International University, VNU-HCMC

School of Computer Science and Engineering

Lecture 2: Relational Database Design Entity-Relationship

Instructor: Nguyen Thi Thuy Loan

nttloan@hcmiu.edu.vn, nthithuyloan@gmail.com
https://nttloan.wordpress.com/

Acknowledgement

Many slides are referenced from Northeastern University and Duke University.

Purpose of the Lecture

- Introduce open-source tools for designing ER diagrams
- Explain the concept of keys in ERD (primary, foreign, composite)
- Teach how to design an ERD.

Purpose of the ER Model

- Provide a clear way to design and represent database schemas
- Capture key constraints between data entities
- Use diagrams (ERDs) for easy visualization of designs
- Serve as a blueprint for conversion into relational database structures

Framework for E/R

- Database design is an essential and careful process.
- Stakeholders may know they need a database, but often are unclear about the exact requirements.
- Sketching the main components through an E/R model helps build an effective database structure.

Warm-up Question

- If you were designing a database for a university, what information (data) would you need to store, and how would you represent the relationships among students, courses, and teachers?

Outlines

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

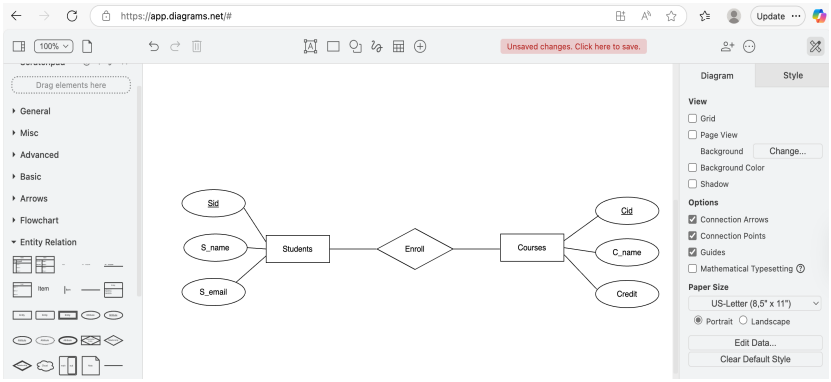
- Open-source tools
- Introduction of the ER Model
- Notation
  - Entity/ Weak entity
  - Attributes
  - Relationship
- ERD design steps

Open-source tools

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Website: Draw.io

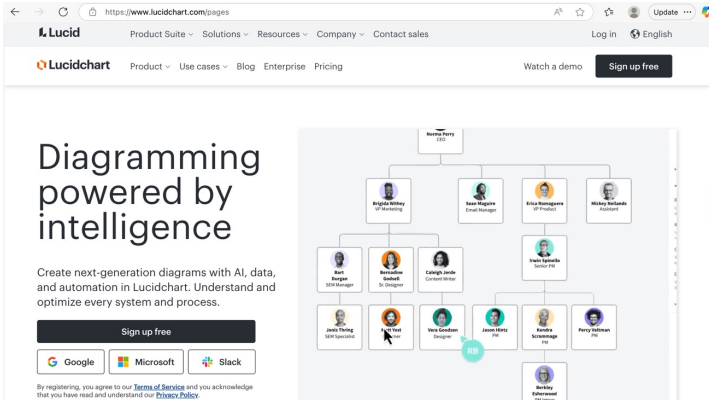
- Free, open-source tool for creating diagrams (e.g., ERDs)
- Easy to use and supports saving locally or on the cloud



Open-source tools

Website: Lucidchart

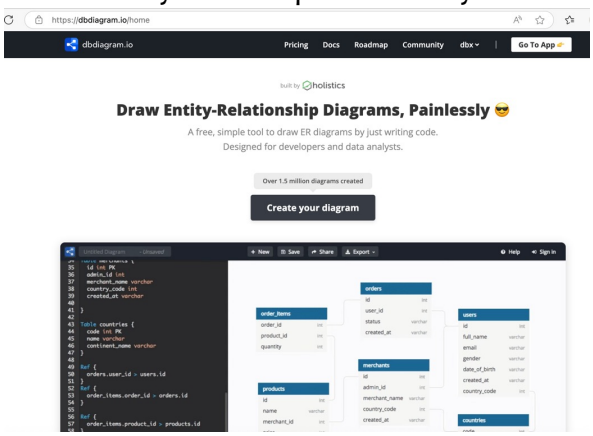
- Web-based diagramming tool with a free, limited version
- Supports ERDs and many other diagram types
- Enables real-time collaboration for teams



Open-source tools

Website: dbdiagram.io

- Online tool focused on database diagrams (ERDs)
- Define structures easily with simple textual syntax

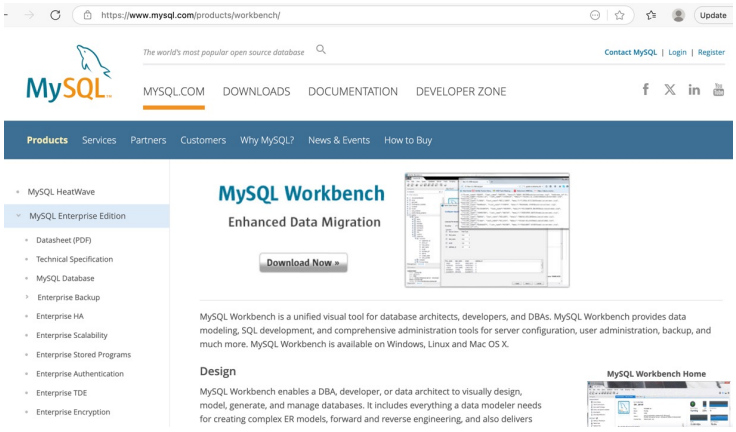


Open-source tools

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Website: MySQL Workbench

- IDE for MySQL with built-in ERD design tool
- Best suited for working directly with MySQL databases

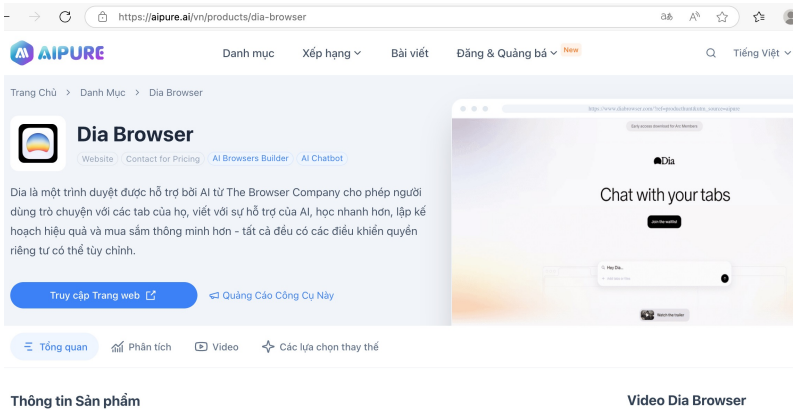


Open-source tools

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Website: Dia

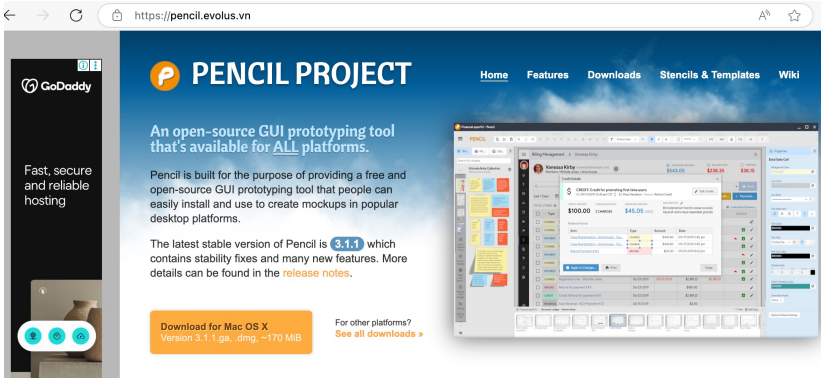
- Open-source diagramming software for Linux, Windows, macOS
- Supports ERDs with many shapes and symbols



Open-source tools

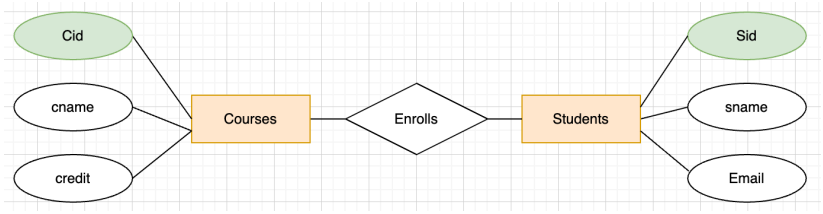
Website: Pencil Project

- Open-source GUI prototyping tool with diagram support (ERDs included)
- Available as a standalone app or a Firefox add-on



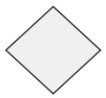
Introduction of the ER Model

- The Entity-Relationship Model (ER Model) is a conceptual model for designing a database.

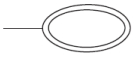
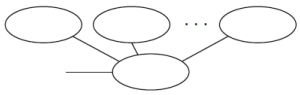
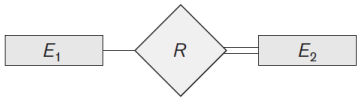
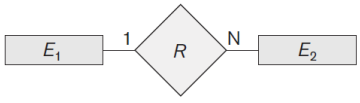
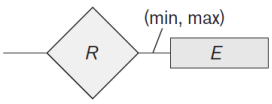


- This model represents the logical structure of a database, including entities, their attributes, and relationships between them.

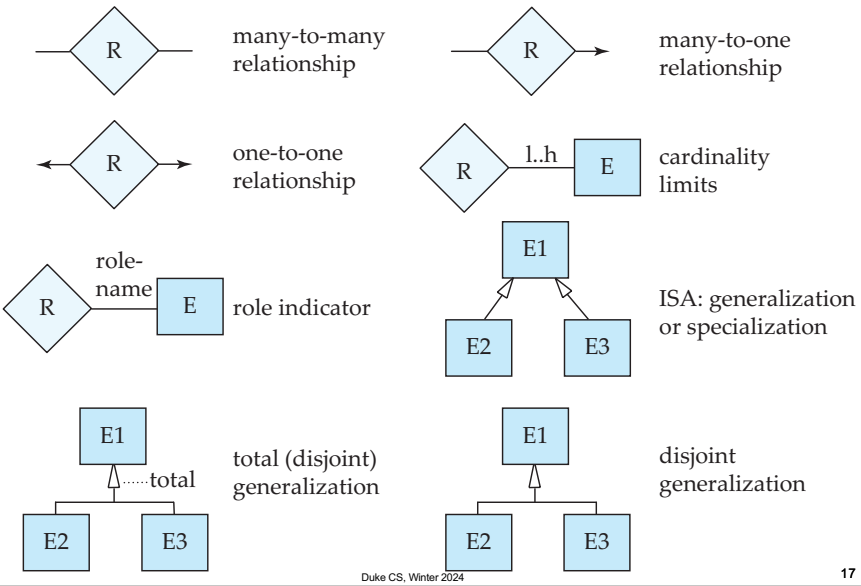
The notation for ER diagrams

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Indentifying Relationship
	Attribute
	Key Attribute

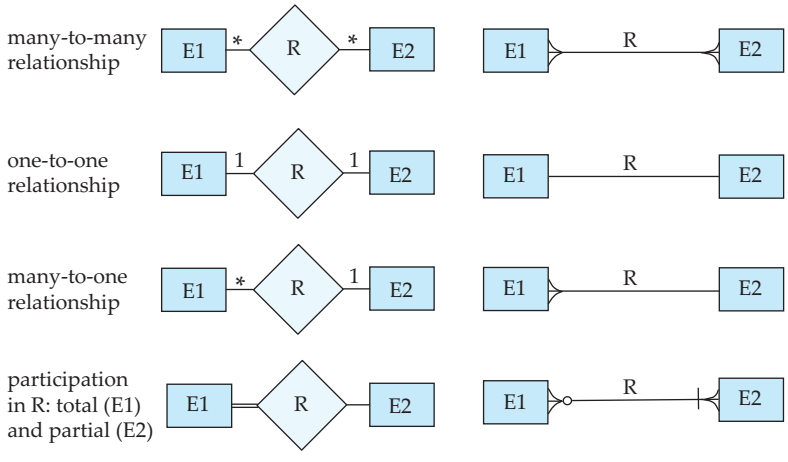
The notation for ER diagrams

	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of E_2 in R
	Cardinality Ratio 1: N for E_1 : E_2 in R
	Structural Constraint (min, max) on Participation of E in R

Summary of the notation for ER diagrams



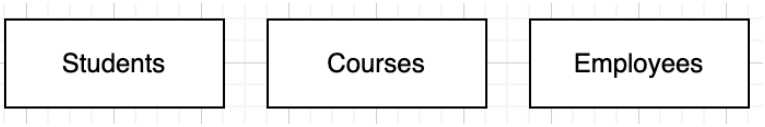
Alternative ER Notations



Entity – Strong Entity

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- An **entity** is an object or thing in the real world that can be distinctly identified.

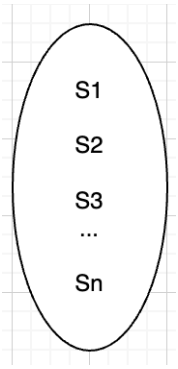


- Represented as a **rectangle** in ERD.
- Example: *Students*, *Courses*, *Employees*.

Entity (set) – Strong Entity

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

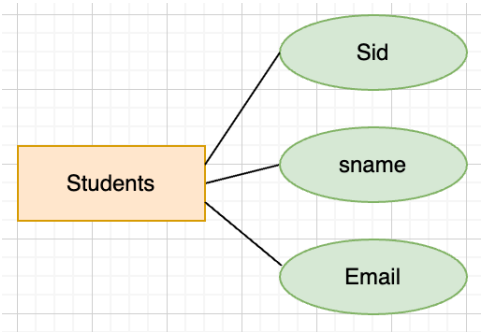
- An **entity set** is a collection of similar entities.



- Example: *Student1*, *Student2*, *student3*…

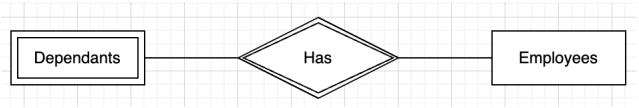
Example

- The entity set Students has three attributes, Sid, sname, and Email.
- Each student entity has values for these three attributes, e.g. (S01, Le Thi A, lethiA@gmail.com)



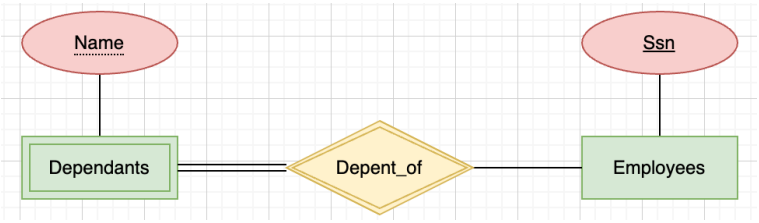
Weak Entity (set)

- A weak entity cannot be uniquely identified by its attributes alone. It depends on a strong entity to be identified.
- Always linked to a strong entity for recognition.

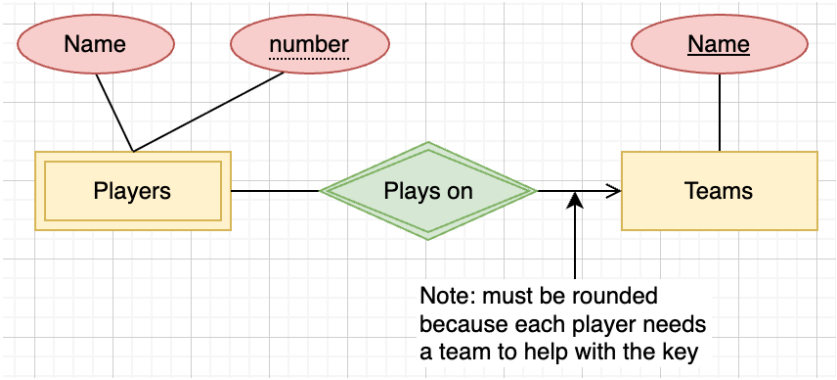


Example

- Employee (Ssn): uniquely identifies each employee.
- Dependent (Name): not unique, since two dependents may share the same name.
- Combination of Dependent's Name + Employee's Ssn (via *Depent_of*) → uniquely identifies each dependent.



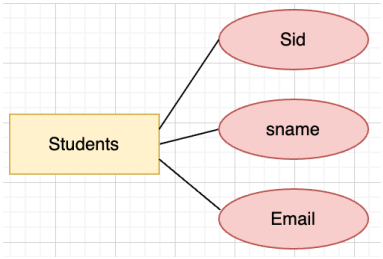
Example



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

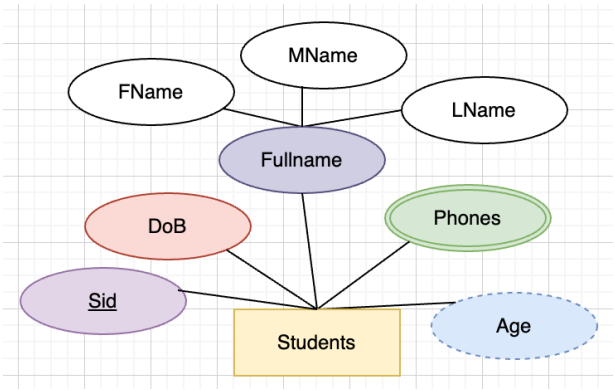
Attributes - details

- **Attributes** are properties or characteristics that describe an entity or a relationship, represented as **ellipses (ovals)** in ERD.
- Example: *Student* entity may have attributes: *Sid, sname, Email*.



Types of attributes

- Entity set Students with simple attribute DoB, composite attribute Fullname, multivalued attribute Phones, derived attribute Age, and primary key Sid



Keys

Here are the main types of keys in ERD:

- Primary Key (PK)
- Candidate Key
- Foreign Key (FK)
- Composite Key

ERD Keys

- In ERD: A key is an attribute (or a set of attributes) that uniquely identifies an entity within an entity set, and is minimal.
- Example: Students (Sid, Sname, Email, Phone, ID-card)
  - Sid is a key
  - Email is not a key (not an identifier)
  - {sid, Sname} is not a key (not minimal)

Keys in ER Diagrams

- Underline the key attribute(s).
- In an *isa* hierarchy, only the root entity set has a key, and it must serve as the key for all entities in the hierarchy.

Primary Key (PK) - Candidate Key

- Primary Key (PK): A unique identifier for each entity instance.
  Example: *Sid* in a *Student* entity.
- Candidate Key:
  An attribute (or group of attributes) that could serve as a primary key because it uniquely identifies an entity.
  Example: In *a Student, an ID-card could be a candidate key.*

Foreign Key (FK) - Composite Key

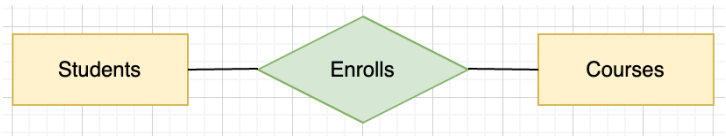
Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Students(Sid, sname, email, phone, ID-card)
- Courses(Cid, cname, credit)
- Enrolls(Sid, Cid, grade)

Relationship

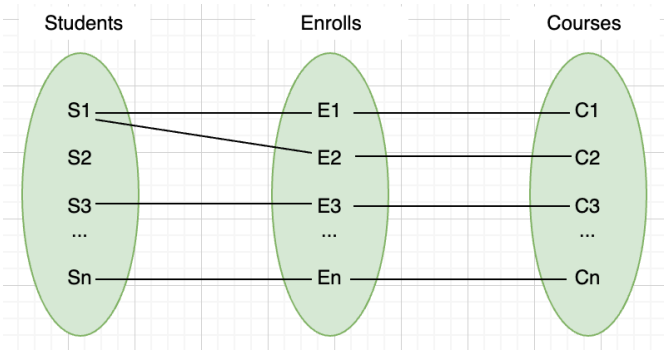
Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- A **relationship** describes how two or more entities are connected.
- Represented as a **diamond** in ERD.
- Example: *Enrolls* (relationship between *Students* and *Courses*).



Relationship set

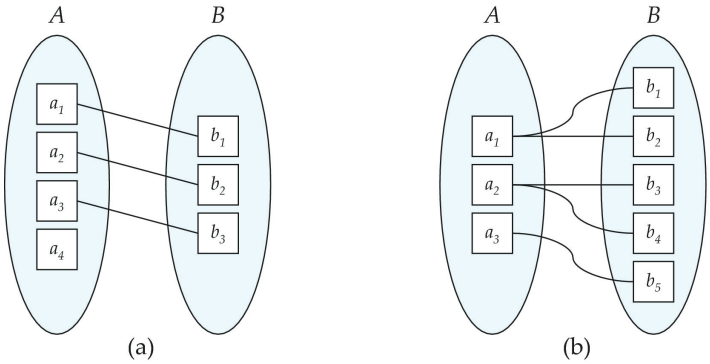
- A set of relationships of the same type is known as a relationship set.



Mapping Cardinality (Binary Relationships)

- Specifies **how many instances** of one entity can be linked to another in a relationship.
- Most relevant for **binary** relationship sets.
- Allowed types:
  - **1:1** – each A relates to at most one B, and vice versa
  - **1:N** – one A relates to many B; each B to one A
  - **N:1** – many A relate to one B
  - **M:N** – many A relate to many B

Mapping Cardinalities

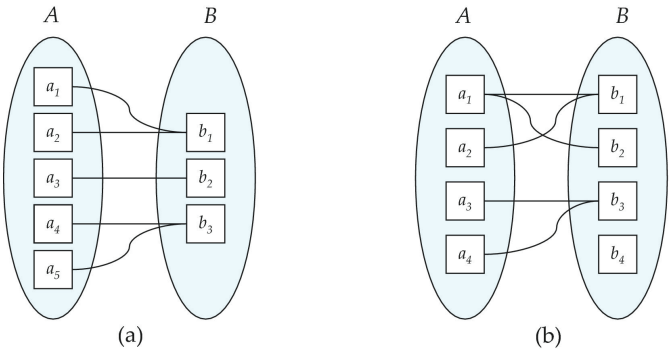


One to one

One to many

Note: Some elements in *A* and *B* may not be mapped to any elements in the other set

Mapping Cardinalities



Many to one

Many to many

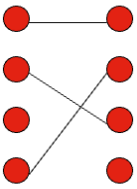
Note: Some elements in A and B may not be mapped to any elements in the other set

One-to-One Relationships

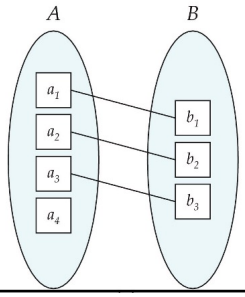
- Definition**:** Each entity in set A is related to **at most one** entity in set B**, and vice versa**.

Example – BestSeller(Manf, Beer):

- Each **manufacturer** has **at most one** best-selling beer.
- A **beer** can be the best-seller of **at most one** manufacturer (assume no ties).

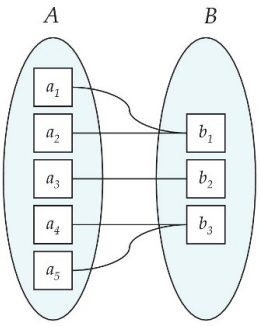


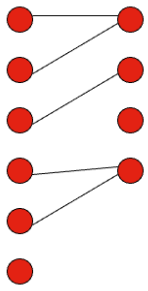
one-one



Many-to-One Relationships

- Definition**:** Each entity in **A** relates to **at most one** entity in **B**; an entity in **B** may relate to **zero, one, or many** in **A**.
- Equivalent view: **Many-to-one** from **B** to **A**.



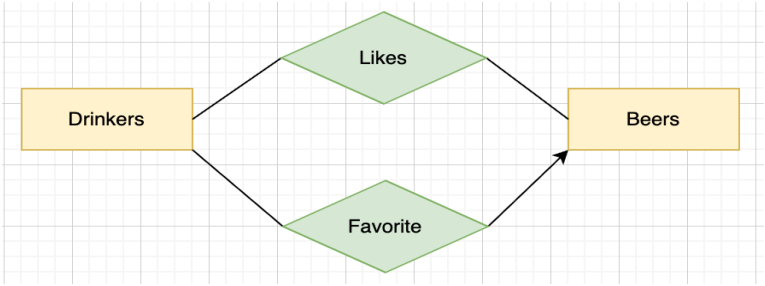


many-one

Example: Many-One Relationship

- Make, from **Manufacturer** to **Beers** is many-to-one.
- Example: Each **Beer** is made by **one Manufacturer;** a **Manufacturer** makes **many Beers.**

Example: Many-One Relationship

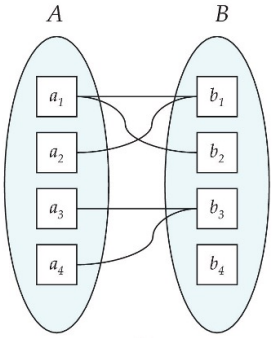


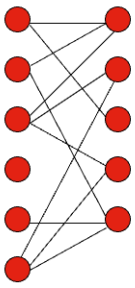
Notice: Although both relationships connect the same entity sets (*Drinkers* and *Beers*), they represent different meanings.

- *Likes:* a drinker may like many beers, and a beer may be liked by many drinkers.
- *Favorite:* each drinker has at most one favorite beer, but a beer can be the favorite of many drinkers.

Many-to-Many Relationships

- **Focus:** binary relationships where **both sides** can have multiple matches.
- **Meaning:** one A **many** B, and one B **many** A.





many-many

Example

- **Sells(Bar, Beer)**, a bar sells many beers; a beer is sold by many bars.
- Note: M:N relationships often carry attributes (e.g., price, grade) and are conveniently modeled as an associative entity when needed.

Representing "Multiplicity"

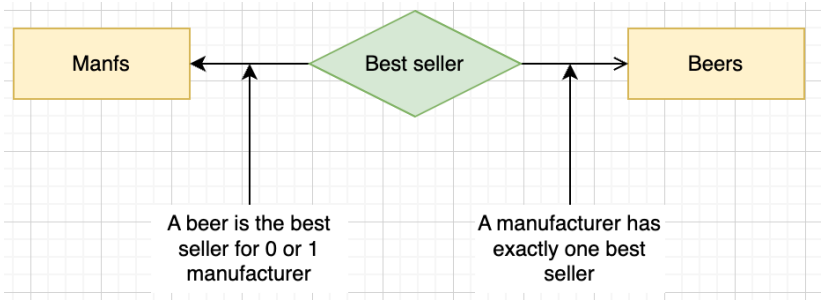
- An arrow **entering an entity** indicates **"at most one"** of that entity per matching entity on the other side (like *functional dependency*).

Representing "Multiplicity"

- No arrows: **M:N** (many-to-many).
- One straight arrow into B: **N:1** from A to B.
  Each A relates to at most one B; a B can relate to many A.
- Arrows into both A and B: **1:1**.
  Each A is to at most one B, and each B is to at most one A.
- Rounded arrow into B: **exactly one** B per A (total participation on that side).
  Straight arrow: ≤ 1 (optional); rounded arrow: =1 (mandatory).

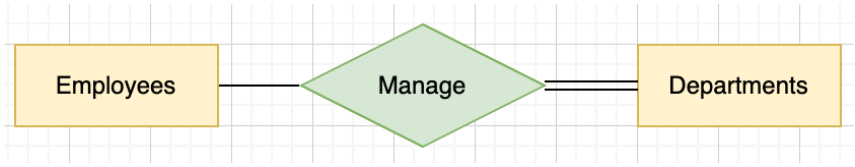
Example

- Beer → Manf (arrow into **Manf**)
- Beer →Manf (rounded arrow)
- Arrows into both **Passport** ↔ **Person: 1:1.**



Participation Constraints

- Total Participation/ Partial participation



Relationship Set

- Entity set value: the list of all entities in that set.

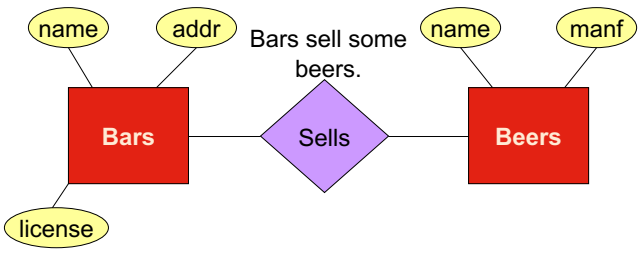
Example: all the *bars* stored in the database

- Relationship set value: the list of all current connections (associations) between entities.
- Formally: a set of tuples, one element from each entity set.

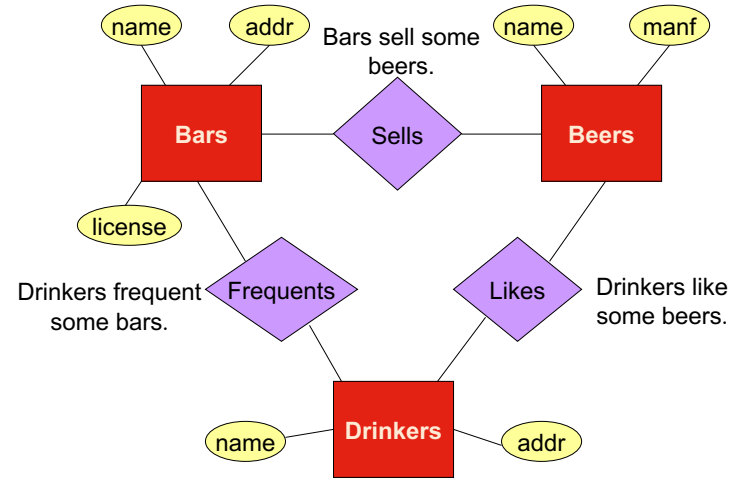
Example: Serves(Bar, Beer) = { (Bar_i, $Beer_j$), … }.

- Think of it this way:
- Entities = rows in a table.
- Relationship set = rows that show how entities are linked.

Example: Relationships



Example: Relationships



Note: license = beer, full, none

Multiway (N-ary) Relationships

- Some constraints require a relationship among **two or more entity sets.**

Example:

- A **Drinker** only drinks a specific **Beer** at a specific **Bar.**
- Binary links Likes(Drinker, Beer), Sells(Bar, Beer), and Frequents(Drinker, Bar) **can't express** this joint condition.

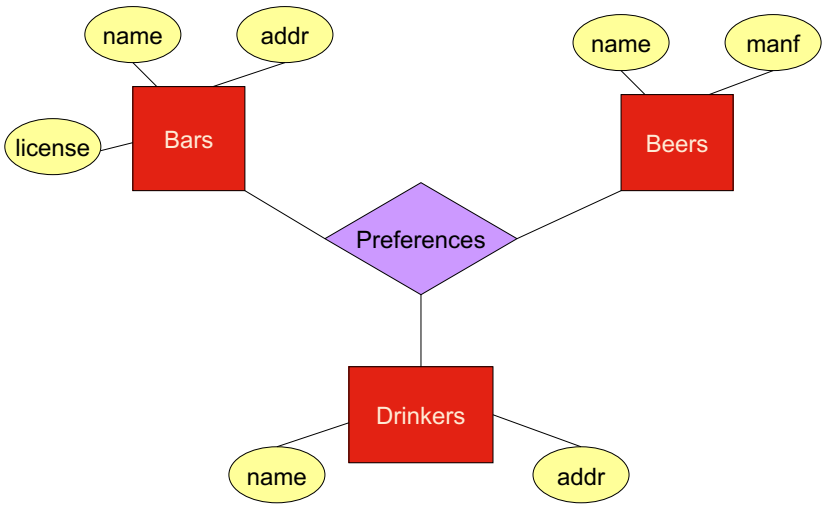
Multiway (N-ary) Relationships

- Use a **ternary** relationship:

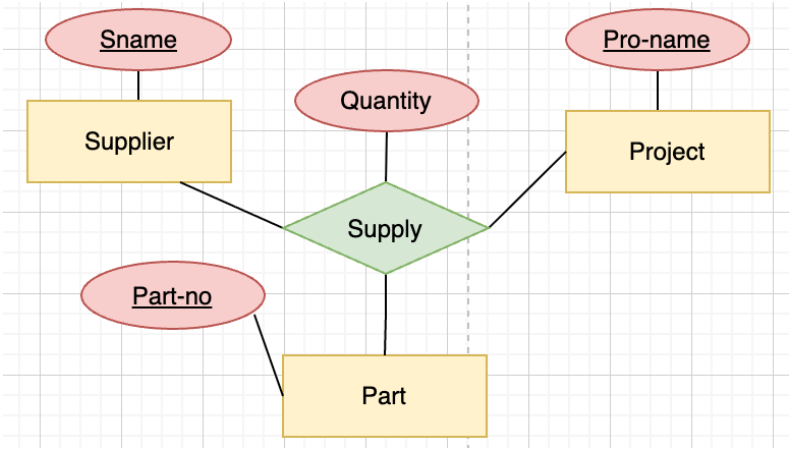
Drinks(Drinker, Beer, Bar)

- Key idea: N-ary relationships capture facts that can't be decomposed into independent binary ones without losing meaning.

Example: 3-Way Relationship



Ex. Ternary relationship type



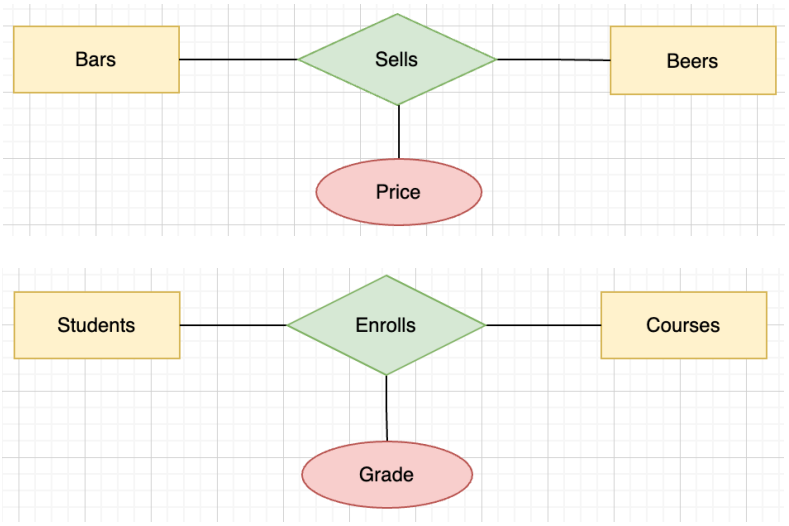
Attributes on Relationships

- Sometimes the **relationship itself** has properties.
- Treat these as attributes of **each relationship tuple/instance**, not of the entities.

Example:

- Sells(Bar, Beer) may have **Price;**
- Enrolls(Student, Course) may have **EnrollDate/Grade.**

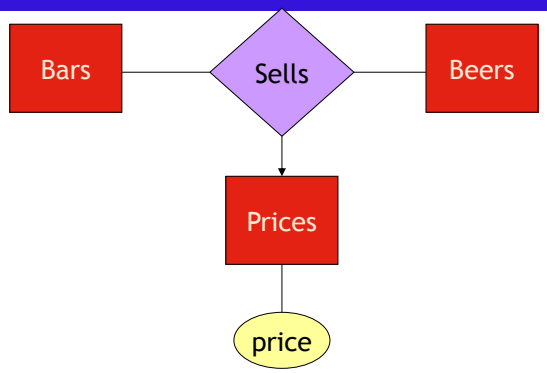
Example: Attribute on Relationship



Without Attributes on Relationships

- Create an entity set that represents the values of the attribute.
- Make that entity set participate in the relationship.

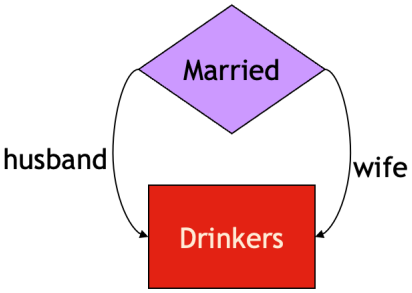
Ex. Removing an Attribute from a Relationship



- Move the attribute **price** out of the relationship **Sells** into a new entity **Prices**.
- The **arrow** shows: each *(Bar, Beer)* pair has **exactly one Price**.
- This makes the design **clearer** and avoids treating price as just a simple attribute.

Roles

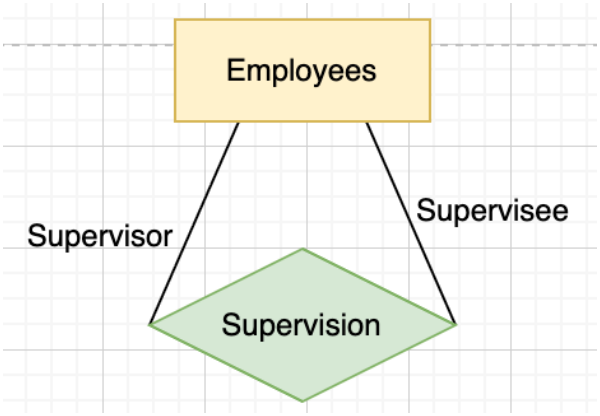
- Sometimes an entity set appears **multiple times** in the same relationship.
- To avoid confusion, we give each connection a **role name** (label on the edge).



Relationship Set

Husband	wife
Bob	Ann
Joe	Sue
…	…

Example: Roles



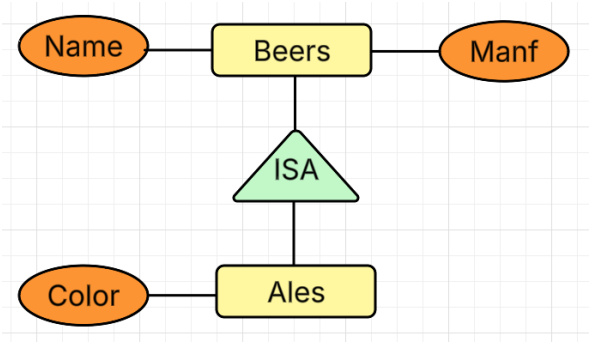
Subclasses (Specialization)

- A **subclass** is a more specific type of a supertype → **fewer instances, more properties**.
- Subclasses **inherit** all attributes/relationships of the supertype and may **add their own**.

Subclasses (Specialization)

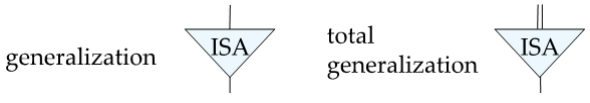
Example: *Ale* ⊆ *Beer*

- Not every beer is an ale (**partial** classification).
- *Ale* adds the attribute **Color** in addition to all Beer properties.



Subclasses in ER Diagrams

- Model subclasses as a **hierarchy/tree, with no multiple inheritance**.
- Use an **ISA triangle** to show specialization/ generalization.
- **Subclasses connect to the triangle; the triangle points to the superclass.**
- (Optional) Annotate the ISA with **disjoint/overlap and total/partial as** needed.

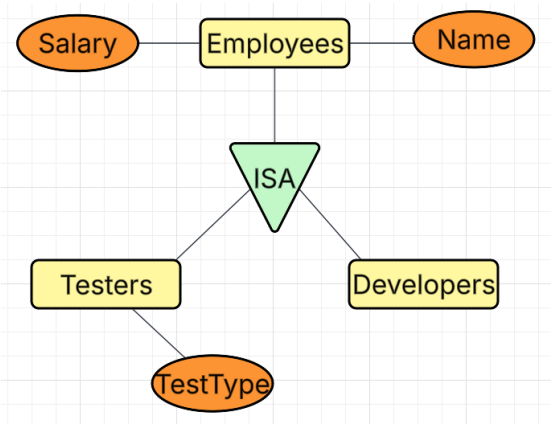


Specialization/Generalization

- **Idea:** Only a subset of a supertype needs extra attributes/roles.
- **Specialization (top-down):** Supertype → subtypes; subtypes **inherit** all properties and may add their own.
- **Generalization (bottom-up):** Merge similar subtypes into a common supertype with shared properties.
- **Constraints:** mark **total vs. partial** participation and **disjoint vs. overlap** among subtypes.

Specialization/Generalization

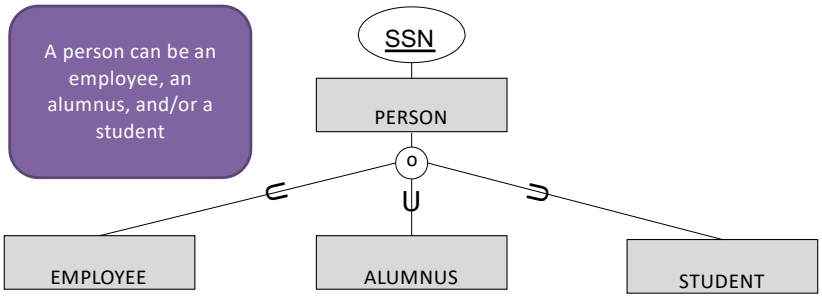
Example: ISA where EMPLOYEES specializes in TESTERS and DEVELOPERS.



Multiple Subtypes: Disjointedness

(o)verlap: may be more than one

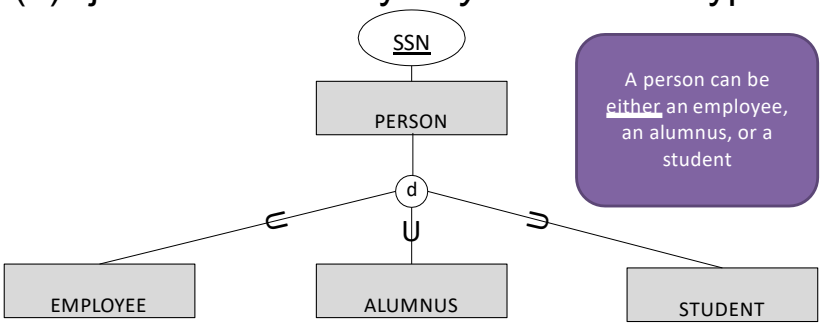
(d)isjoint: entities may *only be one* subtype



Multiple Subtypes: Disjointedness

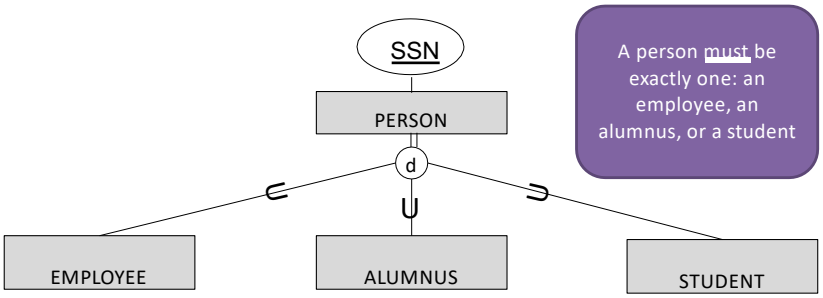
(o)verlap: may be more than one

(d)isjoint: entities may *only be one* subtype



Multiple Subtypes: Completeness

Similar to relationships; can be total (must belong to subtypes) or partial (can belong)

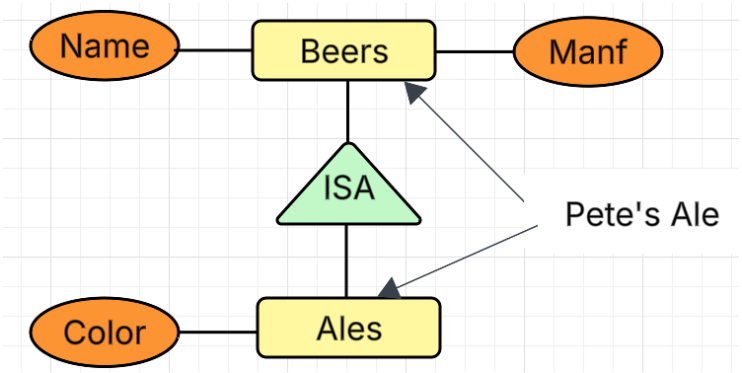


ER Vs. Object-Oriented Subclasses

- In OO, objects are in one class only.
  - Subclasses inherit from superclasses.
- In contrast, E/R entities have *representatives* in all subclasses to which they belong.
  - Rule: if entity *e* is represented in a subclass, then *e* is represented in the superclass (and recursively up the tree).

Example: Representatives of Entities

Assoc. Prof. Nguyen Thi Thuy Loan, PhD



ERD design steps

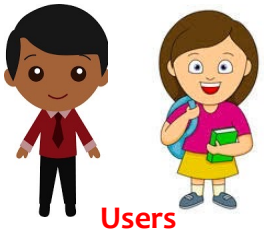
Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Define scope & rules
- Identify entities (include weak entities)
- List attributes & pick key(s)
- Identify relationships & their attributes
- Set cardinality & participation (1–1, 1–N, M–N; total/partial)
- Model hierarchies (ISA: total/partial, disjoint/overlap)

Design ERD

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Example: Users, Groups, Members



Users

Each has uid (unique id), name, age, pop (popularity)

Groups

Each has gid (unique id), name

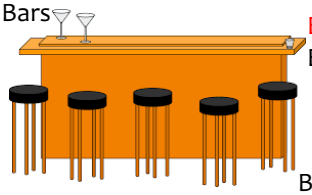
Member

Records fromDate (when a user joined a group)

The famous "Beers" database

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Drinkers **Frequent** Bars "X" times a week



Bars
Each has an address

Bars **Serve** Beers At price "Y"

Drinkers
Each has an address

Drinkers **Likes** Beers

Beers
Each has a brewer

(Later in ER diagram – how to design a relational database)

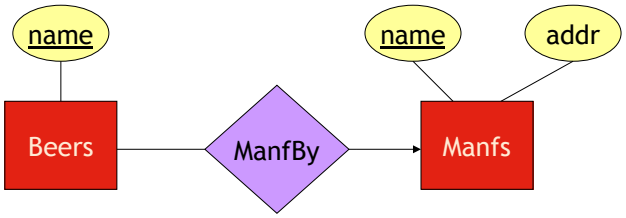
Design Techniques

1. Avoid redundancy.
2. Limit the use of weak entity sets.
3. Don't use an entity set when an attribute will do.

Avoiding Redundancy

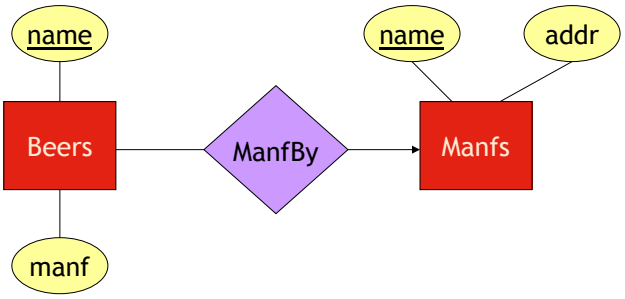
- *Redundancy* = saying the same thing in two (or more) different ways.
- Wastes space and (more importantly) encourages inconsistency.
  - Two representations of the same fact become inconsistent if we change one and forget to change the other.
  - Recall anomalies due to FD's.

Example: Good



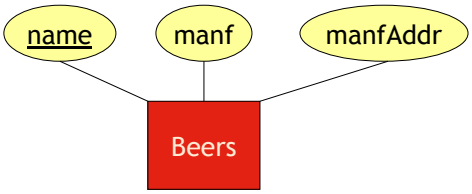
This design gives the address of each manufacturer exactly once.

Example: Bad



This design states the manufacturer of a beer twice: as an attribute and as a related entity.

Example: Bad



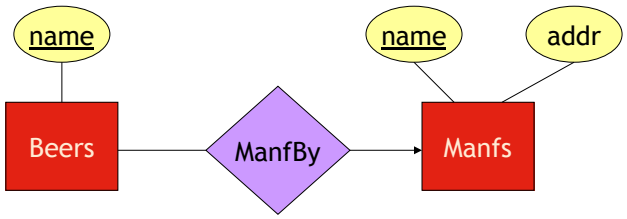
This design repeats the manufacturer's address once for each beer and loses the address if there are temporarily no beers for a manufacturer.

Entity Sets Versus Attributes

- An entity set should satisfy at least one of the following conditions:
  - It is more than the name of something; it has at least one non-key attribute.

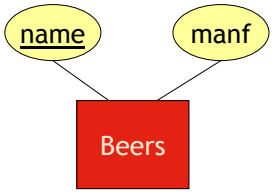
    or
  - It is the "many" in a many-one or many-many relationship.

Example: Good



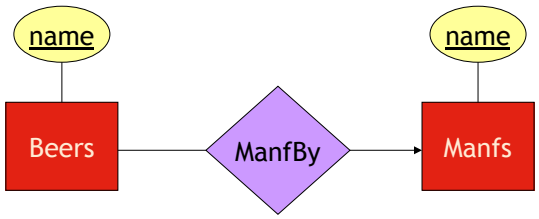
- Manfs deserves to be an entity set because of the non-key attribute addr.
- Beers deserves to be an entity set because it is the "many" of the many-one relationship ManfBy.

Example: Good



There is no need to make the manufacturer an entity set, because we record nothing about manufacturers besides their name.

Example: Bad



Since the manufacturer is nothing but a name, and is not at the "many" end of any relationship, it should not be an entity set.

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 license's etc.

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.

Exercise

Draw an ERD for the following description:

You are the Editor-in-Chief of a journal. The database stores information about papers submitted to the journal (Papers). Each paper has a unique ID (PID), a title, and a status (with editor, under review, accepted, or rejected).

After submission, each paper is assigned to members of the program committee (PC) as reviewers. Each reviewer has a unique ID (RID), a full name (including first, middle, and last names), one email address, and one or more affiliations.

Each reviewer may review zero or more papers. Each paper must be assigned to at least two reviewers.

Exercise

Draw an ERD for the following description:

You are the Editor-in-Chief of a journal. The database stores information about papers submitted to the journal (Papers). Each paper has a unique ID (PID), a title, and a status (with editor, under review, accepted, or rejected).

After submission, each paper is assigned to members of the program committee (PC) as reviewers. Each reviewer has a unique ID (RID), a full name (including first, middle, and last names), one email address, and one or more affiliations.

Each reviewer may review zero or more papers. Each paper must be assigned to at least two reviewers.

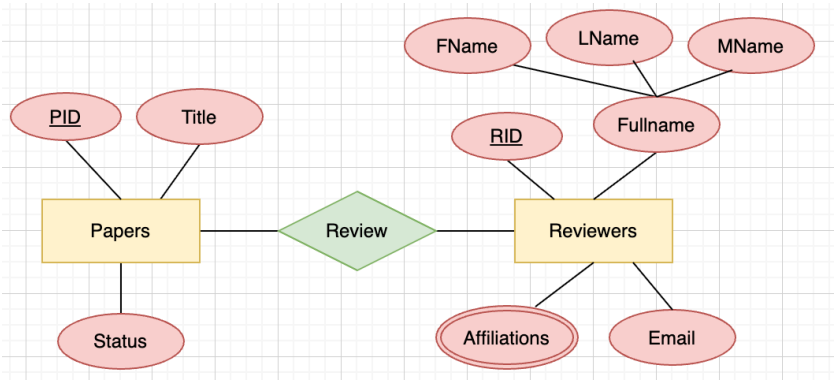
Exercise

Paper (Entity)	Property	Reviewer (Entity)	Property
PID	Key	RID	Key
Title	Normal	Fullname (FName, LName, MName)	Composite
Status	Normal	Email address	Normal
		Affiliations	Multivalue

Relationship: Paper (2, N) - Reviews - (0, N) Reviewer

Exercise

Assoc. Prof. Nguyen Thi Thuy Loan, PhD



Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Thank you for your attention!