CS2102 Database Systems

Slides adapted from Prof. Chan Chee Yong

LECTURE 04

ENTITY RELATIONSHIP DATA MODEL

Null values

Three-valued logic system: `TRUE, FALSE, UNKNOWN`

x	y	x AND y	x OR y	NOT x
FALSE FALSE FALSE	FALSE UNKNOWN TRUE	FALSE FALSE FALSE	FALSE UNKNOWN TRUE	TRUE
UNKNOWN UNKNOWN UNKNOWN	FALSE UNKNOWN TRUE	FALSE UNKNOWN UNKNOWN	UNKNOWN UNKNOWN TRUE	UNKNOWN
TRUE TRUE TRUE	FALSE UNKNOWN TRUE	FALSE UNKNOWN TRUE	TRUE TRUE TRUE	FALSE

x	x IS NULL	x IS NOT NULL
null	TRUE	FALSE
non-null	FALSE	TRUE

x	y	x IS DISTINCT FROM y
null	null	FALSE
null	non-null	TRUE
non-null	null	TRUE
non-null	non-null	x <> y

Data definition language (DDL)

Create table

```
CREATE TABLE [ IF NOT EXISTS ] table_name ( [

    { column_name data_type

        [ column_constraints [ ... ] ]

      | table_constraints }

    [, ...]

] );
```

Drop table syntax

```
DROP TABLE [ IF EXISTS ] table_name
```

Data definition language (DDL)

Alter table

- Add/remove/modify columns
  - `ALTER TABLE students ALTER COLUMN dept DROP DEFAULT;`
  - `ALTER TABLE students DROP COLUMN dept;`
  - `ALTER TABLE students ADD COLUMN faculty varchar(20);`
  - etc
- Add/remove constraints
- etc

Data manipulation language (DML)

Insert into syntax

```
INSERT INTO table_name

[ ( column_name [, ...] ) ]

VALUES ( { expression | DEFAULT } [, ...] );
```

Delete from syntax

```
DELETE FROM table_name

[ WHERE condition ];
```

Update syntax

```
UPDATE table_name

SET column_name = { expression | DEFAULT }

[ WHERE condition ];
```

Simple queries

Basic syntax

◦ Basic form of SQL query consists of *three clauses*

```
SELECT [ DISTINCT ]  select_list  -- select clause
FROM                 from_list    -- from clause
[ WHERE              condition ]  -- where clause
```

◦ `select_list`   specifies columns to be included in output

◦ `from_list`   specifies list of relations

◦ `condition`   specifies conditions on relations

❖ **Output:** relation generated from `from_list` containing attributes based on `select_list` that satisfies `condition`

➢ Output relation could contain duplicate record if `DISTINCT` is not used in the `SELECT` clause

- Entity Relationship Diagram
  - ER model
  - Relationship constraints
  - Participation constraints
  - Weak entity sets

- ER to SQL
  - ER diagram to SQL
  - Additional ER concepts
  - ER design and relational mapping

Overview

- Entity Relationship Diagram
  - ER model
  - Relationship constraints
  - Participation constraints
  - Weak entity sets

- ER to SQL
  - ER diagram to SQL
  - Additional ER concepts
  - ER design and relational mapping

Entity Relationship Diagram

Conceptual data models

Introduction

- **Entity-relationship (ER) model**
  - Developed by Peter Chen in 1976
  - Designed for conceptual data model specifications
  - Most common data model used for database design
- **Unified modelling language (UML)**
  - Developed by Grady Booch & James Rumbaugh in 1997
  - Goes beyond conceptual data modelling
    - Software design specifications
  - Standardized by Object Management Group (OMG)

Database design process

Steps

1. **Requirement analysis**
   - Find out the data/application/performance requirement of the enterprise

2. **Conceptual database design**
   - Capture data requirements using a conceptual schema

3. **Logical database design**
   - Map conceptual schema to logical schema supported by DBMS

4. **Schema refinement**
   - Improve logical schema design using data constraints

5. **Physical database design**
   - Use performance requirements to design physical schema

6. **Application & security design**
   - Specify access control policies

Database design process

Requirement analysis

I would like my customers to be able to browse my catalog of books and place orders over the Internet. Currently, I take orders over the phone. I have mostly corporate customers who call me and give me the ISBN number of a book and a quantity; they often pay by credit card. If I don't have enough copies in stock, I order additional copies and delay the shipment until the new copies arrive; I want to ship a customer's entire order together. My catalog includes all the books I sell. For each book, the catalog contains its ISBN number, title, author, purchase price, sales price, and the year the book was published. Most of my customers are regulars, and I have records with their names and addresses. New customers have to call me first and establish an account before they can use my website. On my new website, customers should first identify themselves by their unique customer identification number. Then they should be able to browse my catalog and to place orders online.
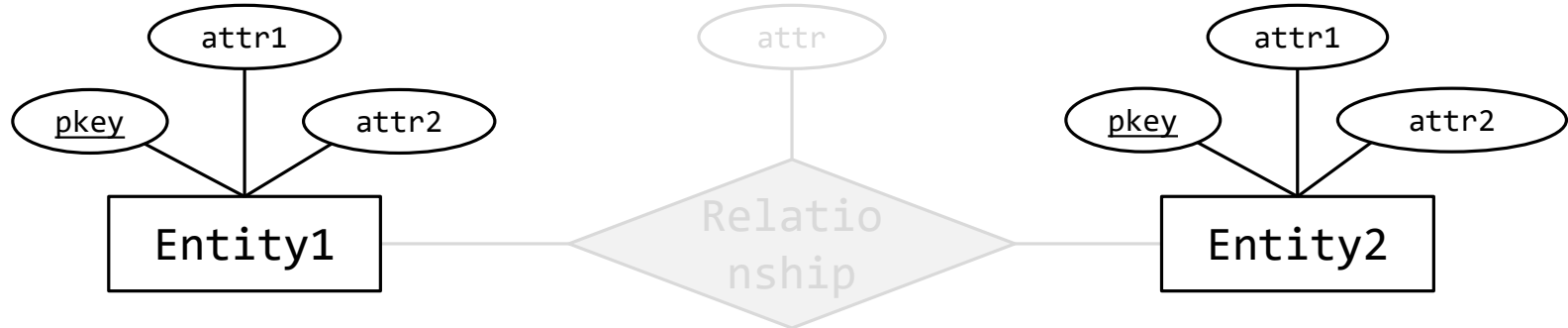
ER model

Introduction

- Data is described in terms of entities and their relationships
- Information about entities and relationships are described using attributes
- Certain data constraints are represented using additional annotations
- ER schemas are presented as ER diagrams
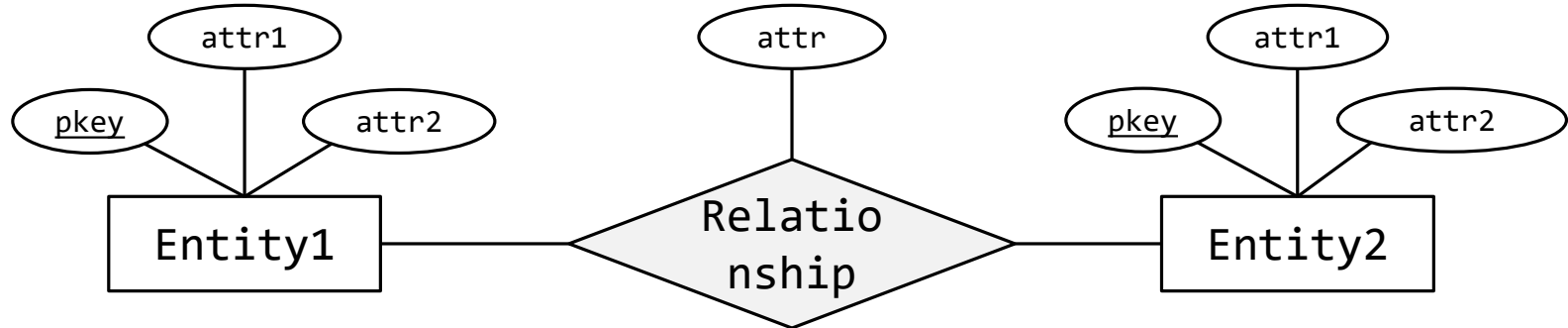
ER model

Basic

- **Entity** *Real-world object distinguishable from other objects*
- **Attribute** *Specific information describing an entity*
  each attribute has an atomic domain (e.g, integer, string)
  - *Represented by ovals*
- **Entity set** *A collection of similar entities*
  - *Represented by rectangles*

ER model
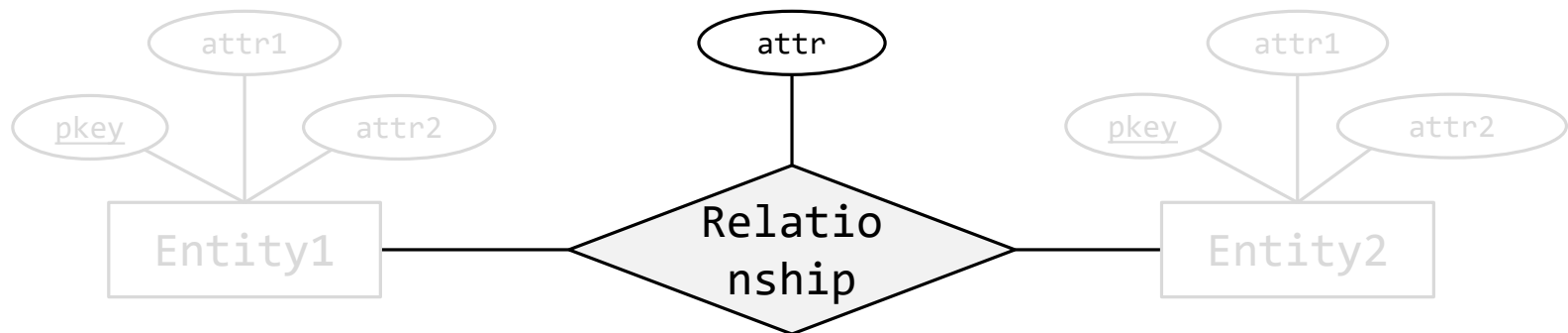
Keys

- Each entity set has a key (*i.e., minimal set of attributes whose values uniquely identify an entity*)
- An entity set could have multiple keys called candidate keys
- One of the candidate keys is chosen as the primary key
- The attributes that formed a primary key are underlined
- Example: `sid` in `Students` and `cid` in `Courses`
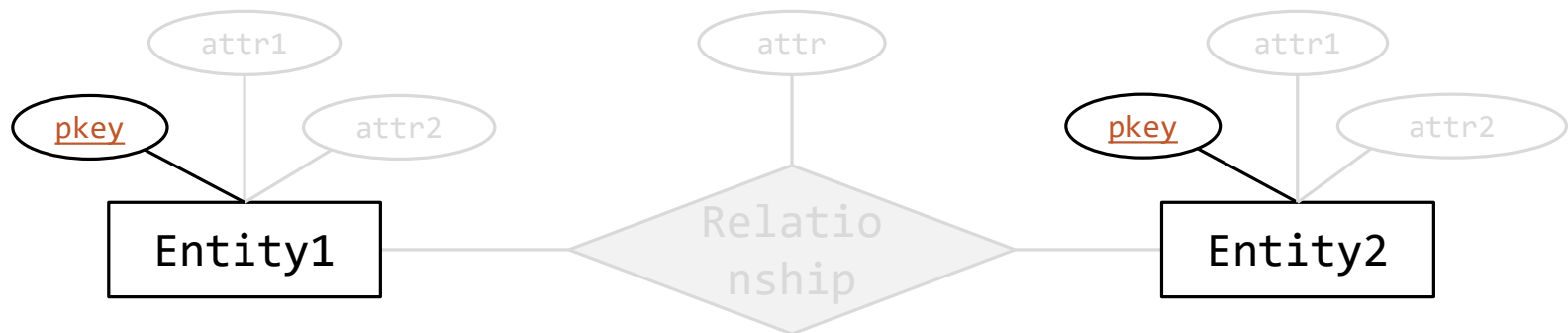
ER model

Relationships

- **Relationship**   *An association among two or more entities attributes are used to describe information about relationships*

- **Relationship set** *A collection of similar relationships*
  - *Represented by diamonds*

ER model

Entities, relationships, and attributes

I would like my customers to be able to browse my catalog of books and place orders over the Internet. Currently, I take orders over the phone. I have mostly corporate customers who call me and give me the ISBN number of a book and a quantity; they often pay by credit card. If I don't have enough copies in stock, I order additional copies and delay the shipment until the new copies arrive; I want to ship a customer's entire order together. My catalog includes all the books I sell. For each book, the catalog contains its ISBN number, title, author, purchase price, sales price, and the year the book was published. Most of my customers are regulars, and I have records with their names and addresses. New customers have to call me first and establish an account before they can use my website. On my new website, customers should first identify themselves by their unique customer identification number. Then they should be able to browse my catalog and to place orders online.
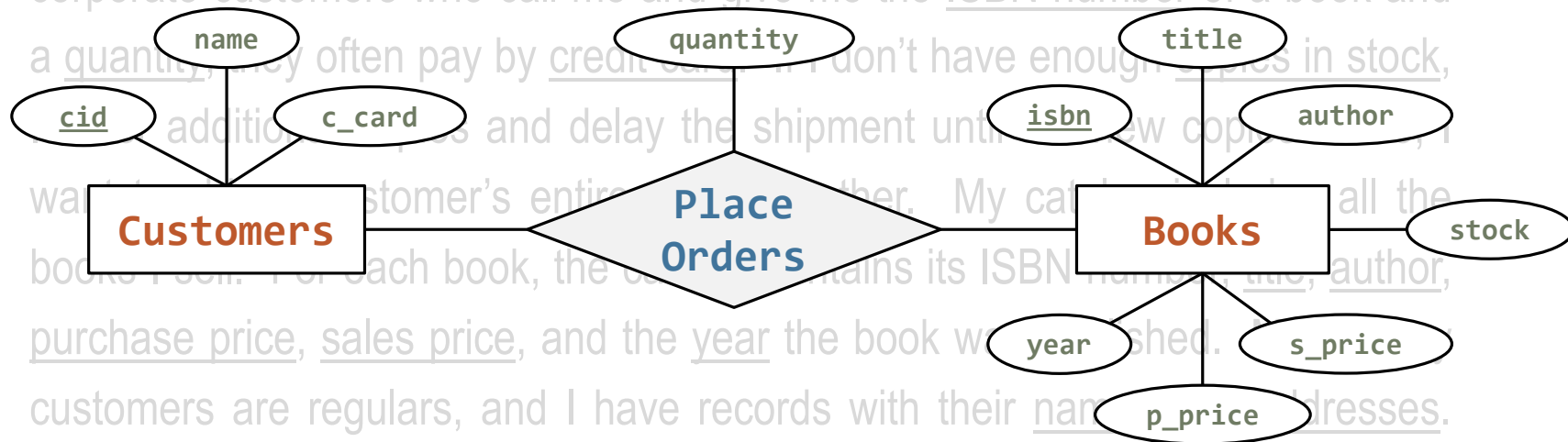
ER model

Entities, relationships, and attributes

I would like my <u>customers</u> to be able to browse my catalog of <u>books</u> and <u>place orders</u> over the Internet. Currently, I take orders over the phone. I have mostly corporate customers who call me and give me the <u>ISBN number</u> of a book and a <u>quantity</u>; they often pay by <u>credit card</u>. If I don't have enough <u>copies in stock</u>, I order additional copies and delay the shipment until the new copies arrive; I want to ship a customer's entire order together. My catalog includes all the books I sell. For each book, the catalog contains its ISBN number, <u>title</u>, <u>author</u>, <u>purchase price</u>, <u>sales price</u>, and the <u>year</u> the book was published. Most of my customers are regulars, and I have records with their <u>names</u> and <u>addresses</u>. New customers have to call me first and establish an account before they can use my website. On my new website, customers should first identify themselves by their unique <u>customer identification number</u>. Then they should be able to browse my catalog and to place orders online.

ER model

Entities, relationships, and attributes

I would like my customers to be able to browse my catalog of books and place orders over the Internet. Currently, I take orders over the phone. I have mostly corporate customers who call me and give me the ISBN number of a book and a quantity; they often pay by credit card. If I don't have enough copies in stock, I ~~addition copies~~ and delay the shipment until new copies, I ~~wa~~/~~stomer's entire~~ ~~ther~~. My cat~~~~ all the books I sell. For each book, the ~~contains~~ its ISBN number, title, author, purchase price, sales price, and the year the book w~~published~~. customers are regulars, and I have records with their na~~~~dresses. New customers have to call me first and establish an account before they can use my website. On my new website, customers should first identify themselves by their unique customer identification number. Then they should be able to browse my catalog and to place orders online.

name cid c_card

quantity

title isbn author stock

Customers — **Place Orders** — **Books**

year p_price s_price

Relationship constraints

Many-to-many relationship sets

◦ Example
- ◦ Each student can enroll in *0 or more* courses
- ◦ Each course can be enrolled by *0 or more* students
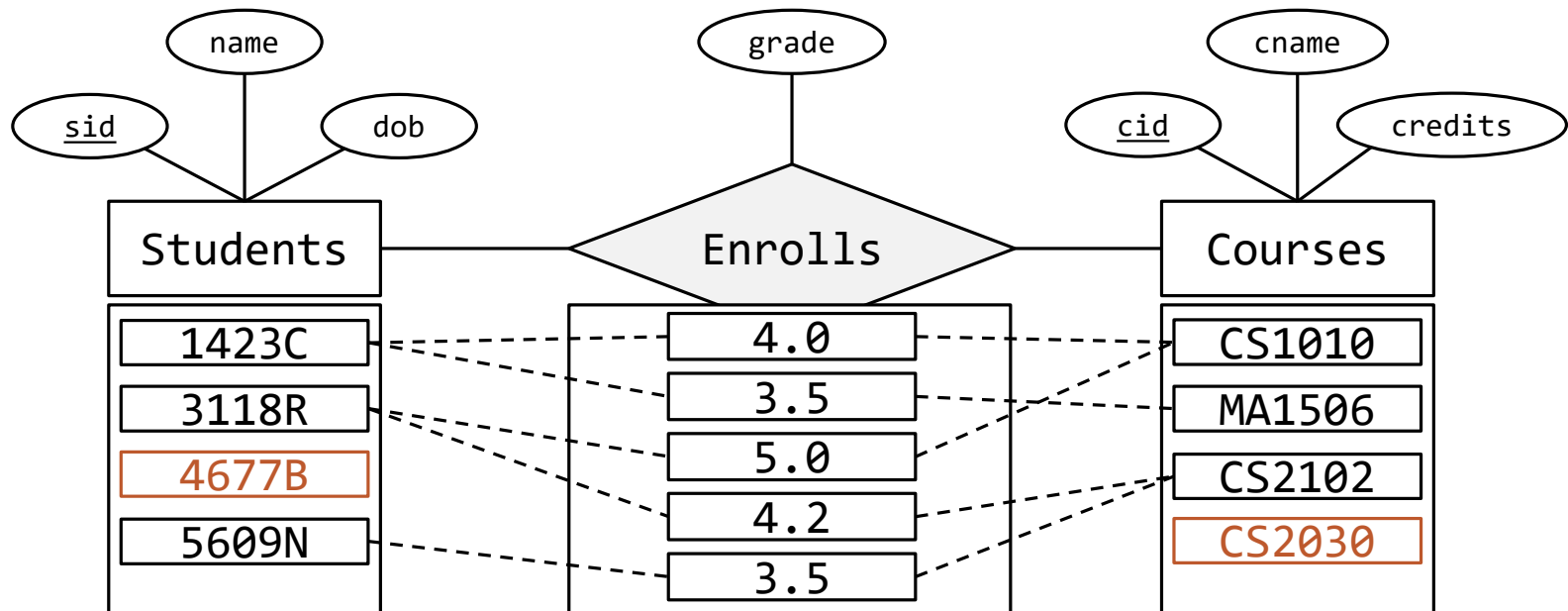
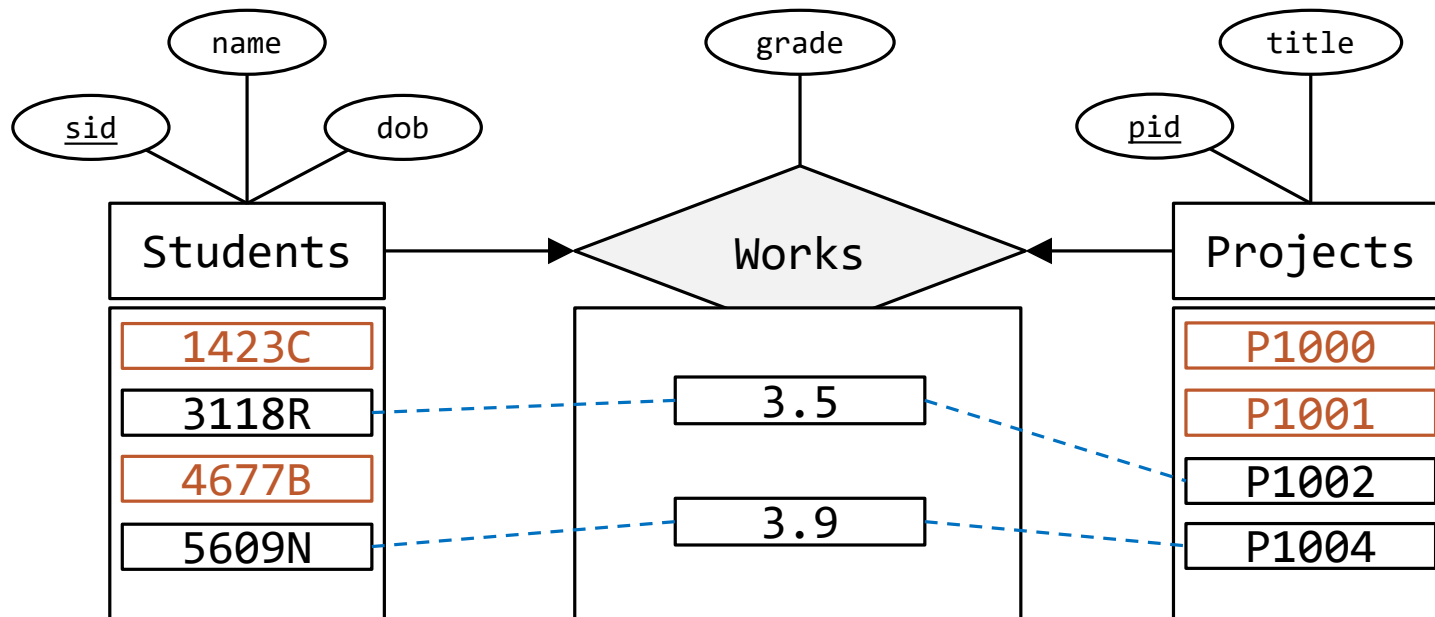# Relationship constraints

Key constraints: one-to-many relationships

◦ Let R be a relationship set that involves entity set E

◦ Key constraint on E w.r.t. R

  ◦ Each instance E can participate in *at most one* instance of R

◦ Example:

  ◦ Each professor can supervise *0 or more* students

  ◦ Each student can be supervised by *at most one* professor
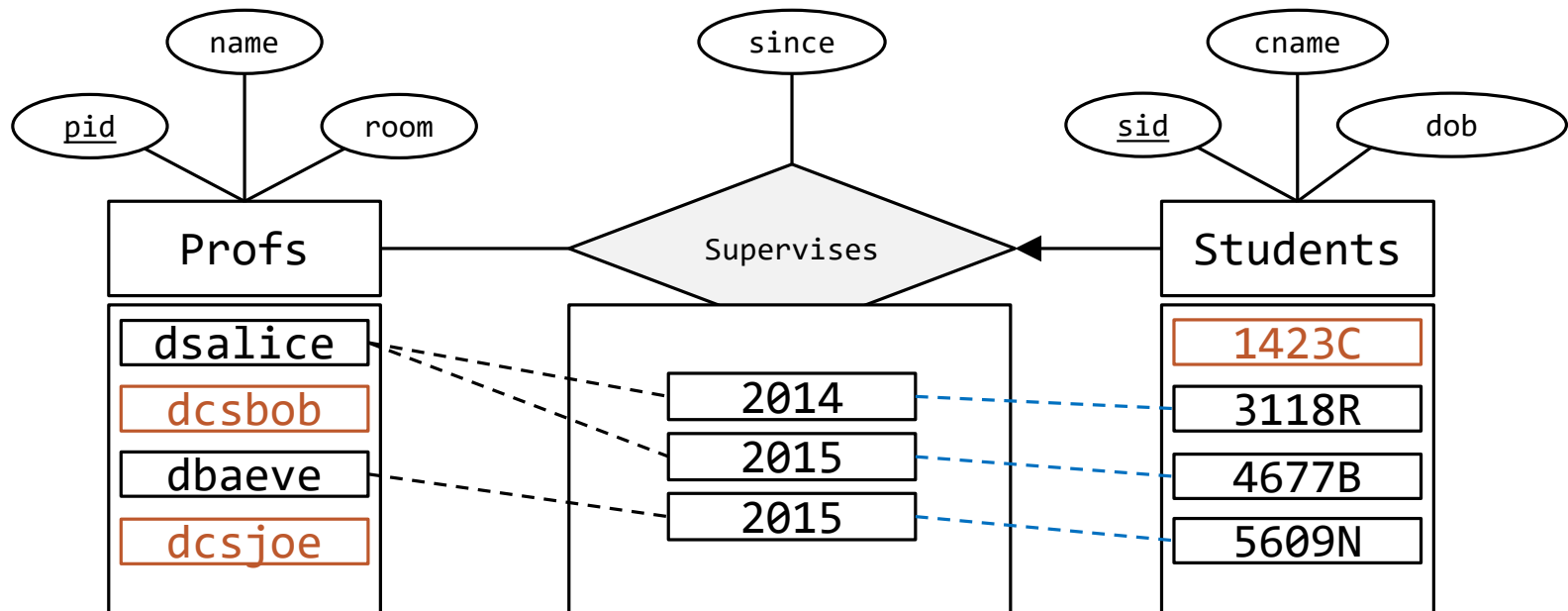
Relationship constraints

Key constraints: one-to-one relationships

- Example:
  - Each student can work on *at most one* project
  - Each project can be worked on by *at most one* student

Relationship constraints

Key constraints: N-ary relationships

◦ Example:

  ◦ Each professor can work in *at most one* department located at *some* building
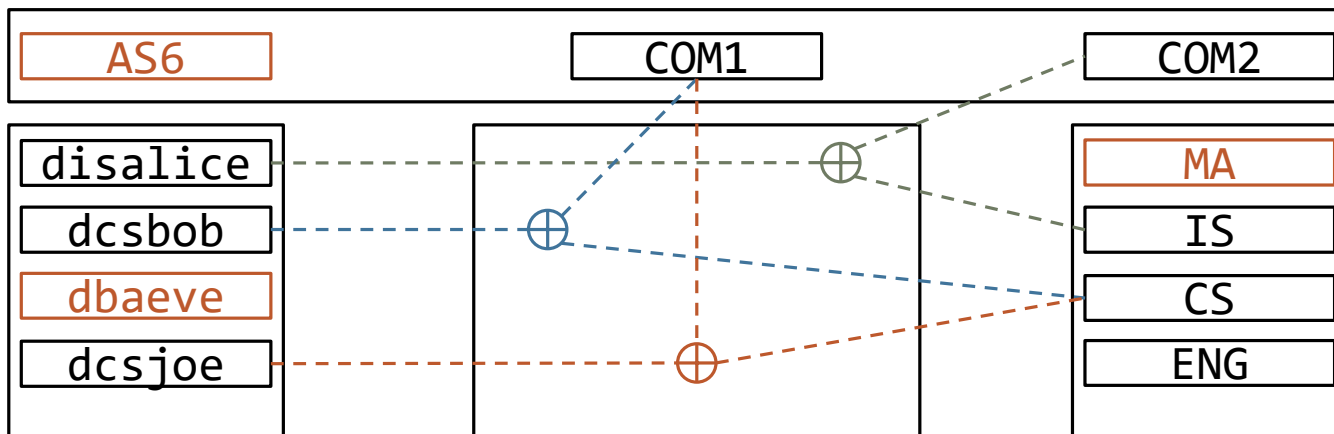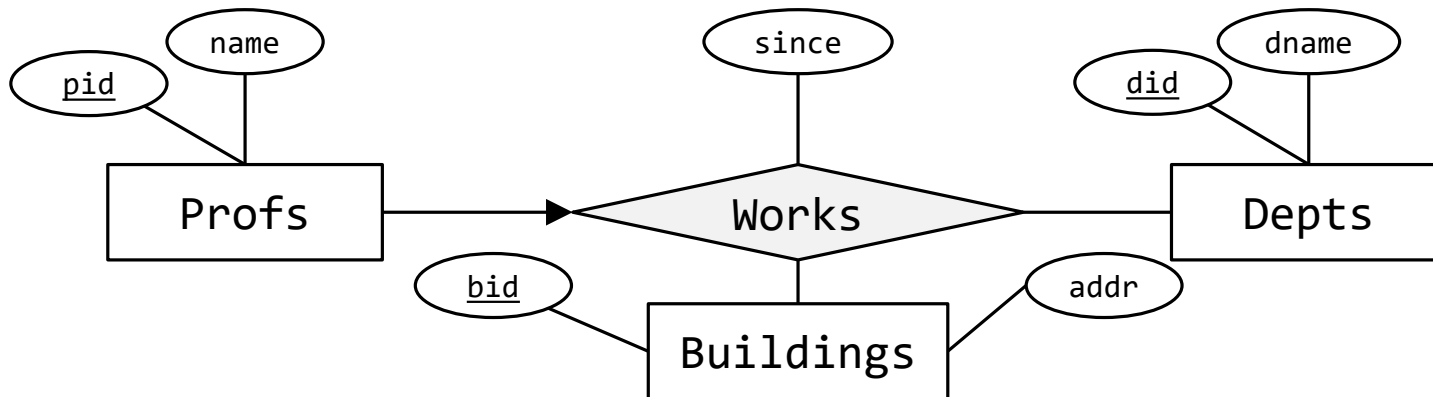
Participation constraints

Introduction

- **Participation constraints:** is the participation of an entity set in a relationship set _mandatory_?

  - **Partial participation constraint**

    - Example: each student can enroll in _0 or more_ courses
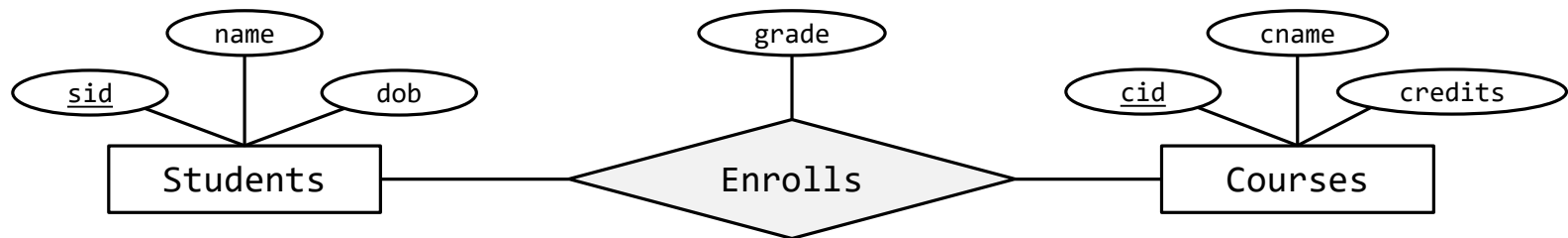
  - **Total participation constraint**

    - Example: each student must enroll in _at least one_ course
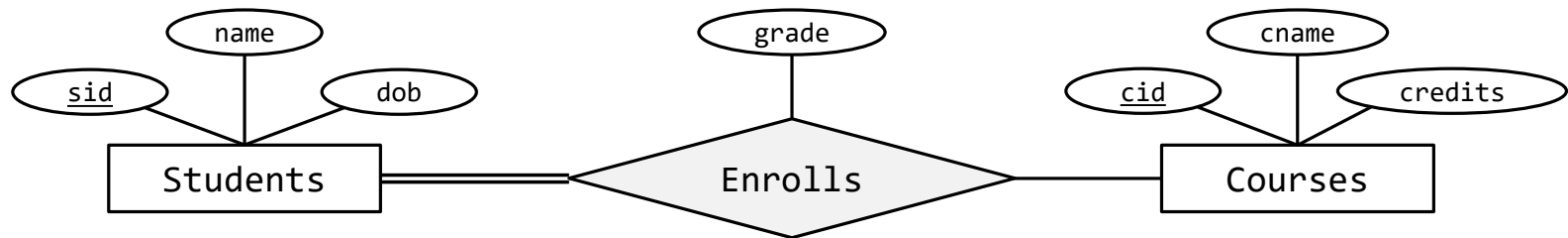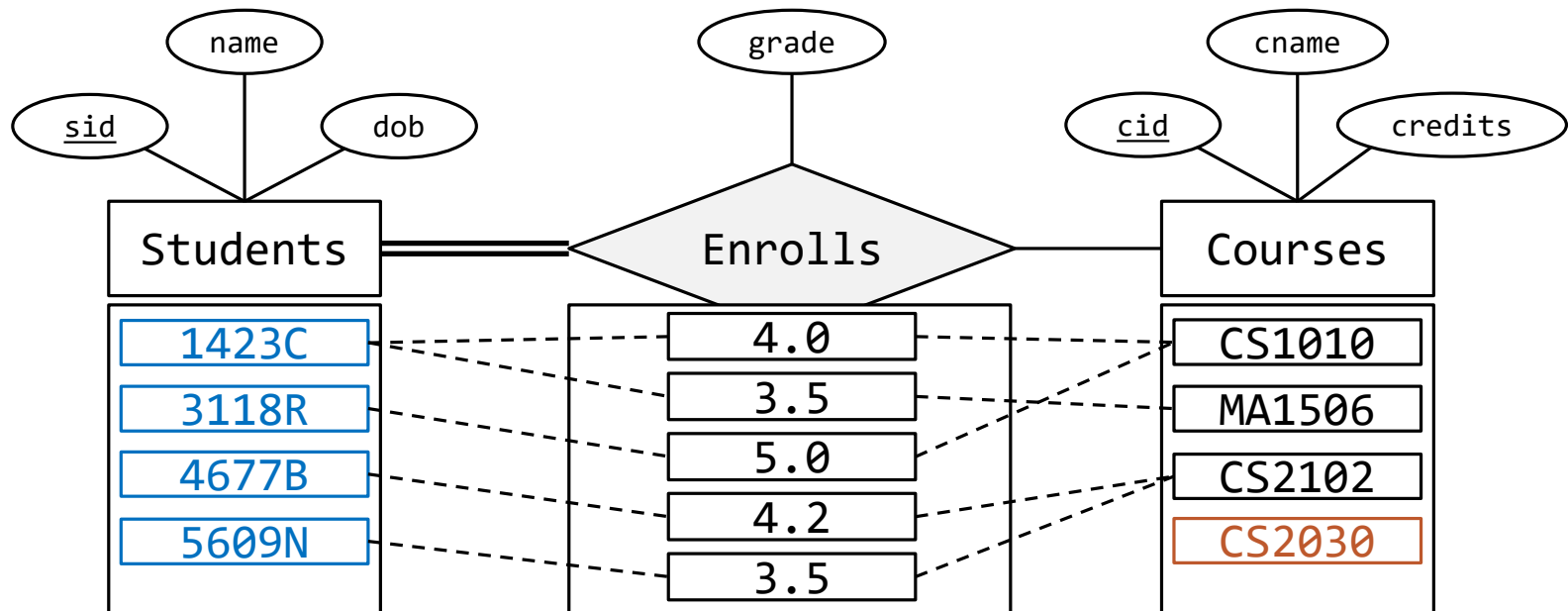
Participation constraints

Partial participation constraints

- Example
  - Each student can enroll in *0 or more* courses
  - Each course can be enrolled by *0 or more* students
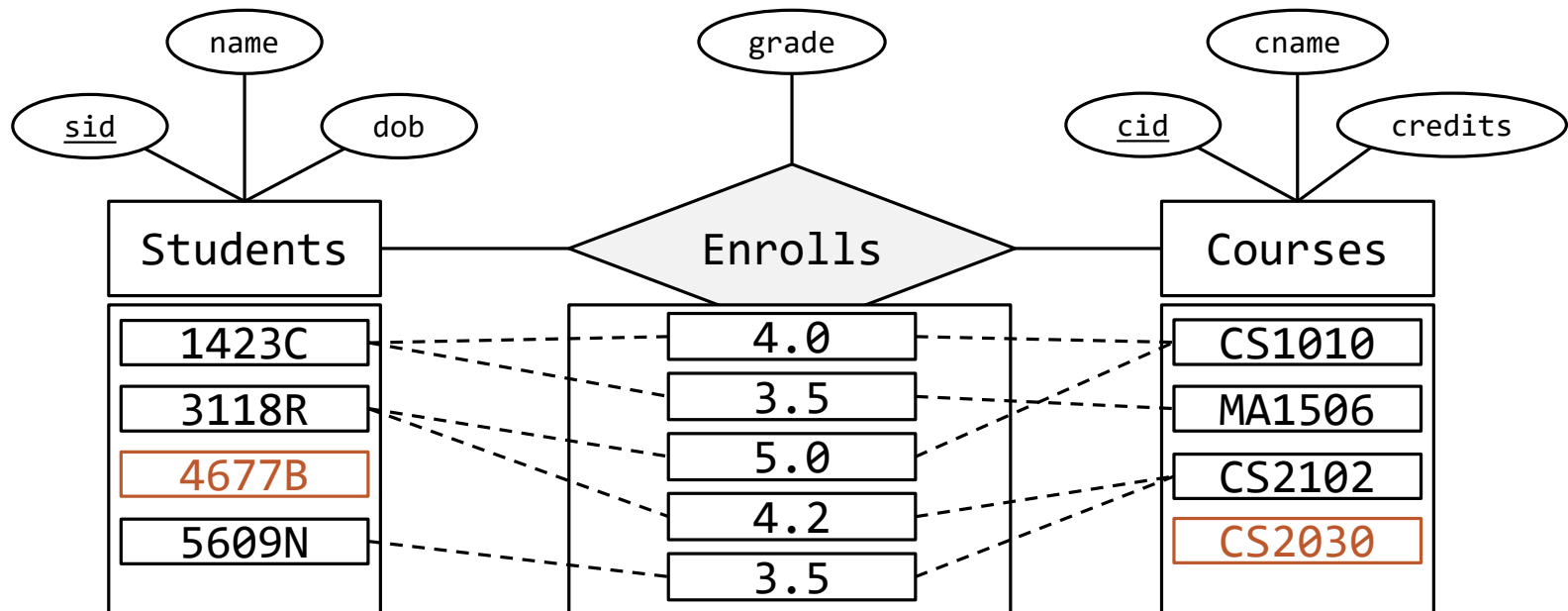
Participation constraints

Total participation constraints
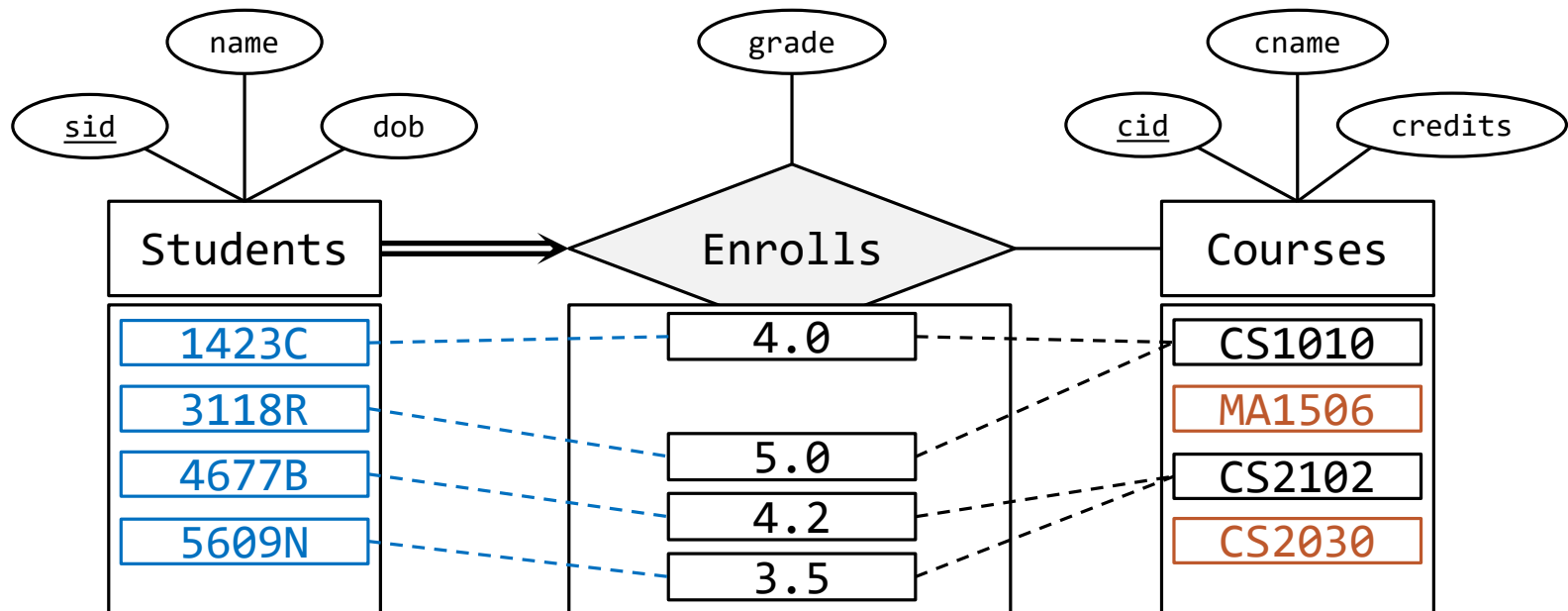
- Example
  - Each student must enroll in *at least one* courses
  - Each course can be enrolled by *0 or more* students

Participation constraints

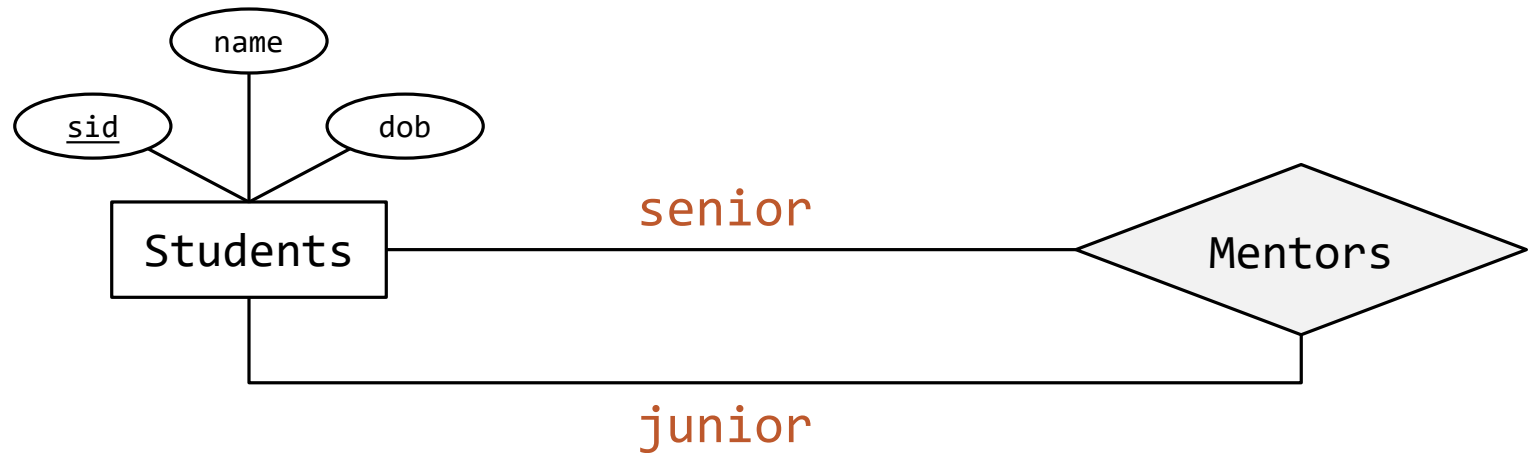## Key & Total participation constraints

- Example
  - Each student must enroll in *at most one* & *at least one* courses
  - Each course can be enrolled by *0 or more* students

Roles in relationship

Roles

- Roles are used when one entity set appears *two or more times* in a relationship set

Weak entity sets

Definition

- Weak entity set is an entity set that _does not have its own key_
- A weak entity can only be uniquely identified by considering the _primary key of another entity_ (called identifying owner)
    - There must be a _many-to-one relationship_ (called identifying relationship) from the weak entity set to an owner entity set
    - Weak entity set must have _total participation_ in identifying relationship
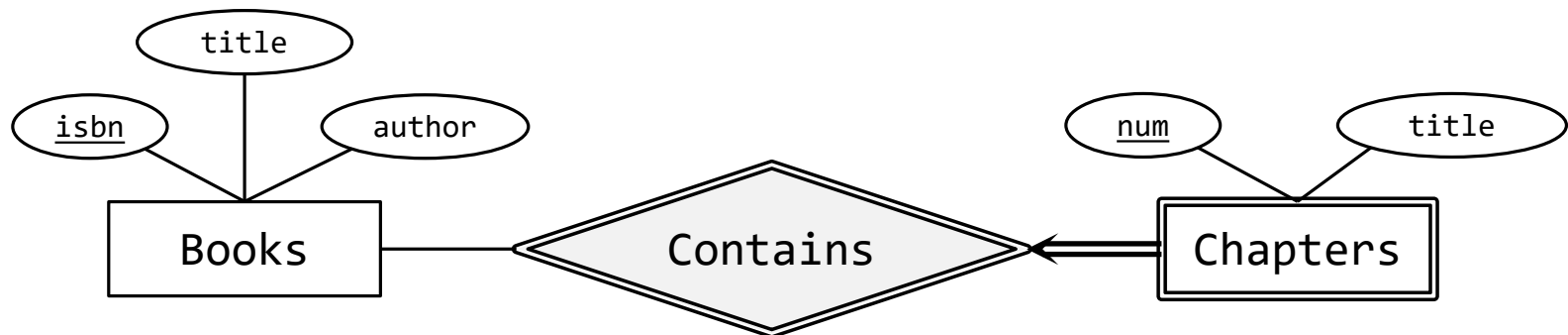
Weak entity sets

Properties

- Partial key of a weak entity set is a set of attributes of weak entity set that uniquely identifies a weak entity *for a given owner entity*

- A weak entity's existence is <u>*dependent on the existence*</u> of its owner entity

- Weak entity sets are represented by double-lined rectangles

- Identifying relationship sets are represented by double-lined diamonds
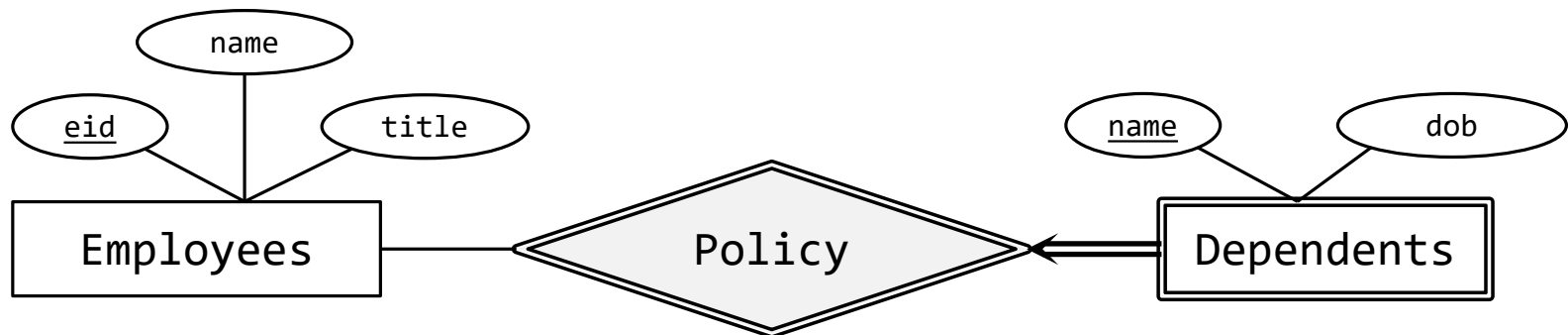
Summary

Relationship constraints

- Many-to-many

  Each instance of E participates in *0 or more* instance of R

- Key

  Each instance of E participates in *at most 1* instance of R

- Total

  Each instance of E participates in *at least 1* instance of R

- Key & total

  Each instance of E participates in *exactly one* instance of R

- Weak entity

  E is a weak entity set with identifying owner E' and identifying relationship set R