Data Modeling

R&G Chapter 2

(slides adapted from content by J.Gehrke, J.Shanmugasundaram, and/or C.Koch)

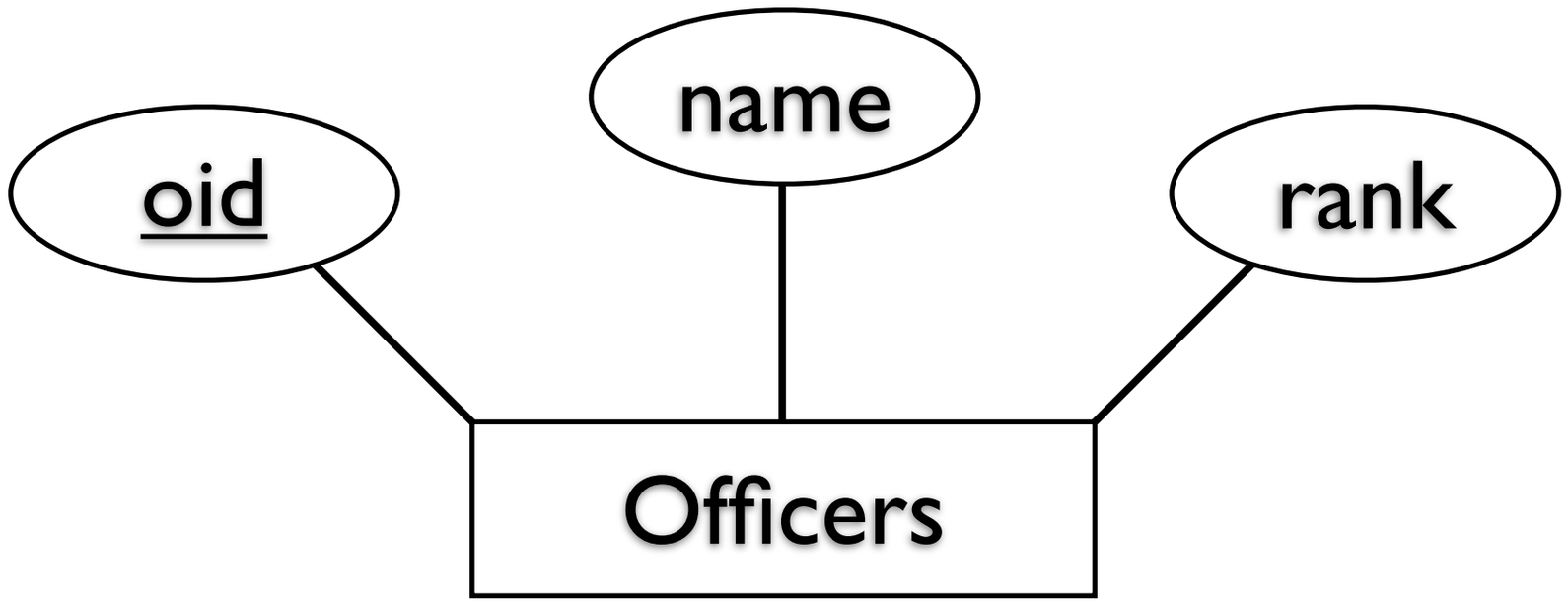
Conceptual Design

- We have discussed schemas in the abstract.
- How do you build a schema for your data?
  - What concepts appear in your data?
  - How are data values related?
  - What information about each data value do you need?

Entity-Relation Model

- A pictorial representation of a schema
  - Enumerates all entities in the schema
  - Shows how entities are related
  - Shows what is stored for each entity
  - Shows restrictions (integrity constraints)

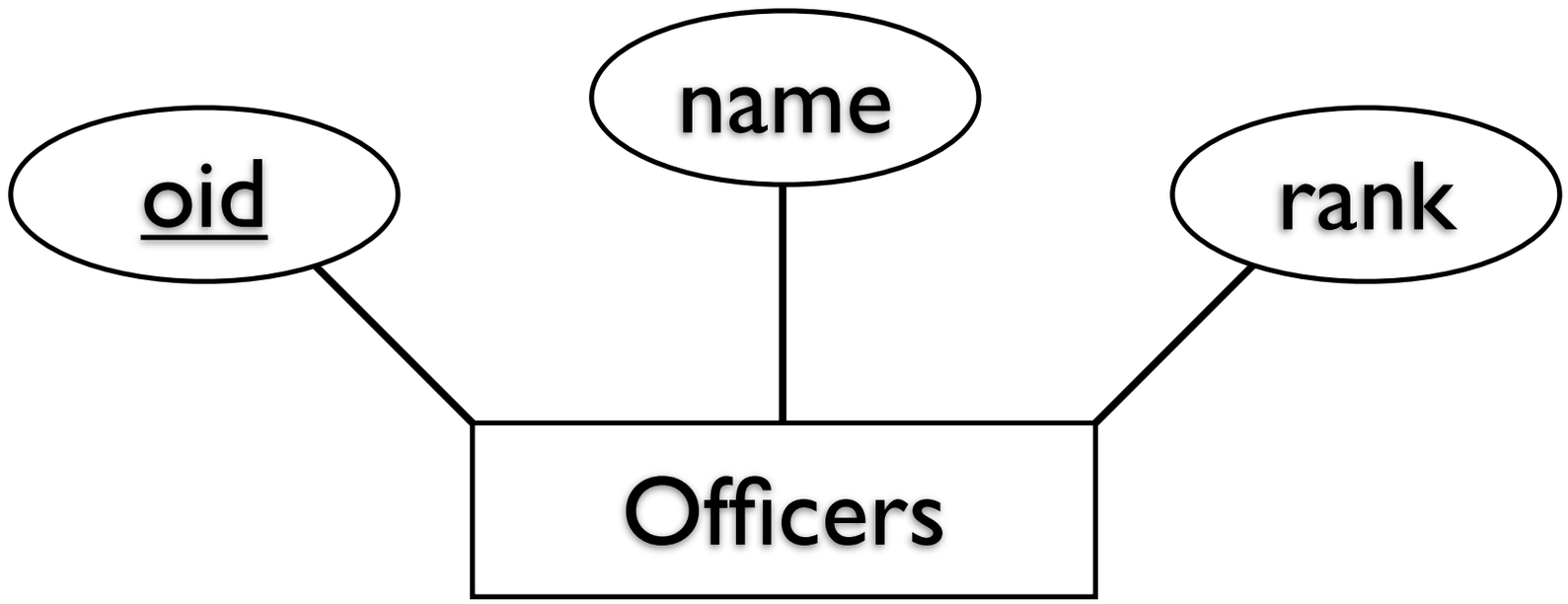
ER Model Basics



Entity: A real-world object distinguishable from other objects. (e.g., a single Officer)

An entity is described through a set of **attributes**

ER Model Basics

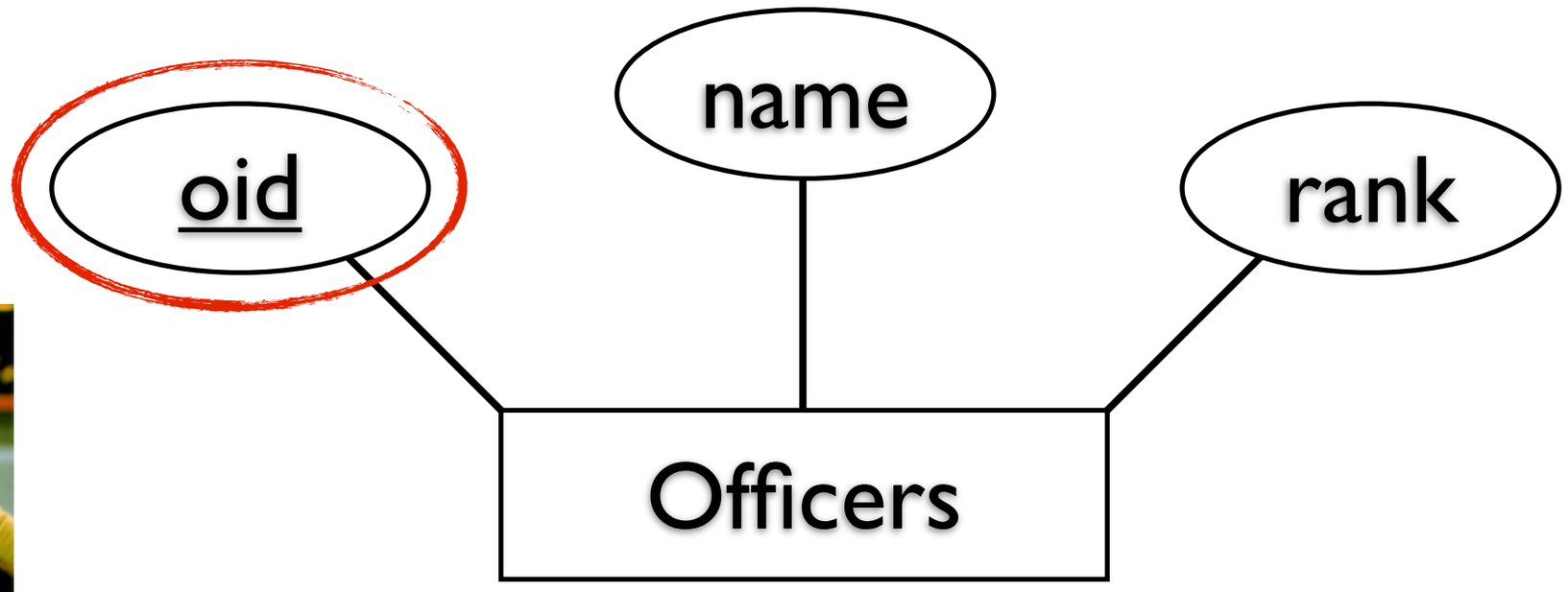


Entity Set: A collection of similar entities. (e.g., all Officers)

Entities in an entity set have the same set of attributes

Each attribute has a **domain** (e.g., integers, strings)

ER Model Basics

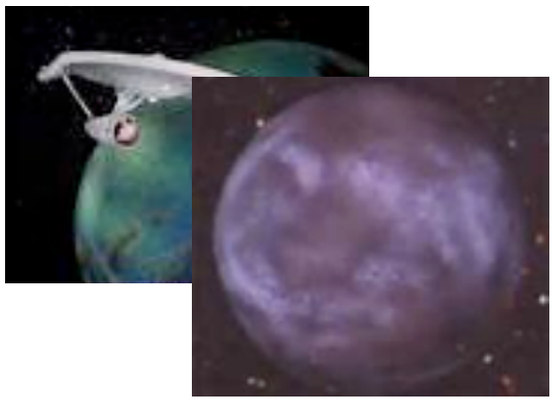


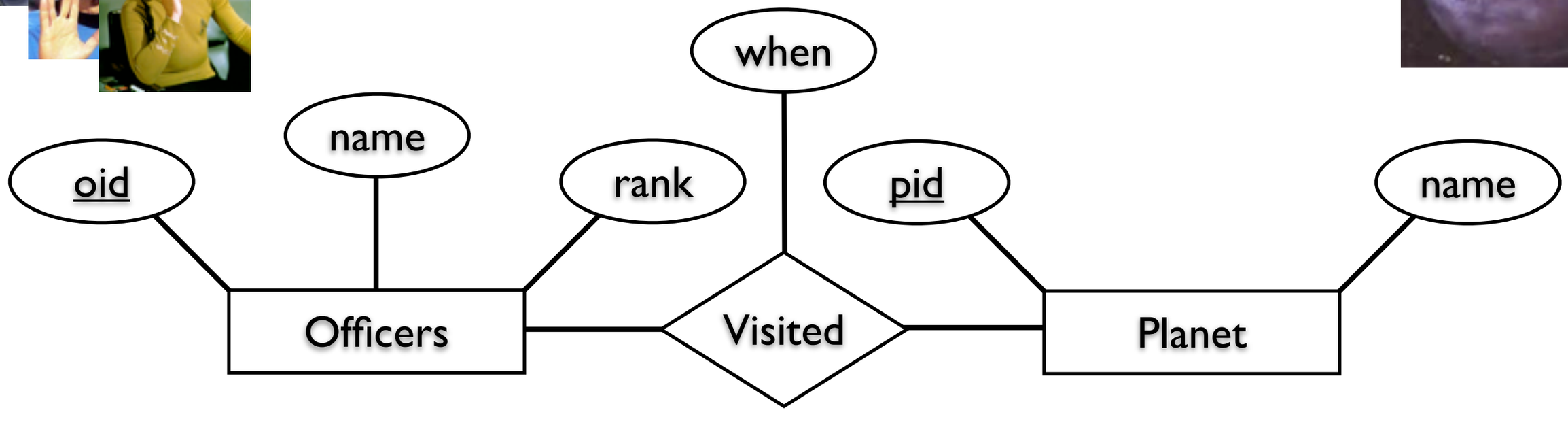
Entity sets must have a **key**, an attribute (or combination of attributes) guaranteed to be unique for every entity in the set.

- Officer ID for officers
- Ship ID for ships
- UBIT for UB students
- Course Code+Semester for courses

Keys are underlined in ER Diagrams

ER Model Basics

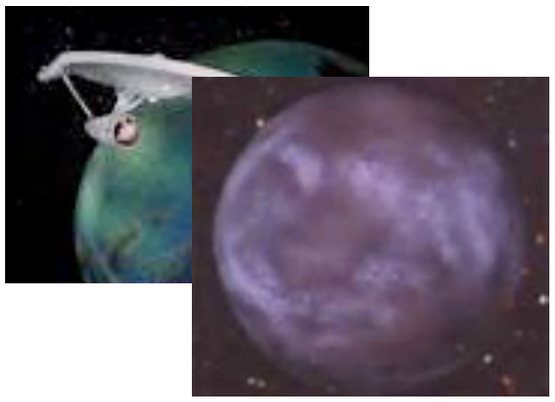


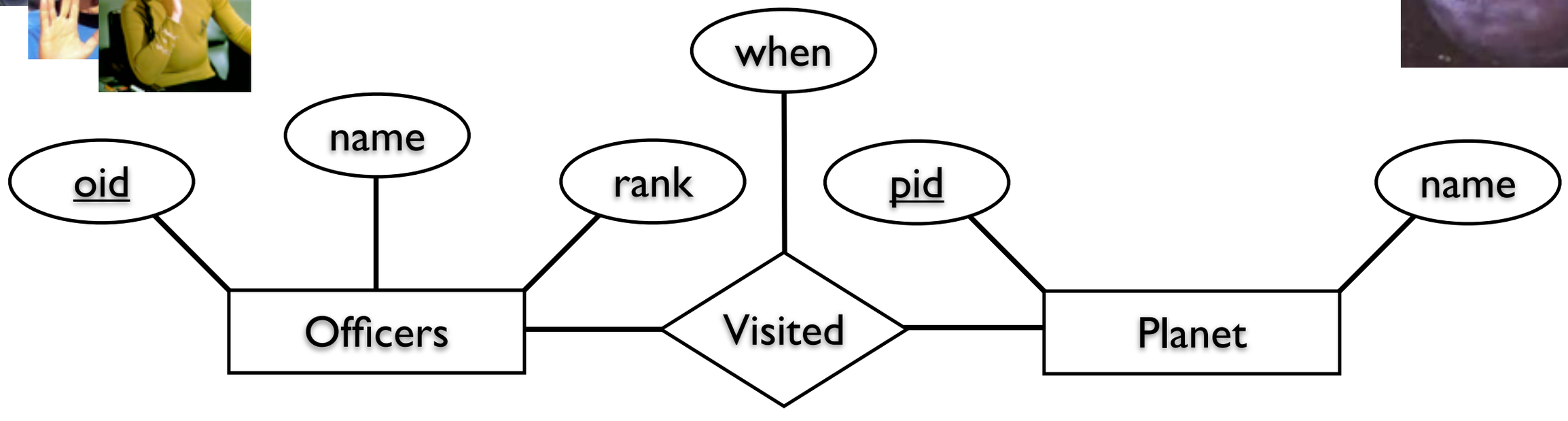


Relationship: Associations between 2 or more entities.
Relationship Set: A collection of similar relationships.
(an n-ary relationship set relates Entity sets E_1-E_n)

Relationships may have their own attributes.

ER Model Basics

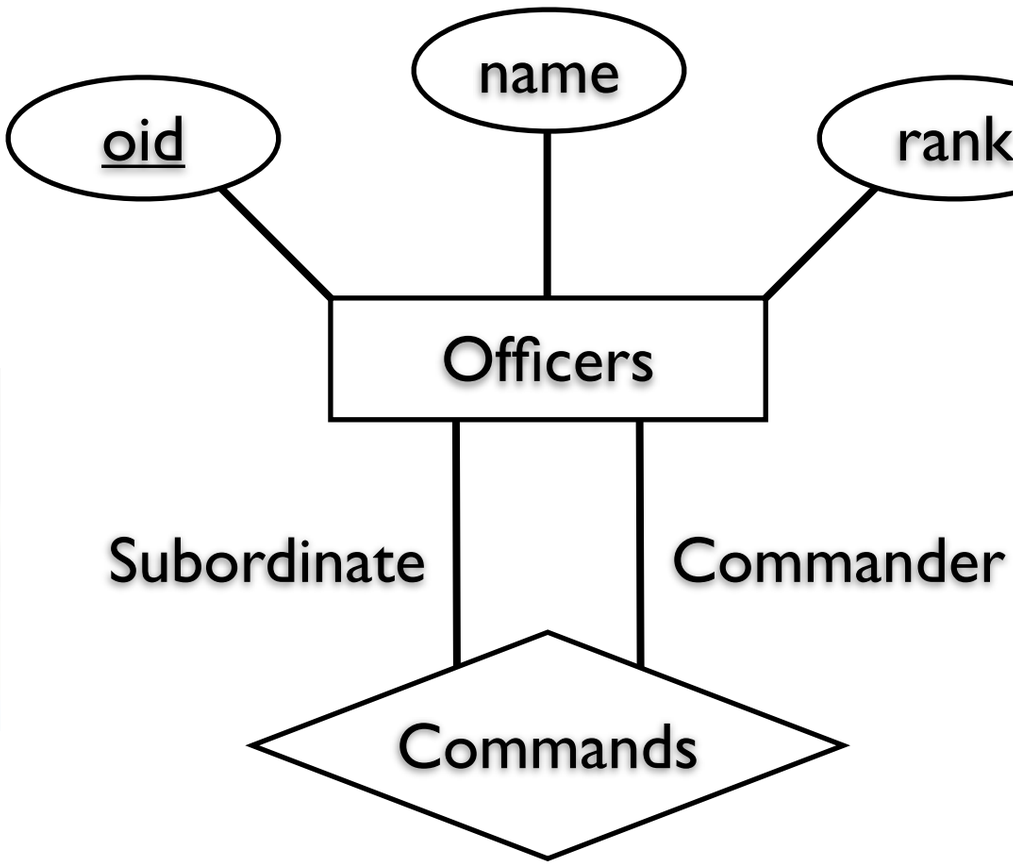




Relationship: Associations between 2 or more entities.
Relationship Set: A collection of similar relationships.
(an n-ary relationship set relates Entity sets E_1-E_n)

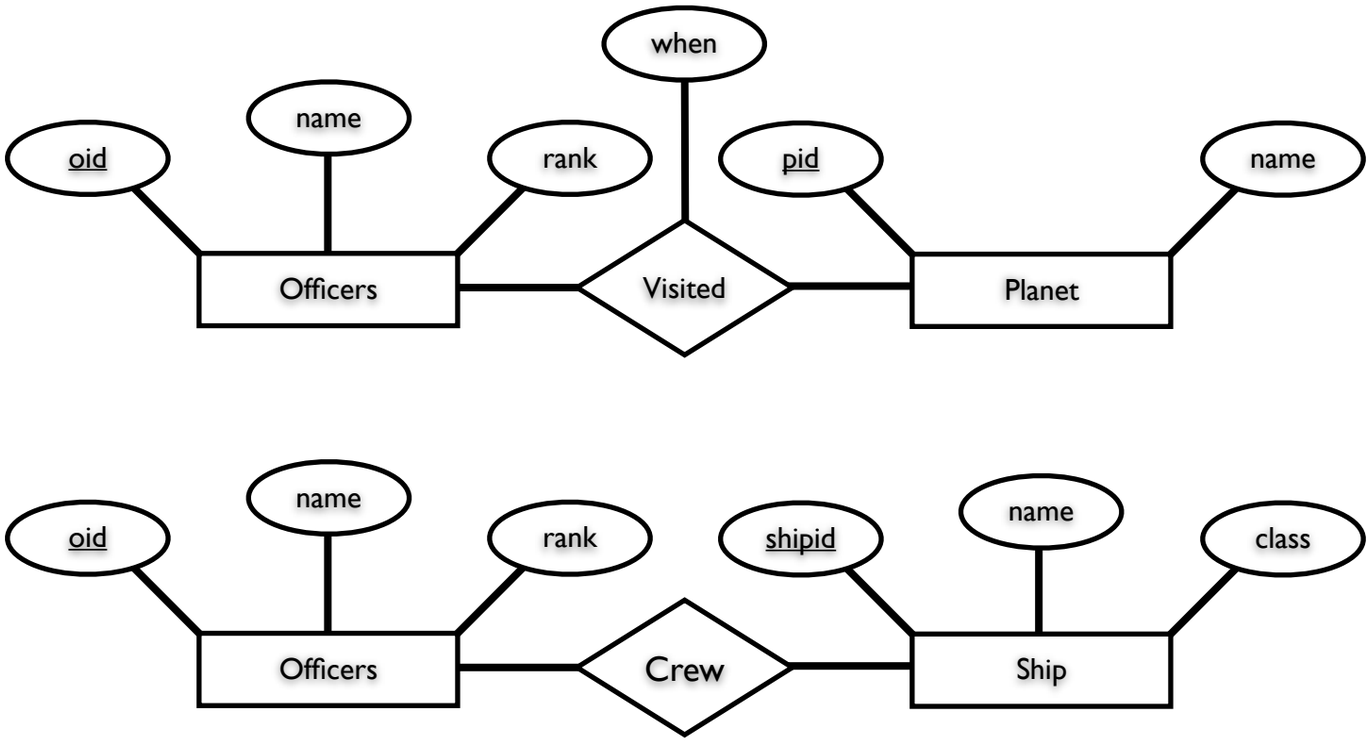
Relationships may have their own attributes.

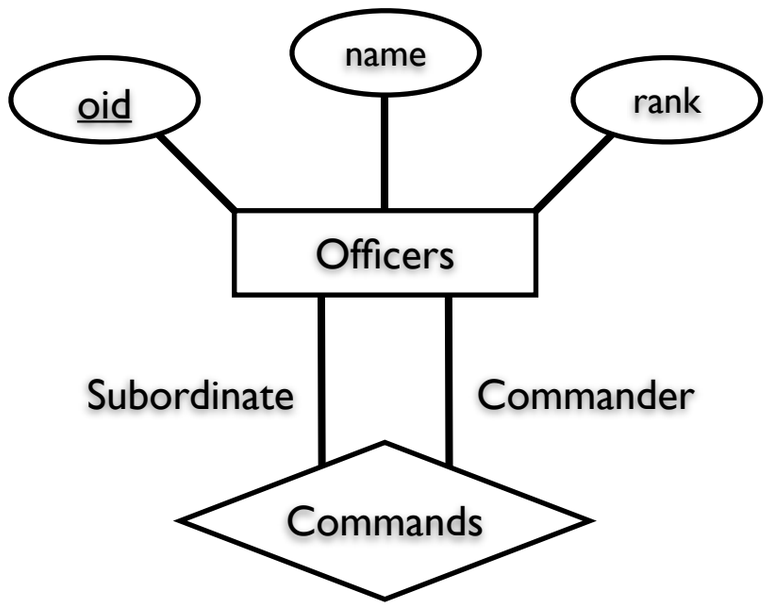
ER Model Basics



There can be relationships between entities in the same entity sets

Key Constraints

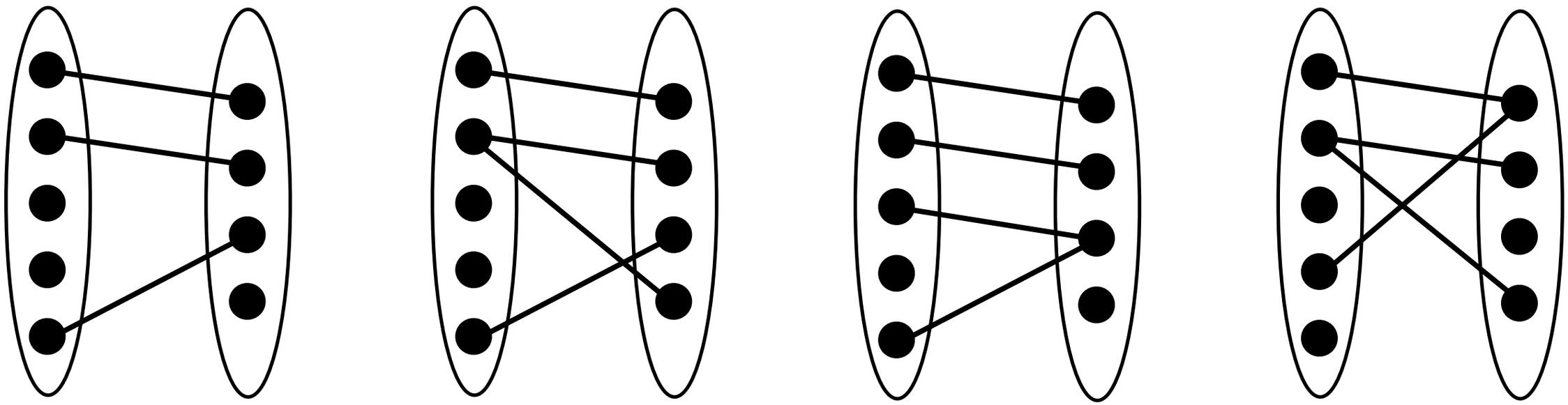




Consider these relationships

- One ship can have many crew, but each crew member has only one ship
- Each officer has one commander, but officers might have many subordinates
- Each planets may have been visited by many officers, and each officer may have visited many planets

Key Constraints

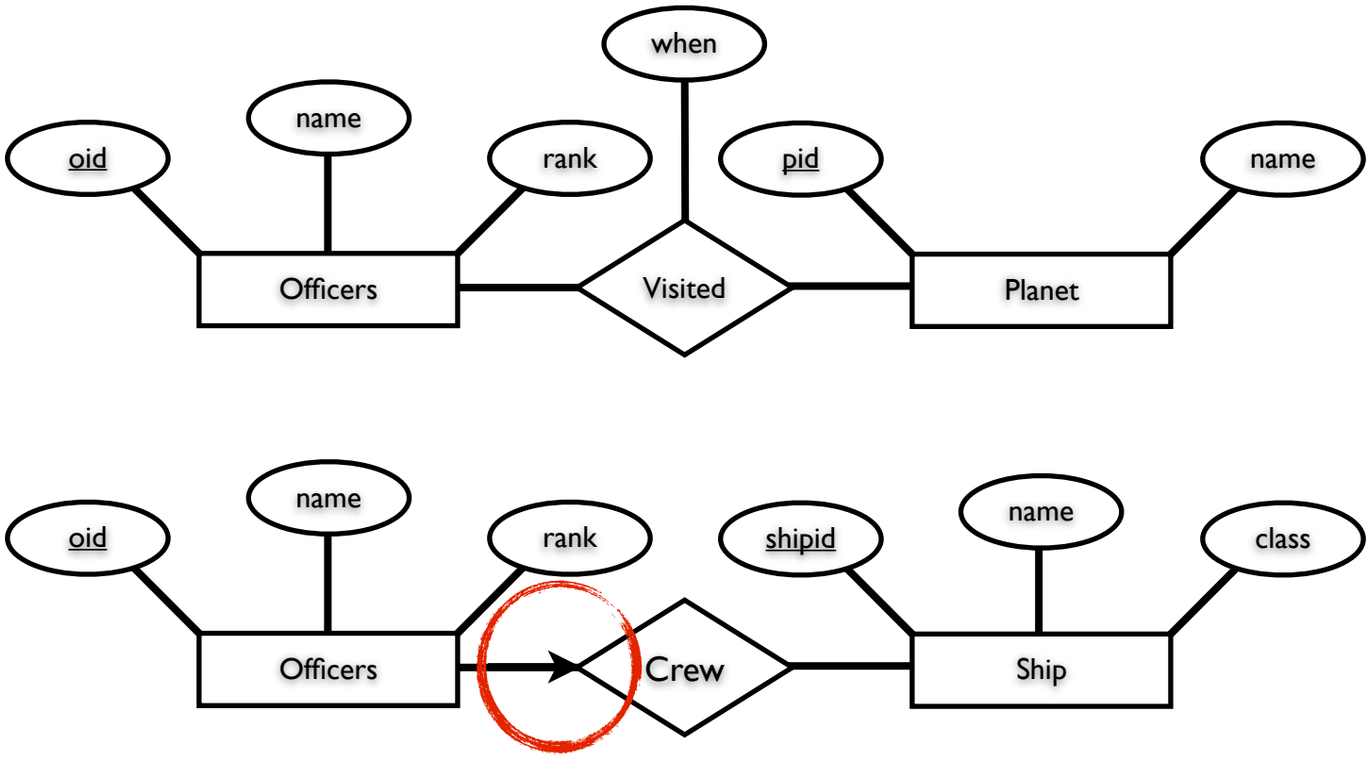


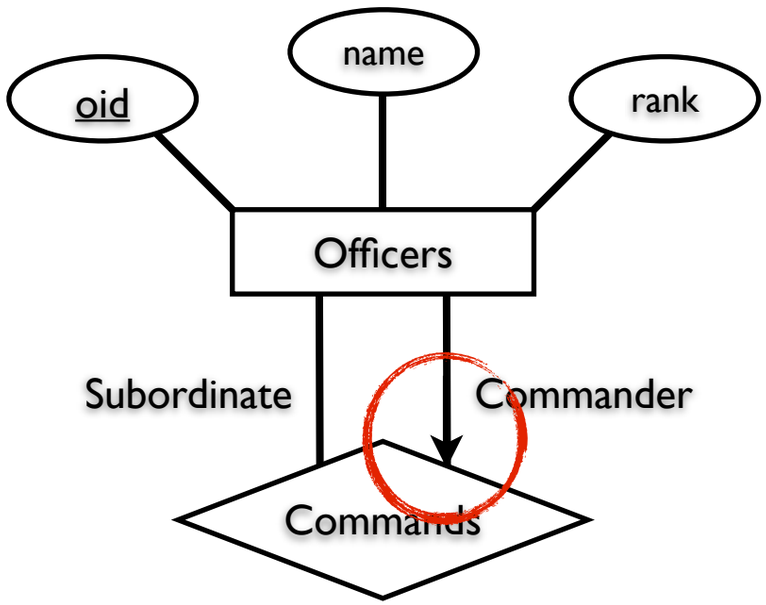
1-to-1 1-to-Many Many-to-1 Many-to-Many

Consider these relationships

- One ship can have many crew, but each crew member has only one ship
- Each officer has one commander, but officers might have many subordinates
- Each planets may have been visited by many officers, and each officer may have visited many planets

Key Constraints

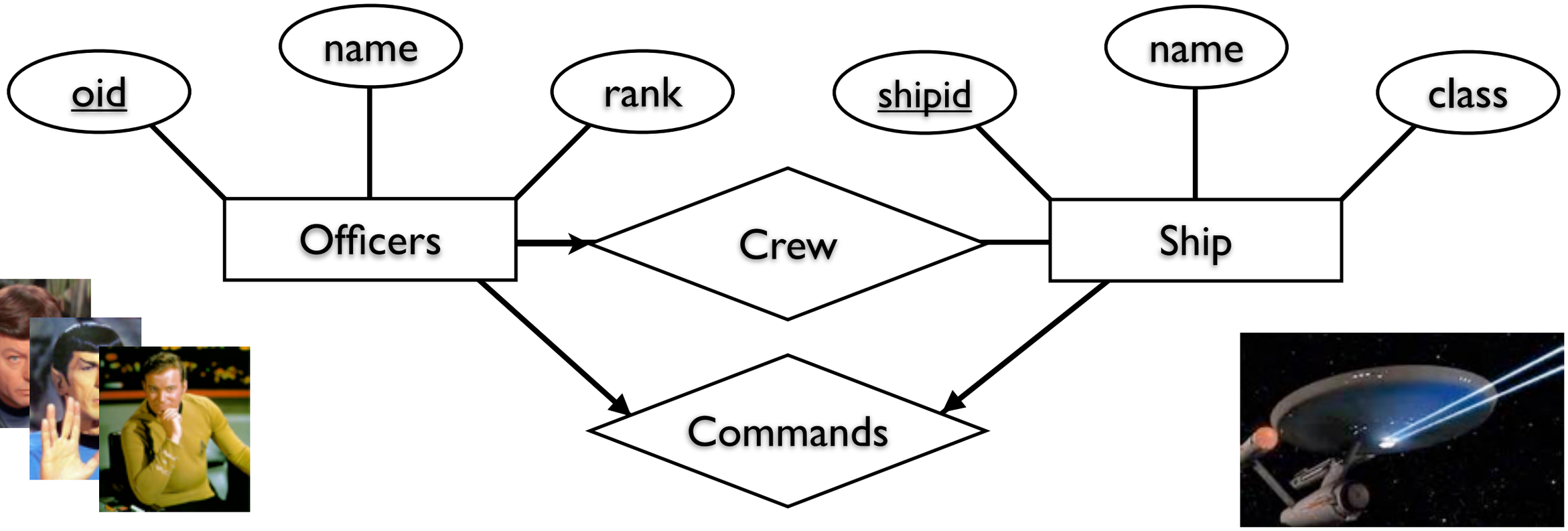




Key constraints identify entities that participate in **at most one** relationship in a relationship set

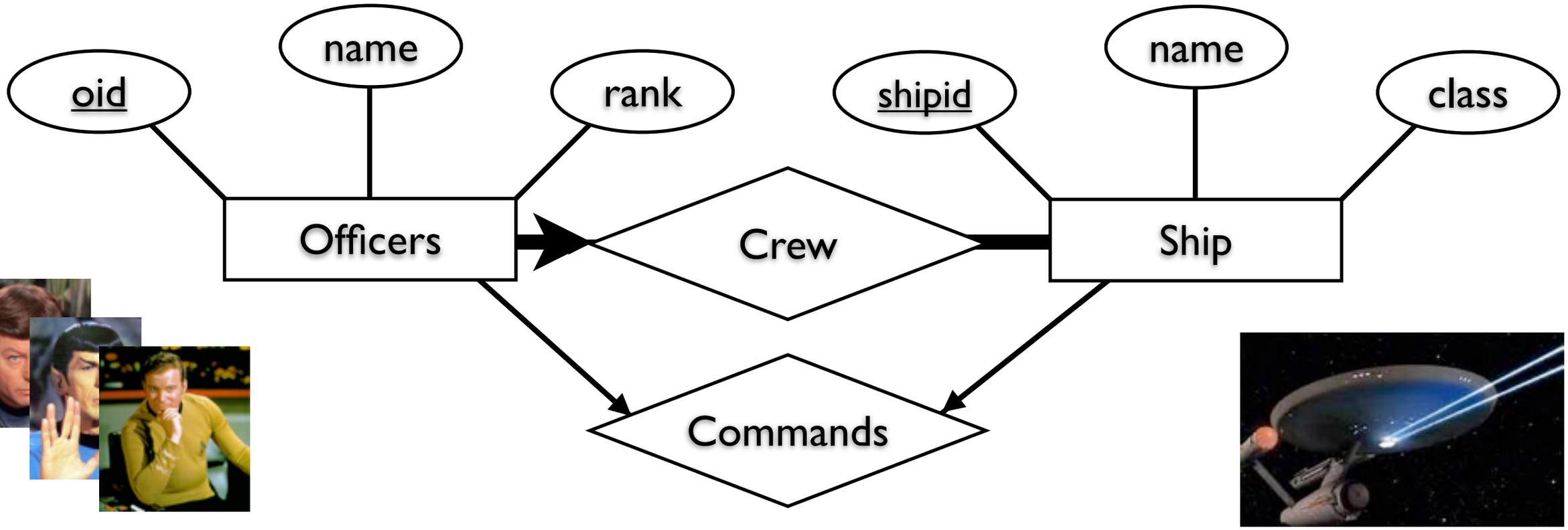
We denote key-constraints with an arrow

Participation Constraints



Participation constraints require participation in a relationship (and are denoted as bold lines)

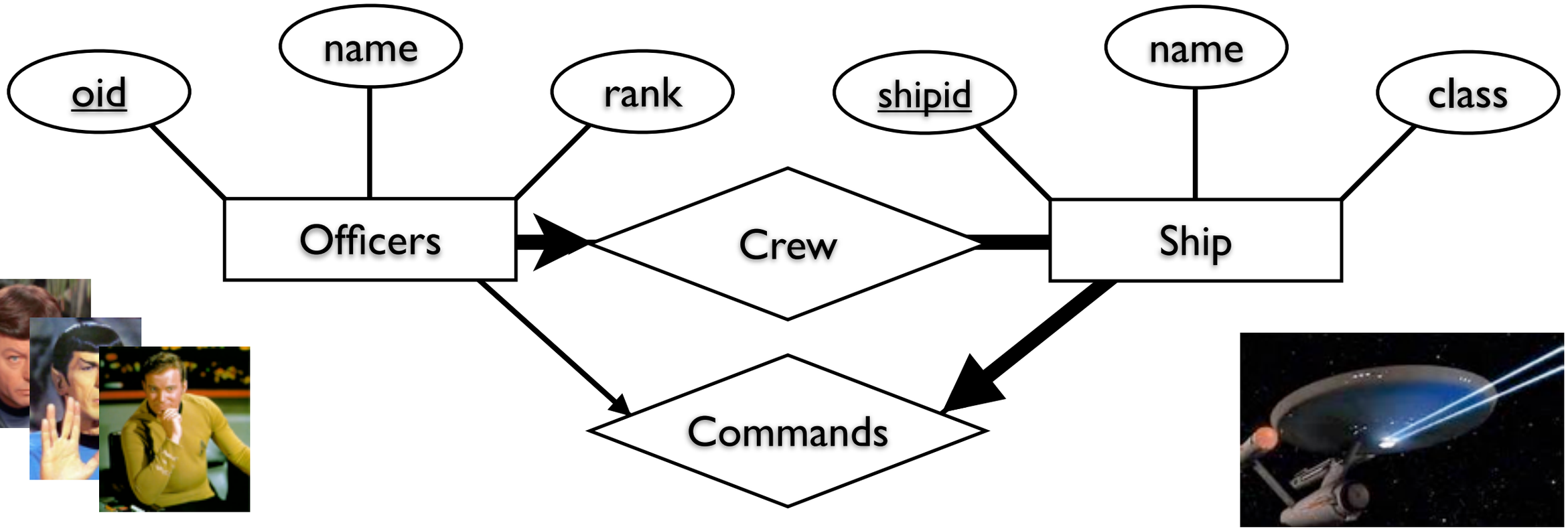
Participation Constraints



Every Ship **must** have crew, and every officer **must** crew a ship.

Participation constraints require participation in a relationship (and are denoted as bold lines)

Participation Constraints



Every Ship **must** have crew, and every officer **must** crew a ship.
Every Ship **must** have a commander.

Participation constraints require participation in a relationship
(and are denoted as bold lines)

Weak Entities

A weak entity can be identified uniquely only relative to the primary key of another (owner) entity.

The weak entity **must** participate in a one-to-many relationship (one owner, many weak entities)

ISA ('is a') Hierarchies

ISA Hierarchies define entity inheritance

If we declare A **ISA** B, then every A is also considered to be a B

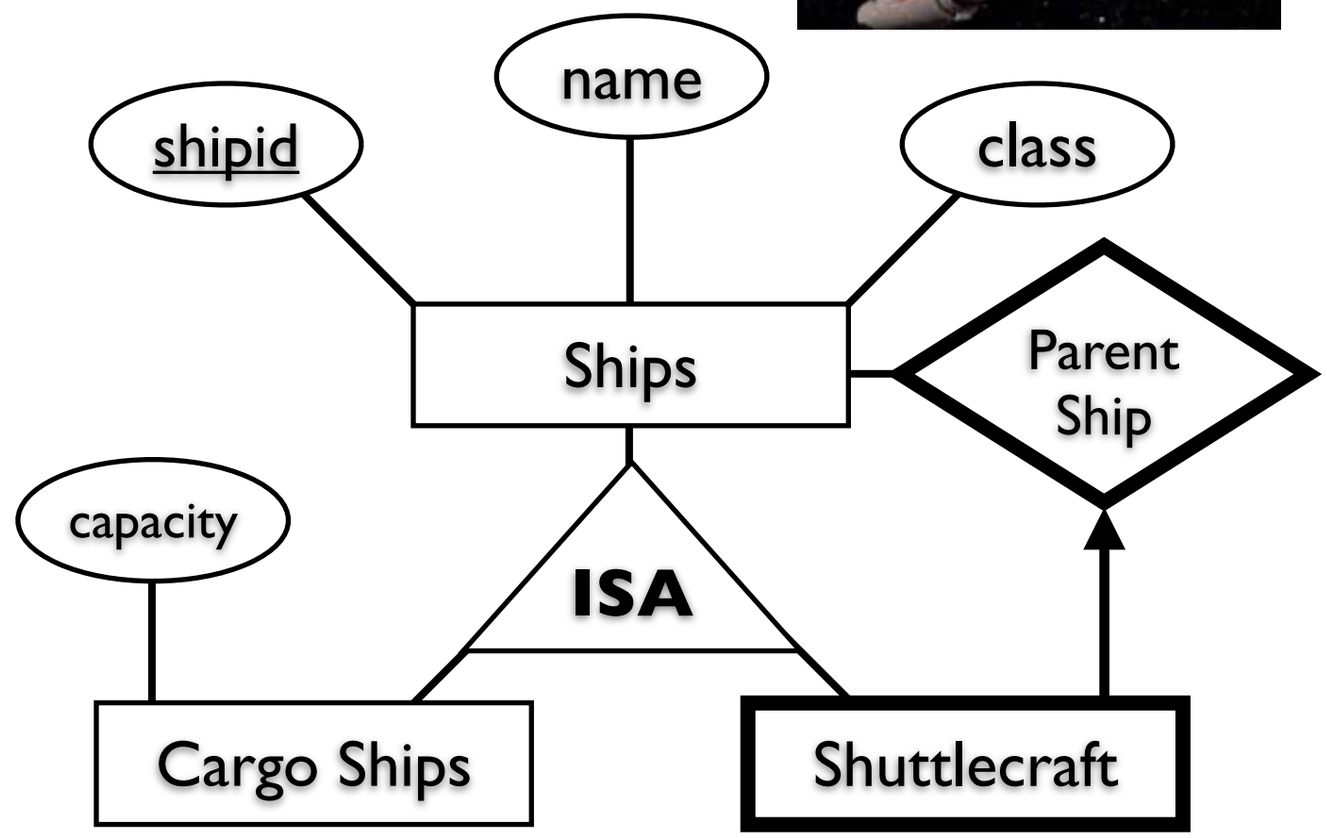
Overlap constraints: Can a ship be a cargo ship and a shuttlecraft?

Covering constraints: Does every ship have to be a cargo ship or a shuttlecraft?

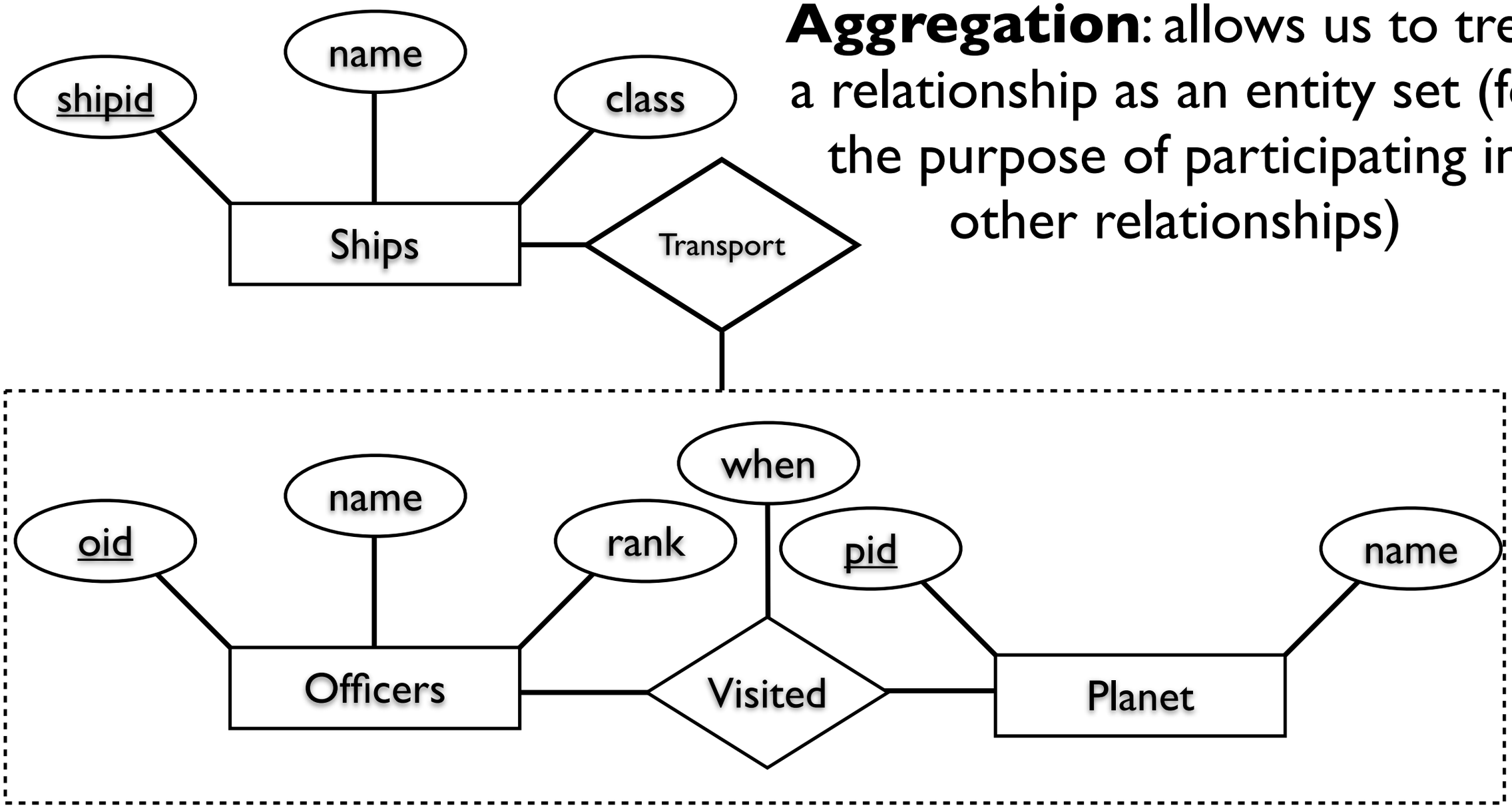
Reasons for using ISA:

Adding descriptive attributes specific to a subclass (cargo ship capacity)

Identifying entities in a specific type of relationship (shuttlecraft of a big ship)



Aggregation



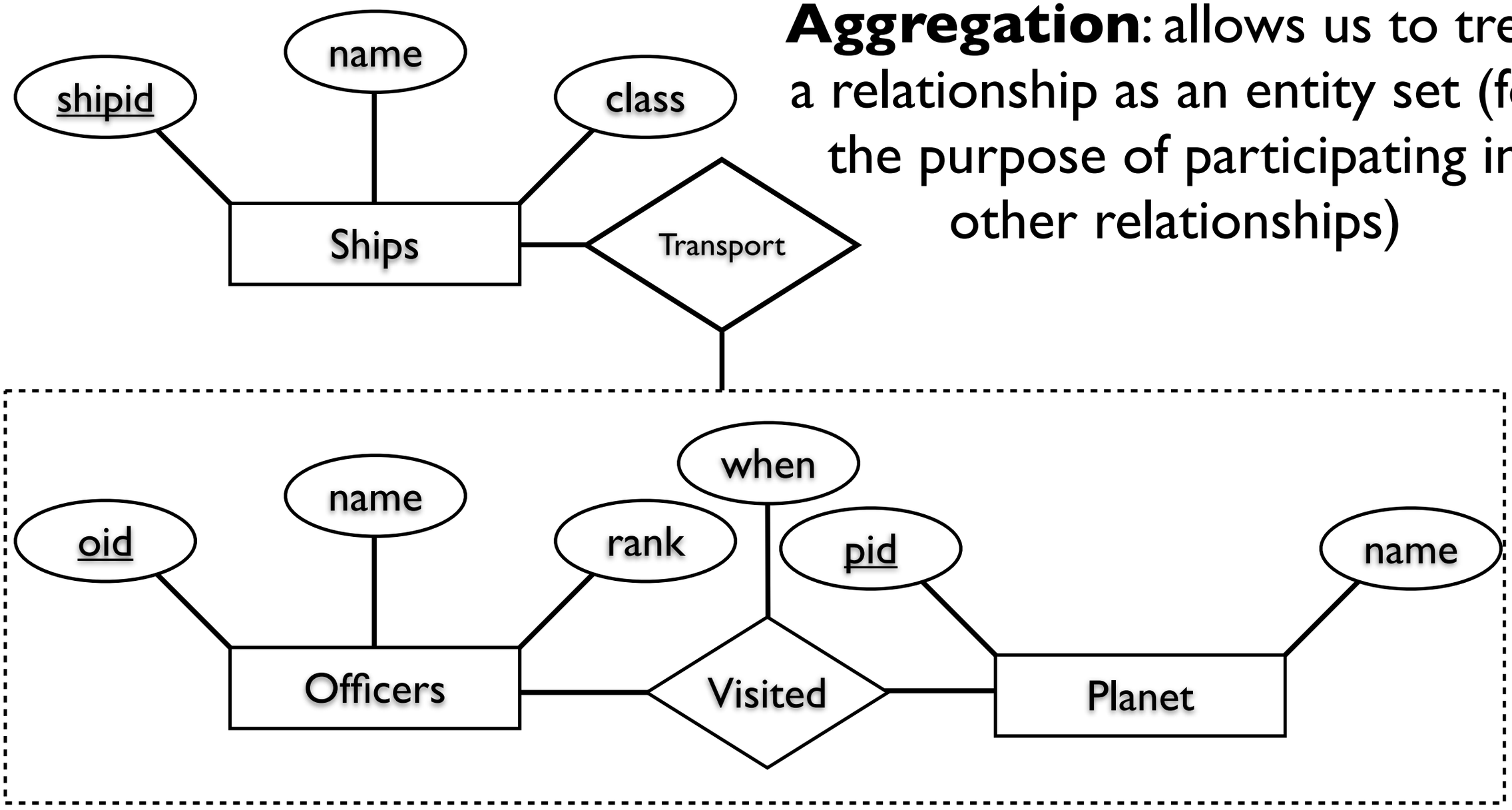
Aggregation: allows us to treat a relationship as an entity set (for the purpose of participating in other relationships)

Contrast with ternary relationship

Thursday, February 14, 13

Multiple ships may be used to transport an officer to a particular planet. Different ships may be used for different visits. Monitors is its own distinct relationship.

Aggregation



Aggregation: allows us to treat a relationship as an entity set (for the purpose of participating in other relationships)

Contrast with ternary relationship

Thursday, February 14, 13

Multiple ships may be used to transport an officer to a particular planet. Different ships may be used for different visits. Monitors is its own distinct relationship.

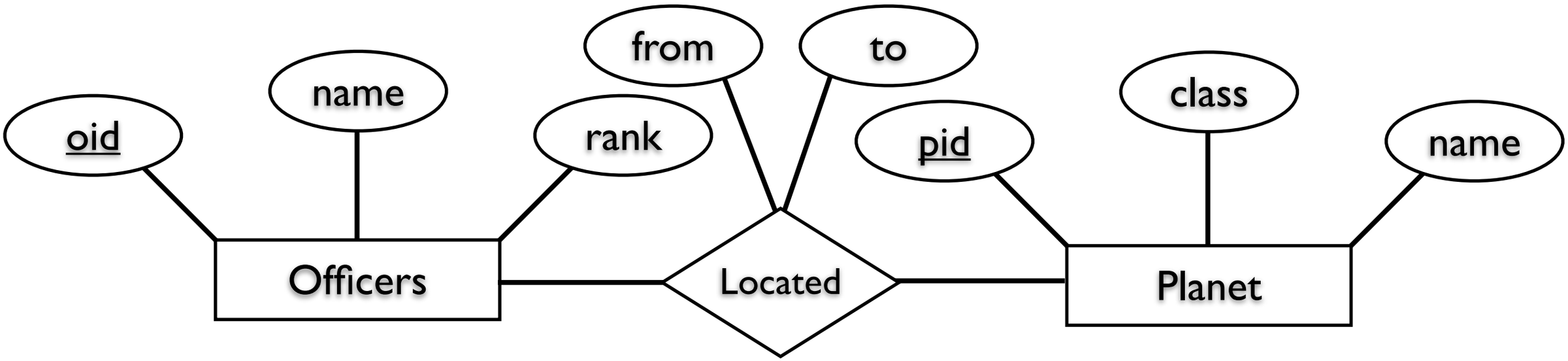
Conceptual Design in ER

- Design choices
  - Should a concept be modeled as an entity or an attribute of another entity?
  - Should a concept be modeled as an entity or a relationship between entities?
  - What kind of relationship: Binary, Ternary, N-ary, Aggregation?
- Constraints
  - A lot of data semantics can (and should) be captured.
  - Not all constraints are expressible in ER diagrams.

Entity vs Attribute

- Expressing the **Location** of an **Officer**
  - **Option 1**: An attribute of **Officers**
  - **Option 2**: A **Planets** entity set and a relationship set **Location**
- Which we use depends on the semantics of the data.
  - Can an Officer have multiple locations? (e.g., transporter accidents, time travel, etc…)
    - Attributes are single-valued, model Planets as entities.
  - Are the details of locations relevant to queries? (i.e., Find all officers on a Class-M planet).
    - Attributes are atomic, model Planets as entities.

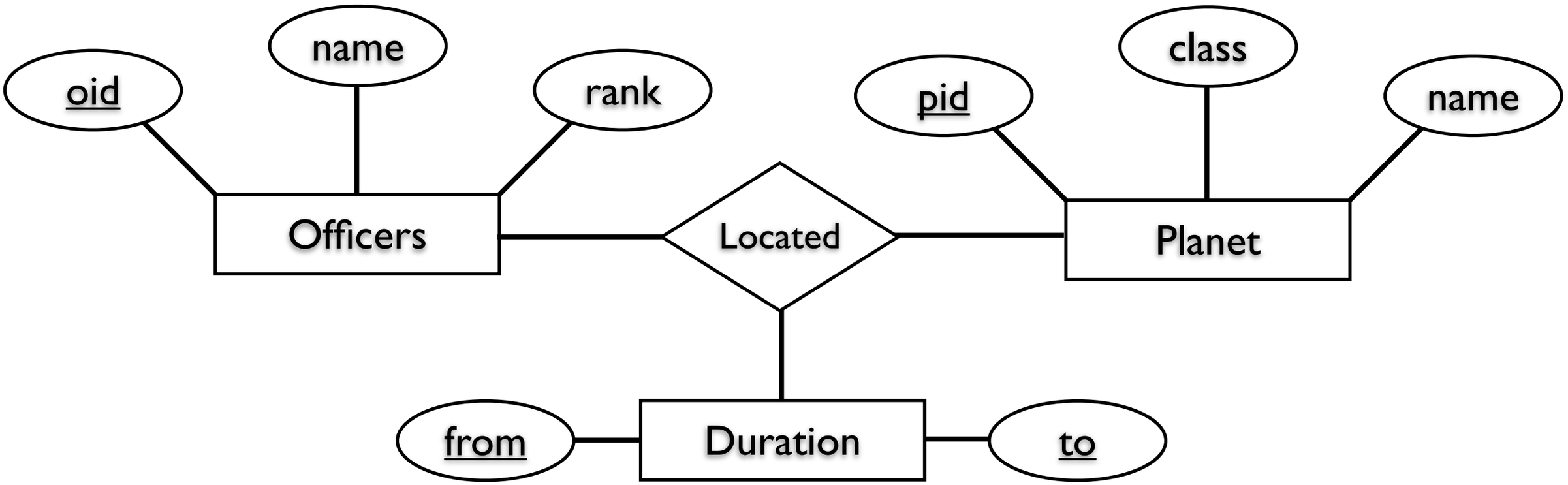
Entity vs Attribute



Problem: Can only have one location for each officer (no time ranges)

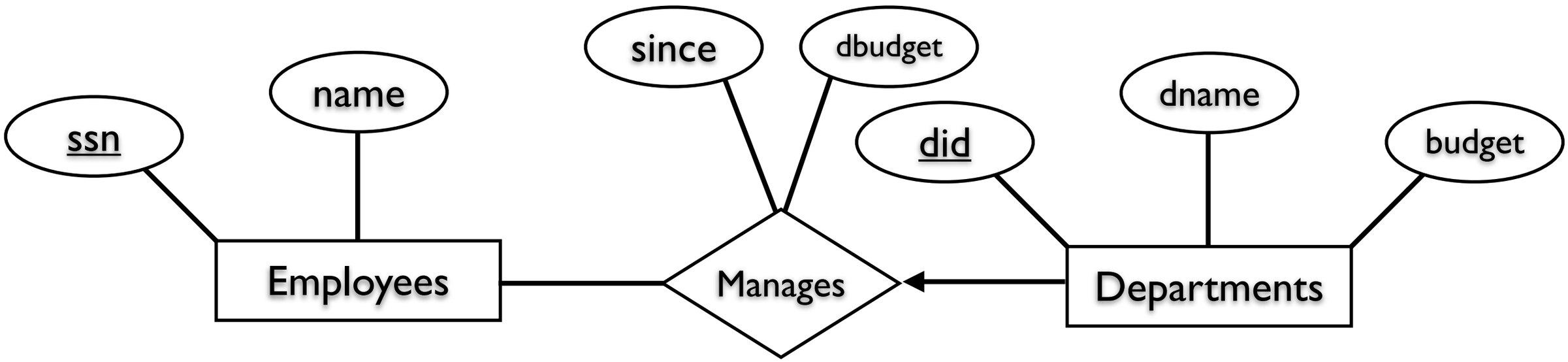
We want to encode multiple instances of the descriptive attributes of the relationship instance

Entity vs Attribute



Solution: Add a duration entity and make location a ternary relationship

Group Work

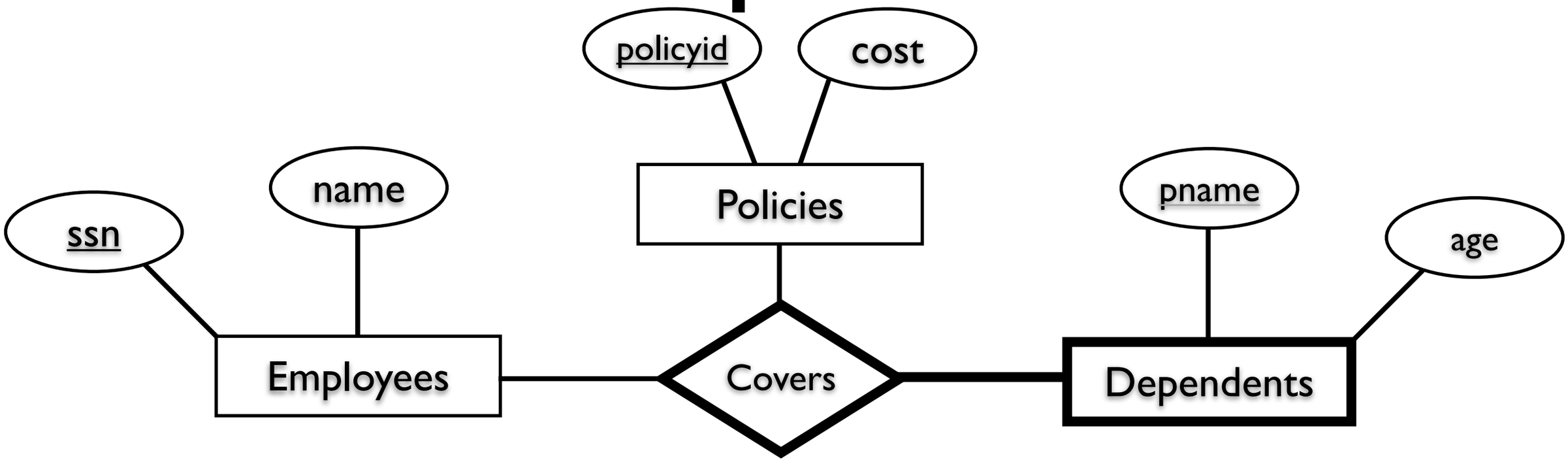


Managers have a discretionary budget (dbudget) for each dept.

How would we modify this ER diagram if the budget were per-manager, rather than per-department

One possible solution would be to use an ISA relationship to define a Managers subset of Employees. The manager entity would have a dbudget attribute, and only managers could participate in the manages relationship

Group Work



1) What are some limitations of this ER Diagram?

2) Design an ER Diagram that addresses these issues.

Each policy has exactly one owner and one dependent. This does not cover the case where multiple employees could co-own a single policy, or the case where one policy could cover multiple dependents. A better solution would be to have 2 binary relationships (owner, beneficiary) connected by the policies entity set.

Summary

- The ER Model is a popular way to design schemas (and maps nicely to SQL)
  - **Basic Constructs**: Entities, Relationships, and Sets of both.
  - **Additional Constructs**: Weak Entities, ISA hierarchies, Aggregation
- There is no one 'right' model for a given scenario.
- Understanding how to design a schema is important.

Summary

- Constraints (as we will see) can be used to optimize query evaluation.
  - Key Constraints
  - Participation Constraints
  - Overlap/Covering Constraints for ISA.