is a good model?

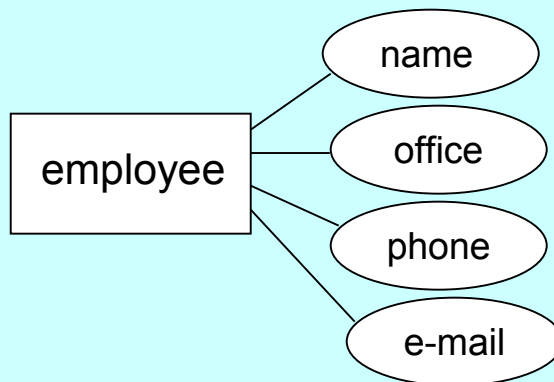
Conceptual Design –

- to focus on the **correctness** of representation;
- to understand business in **application domain**;
- do *not* have to worry about performance issues;
- still, there are issues left unclear sometimes...

Example: attribute or entity?

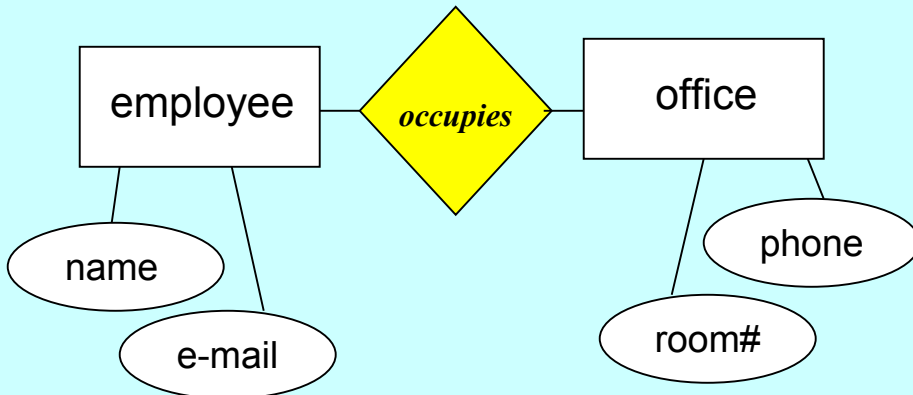
But the employees may be moved around to different offices, and phone numbers are fixed for each office.

- We have **office** as an attribute of each employee...



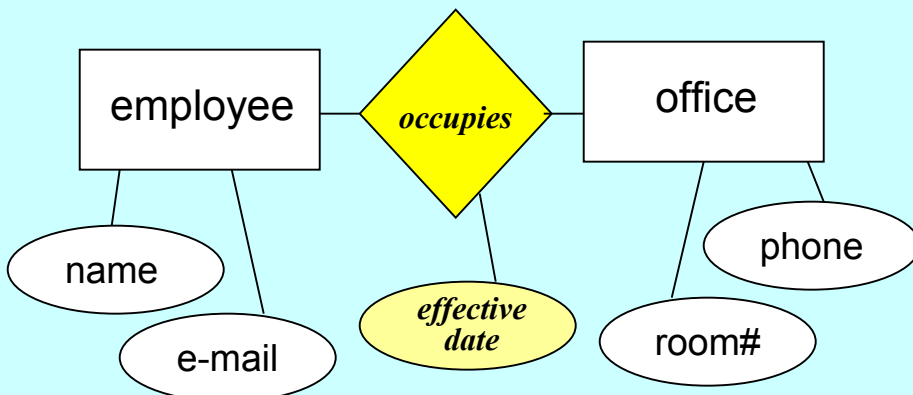
Example: attribute or entity?

We want to build a directory info system for a company. We turn office into an entity...



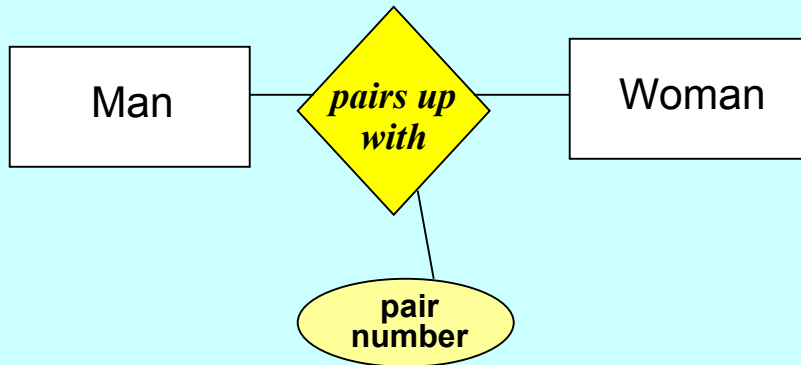
Example: attribute or entity?

So that we may also keep track of changes in the near future in our information system...



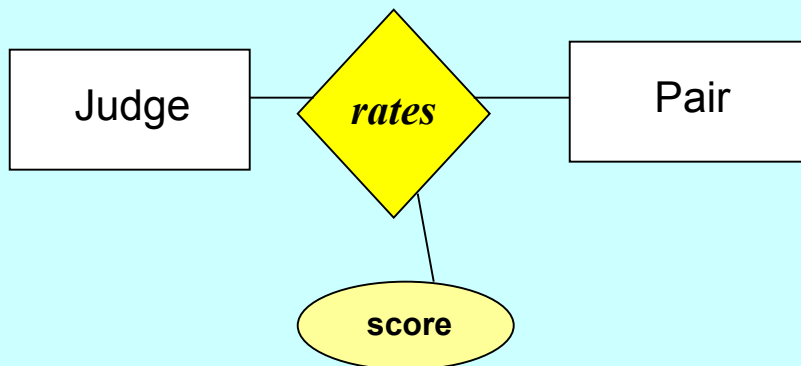
Example: relationship or entity?

Men and Women members of a skating club are going to pair up in a competition....



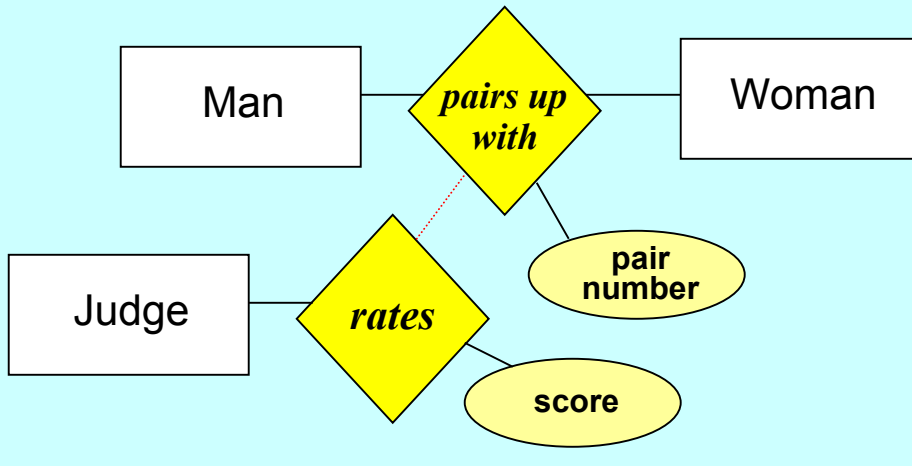
Example: relationship or entity?

Each pair will compete and get rated by three judges, each giving a score...

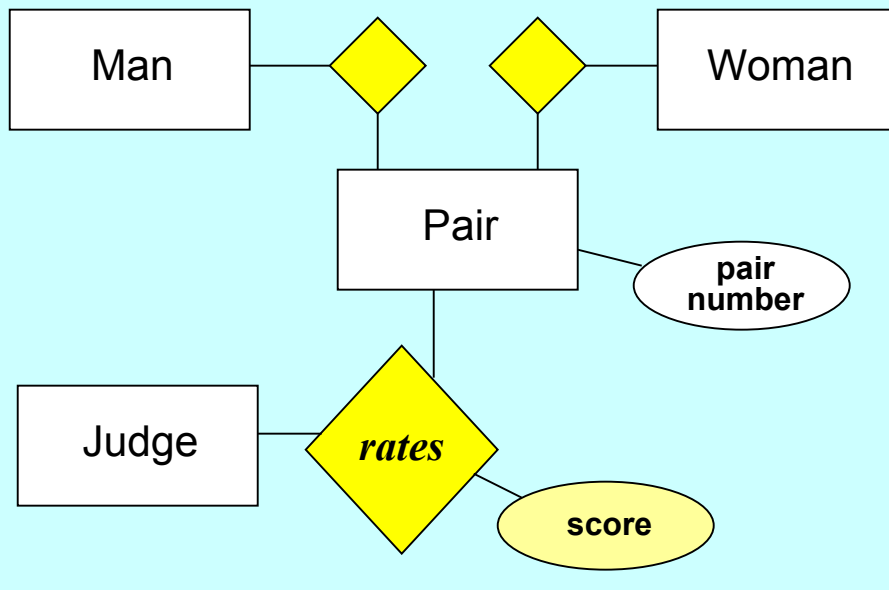


Example: relationship or entity?

- A relationship is the association of entities, **not** that of other relationships!



Example: relationship or entity?



ER Modeling Design Rules

- Each entity is *specific*, and can be identified.
- Each **relationship instance** *relates specific entities*, and therefore can also be identified.
- A relationship set may relate entity sets, **NOT** other relationship sets.
- It is preferred for relationship set to relate only two entity sets: **binary relationships only**.

Q: Relationship with Relationship?

- Graphically in the ER Diagram, it means that we connect a diamond to another diamond, and this is NOT allowed. Why?
- The answer to this may be difficult now; we will try to understand more about ER model and hopefully it will become clearer...
- Let us study the **constraints** on a relationship.

Topics

- Data Modeling in the context of ISAD
- Basic Concepts: Entities and Attributes
- Relationships and Roles
- Design Issues
- **Structural Constraints**
- Uniqueness and Key Attributes
- Weak Entity Set
- Assignment 1

Relationship & Structural Constraints

- The information structure depicted in our ER model has certain constraints, pertaining to the nature of the relationships in the model:
 - **Participation Constraint:** about how the entity sets are participating in the relationship.
 - **Cardinality Constraint:** about the ratio of the number of entities in each entity sets being related.
- Collectively, these are known as the **structural constraints** of the Entity-Relationship model.

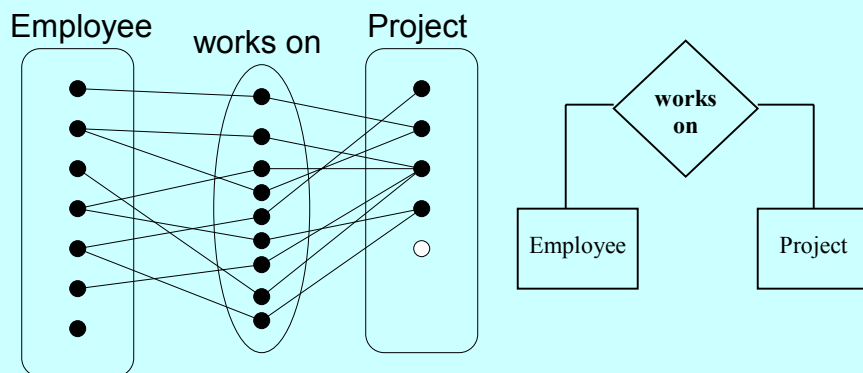
Participation Constraint

When an entity set participates in a relationship, it may participate *partially*, or *totally*.

- **Partial Participation:** some entities of the set do not participate in any relationship instance.
- **Total Participation:** every entity of the set participate in at least one relationship instance.

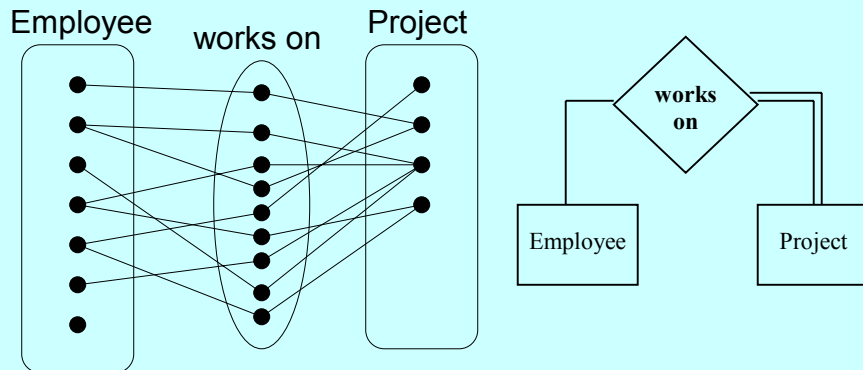
Partial Participation

- Example of *partial participation*: some employees may not work on any project. The entity set Employee participates partially in the relationship “works on”.
- But we *may* not allow a project with no employees.



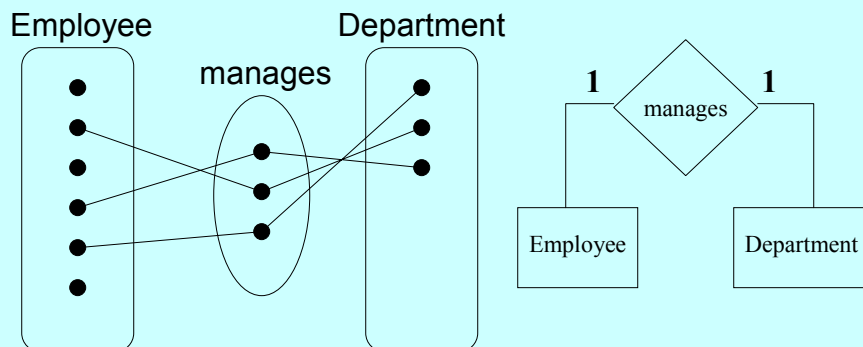
Total Participation

- Every project must have at least one employee.
- The entity set *Project* **participates totally** in the “works on” relationship: a double connection line.



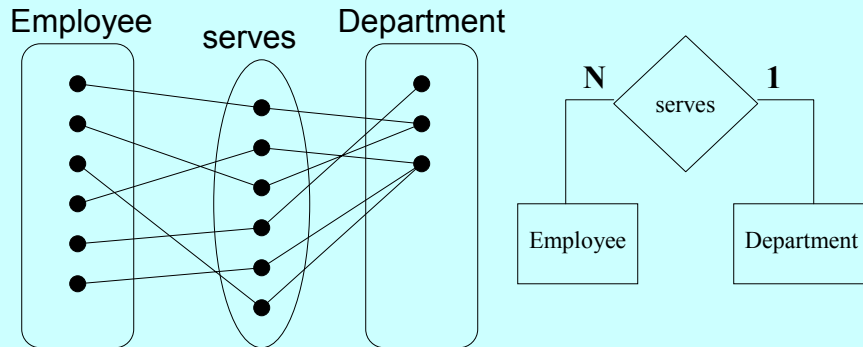
Cardinality Ratio

- When an entity set participates in a (*binary*) relationship, the cardinality ratio specifies the ratio between number of related entities in the relationship.
- For example: *at most* one manager for every department, and a manager manages *no more* than one department.



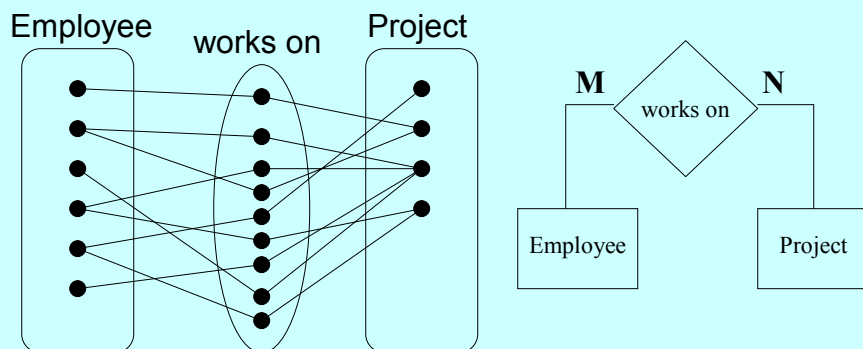
Cardinality Ratios

- There may be other forms of ratios...
- We label only these forms – **1:1**, **1:N**, **N:1**, **M:N**.
- There may be many employees serving one department.



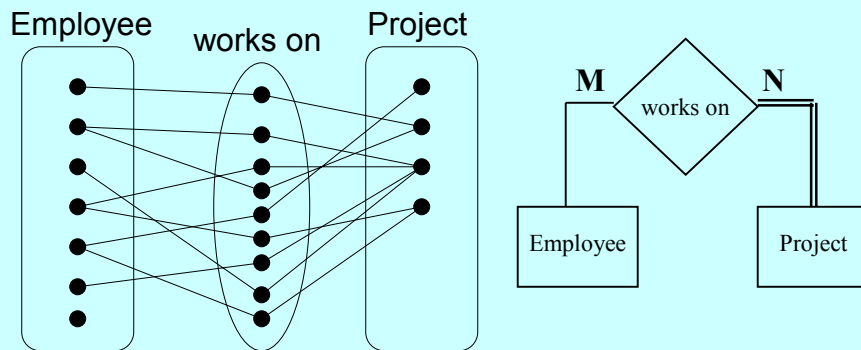
Cardinality Ratios

- An example of **M:N** ratio – an employee may work on more than one project; any project may have more than one employee working on it.



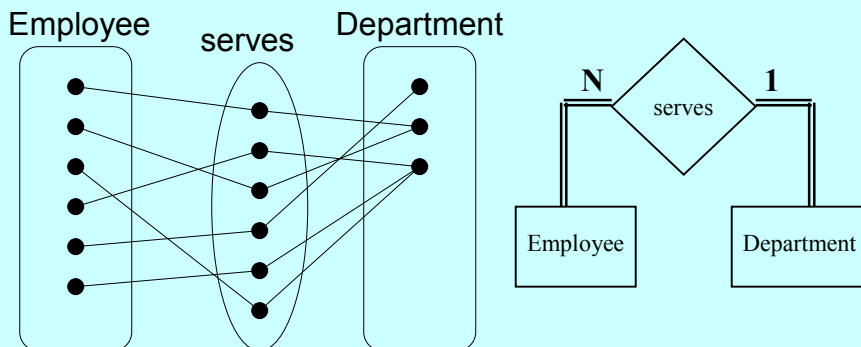
Cardinality and Participation

- Two *different dimensions* of the constraints on a relationship: **any** combination is possible.



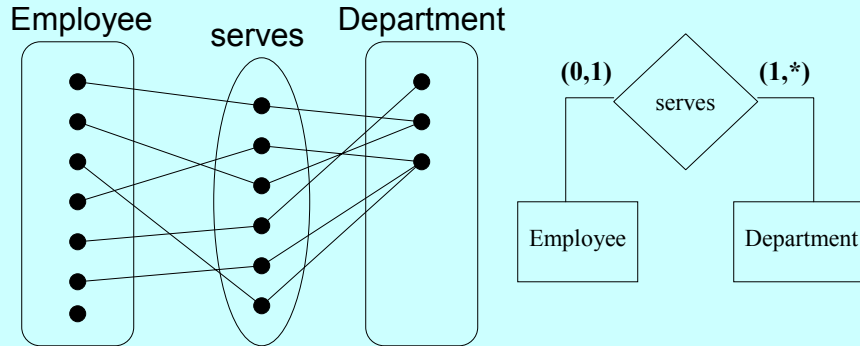
Cardinality and Participation

- Every department must have at least one employee.
- Every employee must serve exactly one department.
- There may be many employees serving one department.



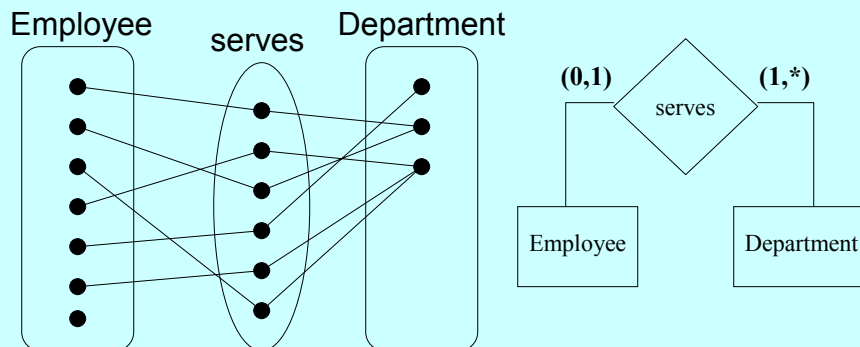
Structural Constraints: (min,max) notation

- We may label participation in a relationship (**min,max**) to specify the range for each entity in the participation: min/max is an integer value or * - an arbitrary number.



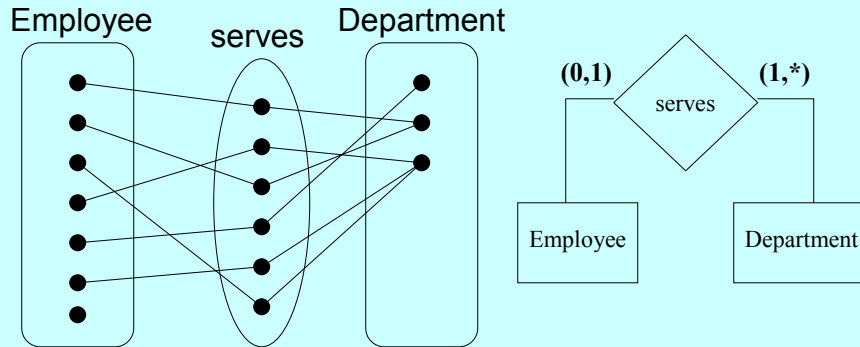
Structural Constraints: (min,max) notation

- An employee may serve no department; **min=0**.
- An employee may serve at most one department; **max=1**.
- A department must have at least one employee; **min=1**.
- A department may have up to any number of employees; **max=*.**



Two notations are related...

- The **min** value of **0** implies partial participation.
- The **min** value of **1 or larger** implies total participation.
- What can we say about the **max values** and *cardinality ratios*?

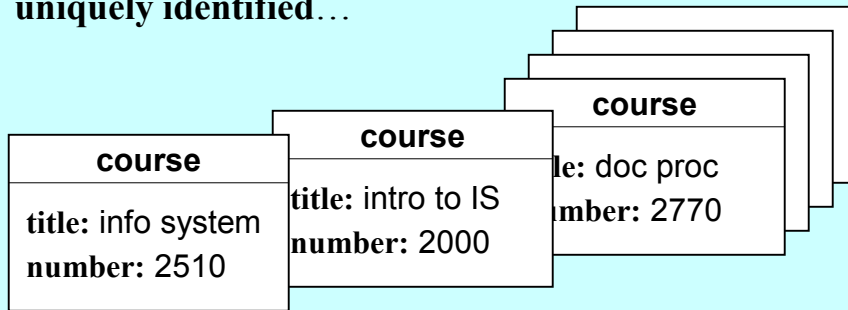


Topics

- Data Modeling in the context of ISAD
- Basic Concepts: Entities and Attributes
- Relationships and Roles
- Design Issues
- Structural Constraints
- **Uniqueness and Key Attributes**
- Weak Entity Set
- Assignment 1

Uniqueness Constraint

- An entity set consists of a collection of entities.
- Entity-Relationship modeling requires *each entity* to be **uniquely identified**...



- Each card (entity) must somehow be different from other cards (other entities).

Key Attributes

We *should* be able to **identify** each entity...

- How can we distinguish each entity from others?
- Every entity in the set is characterized by the collection of its attribute values.
- If e_1 and e_2 are two different entities, they must differ in value in at least one of the attributes.
- A student has **name** and **SSN** – the **name** alone is often insufficient to **identify** each student.

Key Attributes - Examples

- In a small study group, we can use first names; but sometime, we need to also use last name.
- In a larger class, it may be sufficient to use the full name (last and first).
- In school, we quite often use the SSN.

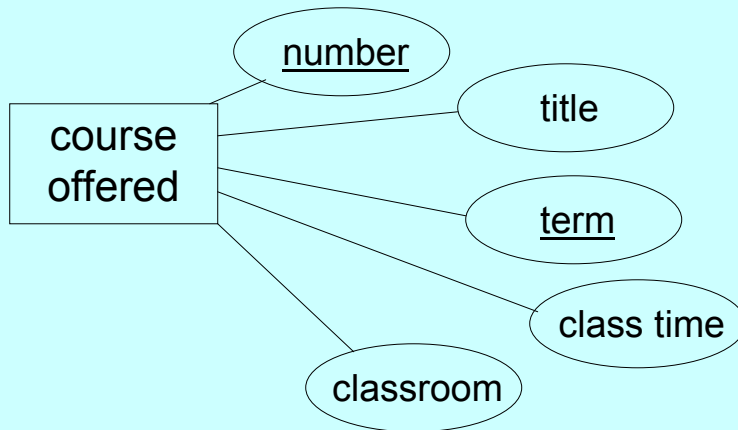
What is the **minimal** set of attributes necessary to *uniquely identify* each entity in an entity set?

Keys Attributes - definitions

- In an entity set, the collection of attributes which is **sufficient to identify** each entity is call a ***superkey*** – a set of attributes such that any two entities in the set must differ in *value in at least one* of those *attributes*.
- A **minimal superkey** is a set of attributes such that any proper subset would make it insufficient to serve as a *superkey*.

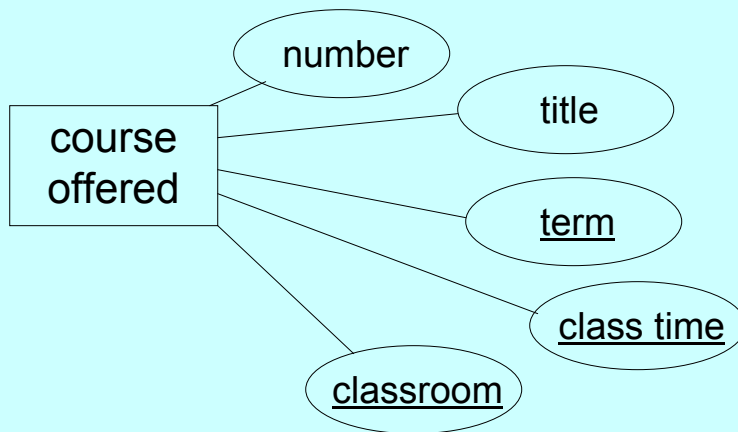
Key Attributes

- ER Model Designer should choose a minimal attribute set to be the key attributes, and indicate the choice: underline the attribute names.



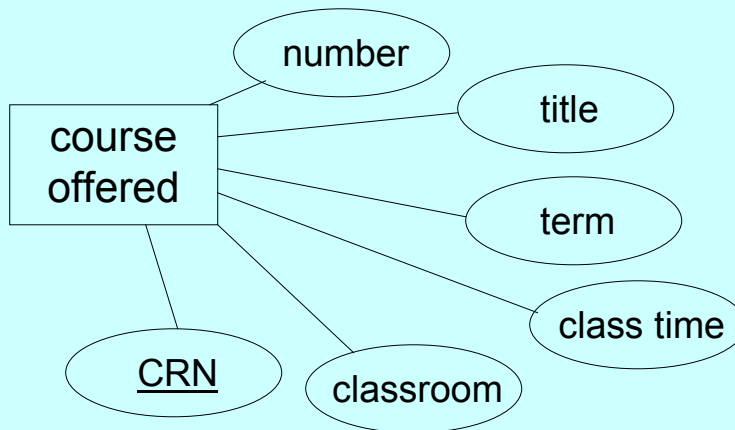
Key Attributes

- Not every set of key attributes makes a good choice...



Key Attributes

- The designer should choose a minimal key attribute set, and indicate the choice...



Q: Key attribute for a relationship?

- A relationship set can have attributes.
- A relationship type can have single-valued as well as multi-valued attributes.
- A relationship type can have stored as well as derived attributes.
- A relationship type should **NOT** any **key attributes**. *Why?*

Topics

- Data Modeling in the context of ISAD
- Basic Concepts: Entities and Attributes
- Relationships and Roles
- Design Issues
- Structural Constraints
- Uniqueness and Key Attributes
- **Weak Entity Set**
- Assignment 1

Weak Entity Set

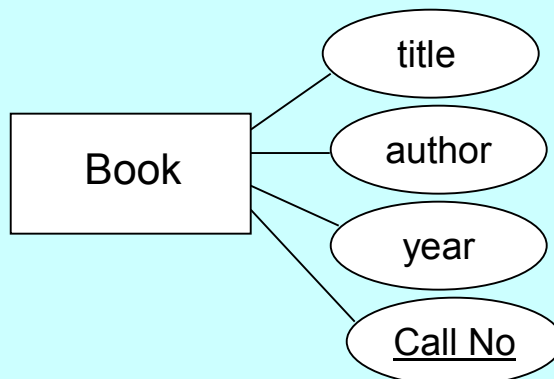
- Every *normal* entity set should have a set of key attributes specified. It is called the **uniqueness constraint** – required in an entity set, so that we can **identify each specific entity**.
- However, sometimes we *allow* an entity set to have NO key attributes but only **Partial** key attributes – called a **Weak Entity Set**.
- An entity set with key attributes specified is therefore sometimes called a **Strong Entity Set**.

Weak Entity Set

- A **strong entity set** should have at least one set of key attributes – so that every entity can be uniquely identified, fulfilling the uniqueness constraint.
- A **weak entity set** does not have a set of key attributes (but a set of partial key attributes), but the uniqueness constraint still requires every entity to be uniquely identified.
- A **weak entity set** must participate totally in a *dependency relationship*, and, if necessary, a set of *partial key attributes*.

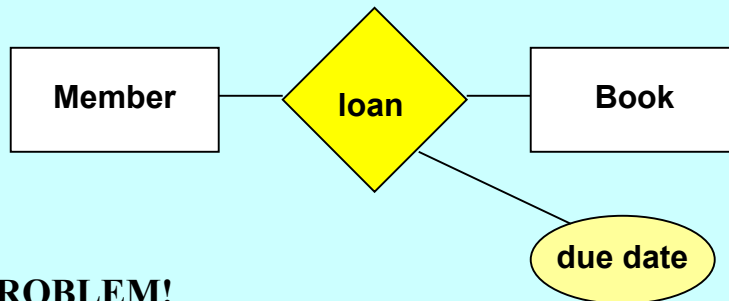
Weak Entity Set: Example

- Consider modeling the information for a library: we have the entity set **Book** for books...



Weak Entity Set: Example

- Members of the library may borrow books: we have a relationship between **Book** and **Member**...

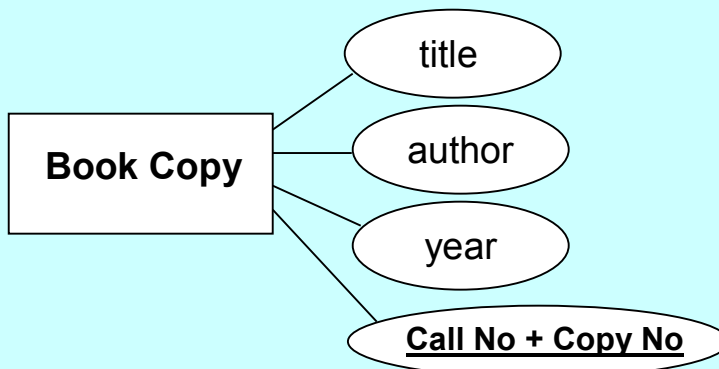


PROBLEM!

- Can we have two different members borrowing the same book? OR two different copies of the same book?

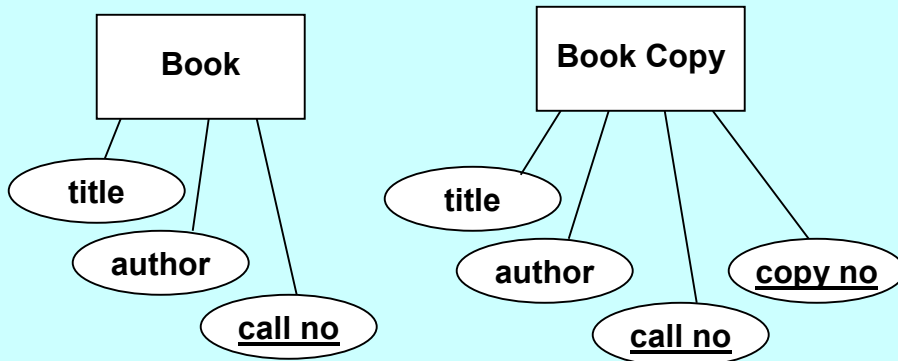
Weak Entity Set: Example

- The Book entity set: does each entity represent a book? OR a copy of a book? Need copy number!



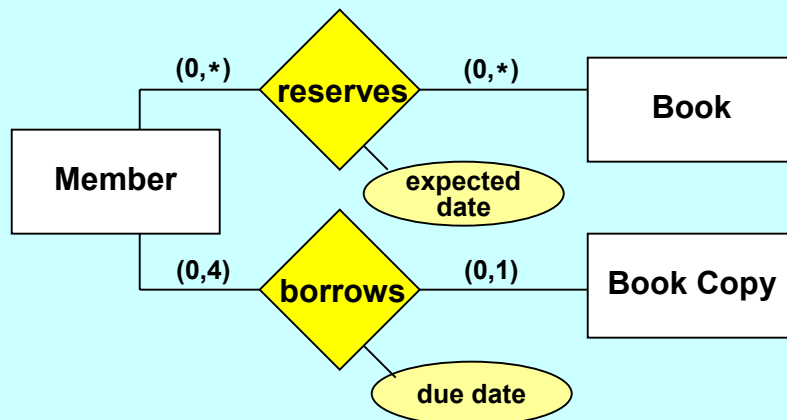
Weak Entity Set: Example

- We need to deal with two concepts in a library: **book**, and **copy of a book**:



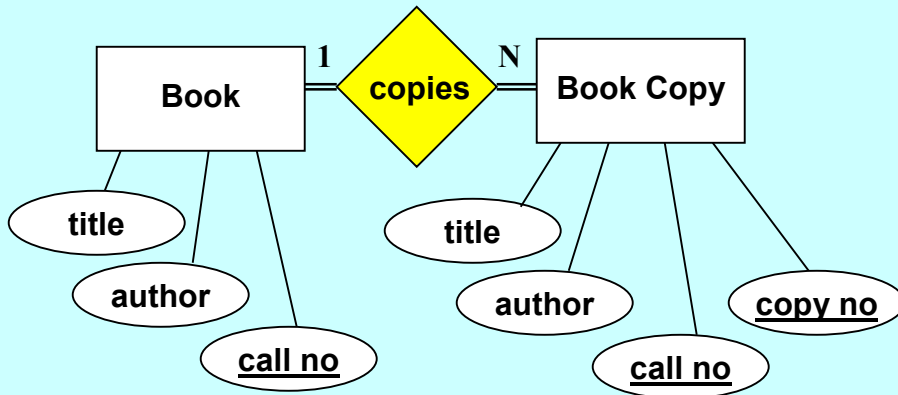
Weak Entity Set: Example

- With both **Book** and **Book Copy**, we can properly model the relationships with **Member**.



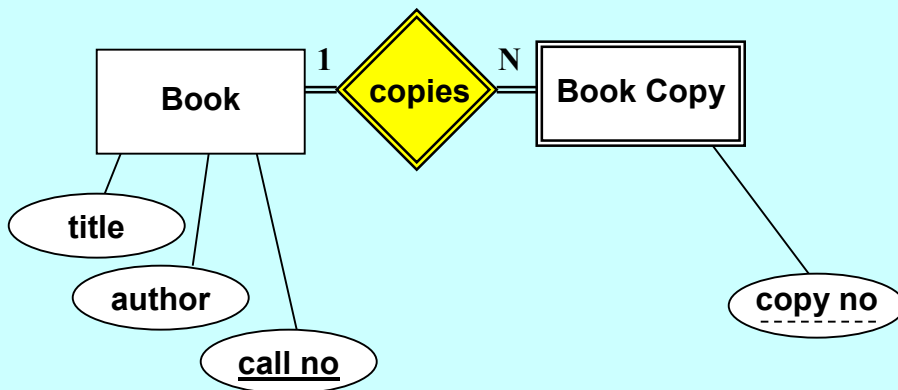
Weak Entity Set: Example

- We duplicate a good deal of information in the entity sets: **Book** and **Book Copy** – the two should be related!



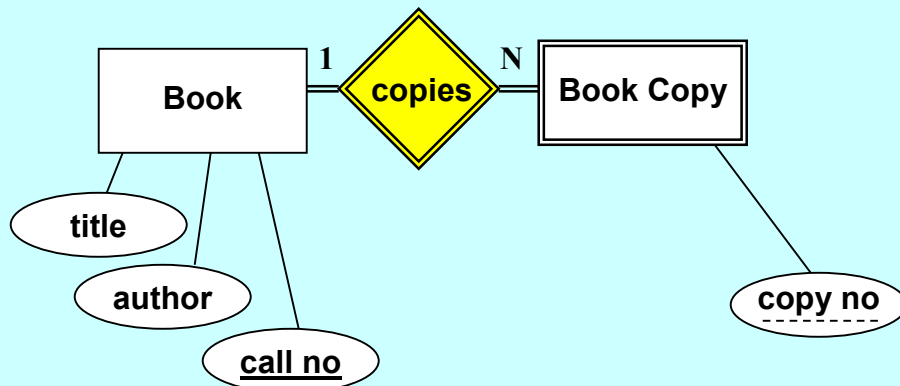
Weak Entity Set: Example

- We will allow a **weak entity set** for **Book Copy**, with partial key **copy no** (underlined by dash), and a dependency relationship **copies** to a strong entity set.



Weak Entity Set: Example

- Each book copy (entity in a weak entity set) is then uniquely identified by the copy no (partial key) for each book (strong entity it depends on).

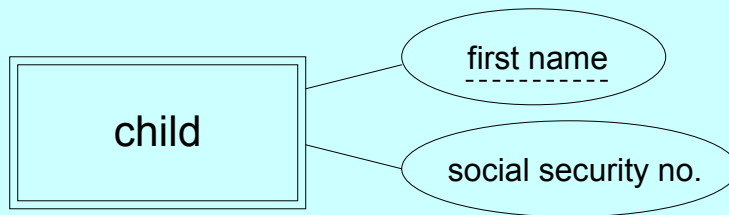


Weak Entity Set

- A weak entity set must **participate totally** in *at least* one **1-to-1** or **N-to-1** relationship which is the **dependency relationship**.
- A weak entity set may have a set **partial key** attributes (each underlined by dashed line).
- The partial key must be sufficient to identify each entity in the weak entity set, within each group (of N) related to the same entity through the dependency relationship, fulfilling the **uniqueness constraint**.

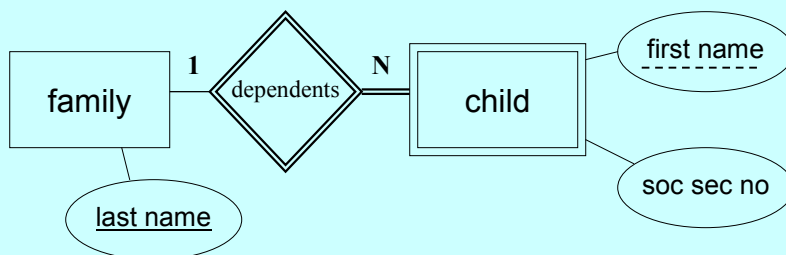
Weak Entity Set: Example

- In census, we model children in family units as entities in an entity set – called **child**.
- Each child has a **first name**, may or may not have a **social security number**; therefore the attributes are not sufficient to identify each child, except within the family.
- A **Weak Entity Set** has a double rectangle, and the attributes for a partial key, underlined in dash.



Weak Entity Set: Example

- The weak entity set depends on the relationship with a strong entity set to identify each entity in the set. It is called a *dependency relationship*.



Topics

- Data Modeling in the context of ISAD
- Basic Concepts: Entities and Attributes
- Relationships and Roles
- Design Issues
- Structural Constraints
- Uniqueness and Key Attributes
- Weak Entity Set
- **Assignment 1**

Assignment 1

- Due next Monday in class.
- OK to work in groups.
- No need to use extended constructs yet.
- Like in an interview, requirements spelled out may be somewhat fuzzy...
- May draw diagrams using Power Point, Visio, or Word, Corel Draw.... hand drawn diagram OK too – as long as they are readable, but please do not use pencil.