The Entity-Relationship Model

Chapter 2

Last Time

Database Models

Determine how data can be stored, organized and manipulated in a database system

Common Models?

- …

Last Time

Database Models

Determine how data can be stored, organized and manipulated in a database system

Common Models:

- Entity Relationship Model
  - Useful for end-users and database designers
- Relational Model
  - Implemented in databases
- Network Model
- Object Model
- …

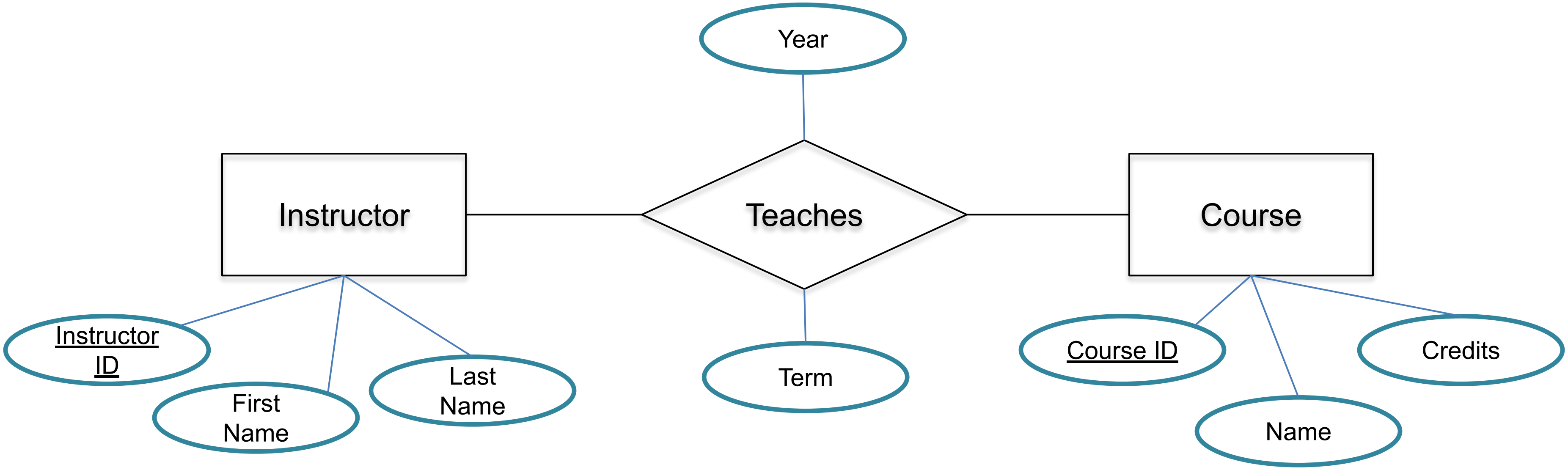
TODAY

Learning Objective

Textual Database requirements □ Entity-Relationship Model

Requirements:

Instructors teach 0 or more courses.
Each course can be taught by 0 or more instructors
Instructors have attributes of first name and last name
Courses have attributes Name and credits
Teaching relationship is valid for a specific term and year.



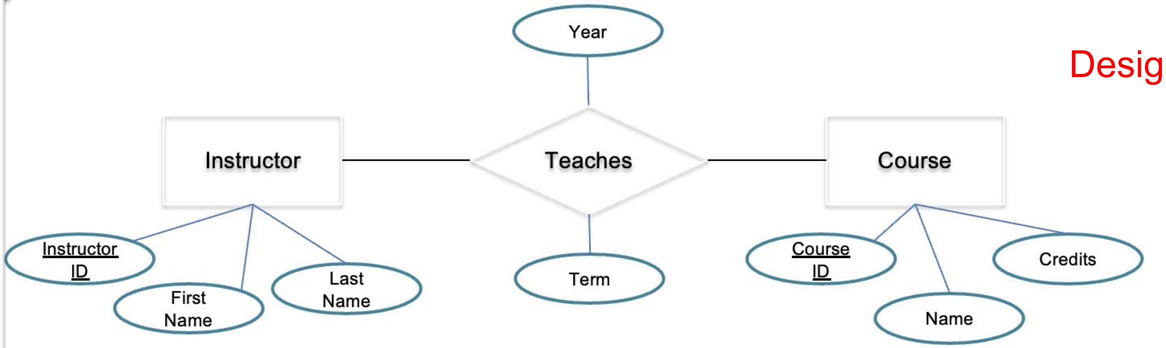
TODAY

We will cover:

- Database design
  - High-level description of data to be stored
- ER modeling
  - Basic constructs (e.g. entities, relationships)
  - Additional constructs (e.g. hierarchies)
  - Integrity constraints : capture real-world properties of the data

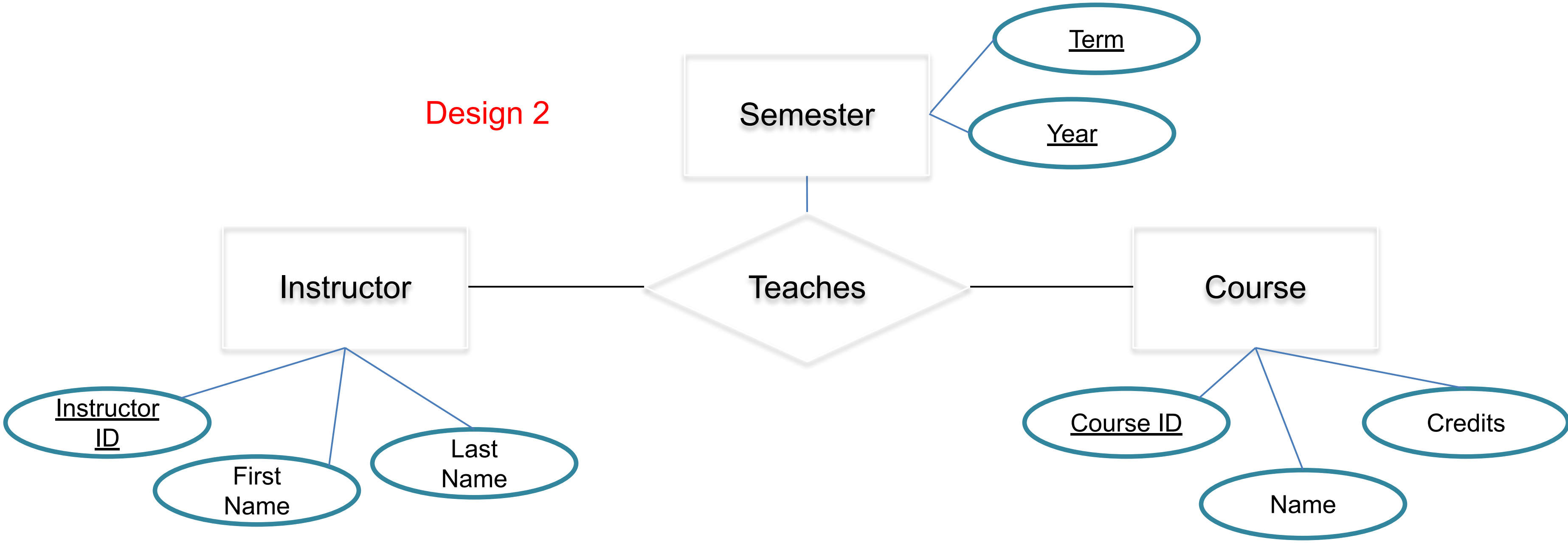
TODAY

Tradeoffs among Designs



Design 1

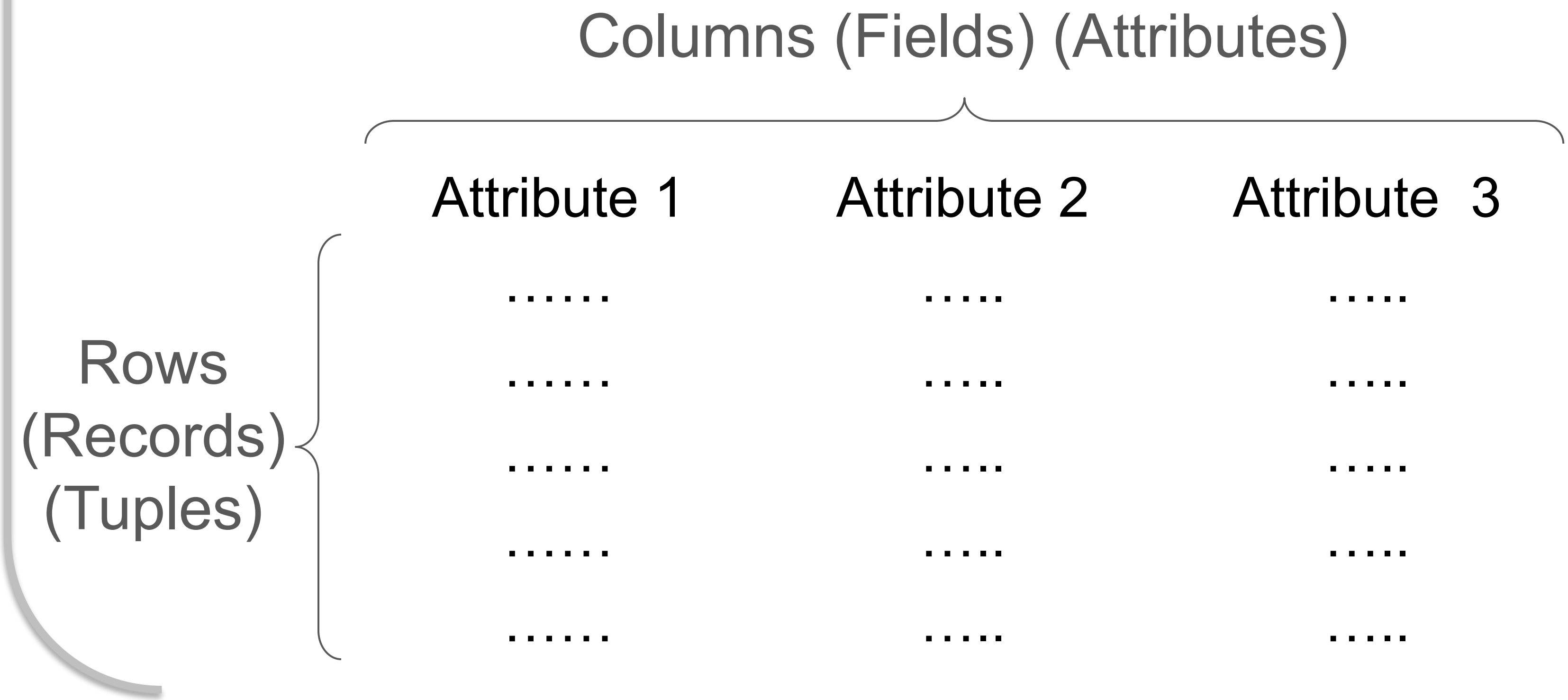
Design 2



Later

Relational Database

- Entities and (most) relationships are translated into Tables



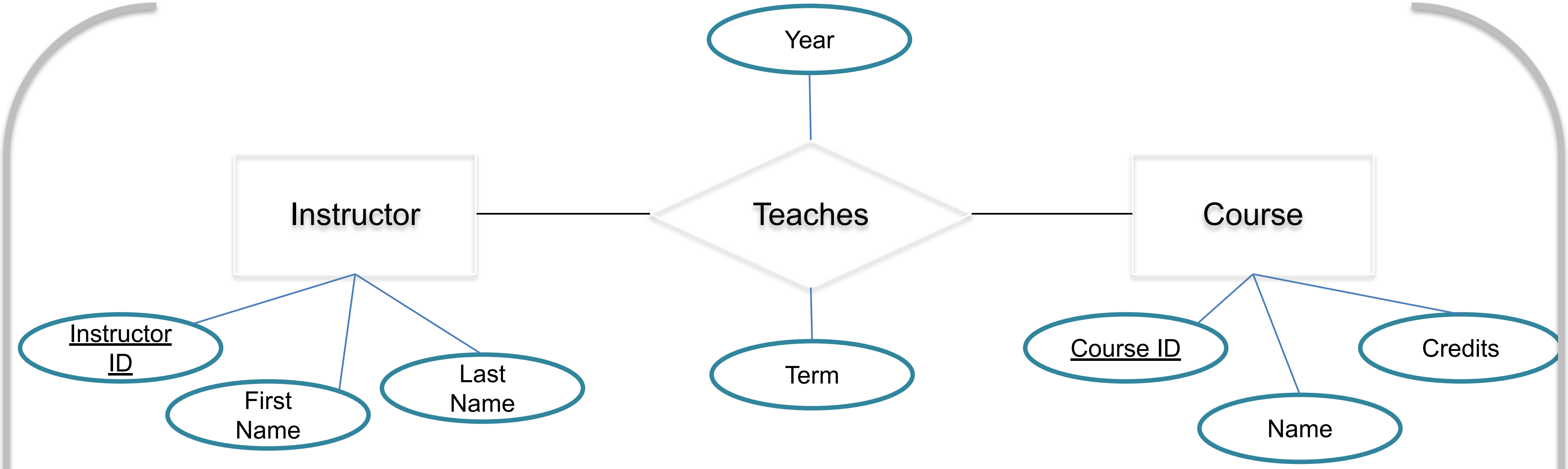
Columns (Fields) (Attributes)

Rows (Records) (Tuples)

Attribute 1	Attribute 2	Attribute 3
……	…..	…..
……	…..	…..
……	…..	…..
……	…..	…..
……	…..	…..

Later

Translation from ER Diagrams to Relations



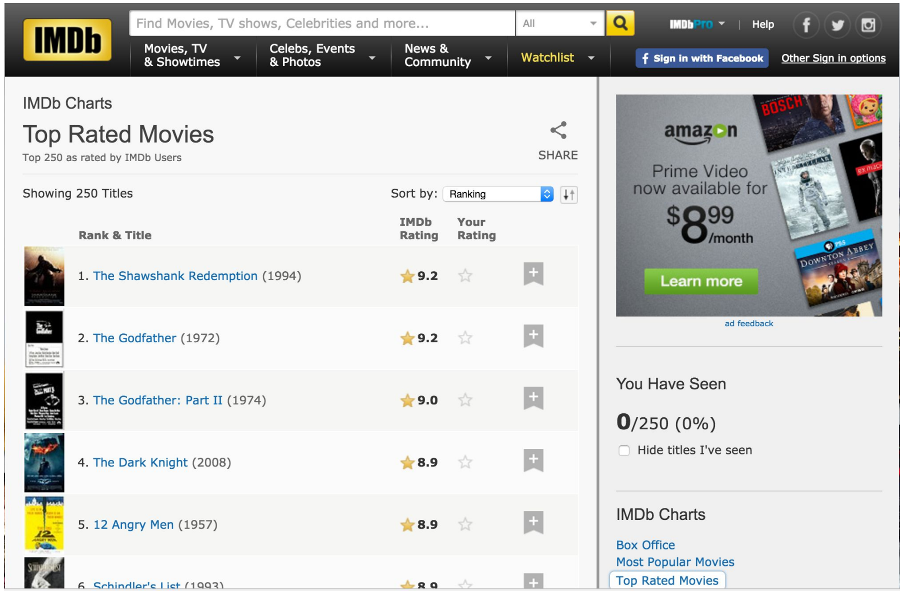
Instructor ID	First Name	Last Name
394953	John	Smith
454544	Sara	King
439849	Alex	Dee
….	….	….
….	….	….

Instructor ID	Course ID	Year	Term
454544	E302	2009	F
394953	C210	2010	W
439849	M184	2010	F
….	….	….	….
….	….	….	….

Course ID	Name	Credits
M184	Calculus	3
C210	Physics	4
E302	Algorithms	4
….	….	….
….	….	….

Scenario 1

- Create IMDb (Internet Movie Database)
  - i.e., a database to store information about movies, casts, directors, ratings, …



IMDb Charts

Top Rated Movies

Top 250 as rated by IMDb Users

Showing 250 Titles

Sort by: Ranking

Rank & Title	IMDb Rating	Your Rating
1. The Shawshank Redemption (1994)	9.2	☆
2. The Godfather (1972)	9.2	☆
3. The Godfather: Part II (1974)	9.0	☆
4. The Dark Knight (2008)	8.9	☆
5. 12 Angry Men (1957)	8.9	☆
6. Schindler's List (1993)	8.9	☆

You Have Seen

0/250 (0%)

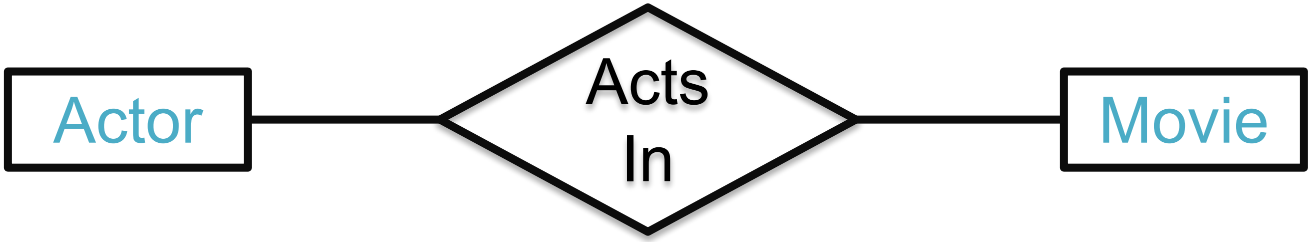
□ Hide titles I've seen

IMDb Charts

Box Office
Most Popular Movies
Top Rated Movies

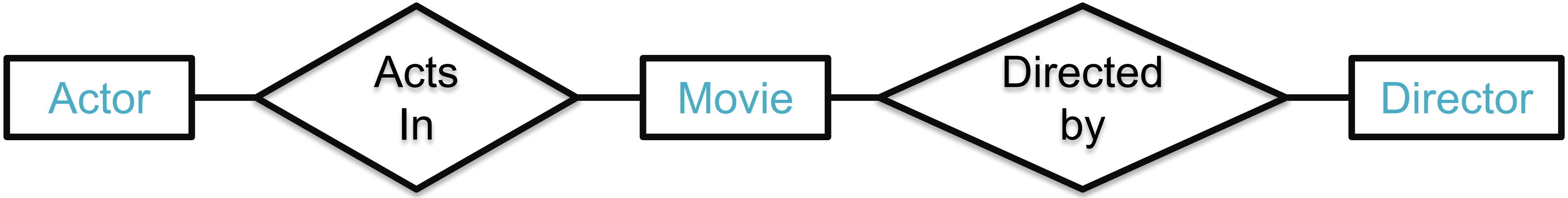
ER Diagram

- ▭ : Entities — ‘Actor’ and ‘Movie’
- ◇ : Relationships — ‘Acts In’
- ⬭ : Attributes



- **Entity sets**: Collection of entity instances
  e.g., set of Actors or set of Movies
- **Relationship set**: e.g. Acts_In:
  {(a1, m1), (a1, m2), (a2, m3), …}

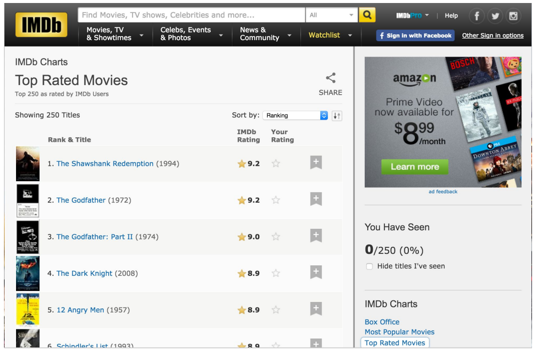
ER Diagram



- Actor *Acts In* Movie
- Movie *Directed by* Director

Scenario 1: Update

- Create IMDB (Internet Movie Database)
  - i.e., a database to store information about the movies, cast, directors, …
- Constraint (they ask you to impose it)
  - A movie can have AT MOST one director
  - A director can direct MULTIPLE movies



IMDb Charts

Top Rated Movies

Top 250 as rated by IMDb Users

Showing 250 Titles

Sort by: Ranking

Rank & Title	IMDb Rating	Your Rating
1. The Shawshank Redemption (1994)	★9.2	☆
2. The Godfather (1972)	★9.2	☆
3. The Godfather: Part II (1974)	★9.0	☆
4. The Dark Knight (2008)	★8.9	☆
5. 12 Angry Men (1957)	★8.9	☆
6. Schindler's List (1993)	★8.9	☆

You Have Seen

0/250 (0%)

□ Hide titles I've seen

IMDb Charts

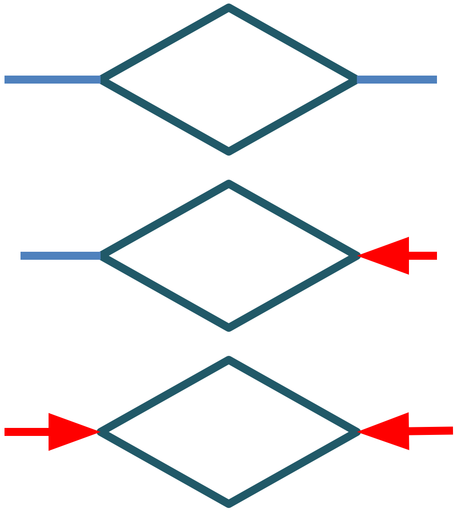
Box Office

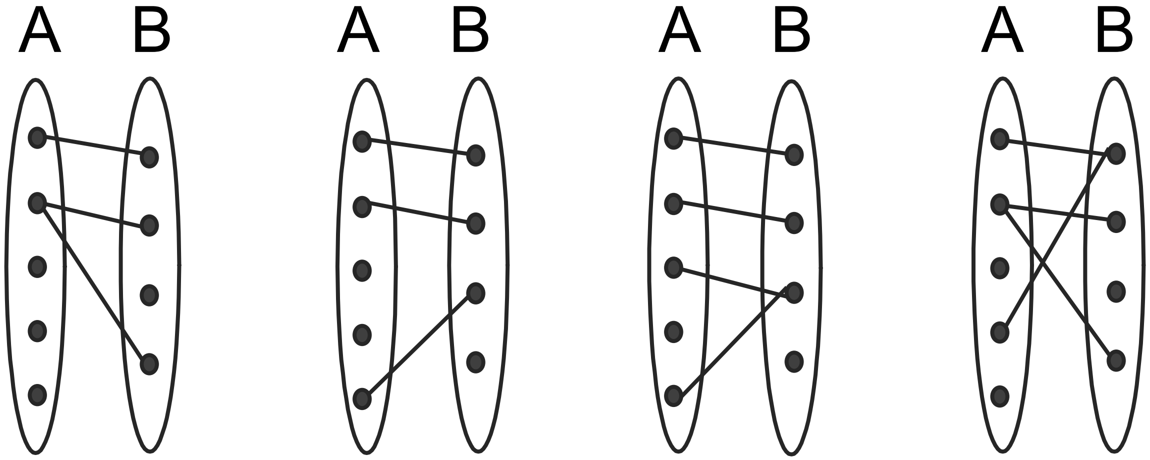
Most Popular Movies

Top Rated Movies

ER Diagram: Relationship Types

- Many-to-Many
- One-to-Many (or Many-to-One)
- One-to-One

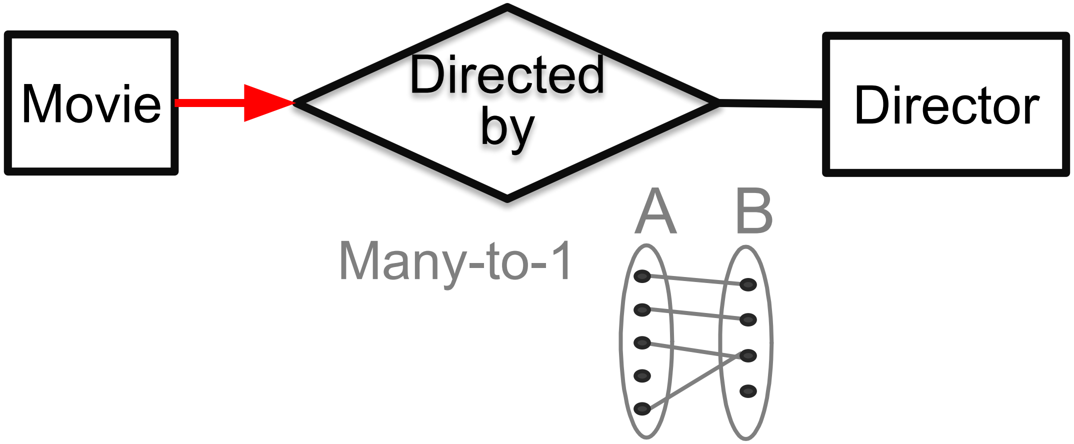




1-to Many 1-to-1 Many-to-1 Many-to-Many

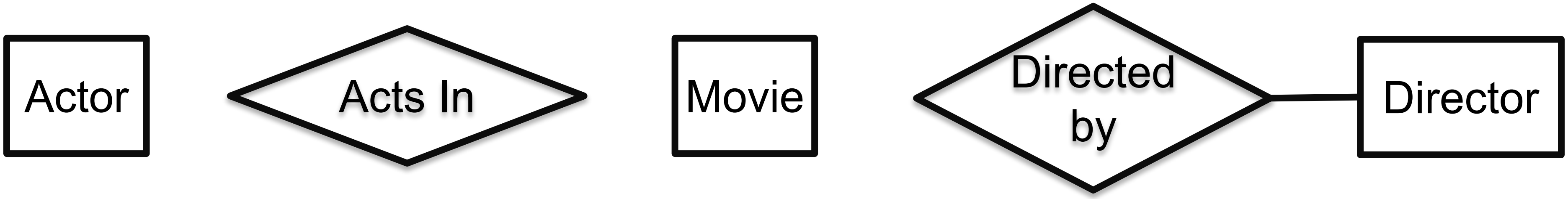
Scenario 1: Update

- Create IMDB (Internet Movie Database)
  - i.e., a database to store information about the movies, cast, directors, …
- Constraint (they ask you to impose it)
  - A movie can have AT MOST one director
  - A director can direct MULTIPLE movies



Types of relationships?

- Acts In:
  - An actor can act in multiple movies and a movie can have multiple actors
- Directed by:
  - A movie has **at most** one director
  - **Arrow**: indicates **Key Constraint** on directed-by relationship: a movie in the relationship must be unique



Actor

Acts In

Movie

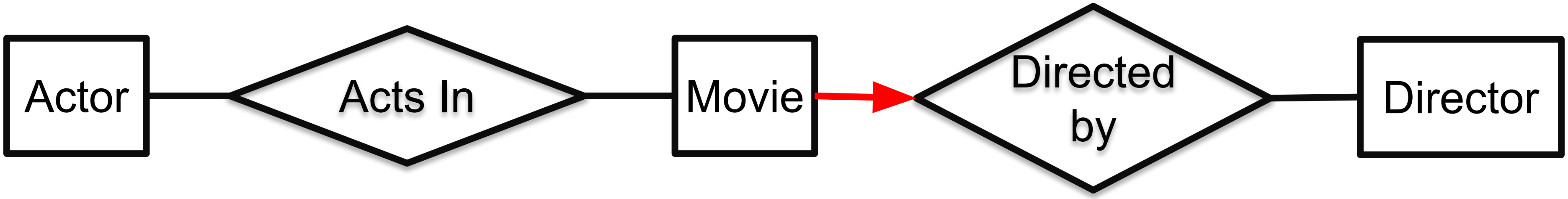
Directed by

Director

Types of relationships?

IMDb

- Acts In: Many-to-Many
  - An actor can act in multiple movies and a movie can have multiple actors
- Directed by: Many-to-1
  - A movie has **at most** one director
  - **Arrow**: indicates **Key Constraint** on directed-by relationship: a movie in the relationship must be unique

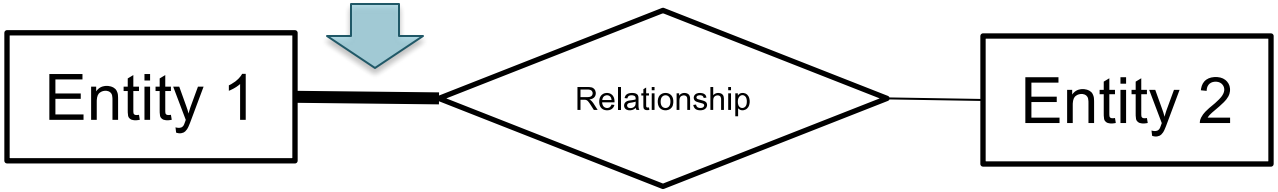


Scenario 1: Update (2)

- Create IMDB (Internet Movie Database)
  - i.e., a database to store information about the movies, cast, directors, …
- Constraint 1
  - A movie can have AT MOST one director
  - A director can direct MULTIPLE movies
- Constraint 2
  - Every Movie entity must participate in a relationship with an Actor
    - i.e., every movie must have ***at least*** one actor

Participation constraints

- **Heavy** line

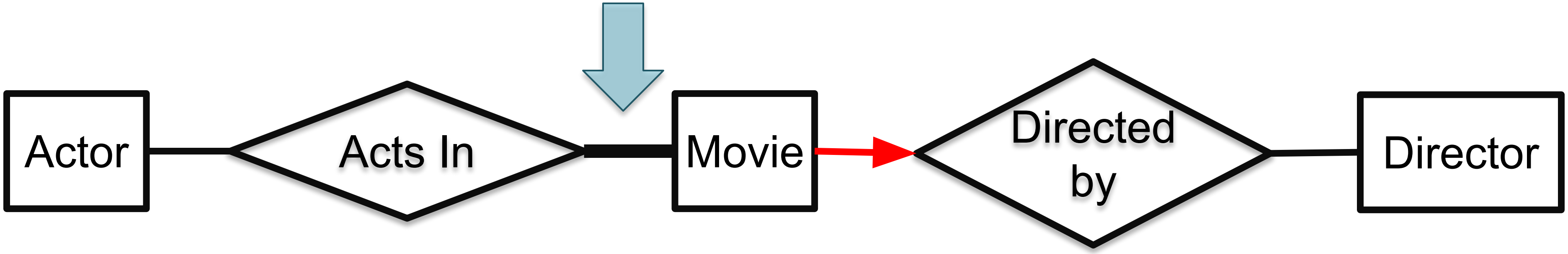


  - Every Entity-1 entity **must participate** in a relationship with an Entity-2 entity.

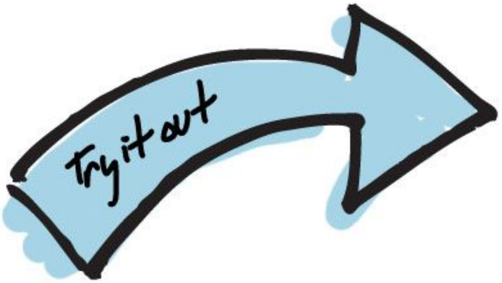
- Light line
  - An Entity-2 can be related to ≥ 0 Entity-1 entities.

Final ER diagram

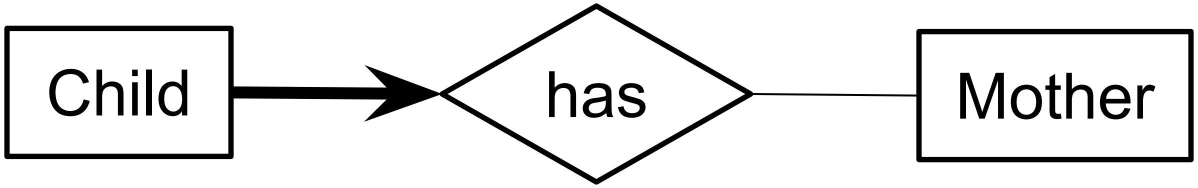
- Participation Constraint
  - Every Movie entity must participate in a relationship with an Actor
    - i.e., every movie must have ***at least*** one actor



- An Actor can participate in ≥ 0 movies.
- There can be actors who never act in movies.



Child and Birth Mother



Child

has

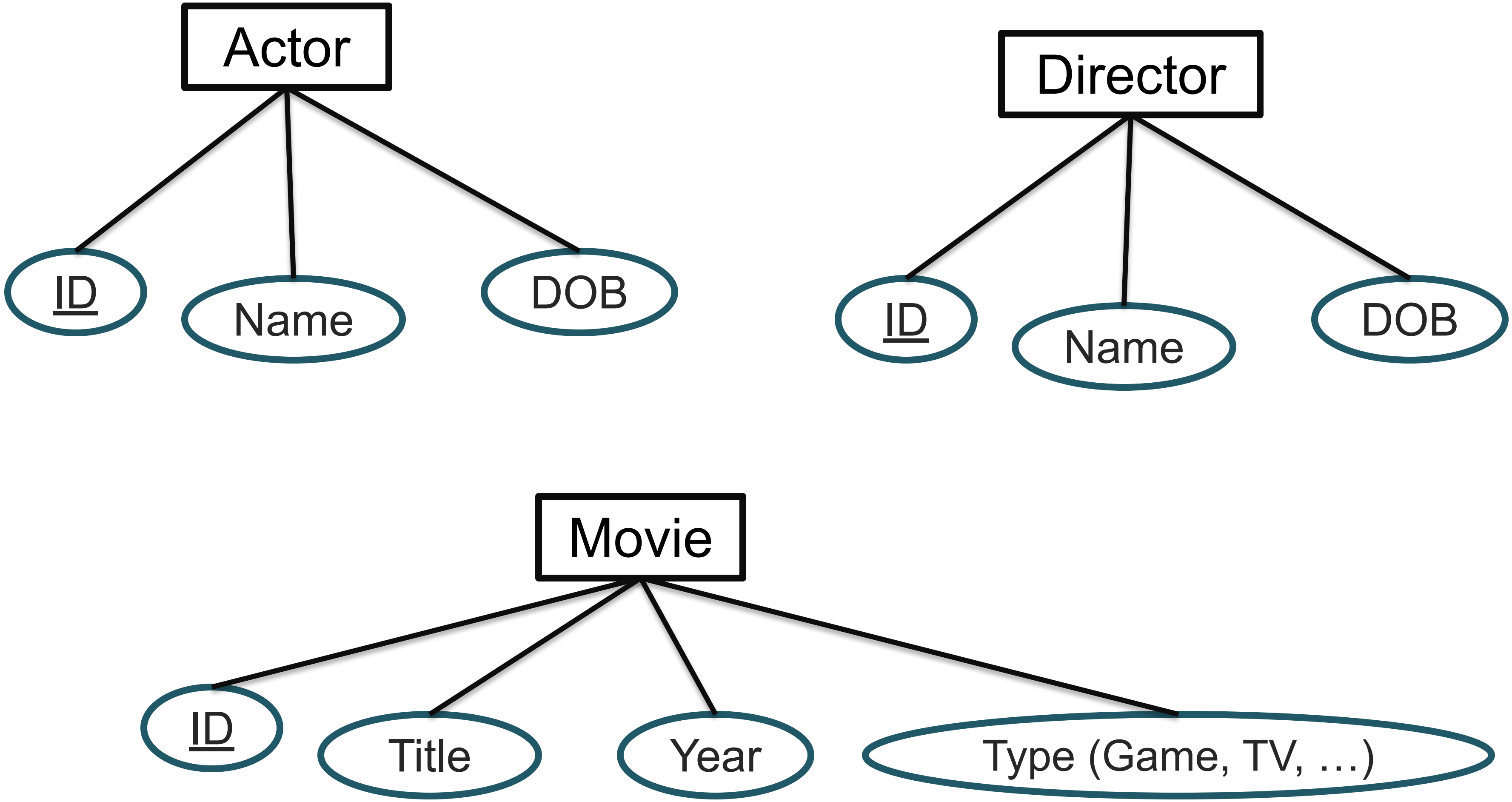
Mother

- **Key** Constraint:
  - A child has **at most** one birth mother but a birth mother may bear several children.
- **Participation** Constraint:
  - Each child must have **at least** one birth mother
- **Net Result**: Every child has exactly one birth mother.

Scenario 1: Update (3)

- Create IMDB (Internet Movie Database)
  - i.e., a database to store information about the movies, cast, directors, …
- Constraint 1 (they ask you to impose it)
  - A movie can have AT MOST one director
  - A director can direct MULTIPLE movies
- Constraint 2
  - Every Movie entity must participate in a relationship with an Actor
    - i.e., every movie must have ***at least*** one actor
- Actors & directors have attributes: ID, Name, DOB
- Movies have attributes: ID, title, year, type

ER Diagram: Attributes



Actor

ID

Name

DOB

Director

ID

Name

DOB

Movie

ID

Title

Year

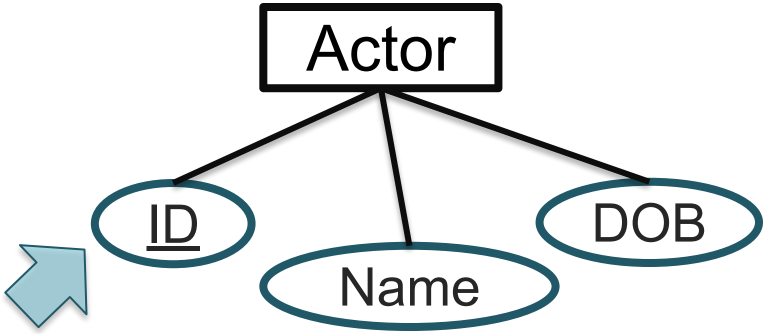
Type (Game, TV, …)

Keys

Key: A minimal set of one or more attributes that has unique value for each record.

- **Candidate keys:** potential keys
- Students in a student database have multiple **potential keys** (they must be unique to be a key):
  - Student ID
  - Login name
  - SSN
  - (Name, address)

Primary Key

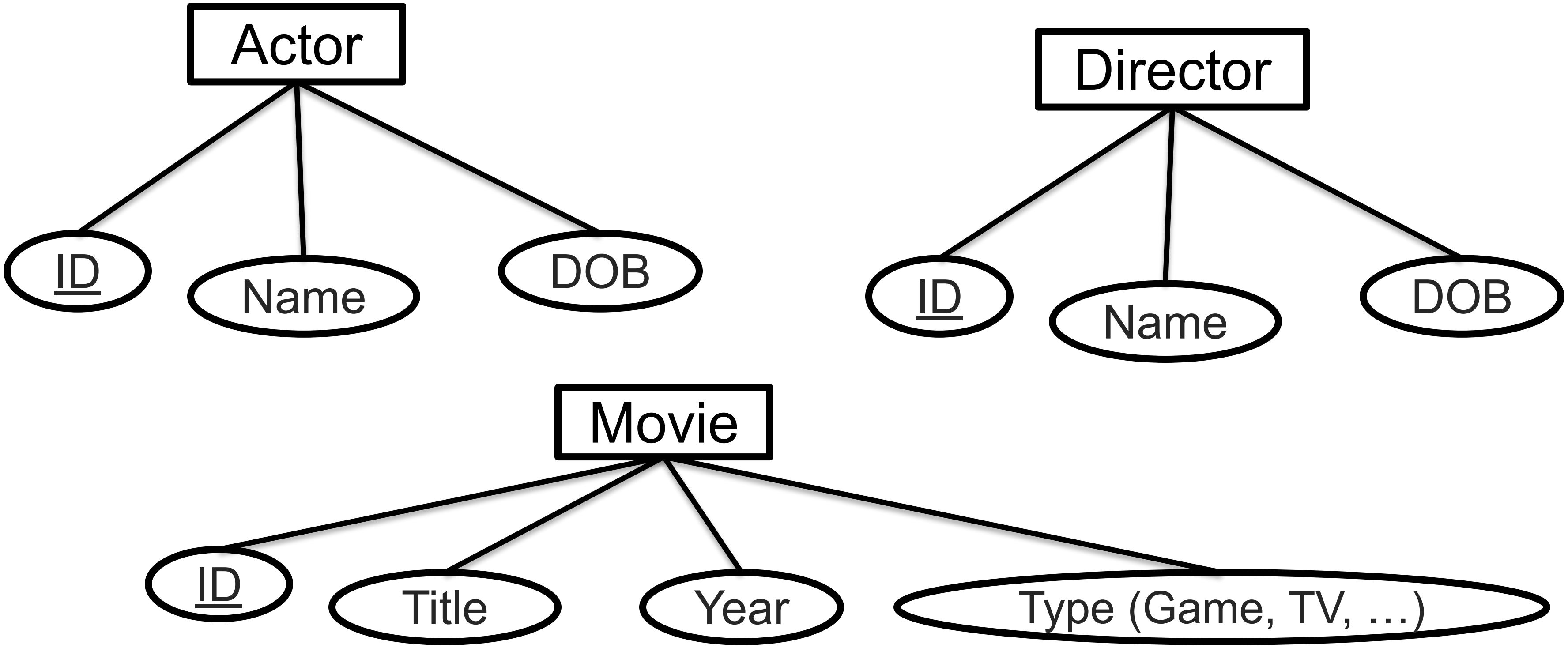


- **Primary Key:** One of the candidate keys
- The primary key attribute(s) is (are) underlined in the ER diagram

- When you design a database, the **primary** key is **cross-referenced** in other tables to represent relationships.
  - e.g., For students, Student ID is a good primary key

Primary Keys

- Often each Entity is assigned a unique ID, which serves as a primary key


Actor

ID

Name

DOB

Director

ID

Name

DOB

Movie

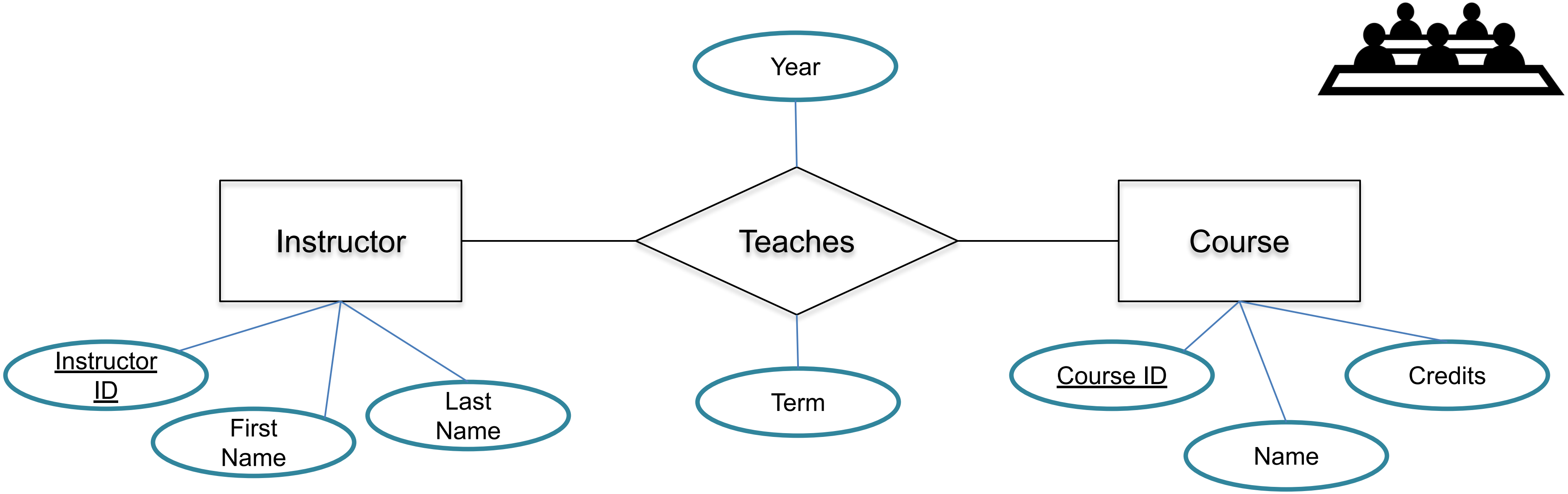
ID

Title

Year

Type (Game, TV, …)

What are the Primary Keys for Instructor? Course? Teaches?

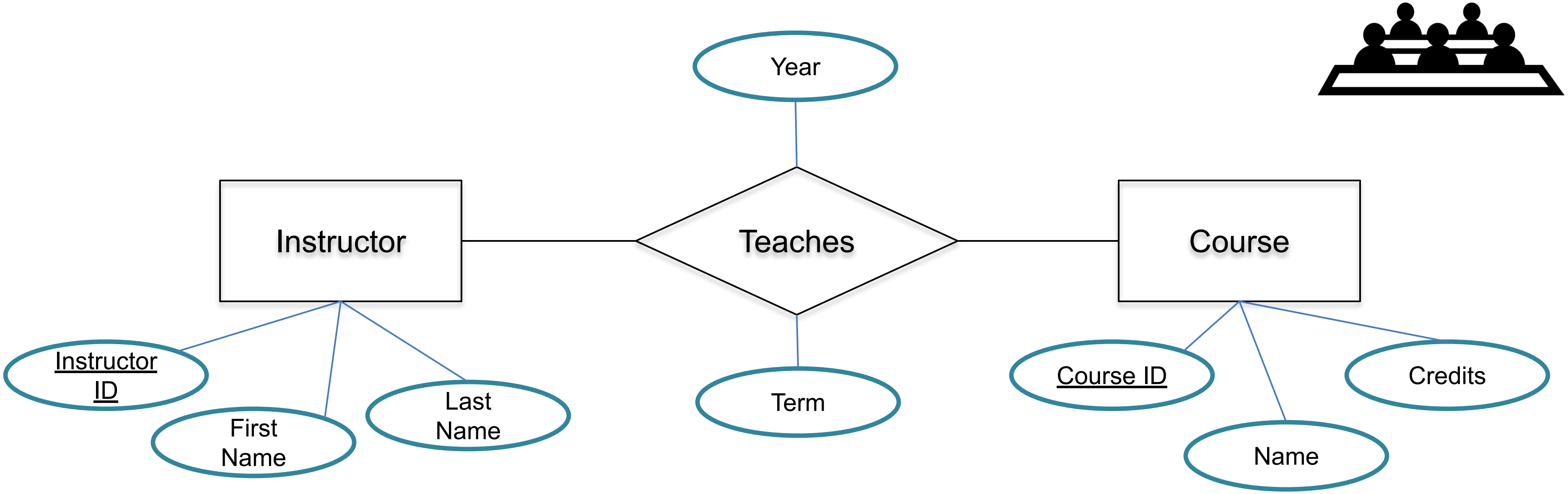


Instructor ID	First Name	Last Name
394953	John	Smith
454544	Sara	King
439849	Alex	Dee
….	….	….
….	….	….

Instructor ID	Course ID	Year	Term
454544	E302	2009	F
394953	C210	2010	W
439849	M184	2010	F
….	….	….	….
….	….	….	….

Course ID	Name	Credits
M184	Calculus	3
C210	Physics	4
E302	Algorithms	4
….	….	….
….	….	….

What are the Primary Keys for Instructor? Course? Teaches?



Instructor ID	First Name	Last Name
394953	John	Smith
454544	Sara	King
439849	Alex	Dee
….	….	….
….	….	….

Instructor Entities

Instructor ID	Course ID	Year	Term
454544	E302	2009	F
394953	C210	2010	W
439849	M184	2010	F
….	….	….	….
….	….	….	….

Teaches Relationship

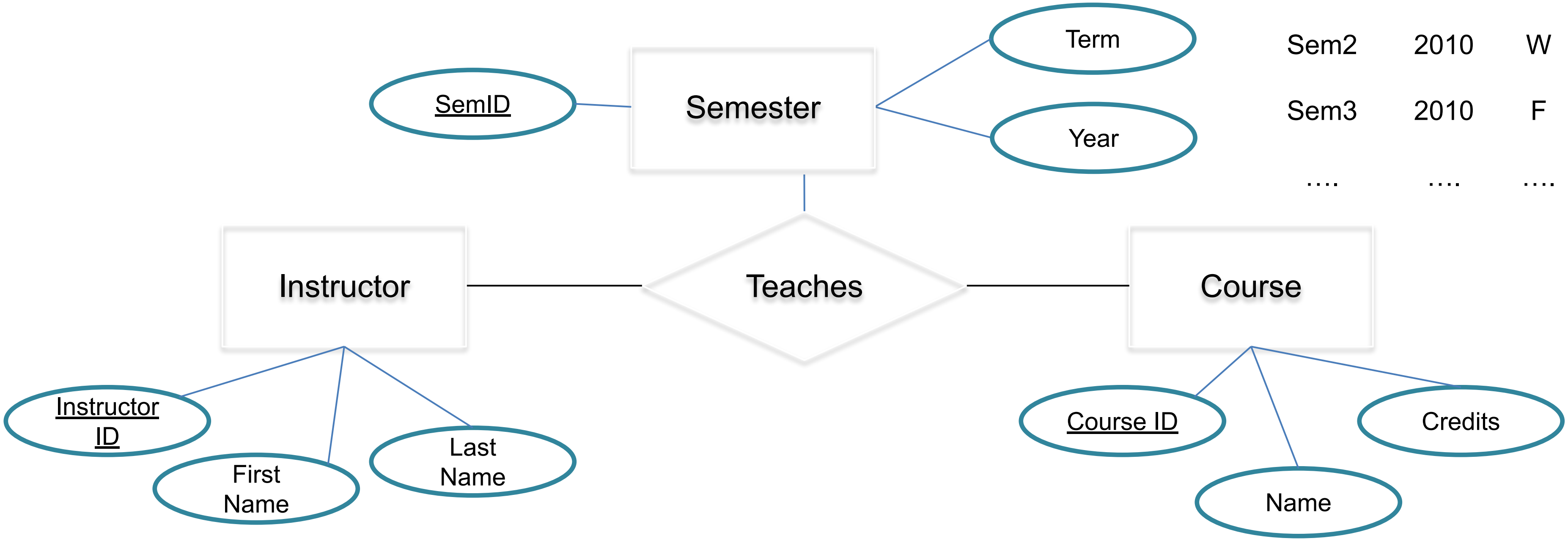
Course ID	Name	Credits
M184	Calculus	3
C210	Physics	4
E302	Algorithms	4
….	….	….
….	….	….

Course Entities

Difference from this design?

Semester Entities

SemID	Year	Term
Sem1	2009	F
Sem2	2010	W
Sem3	2010	F
….	….	….



Instructor ID	First Name	Last Name
394953	John	Smith
454544	Sara	King
439849	Alex	Dee
….	….	….
….	….	….

Instructor Entities

Instructor ID	Course ID	SemID
454544	E302	Sem1
394953	C210	Sem2
439849	M184	Sem3
….	….	….
….	….	….

Teaches Relationship

Course ID	Name	Credits
M184	Calculus	3
C210	Physics	4
E302	Algorithms	4
….	….	….
….	….	….

Course Entities

Practice ER Modeling Problem

Scenario 2

ER Modeling Problem

- Citizens vote on initiatives at polling stations.
- Polling stations have a location and size.
  - Citizen info: SSN, name, bday (SSN is unique)
  - Initiatives info: ID, name, description (Id is unique)
- Citizens are represented by an elected official, who is also a citizen.

Exercise:

1. Identify entities (objects), relationships between entities
2. Attach attributes
3. Keys: something that uniquely identifies an entity

ER Modeling Problem
STEP 1a

- **Citizens** vote on **initiatives** at **polling stations**.
- **Polling stations** have a location and size.
  - Citizen info: SSN, name, bday (SSN is unique)
  - Initiatives info: Id, name, description (Id is unique)
- **Citizens** are represented by an elected official, who is also a **citizen**.

Exercise:

1. **Identify entities** (objects), relationships between entities
2. Attach attributes
3. Keys: something that uniquely identifies an entity

ER Modeling Problem
STEP 1b

- **Citizens** vote on **initiatives** at **polling stations**.
- **Polling stations** have a location and size.
  - Citizen info: SSN, name, bday (SSN is unique)
  - Initiatives info: Id, name, description (Id is unique)
- **Citizens** are **represented by** an elected official, who is also a **citizen**.

Exercise:

1. **Identify** entities (objects), **relationships** between entities
2. Attach attributes
3. Keys: something that uniquely identifies an entity

ER Modeling Problem
STEP 2

- **Citizens** vote on **initiatives** at **polling stations**.
- **Polling stations** have a location and size.
  - Citizen info: SSN, name, bday (SSN is unique)
  - Initiatives info: Id, name, description (Id is unique)
- **Citizens** are **represented by** an elected official, who is also a **citizen**.

Exercise:

1. **Identify entities** (objects) [And relationships between entities]
2. Attach attributes
3. Keys: something that uniquely identifies an attribute

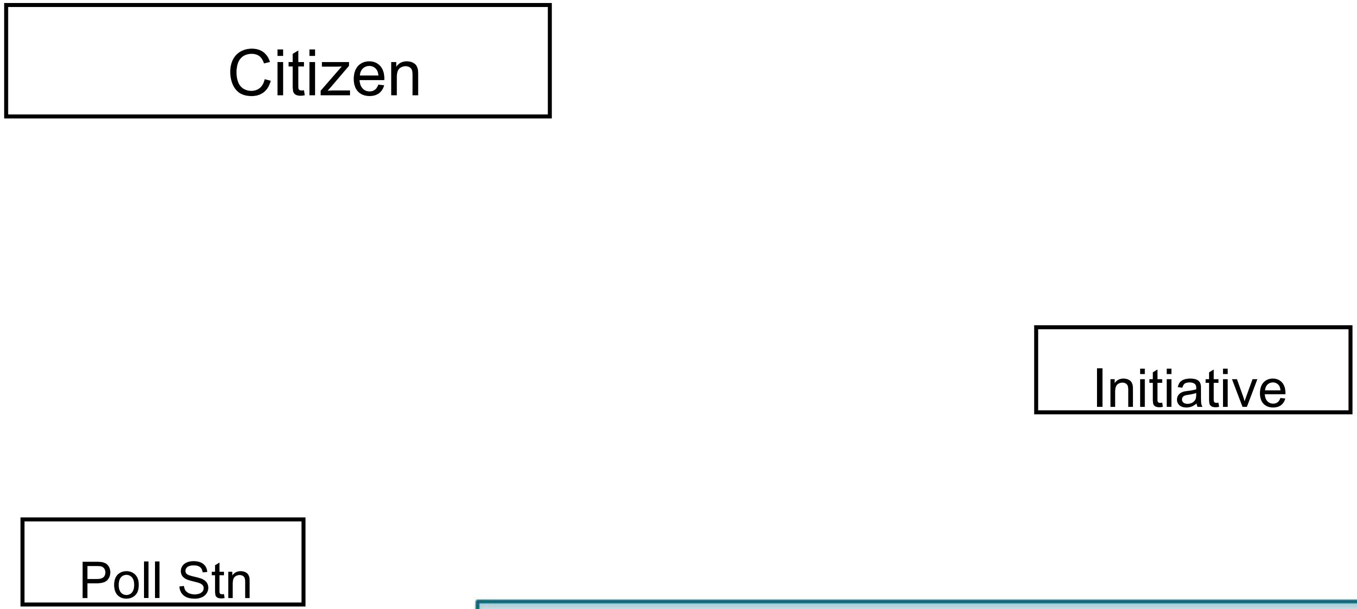
ER Modeling Problem
STEP 3

- **Citizens** **vote** on **initiatives** at **polling stations**.
- **Polling stations** have a location and size. + pid
  - Citizen info: SSN, name, bday (SSN is unique)
  - Initiatives info: Id, name, description (Id is unique)
- **Citizens** are **represented by** an elected official, who is also a **citizen**.

Exercise:

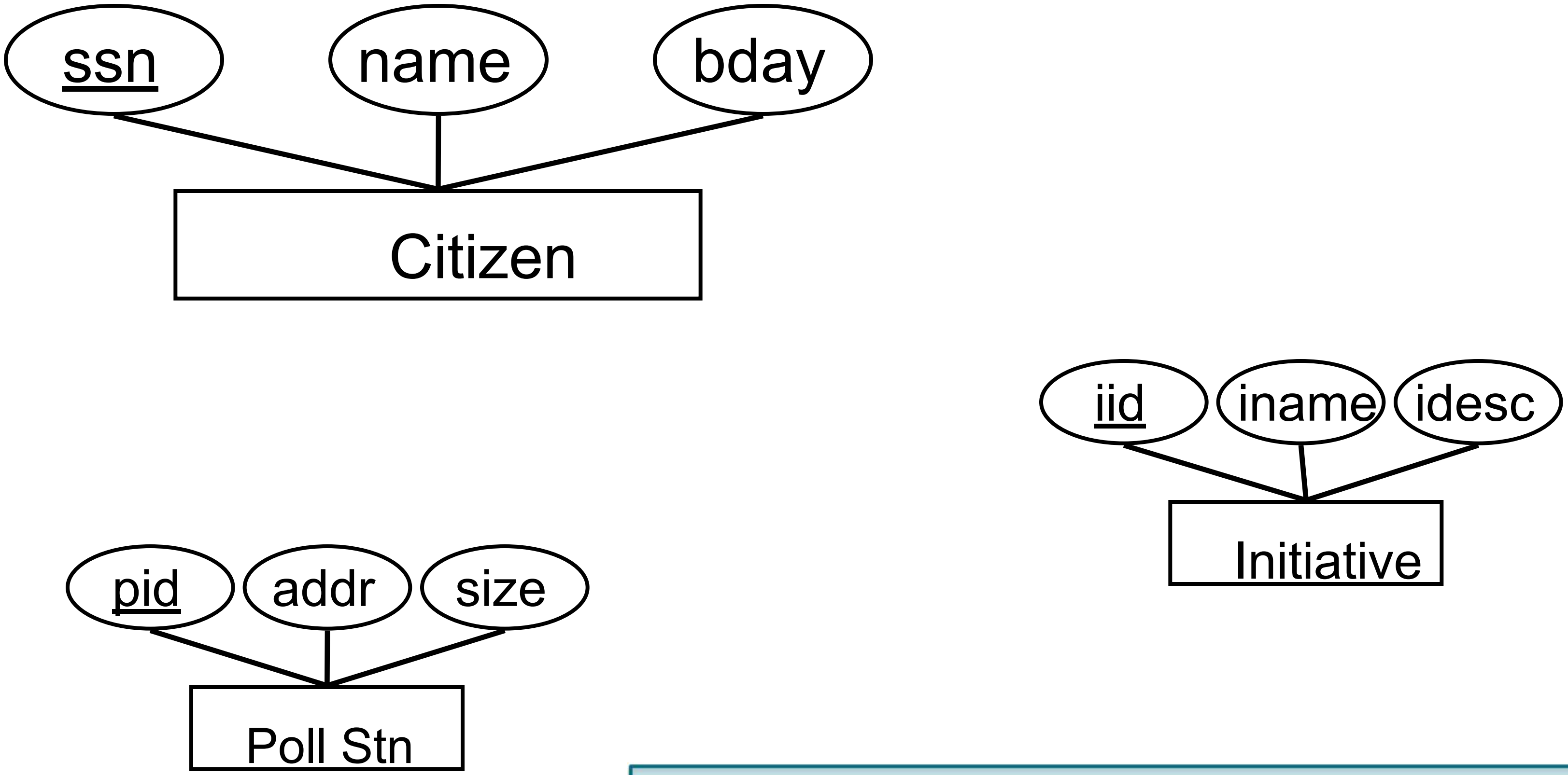
1. **Identify entities** (objects), relationships between entities
2. Attach attributes
3. Keys: something that uniquely identifies an entity

Entities



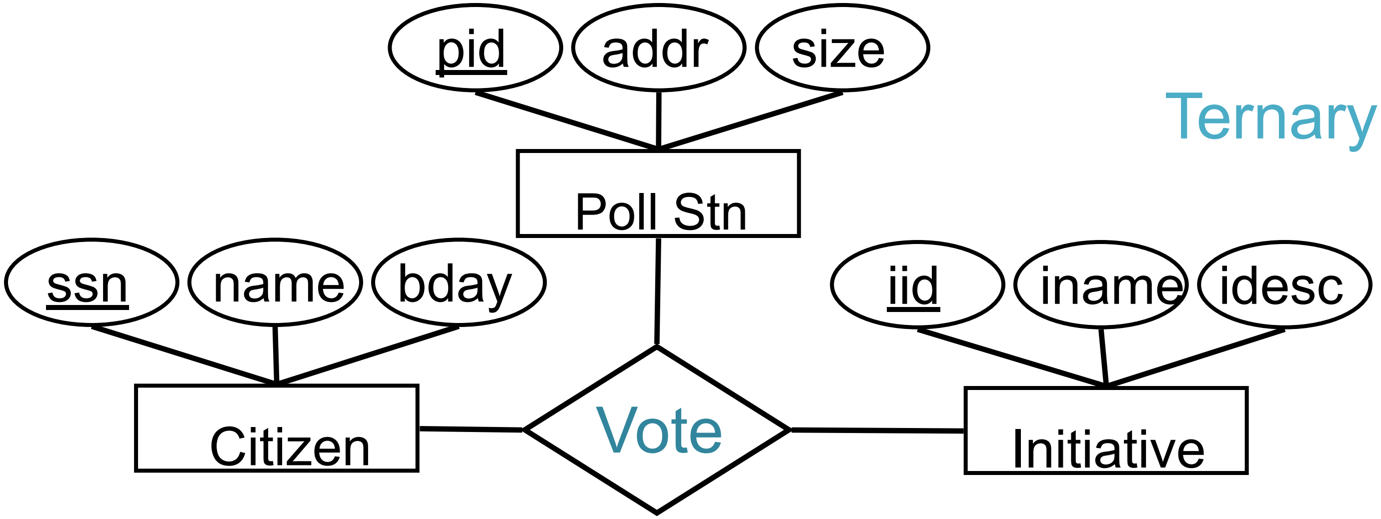
- **Citizens** vote on **initiatives** at **polling stations**.
- **Polling stations** have a location and size. + pid
  - Citizen info: SSN, name, bday (SSN is unique)
  - Initiatives info: Id, name, description (Id is unique)
- **Citizens** are represented by an elected official, who is also a **citizen**.

Add attributes and underline keys



- **Citizens** vote on **initiatives** at **polling stations**.
- **Polling stations** have a location and size. + pid
  - Citizen info: SSN, name, bday (SSN is unique)
  - Initiatives info: Id, name, description (Id is unique)
- **Citizens** are represented by an elected official, who is also a **citizen**.

Add in the relationships



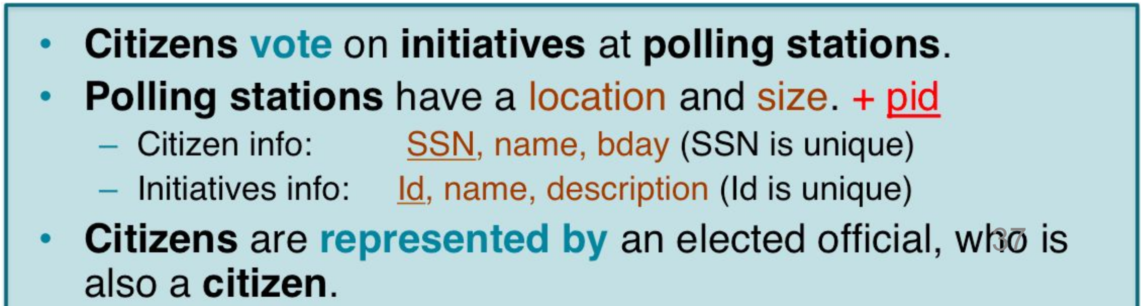
Ternary relationship

- **Relationship set**: $\{(e_1, e_2, \ldots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}$

  e.g., Vote: {(c1, p, i), (c2, p, i), …}

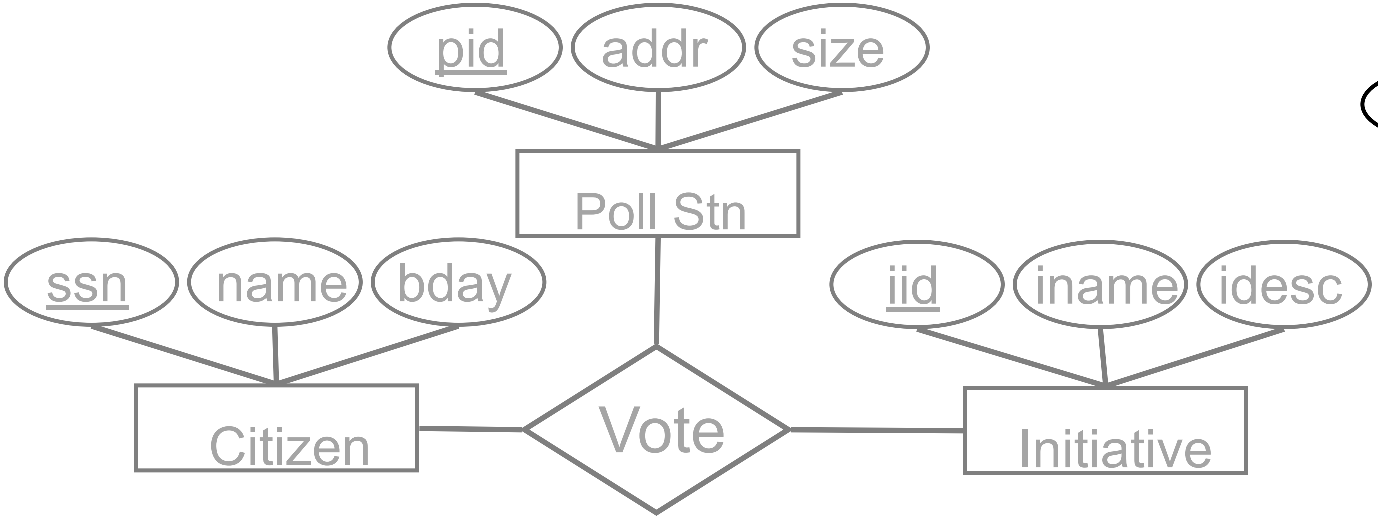
- **Entity sets**: Collection of entity instances

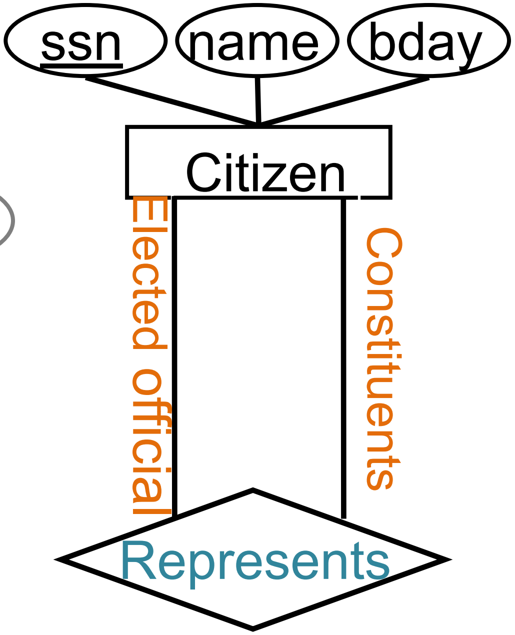
  e.g., set of citizens



- **Citizens** vote on **initiatives** at **polling stations**.
- **Polling stations** have a location and size. + pid
  - Citizen info: SSN, name, bday (SSN is unique)
  - Initiatives info: Id, name, description (Id is unique)
- **Citizens** are represented by an elected official, who is also a **citizen**.

Add in the relationships





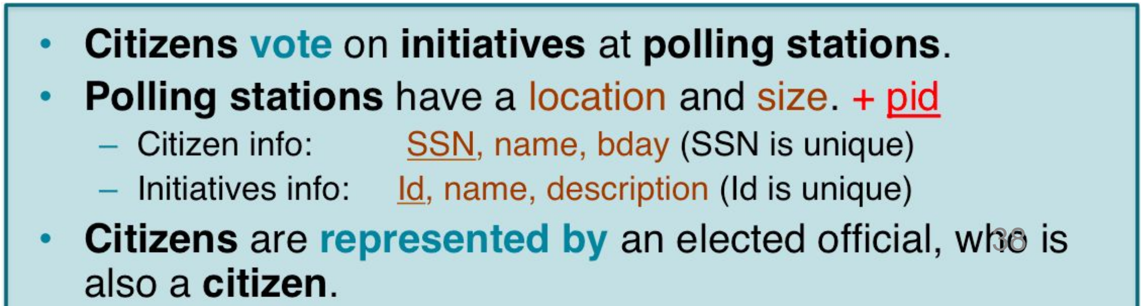
Different roles
In the same entity

- **Relationship set**:

  e.g. Vote: {(c1, p, i), (c2, p, i), …}

- **Entity sets**: Collection of entity instances

  e.g. set of citizens

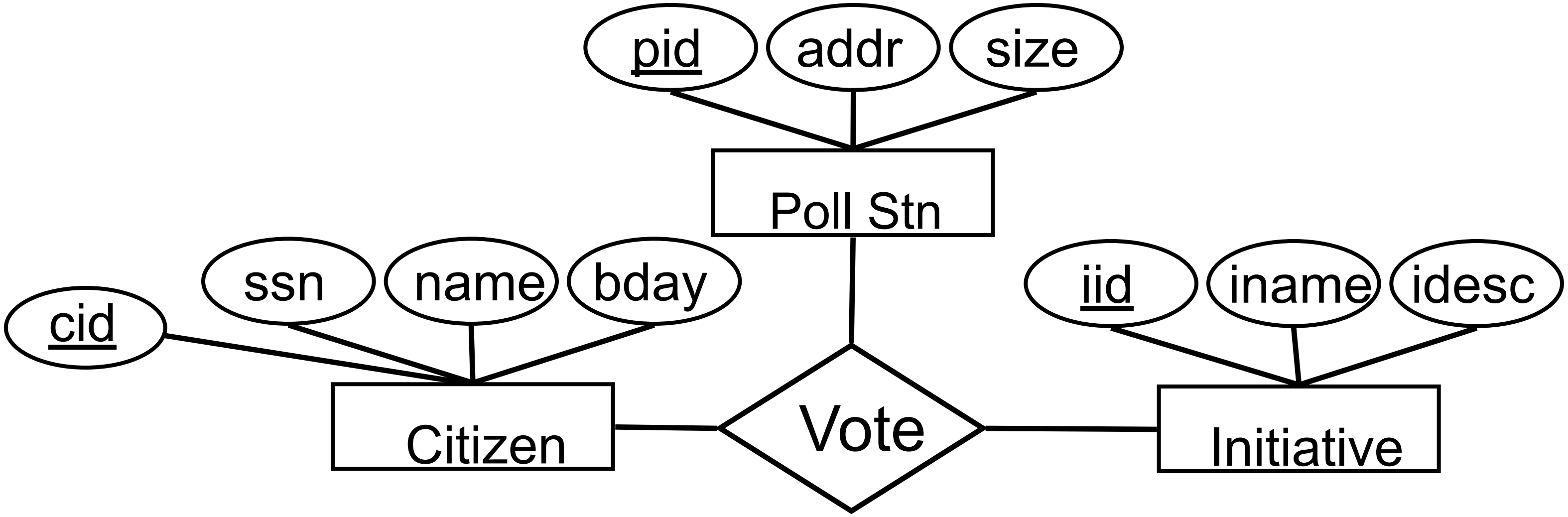


- **Citizens** vote on **initiatives** at **polling stations**.
- **Polling stations** have a location and size. + pid
  - Citizen info: SSN, name, bday (SSN is unique)
  - Initiatives info: Id, name, description (Id is unique)
- **Citizens** are represented by an elected official, who is also a **citizen**.

Advanced ER Modeling

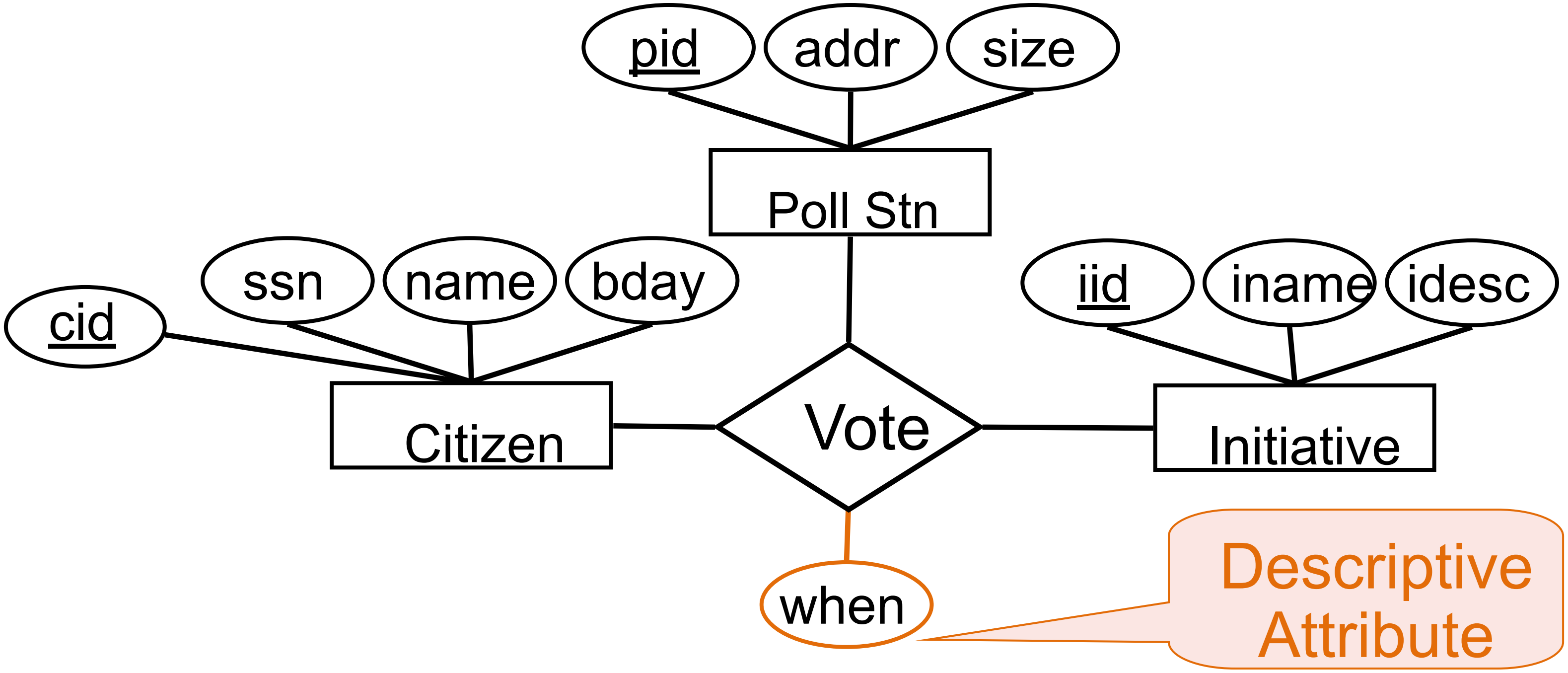
Attributes on Relationships
Key Constraints
Weak Entities
IS-A hierarchy
Aggregation (Relationships with relationships)

Additional requirement 1



- Suppose we want to also record *when* a Citizen voted.
- How should we represent that?

Attributes on Relationships

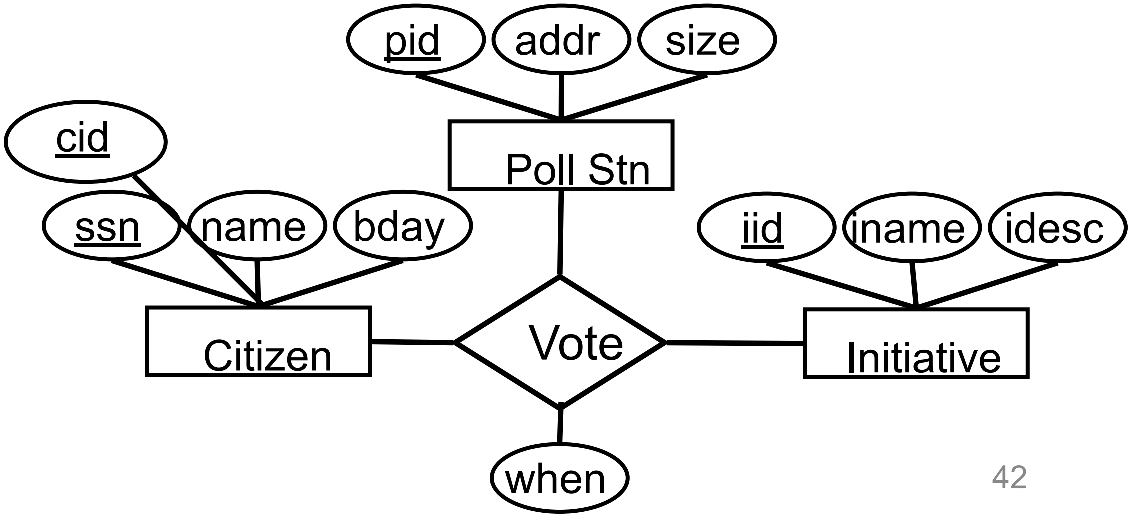


Descriptive Attribute

Additional Requirement 2

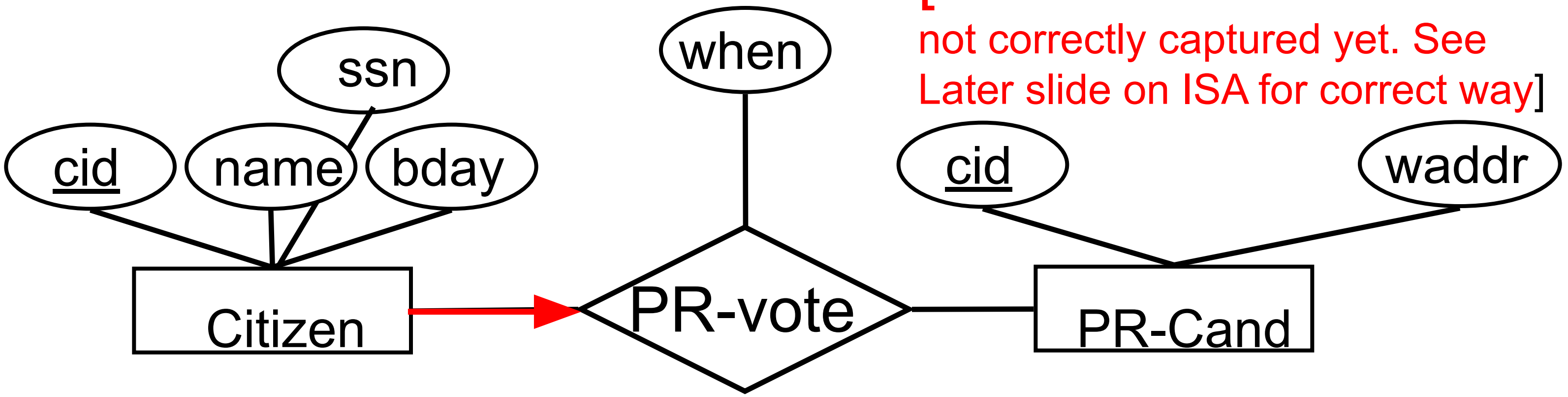
- A citizen also votes (PR-vote) for a presidential candidate.
- They can vote at most once for a presidential candidate, and we record the date of the vote.
- A presidential candidate is also a citizen and has a work address (waddr).

Note: For simplicity, we only focus on the PR-vote relationship in the next slides (and not show Vote).



Additional Requirement 2: Key Constraints

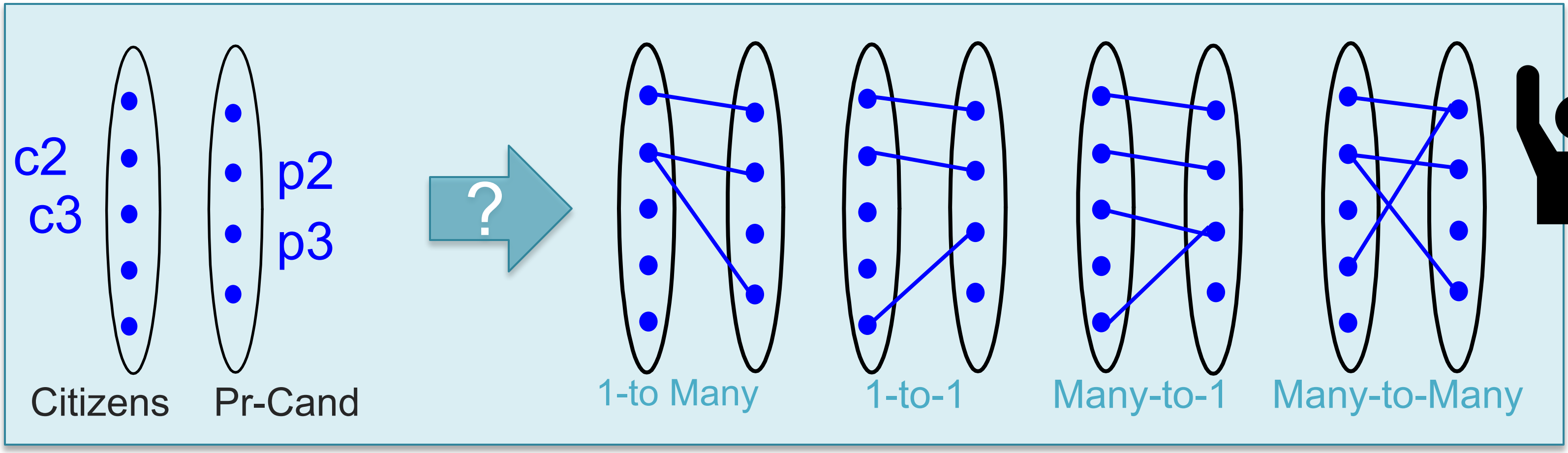
- A citizen also votes for a candidate.
- They can vote **at most once for a candidate**, and we record the date of the vote.
- A candidate is also a citizen and has a work address (waddr).

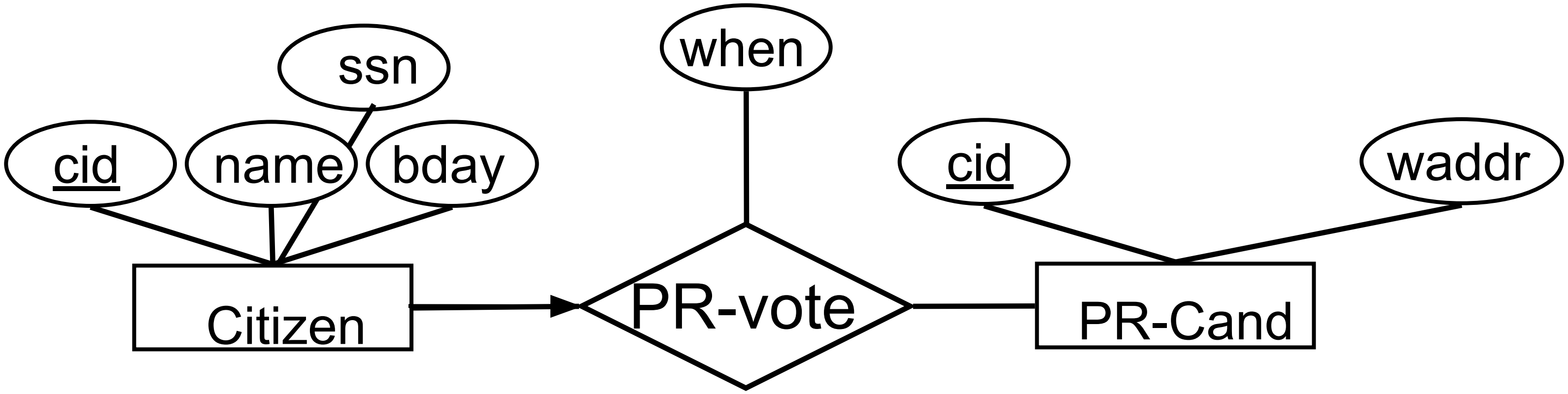


["Candidate is a citizen"" not correctly captured yet. See Later slide on ISA for correct way]

Additional Requirement 2: Key Constraints

- Relationship type between Citizen & PR-Cand?



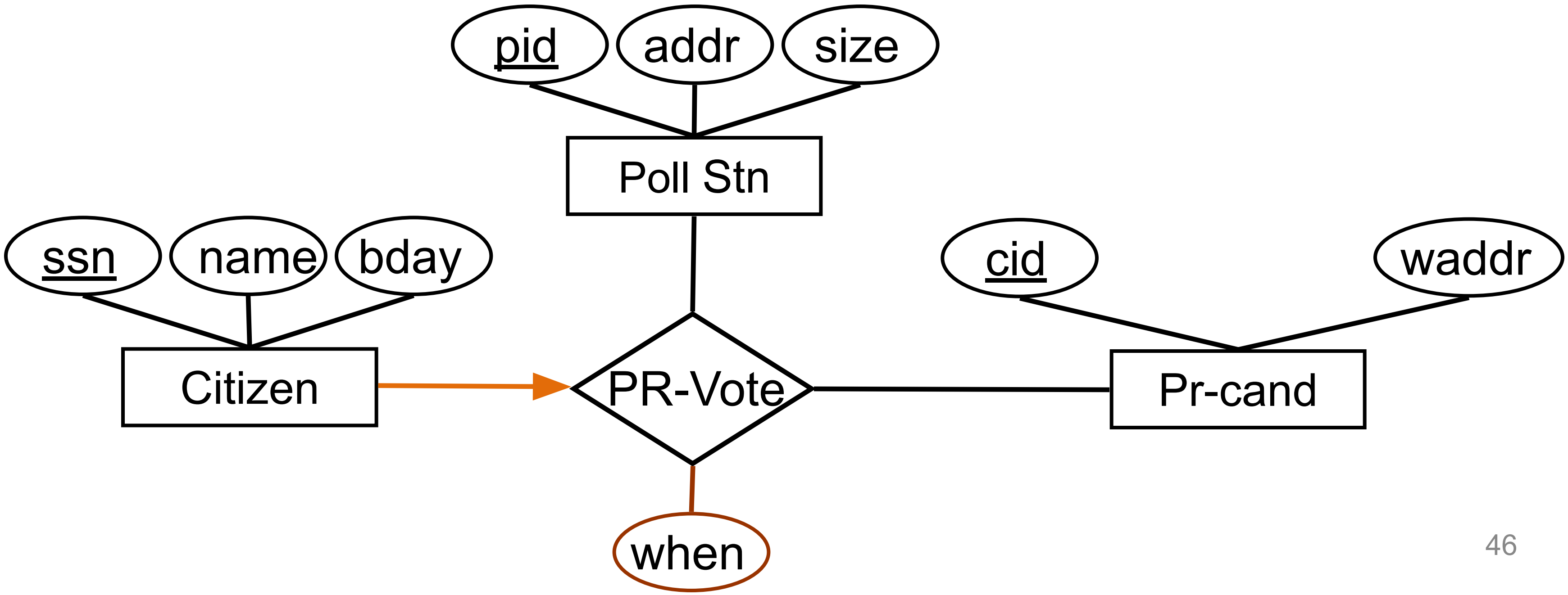


Key Constraints: Generalize

- Each voter votes at most once (for one candidate) and at a polling location.
- A given citizen can appear at most once in (citizen, polling station, candidate) sets

Key Constraints: Generalize

- Each voter votes at most once (for one candidate) and at a polling location.
- A given citizen can appear at most once in (citizen, polling station, candidate) sets

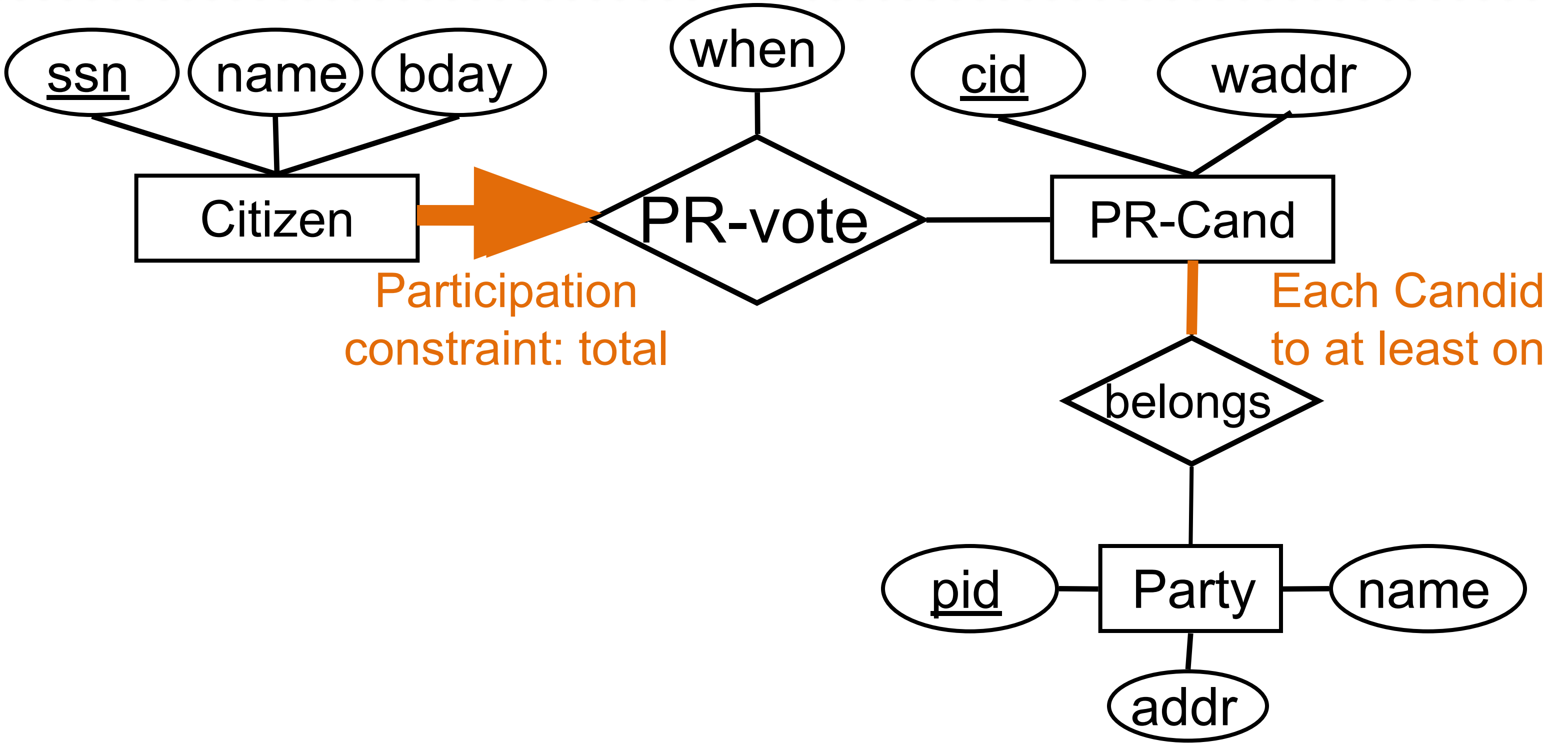


Additional requirements 3

- Every citizen MUST vote for a candidate
- Each candidate belongs to a party. Parties have a name and address
- Each party has exactly one candidate

Additional requirements

- Every citizen MUST vote for a candidate
- Each candidate belongs to a party. Parties have a name and address
- Each party has exactly one candidate

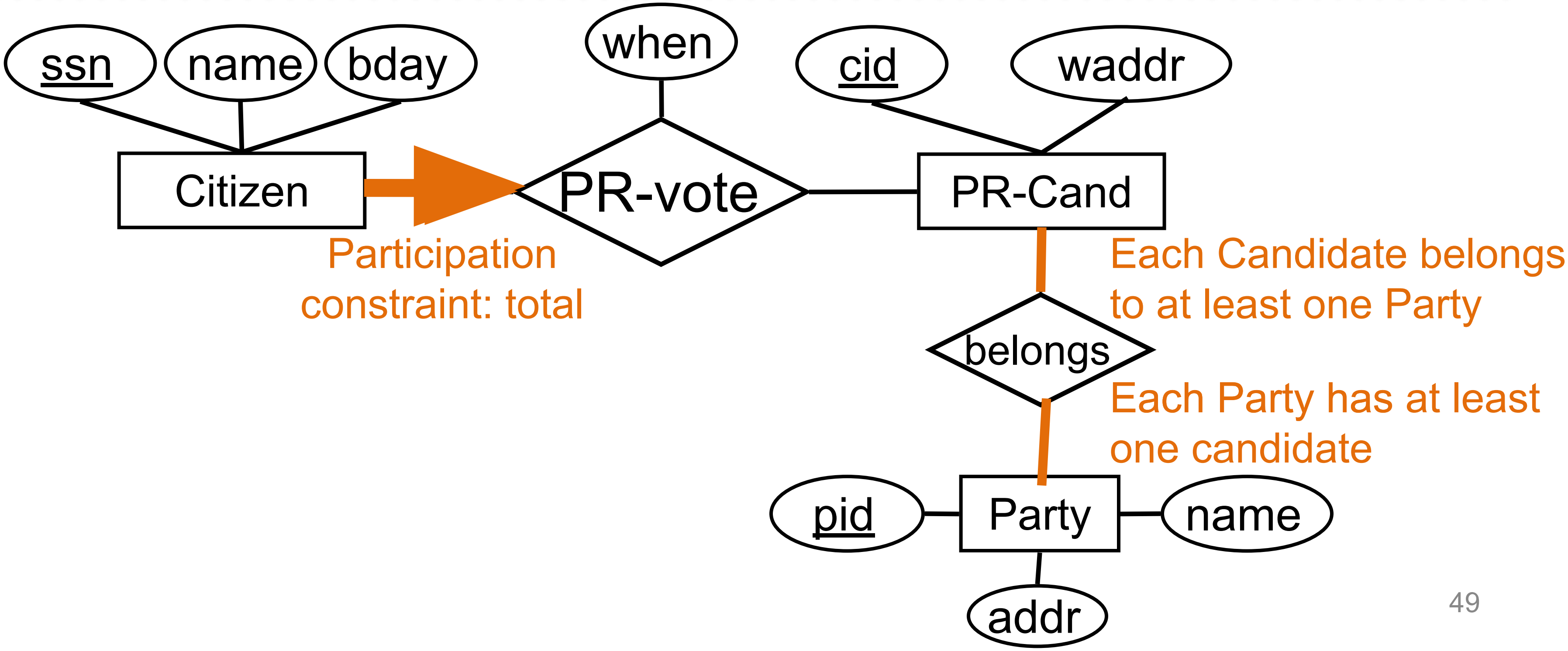


Participation constraint: total

Each Candidate belongs to at least one Party

Additional requirements

- Every citizen MUST vote for a candidate
- Each candidate belongs to a party. Parties have a name and address
- Each party has exactly one candidate



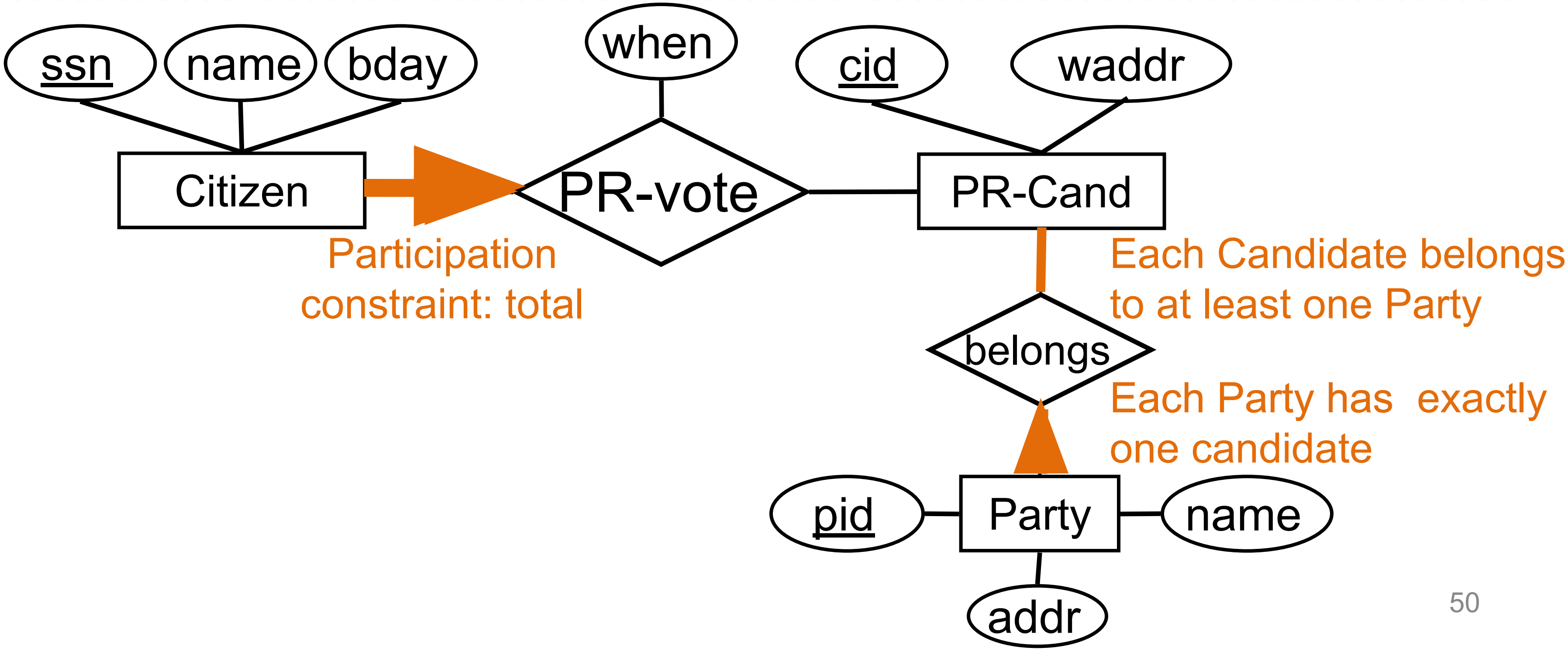
Participation constraint: total

Each Candidate belongs to at least one Party

Each Party has at least one candidate

Additional requirements

- Every citizen MUST vote for a candidate
- Each candidate belongs to a party. Parties have a name and address
- Each party has exactly one candidate

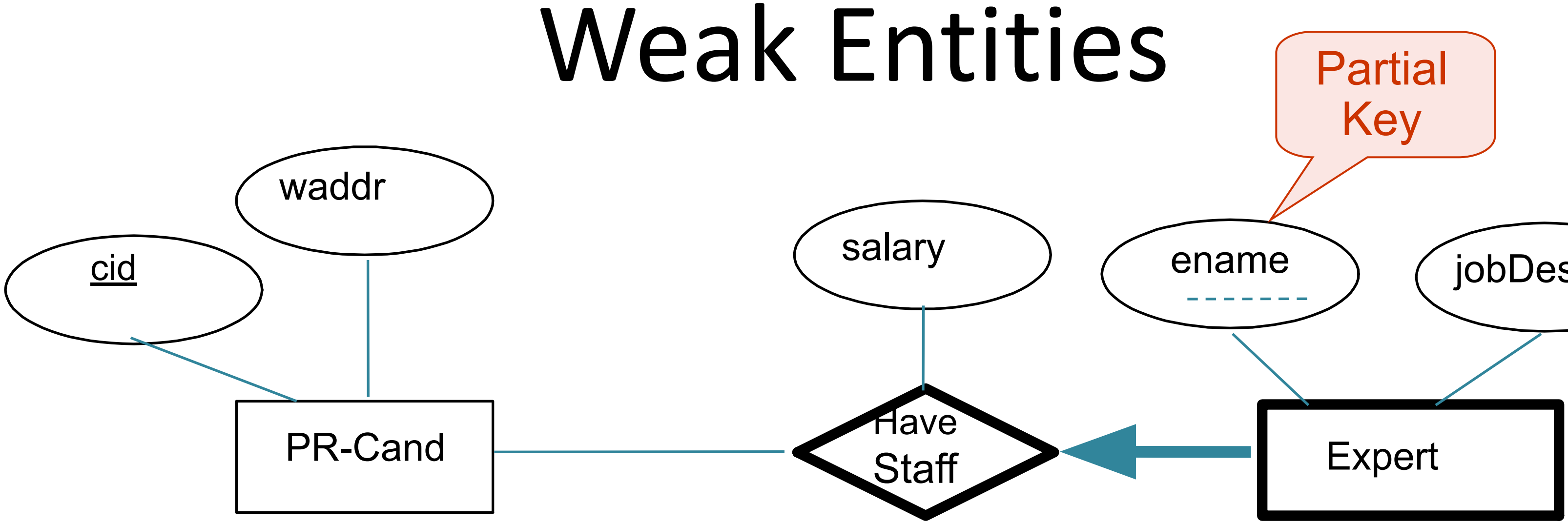


Participation constraint: total

Each Candidate belongs to at least one Party

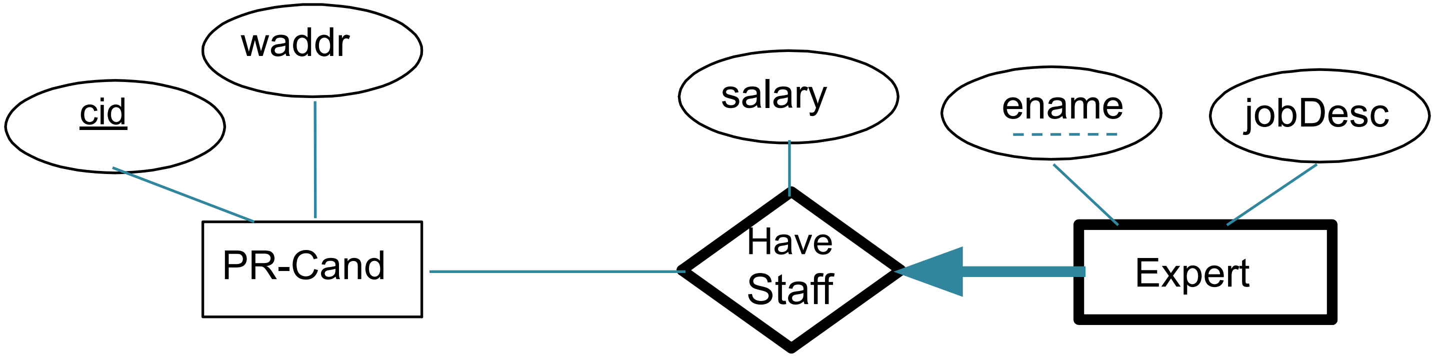
Each Party has exactly one candidate

Weak Entities



- **Example**: Candidates have experts on their staff, identified by their name.
- Experts should be modeled as weak entities
- Expert names are not globally unique. To identify an expert, we need candidate’s ID + expert’s name.

Weak Entities



- A weak entity can be identified uniquely only by considering *some of its attributes in conjunction with* the *primary key* of the another entity (**identifying owner**)
- **Rules:**
  - Weak entity has a single owner (**one-to-many relationship**)
  - Weak entity must have **total participation** in the above identifying relationship set.

ISA (‘is a’) Hierarchies

✓Attributes are inherited (as in C++)

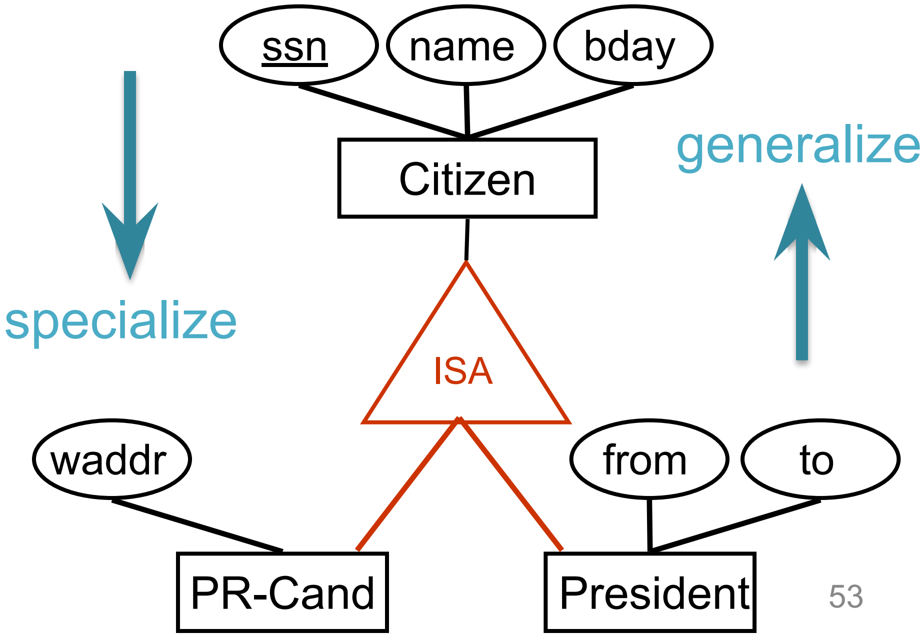
✓If A ISA B, every A entity is also a B entity

✓Specialize superclass (top-down design)

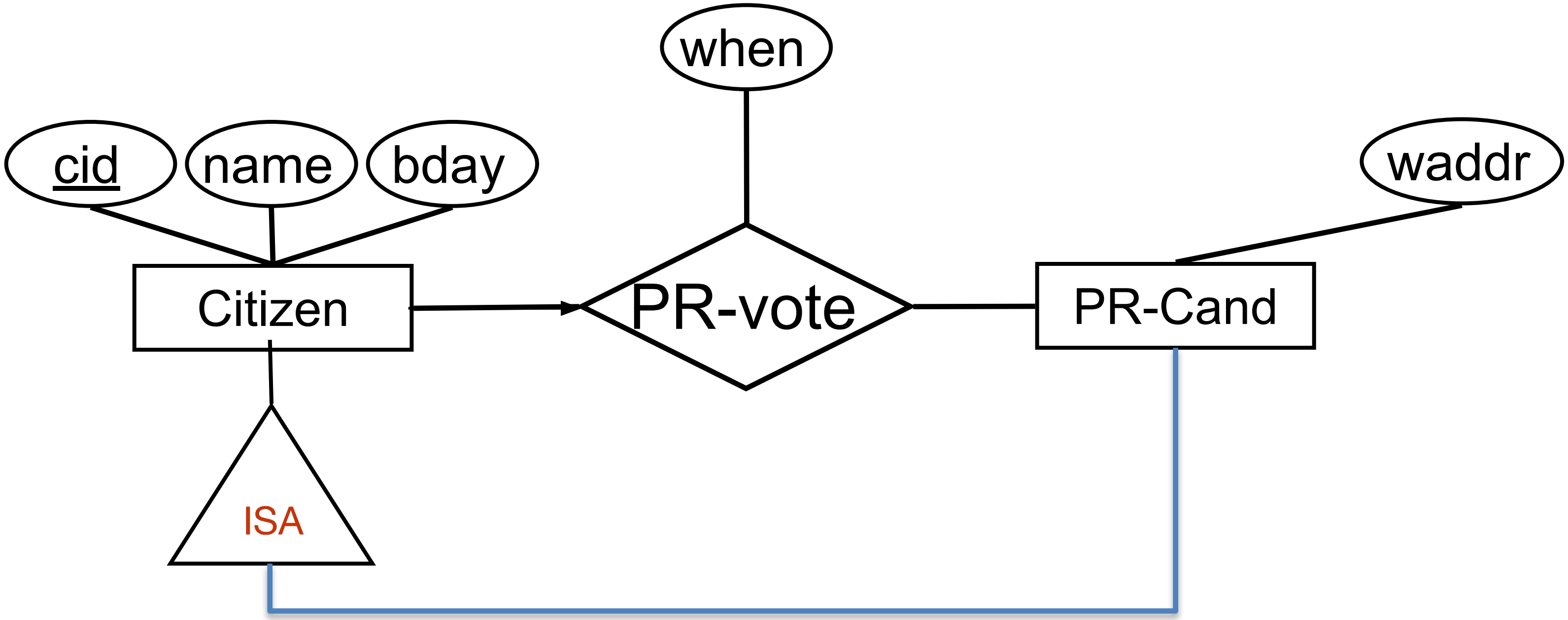
✓Generalize subclasses (bottom-up design)

Example

- Superclass: Citizen
- Subclasses: PR-Cand, President



Correcting Slide 44 using ISA



ISA ('is a') Hierarchies

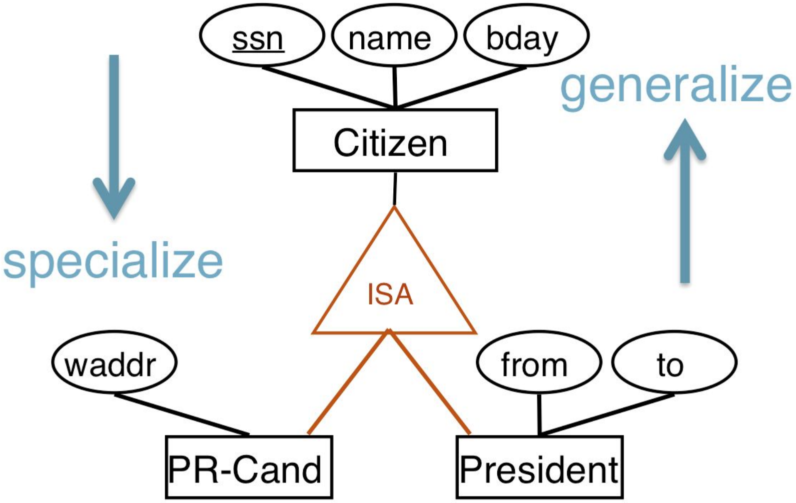
- Do not overuse them!
- Constraint types:
  - **Overlap**: Can >=2 subclasses contain the same entity?
    - Overlapping vs. Disjoint (default)
  - **Covering**: Do the entities in the subclasses include ALL the entities in the superclass?

    i.e., union of subclass entities = the set of superclass entities?
    - Total vs. Partial (default)

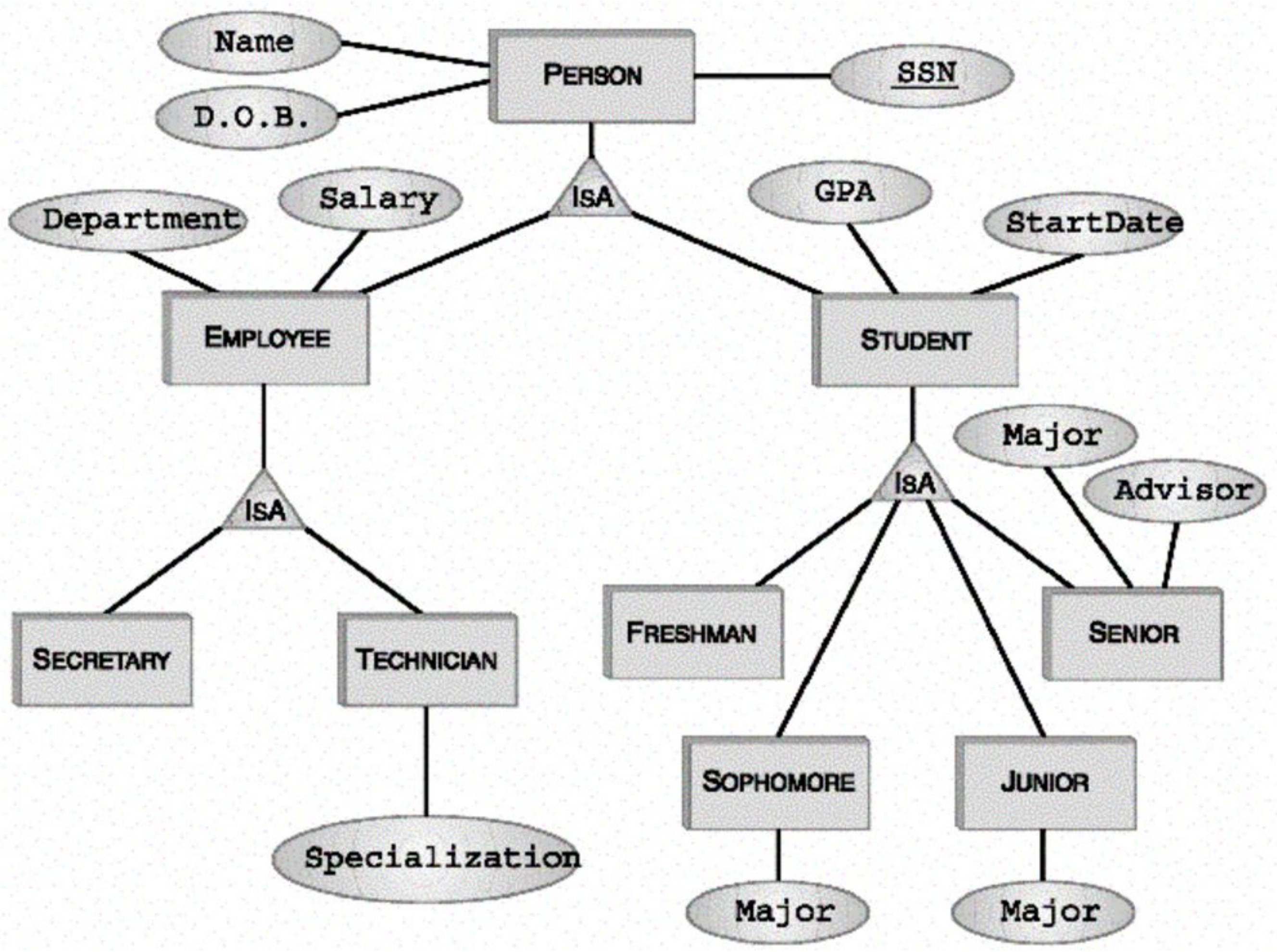
Example

Superclass: Citizen

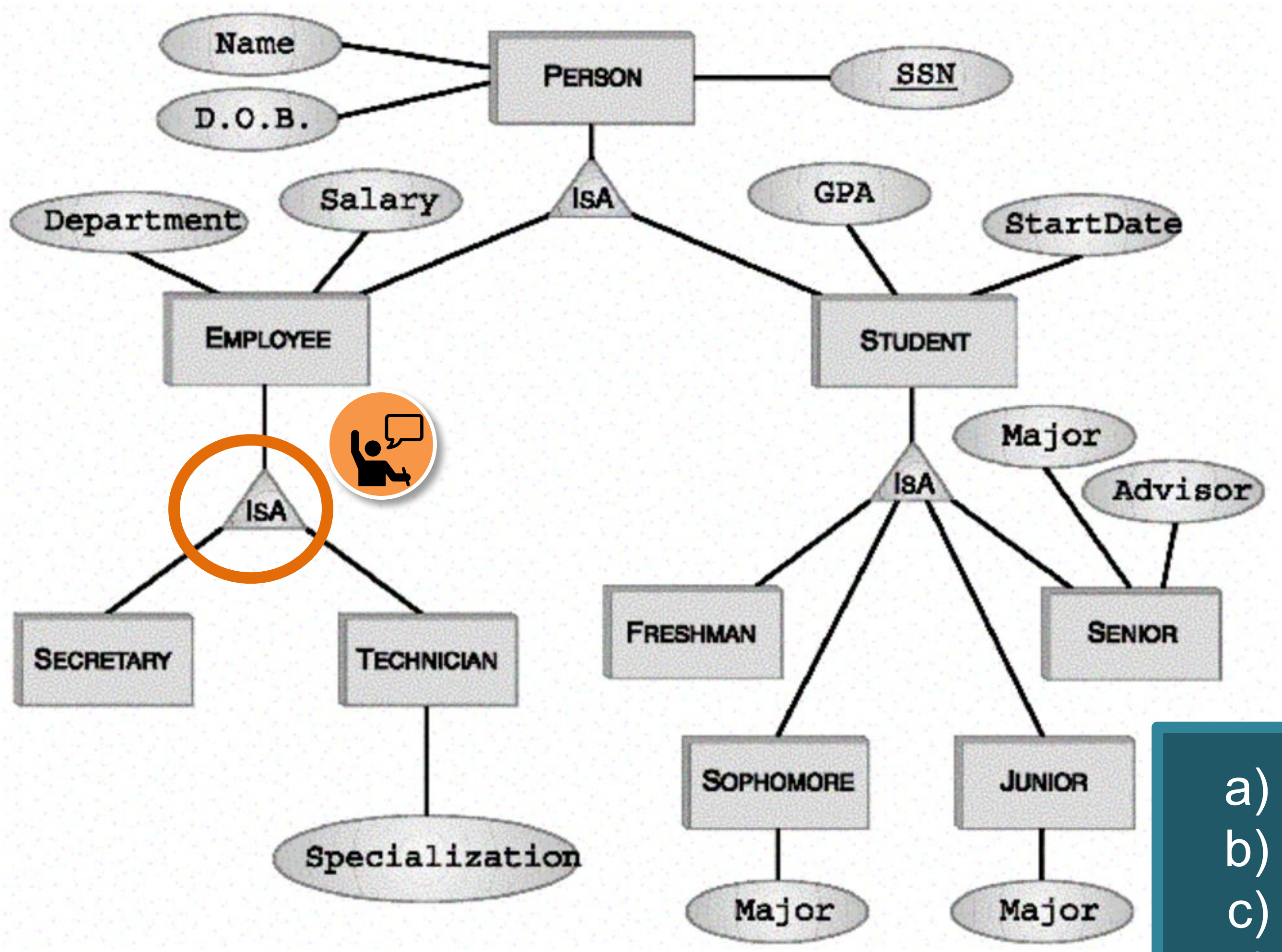
Subclasses: PR-Cand, President



ISA Overlap and Covering Constraints: Example

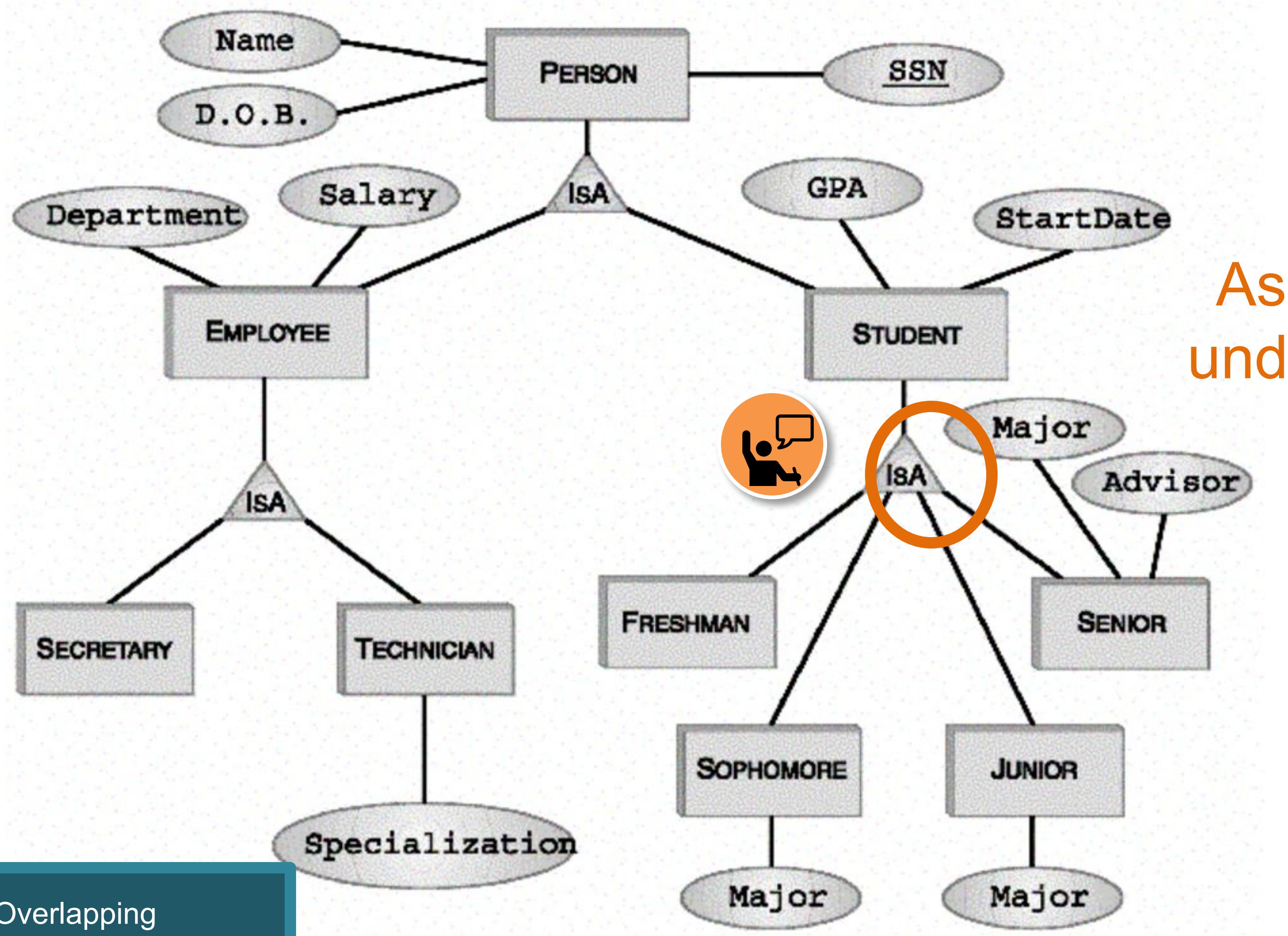


ISA Overlap and Covering Constraints: Example 1



a) Covering + Overlapping
b) Non-covering + Overlapping
c) Covering + Disjoint
d) Non-covering + Disjoint

ISA Overlap and Covering Constraints: Example 2

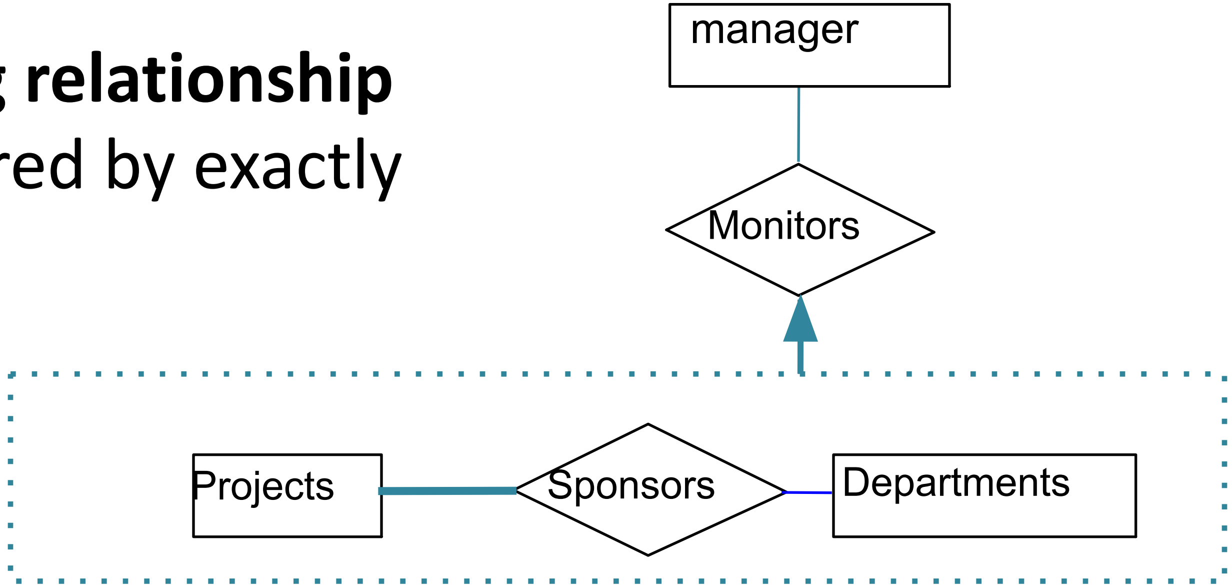


Assume undergrad

a) Covering + Overlapping
b) Non-Covering + overlapping
c) Covering + disjoint

Relationship with relationships

- Each **Project** must be sponsored by at least one **Department**
- Each **sponsoring relationship** must be monitored by exactly one **manager**

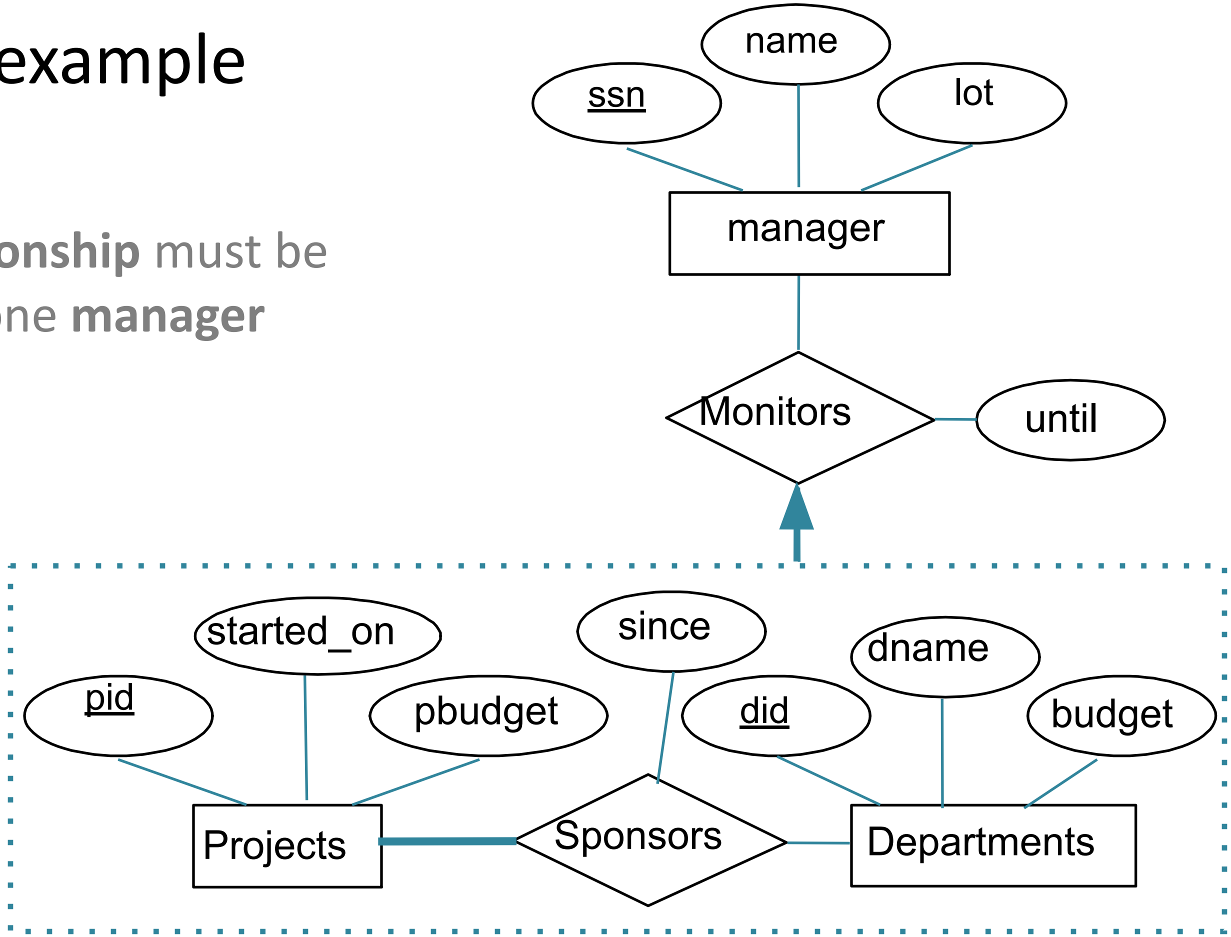


Treating a relationship as an entity for another relationship is called aggregation.

Aggregation

- More complete example

- Each **sponsoring relationship** must be monitored by exactly one **manager**



Choosing among available ER concepts

Conceptual Design Using the ER Model

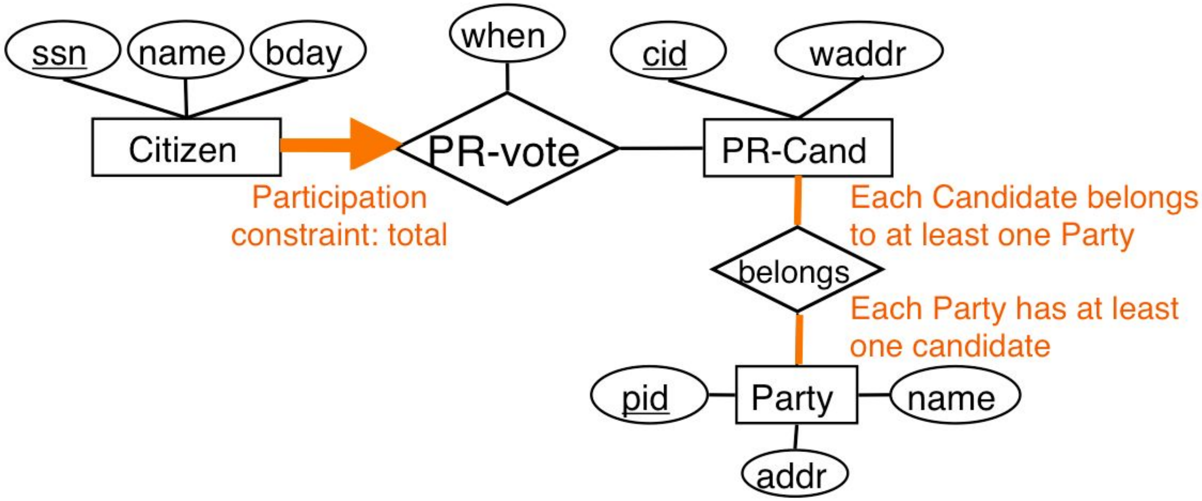
- Design choices:

  1. Model a concept as an **entity** or an **attribute**?
  2. Model a concept as an **entity** or a **relationship**?
  3. **Binary** or **ternary** relationship? **Aggregation**?

Entity vs. Attribute

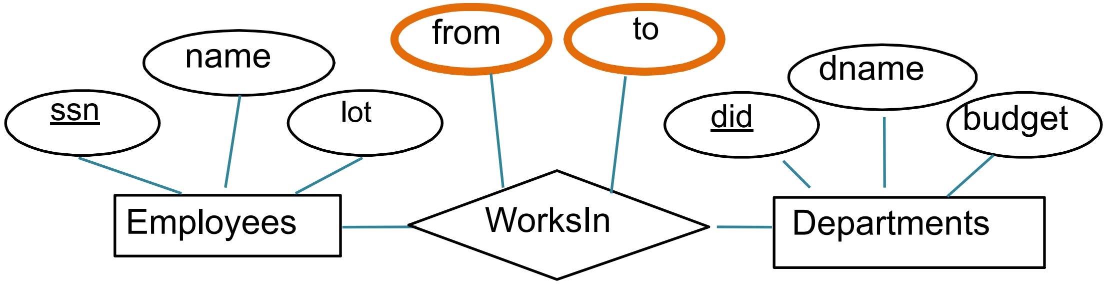
- Address of a Party: entity or attribute?
- Go with **entity** IF you want to:
  - Store several addresses per Party

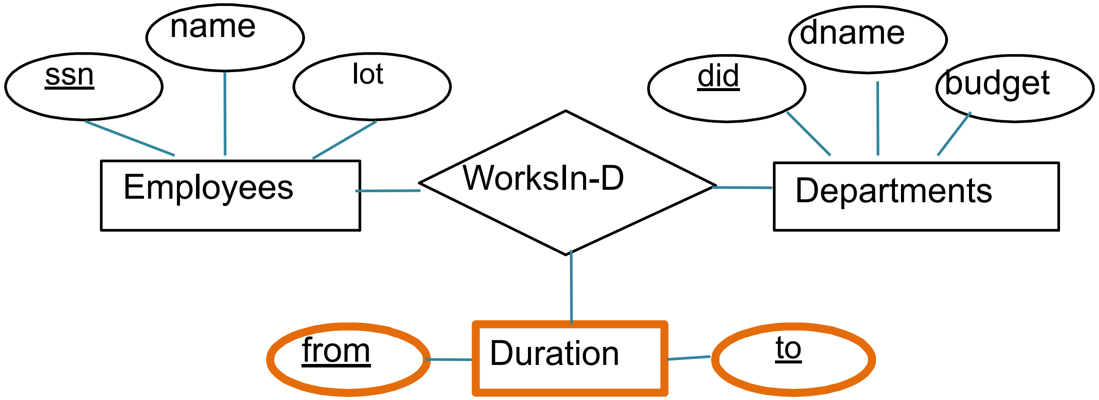
  or
  - Encode the structure of address (city, street, etc.)


ssn
name
bday
when
cid
waddr
Citizen
PR-vote
PR-Cand
Participation constraint: total
Each Candidate belongs to at least one Party
belongs
Each Party has at least one candidate
pid
Party
name
addr

Entity vs. Attribute

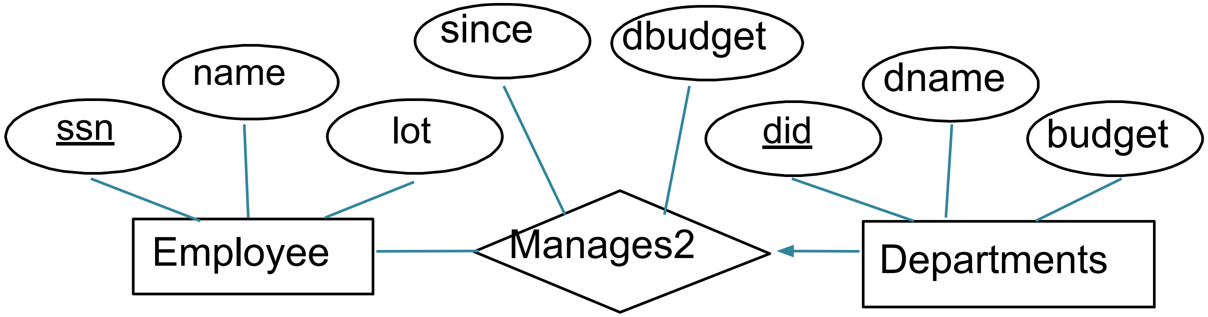
- Can employee work in a given dept. for two or more periods?





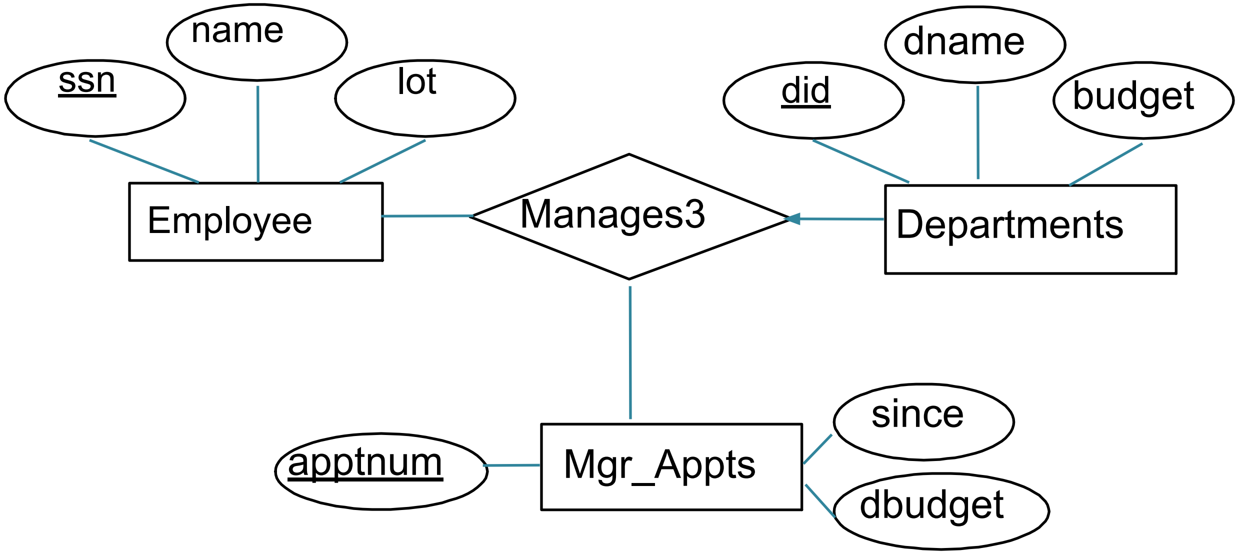
Entity vs. Relationship

- Manager (also an employee) gets a separate discretionary budget for each dept.



- What if we want Manager to get a discretionary budget that covers all managed depts?
  - Redundancy of dbudget, in each Manages2 relationship.
  - Misleading: suggests dbudget tied to relationship, not mngr

One Solution



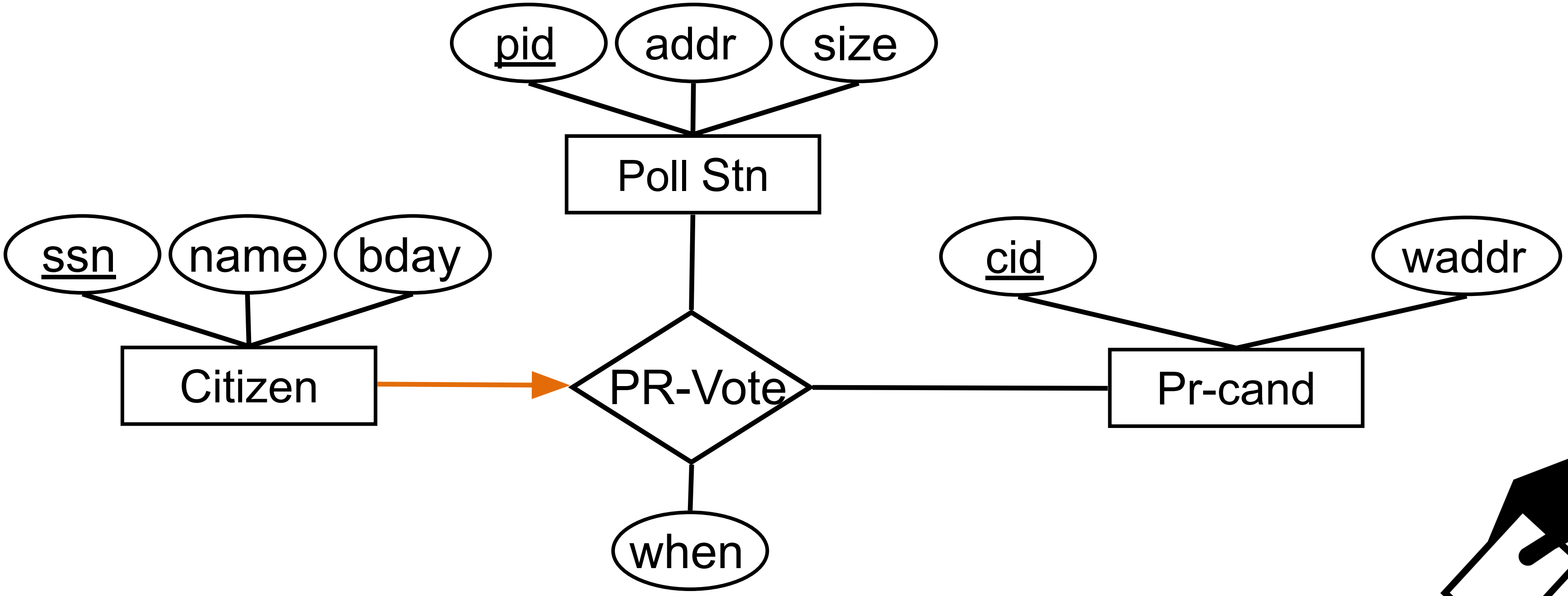
There is another way to do this.
Left as an exercise.

Think ISA hierarchy

- What if we want Manager to get a discretionary budget that covers all managed depts?

Participation constraint in a ternary relationship?

- A citizen votes at most once and at only one polling location (on a specific date)

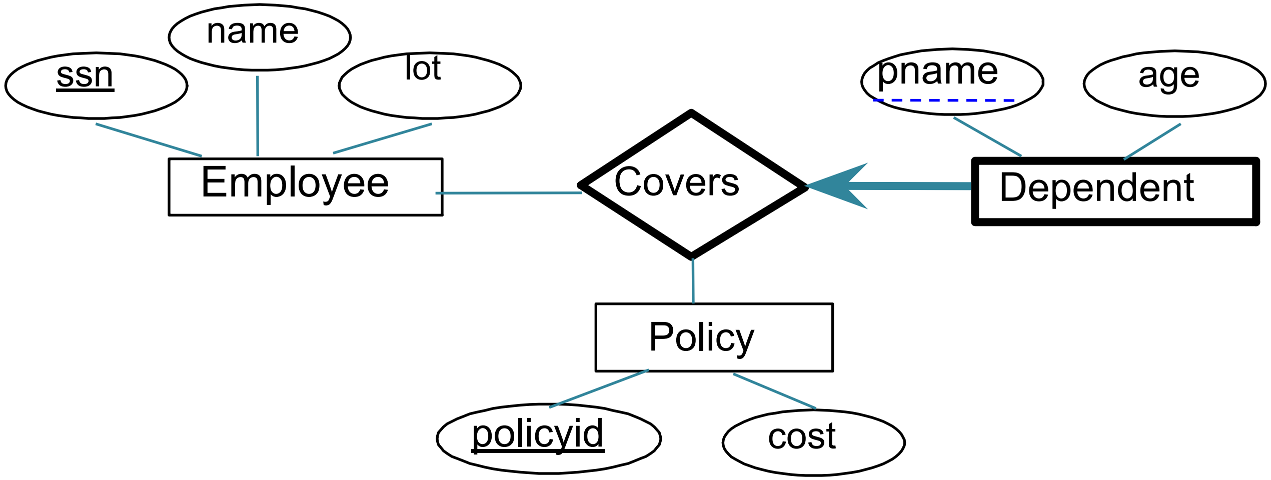


Try it out!

1. An insurance policy cannot be owned by multiple employees
2. Every policy must be owned by an employee
3. Dependents is a weak entity set, identified by the conjunction of pname with policyid.

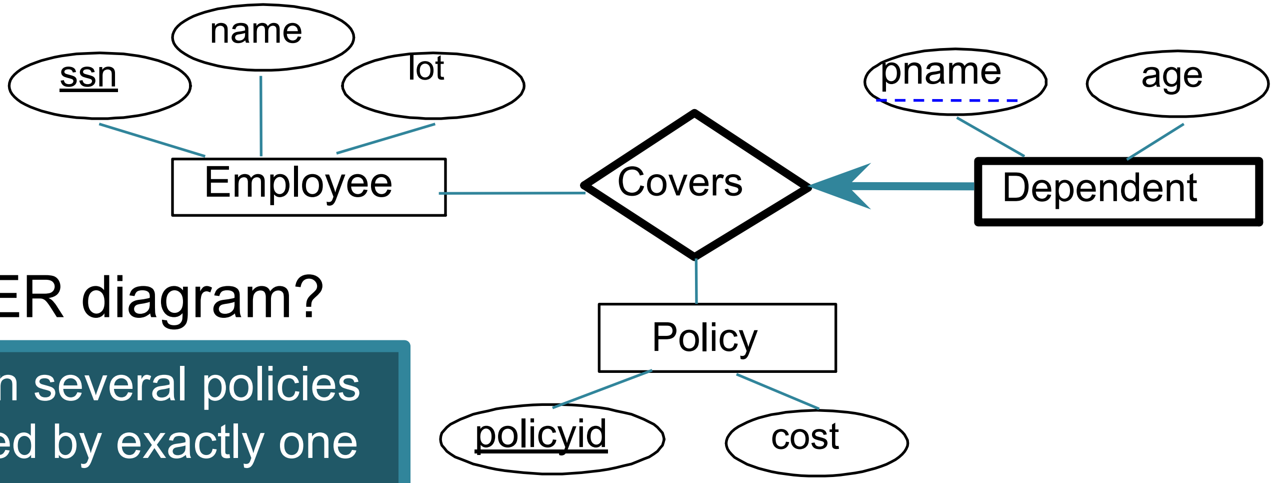
Incorrect Design Example

1. An insurance policy cannot be owned by multiple employees
2. Every policy must be owned by an employee
3. Dependents is a weak entity set, identified by the conjunction of pname with policyid.



Incorrect Design Example

1. An insurance policy cannot be owned by multiple employees
2. Every policy must be owned by an employee
3. Dependents is a weak entity set, identified by the conjunction of pname with policyid.

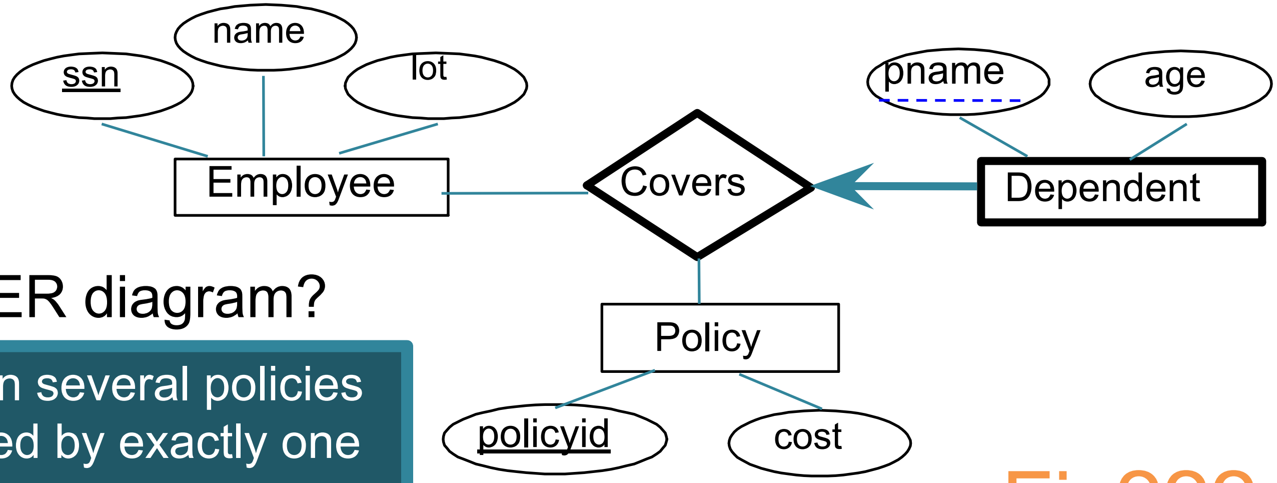


What is true in this ER diagram?

A) Employees can own several policies
B) Each policy is owned by exactly one employee
C) Each dependent can be covered by several policies

Incorrect Design Example

1. An insurance policy cannot be owned by multiple employees
2. Every policy must be owned by an employee
3. Dependents is a weak entity set, identified by the conjunction of pname with policyid.



What is true in this ER diagram?

A) Employees can own several policies

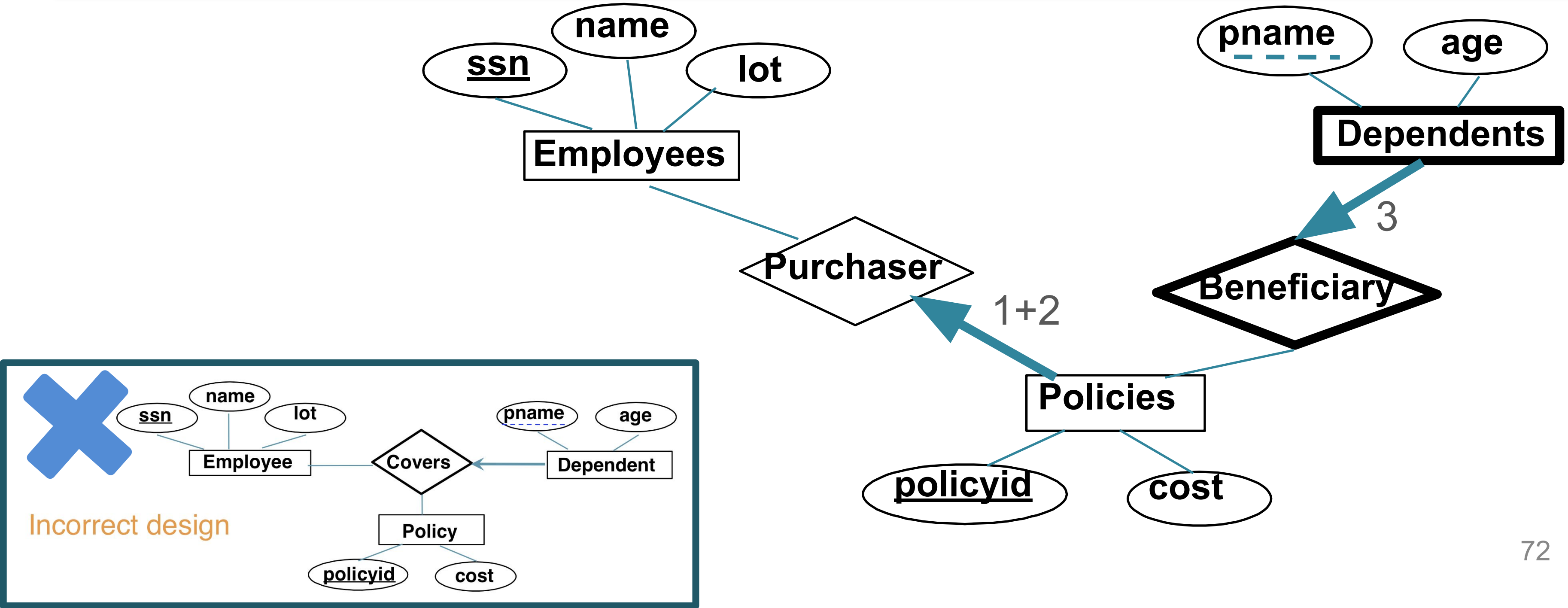
B) Each policy is owned by exactly one employee

C) Each dependent can be covered by several policies

Fix???

Better Design

1. An insurance policy cannot be owned by multiple employees
2. Every policy must be owned by an employee
3. Dependents is a weak entity set, identified by the conjunction of pname with policyid.



Summary of Conceptual Design

- High-level description of data to be stored
- ER model popular for conceptual design
  - Constructs are expressive and natural
  - Basic constructs: **entities, relationships, and attributes** (of entities and relationships).
  - Additional constructs: **weak entities, ISA, aggregation**.
  - Integrity constraints: **key constraints and participation constraints.**
- Note: There are many variations on ER model
- ER designing is subjective!