

Continue exploring our first model for databases:

Entity-relationship (ER) model

Board Example

Entity *Books*: (title, ISBN#, edition, date)

Entity *Authors*:
(name, gender, birth date, place of birth, date of death)

Entity *Publishers*: (name, country, address)

Relationship *written by*: (*books*, *authors*)

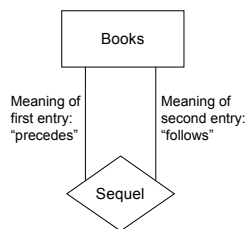
Relationship *published by*:
(*books*, *publishers*, in print)

Another relationship

Relationship *sequel*:
(*books*, *books*)

Any book can be on either
side of relationship
If sequence longer than 2,
middle book appears in
tuple on each side

Example tuples for *Lord of the Rings*:
(*Fellowship of the Ring*, *Twin Towers*)
(*Twin Towers*, *Return of the King*)



Identifying entities

Key: a minimal* set of attributes whose values
uniquely identify each entity in an entity set



□ represents a book

* No subset will do the job

Identifying entities

Key: a minimal set of attributes whose values
uniquely identify each entity in an entity set

Candidate Key: any key

Primary key: a candidate key defined to be primary
by person who defines entity

Superkey: any set of attributes that contains a
candidate key

Denote primary key by underlining attributes

Entity *Books*: (title, ISBN#, edition, date)

Entity *Authors*:
(name, gender, birth date, place of birth, date of death)

Entity *Publishers*: (name, country, address)

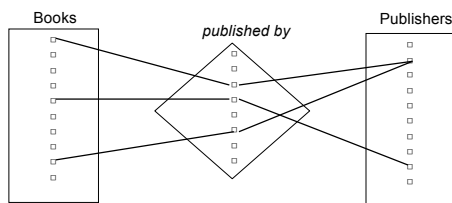
Constraints

- Declaring a candidate key constrains values of attributes
- Example: ISBN# as key
 - No book without an ISBN#
 - No two books with same ISBN#

What about constraints on relationships?

- Constraints are statements about structure

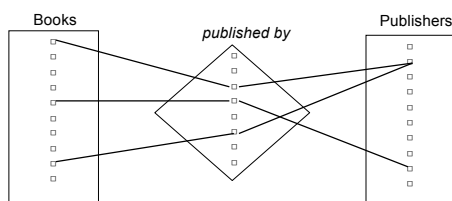
How many entities have link?
How many links per entity?



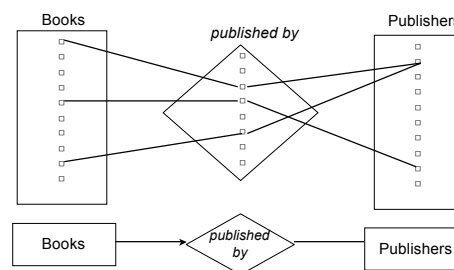
Two major constraints for relationships

- Key constraint on an entity type participating in a relationship type:
 - Each entity of the entity type appears in at most one tuple of the relationship type
 - Equivalently, the value of the key-constrained entity determines the values of the other entities in a relationship tuple
- Total Participation constraint of an entity type participating in a relationship type:
 - Every entity in the entity set of the entity type appears in a tuple of the relationship

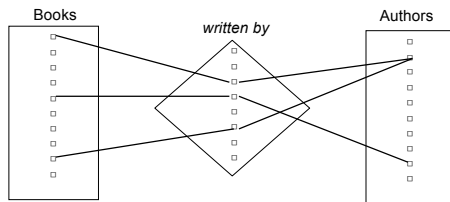
Example: key constraint on books for
published by?



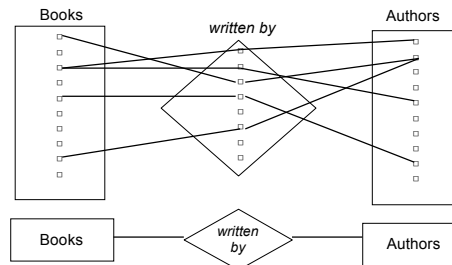
Example: key constraint on books for
published by? **YES**
But remember NOT by inspection!!



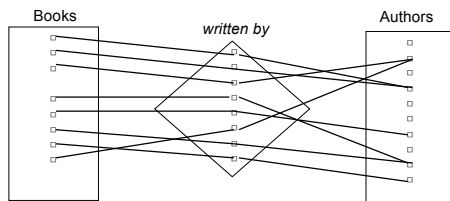
Example: key constraint on books for *written by*?



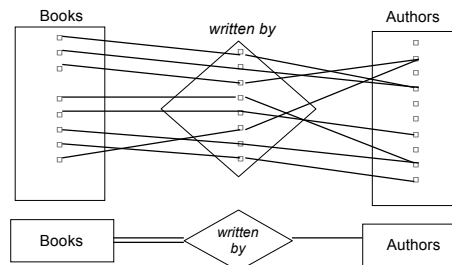
Example: key constraint on books for *written by*? NO
But remember NOT by inspection!!



Example: total participation constraint on books for *written by*?



Example: total participation constraint on books for *written by*? **YES**
all books authored by someone

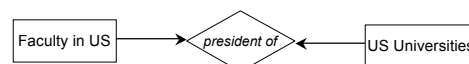


Fill in rest of constraints on board diagram

- watch what constraint means
- watch intended use of database

Multiple Key Constraints

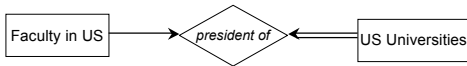
- can have binary relationship where both entity sets satisfy key constraint
– i.e. both entity sets keys for relationship



Total participation constraints?

Multiple Key Constraints

- can have binary relationship where both entity sets satisfy key constraint
 - i.e. both entity sets keys for relationship



Constraints cannot denote in basic ER model

- Domain attribute constraints within entity
 - Need to test values of attributes not simply membership properties in sets
 - Example:
 - Attribute *NJ driver*: yes/no flag
 - Attribute *age*: number
 - Constraint "if age < 17 then *NJ driver* == "no"

Constraints cannot denote in basic ER model

- Functional constraints

Example:

person entity with 6 attributes:
first name, last name, street address, state,
area code, 7-digit phone number.

Constraint:

if *area code* of person 1 = *area code* of person 2
 then *state* of person 1 = *state* of person 2

Equivalently, *area code* determines *state*

Constraints cannot denote in basic ER model

- Functional constraints

General form:

Let A and B be subsets of attributes for an entity type.

For any entities e_j and e_k of the type:

If the values of attributes in set A for tuple e_j equal
 the values of attributes in set A for tuple e_k

Then the values of attributes in set B for tuple e_j equal
 the values of attributes in set B for tuple e_k

Constraints cannot denote in basic ER model

- Functional constraints

More complicated example:

customer entity with 8 attributes:
height, weight, arm length, leg length, color preference,
jacket size, pant size, shirt size

Constraint:

Height, weight, arm length determine *shirt size*
Height, weight, leg length determine *pant size*

n-ary relationships, $n > 2$

Example- scheme:

tutorial offering = Tutorials X Instructors
 X Conferences

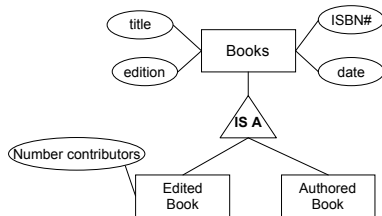
- Do with binary relationships?
- Capturing constraints. Careful!

Richness of ER model

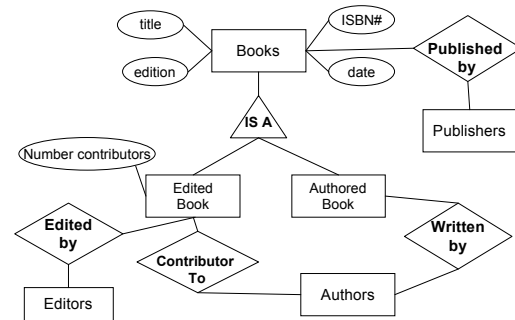
Richness of ER model

1. Inheritance for entities:
 - An entity type can be further refined 
2. Tuples can be composed of tuples – hierarchy:
 - Relationship tuples composed of entity tuples
 - By aggregation, relationship tuples can contain relationship tuples

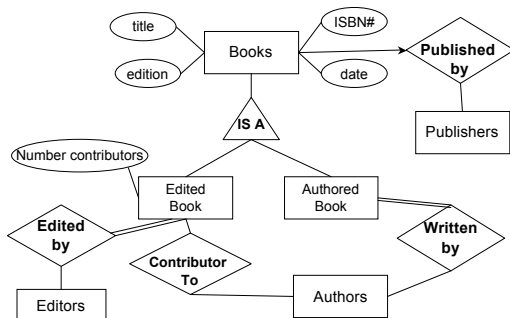
Inheritance hierarchy for entities



Using hierarchy with relationships



Key and total participation constraints possible



We just declared constraints:

1. A book is published by at most one publisher (key constraint)
2. Every authored book is written by some author (total participation constraint)
3. Every edited book is edited by some editor (total participation constraint)

Constraints on “is a” decomposition

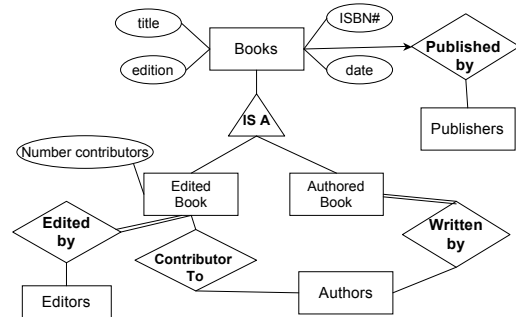
- **Covering**
If union of subclasses = superclass
 $S = S1 \cup S2$



- **Overlapping**
If subclasses can overlap
 $S1 \cap S2 \text{ may } \neq \emptyset$

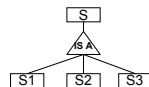


Overlapping? Covering?

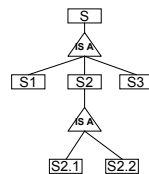


GENERAL “IS A”

- Can have > 2 subclasses for one “IS A”

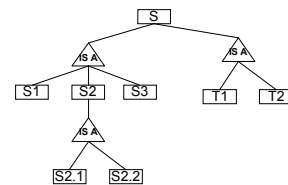


- Subclasses can be further refined with “IS A”
– Deep hierarchy possible



GENERAL “IS A”

- Can have more than one “IS A” refinement for an entity



Another example

