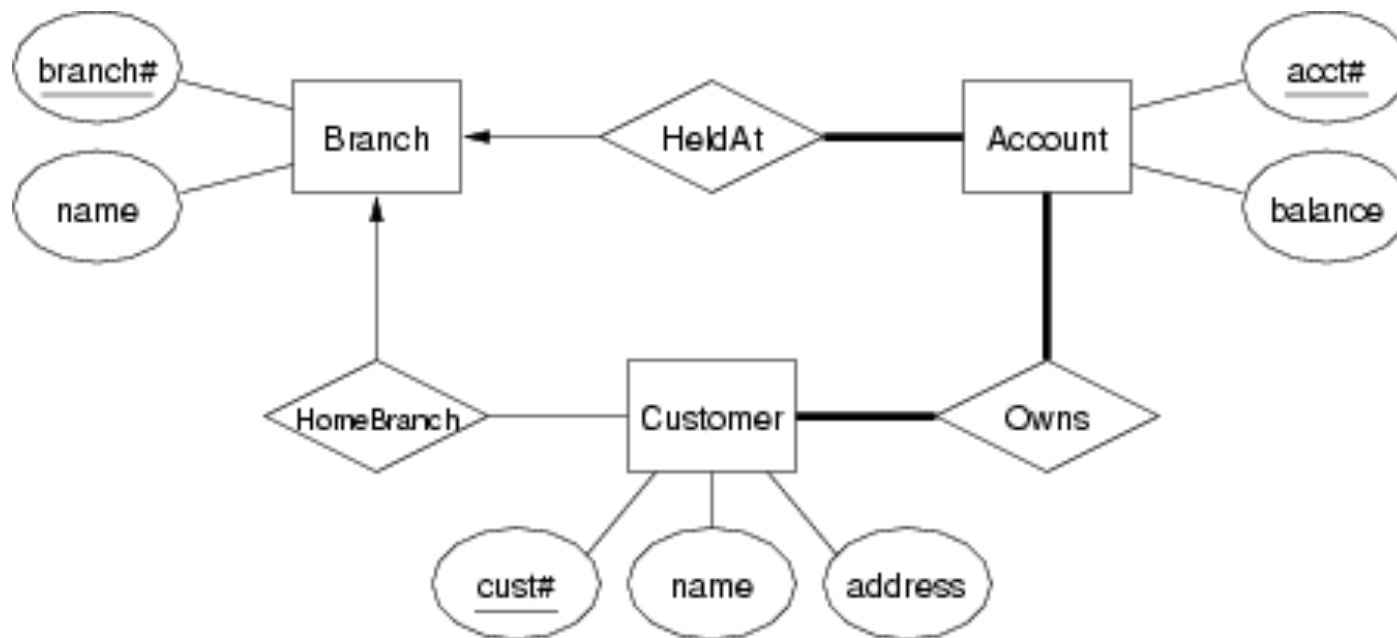


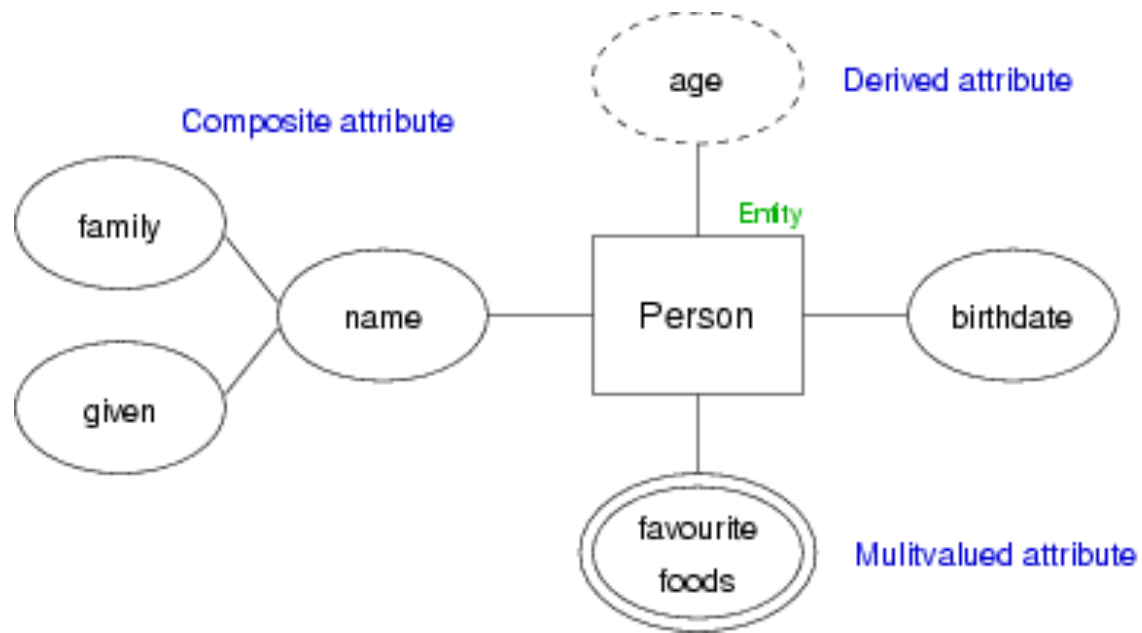
Brief Recap - Entity-Relationships

Example ER Diagram



Entity-Relationships

Example of attribute notations



Entity Sets

An entity set can be viewed as either:

- a set of entities with the same set of attributes (extensional)
- an abstract description of a class of entities (intensional)

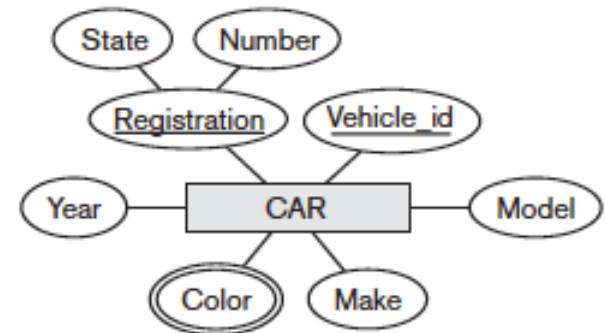
Key (superkey): any set of attributes whose set of values are distinct over entity set

- natural (e.g., name+address+birthday) or artificial (e.g., SSN)

Candidate key = minimal superkey (no subset is a key)

Primary key = candidate key chosen by DB designer

Keys are indicated in ER diagrams by underlining



Relationship Sets

Relationship: an association among several entities

- e.g., Customer(9876) is the owner of Account(12345)

Relationship set: collection of relationships of the same type

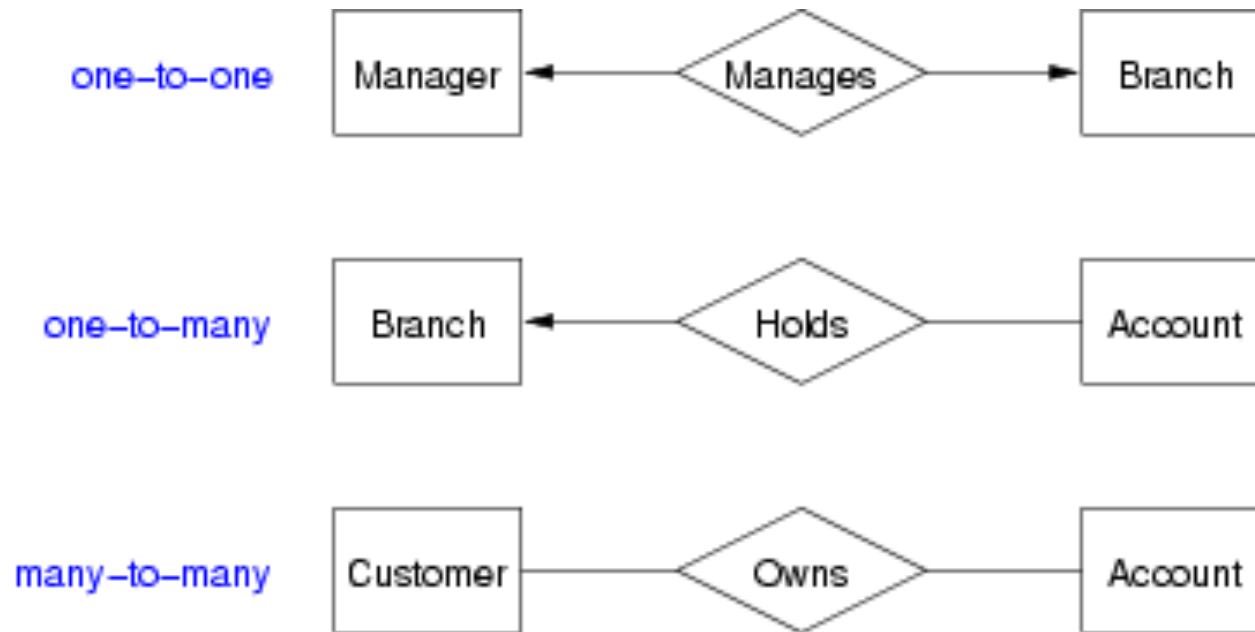
- Degree = # entities involved in reln (in ER model, ≥ 2)
- Cardinality = # associated entities on each side of reln
- Participation = must every entity be in the relationship

Example: relationship participation



Relationship Sets

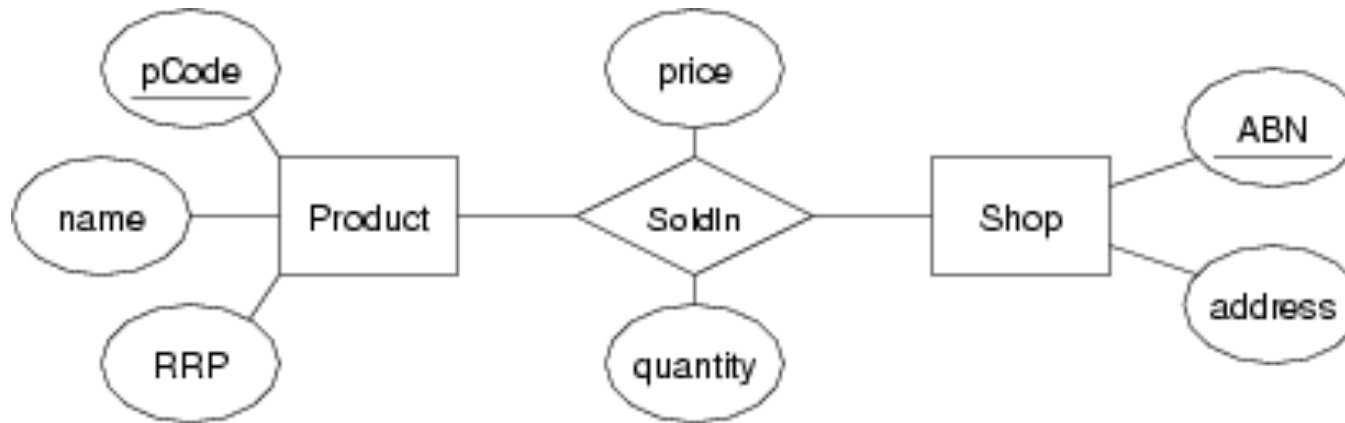
Examples: Relationships



Relationship Sets

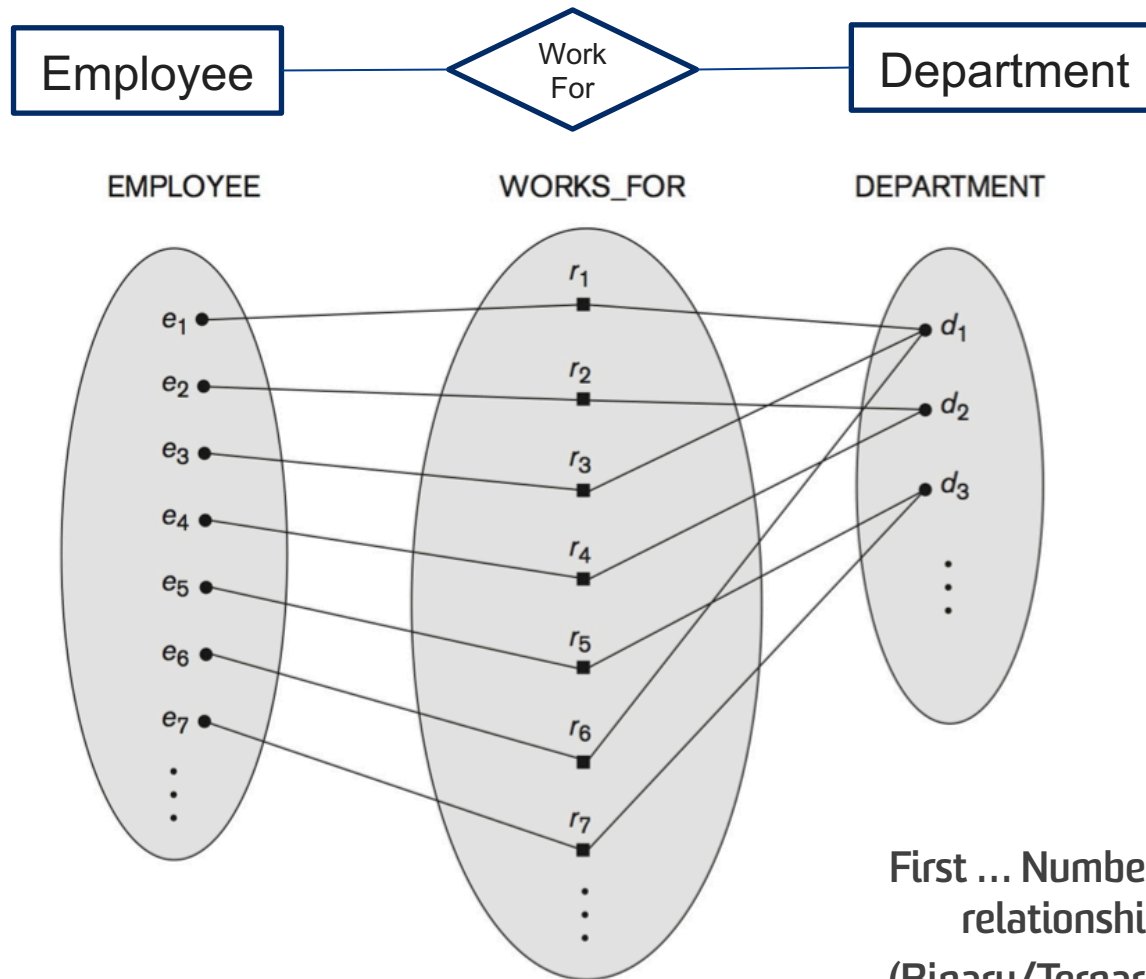
In some cases, a relationship needs associated attributes.

Example:



(Price and quantity are related to products in a particular shop)

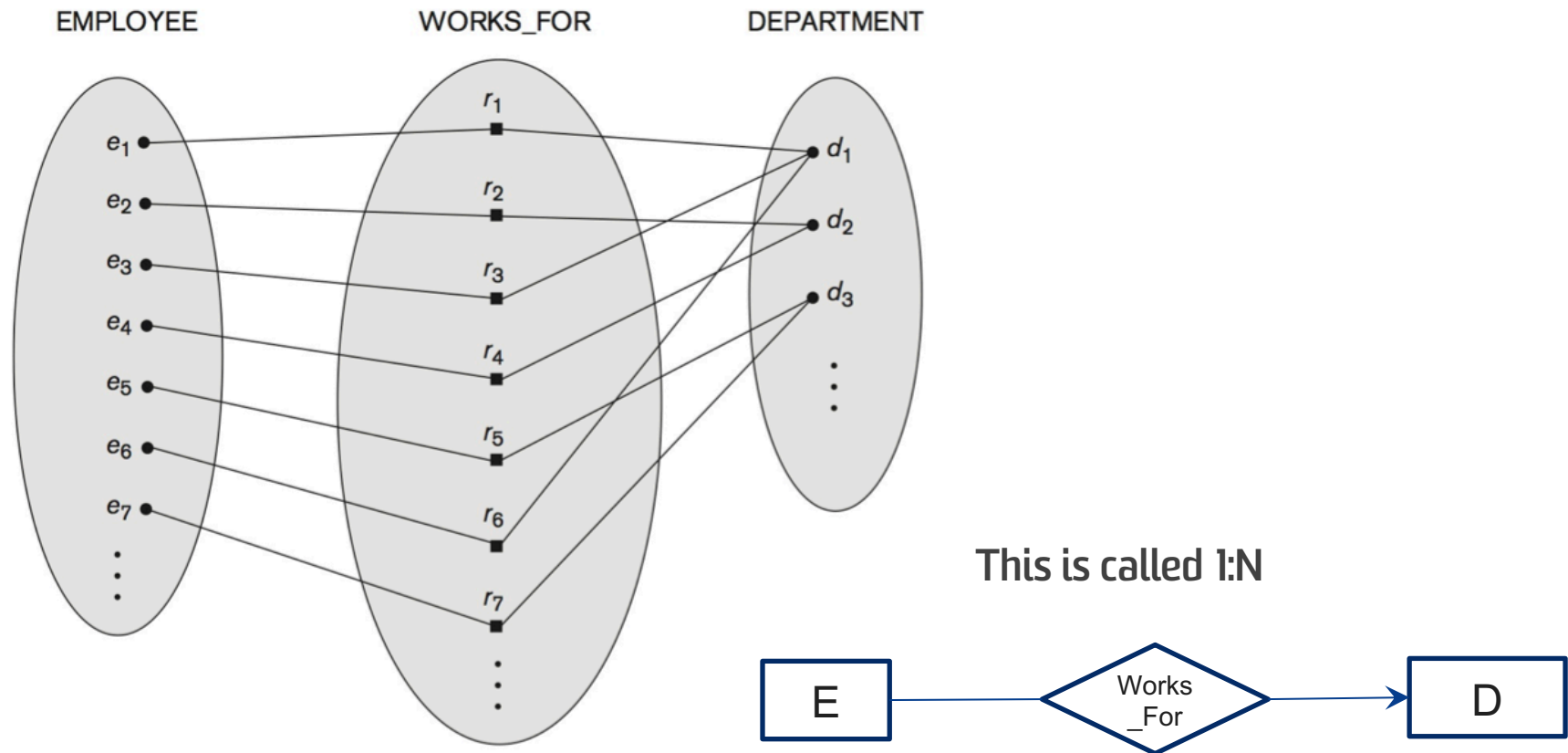
How to think about the relationships more concretely: Entity “sets”, Relationship “sets” and their memberships



First ... Number of Degree in this relationship set?
(Binary/Ternary examples ...)

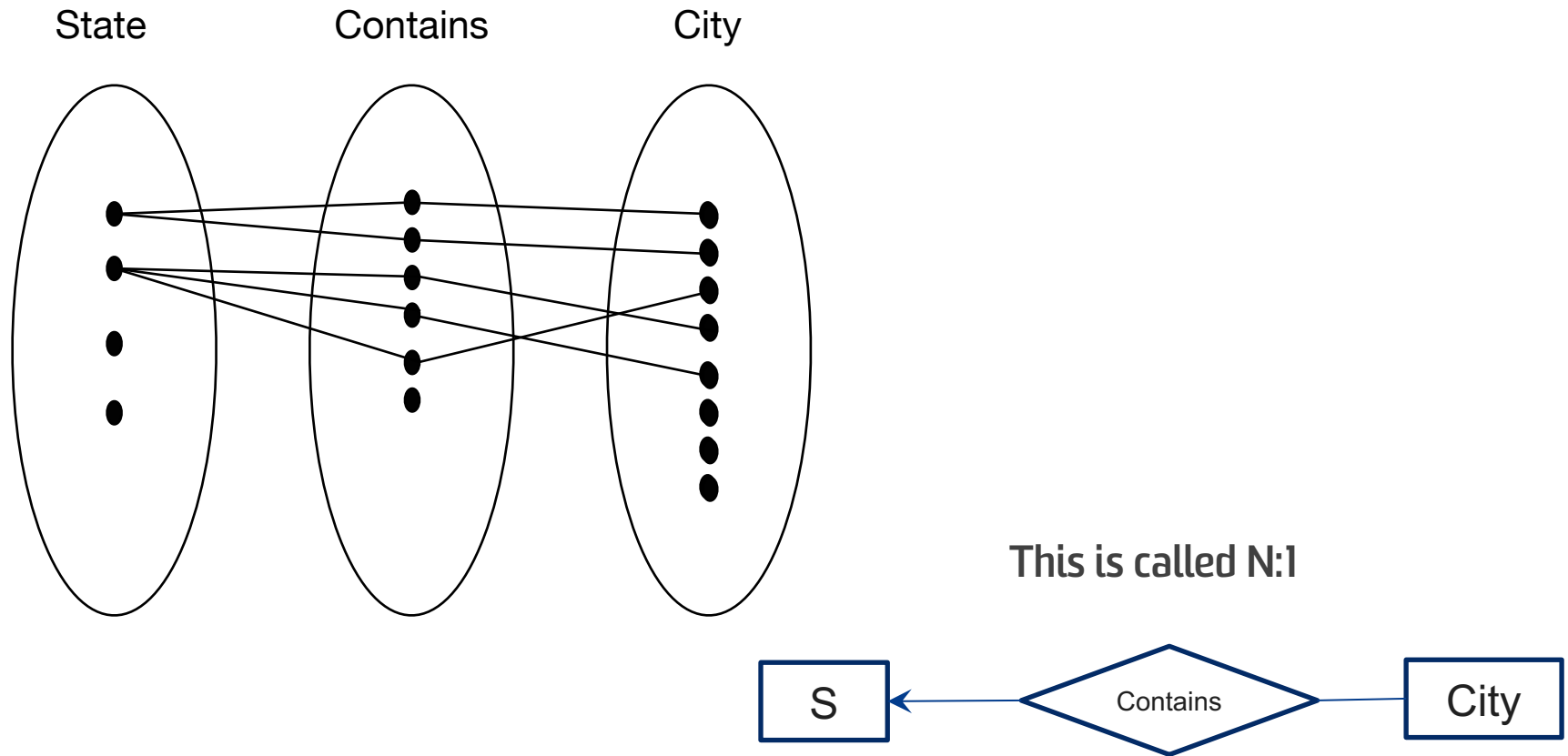
Placing 'constraints' on the relationships

Cardinality (think how many number of *relationship instances* that an entity can participate?) – this should depend on your requirements



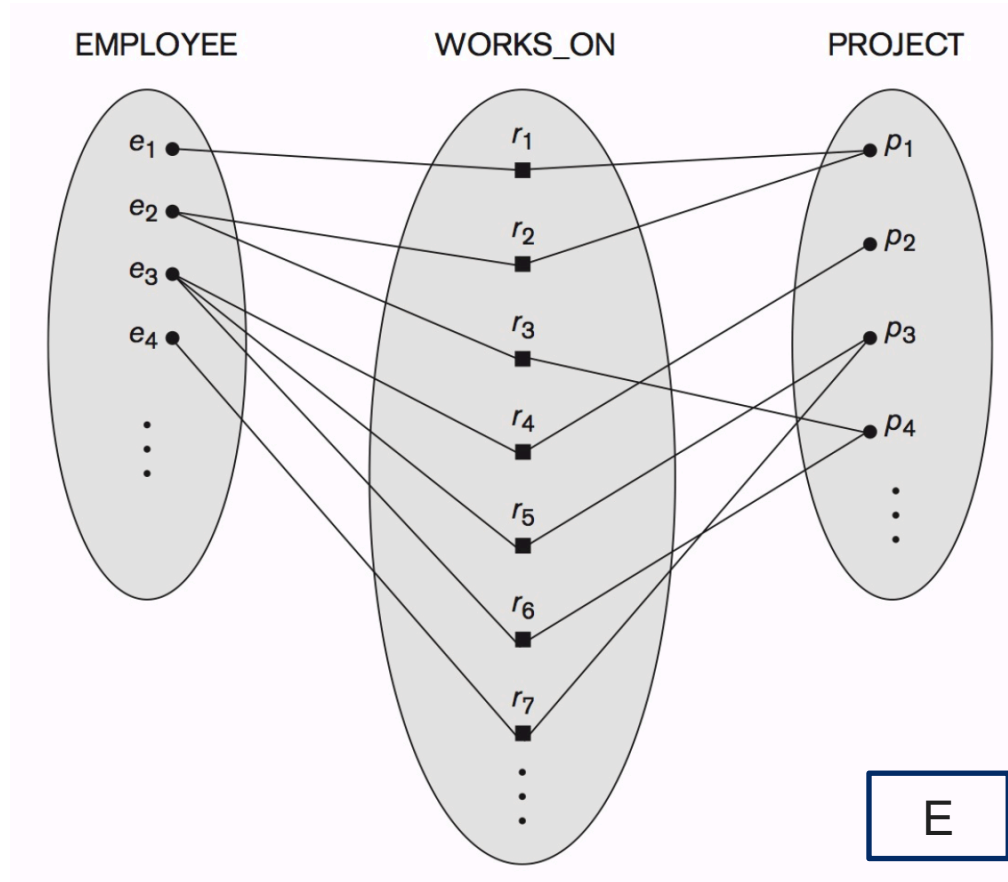
Placing 'constraints' on the relationships

Cardinality (think how many number of *relationship instances* that an entity can participate?) – this should depend on your requirements



Placing 'constraints' on the relationships

Cardinality (think how many number of *relationship instances* that an entity can participate?) – this should depend on your requirements

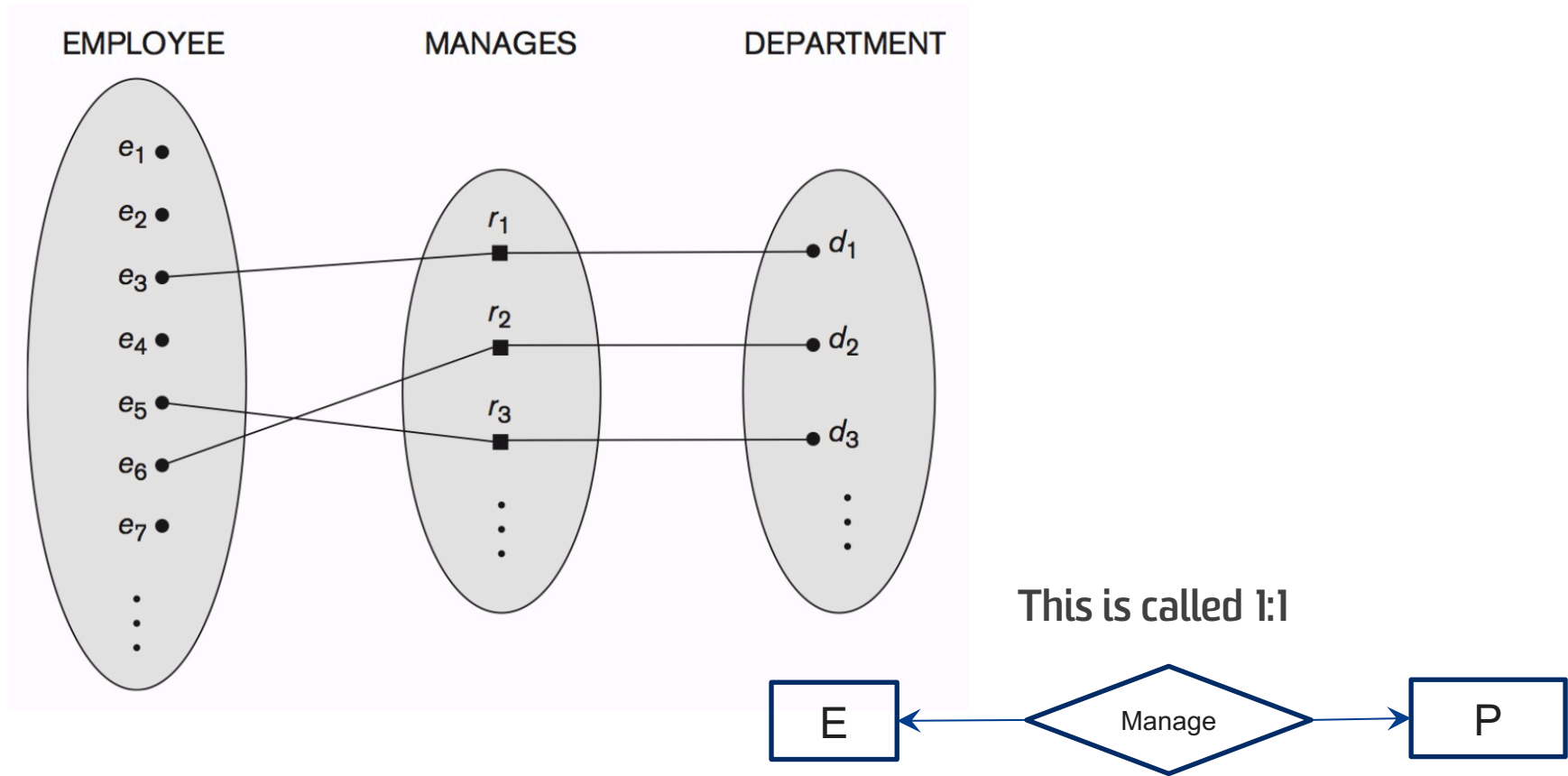


This is called M:N

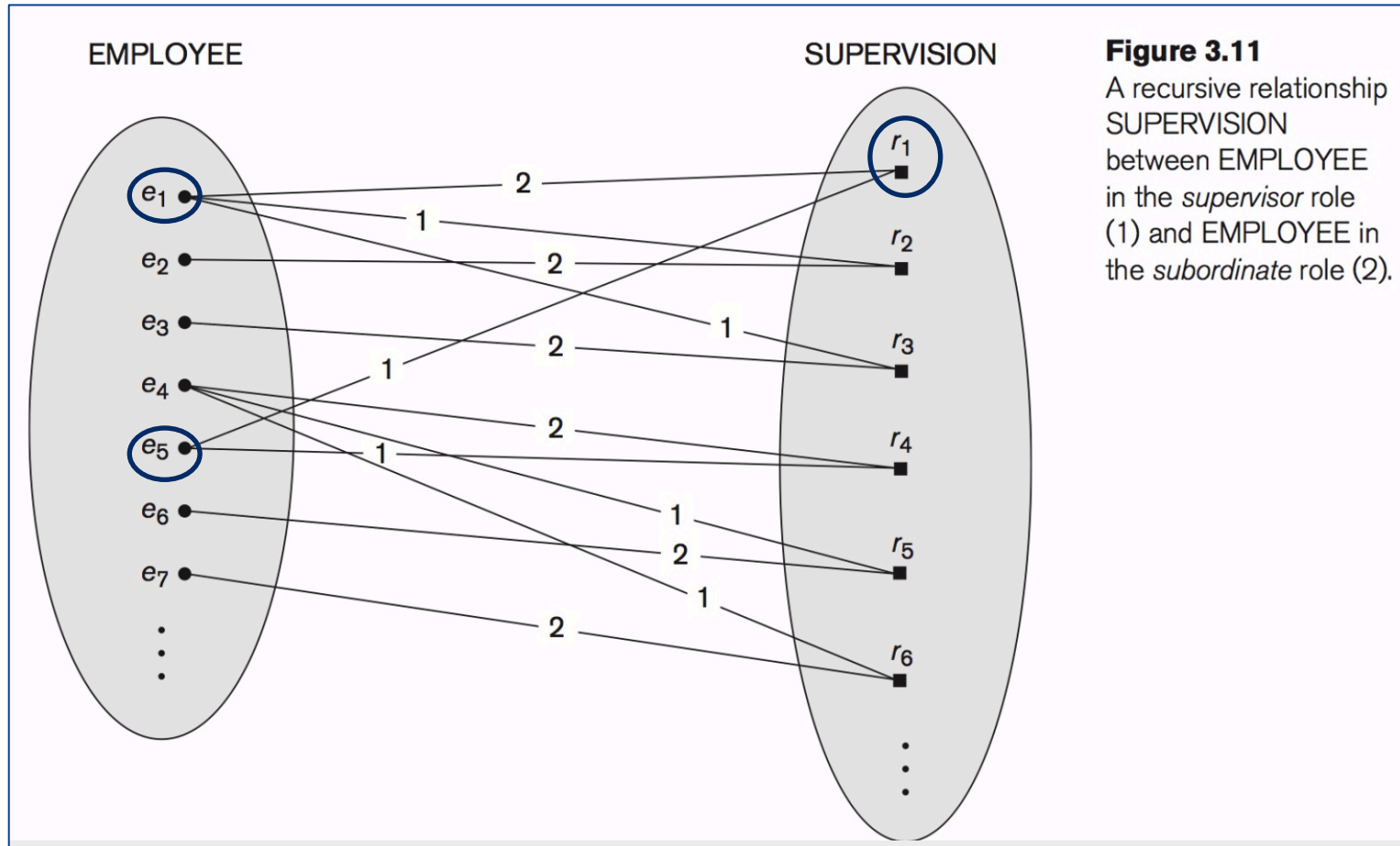


Placing 'constraints' on the relationships

Cardinality (think how many number of *relationship instances* that an entity can participate?) – this should depend on your requirements



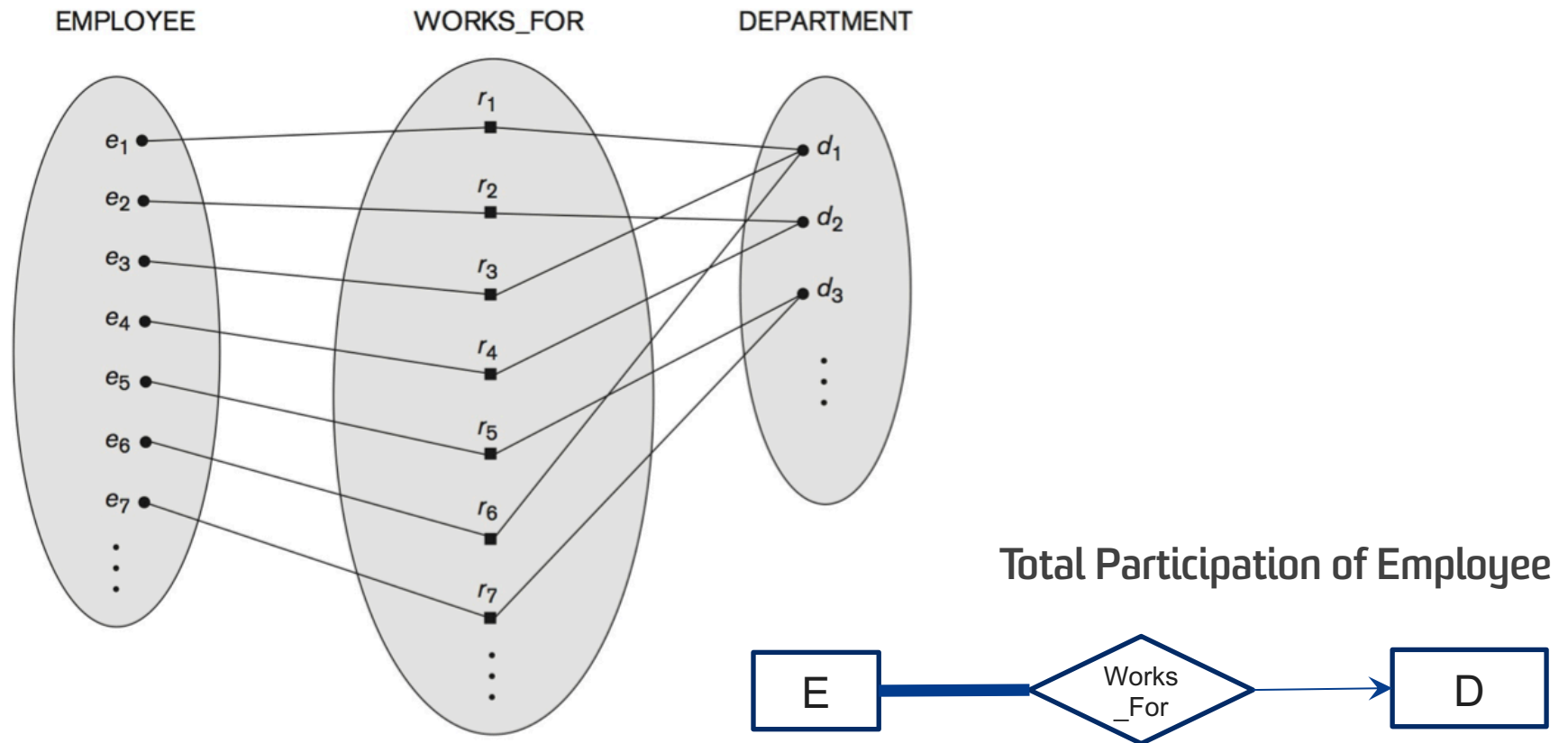
Recursive (or self-referencing) relationships



What would the ER diagram look like in this scenario?

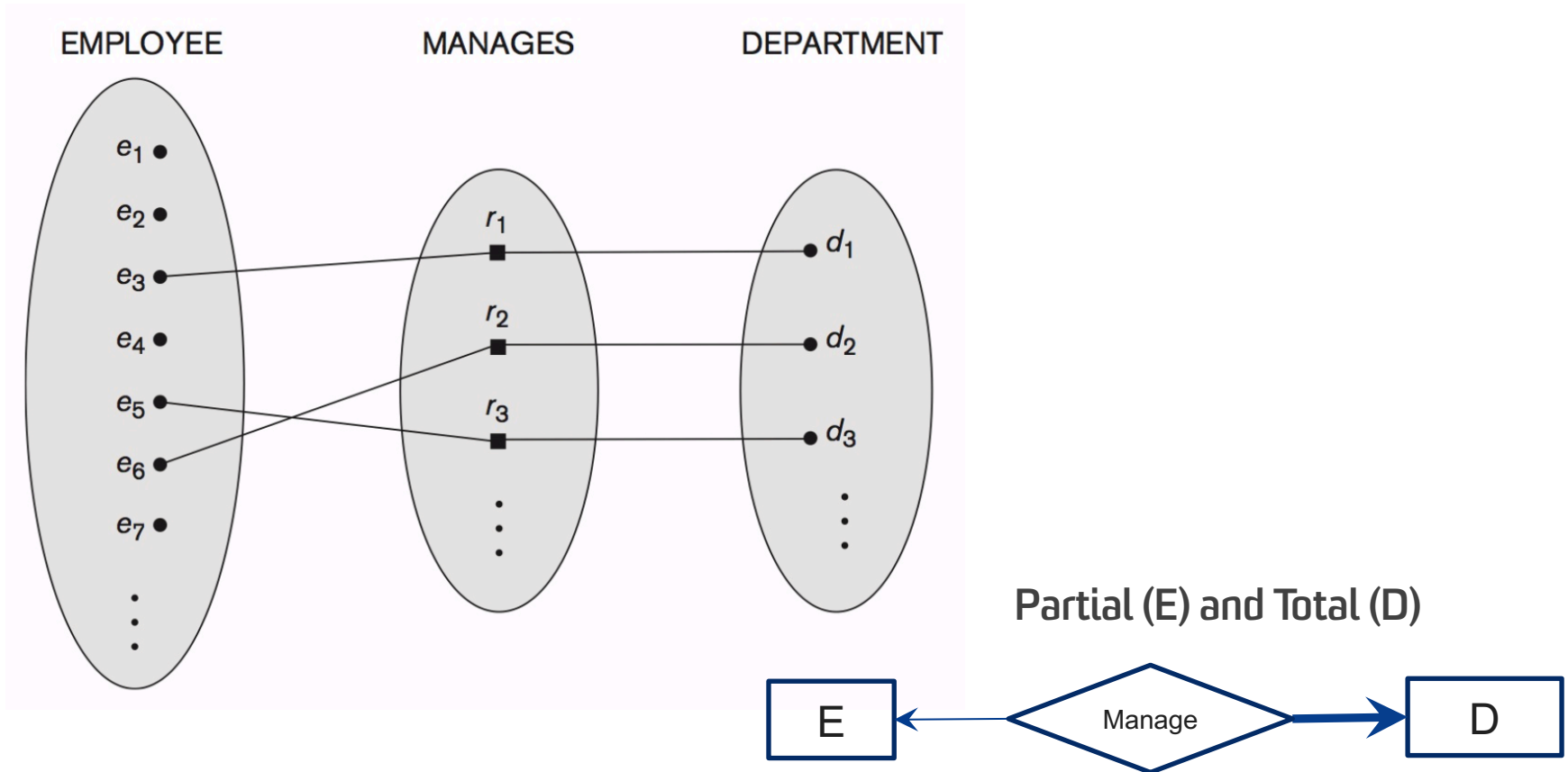
Placing 'constraints' on the relationships

Participation (think is it 'every (or total)' or 'some' entity instances ?) – this should depend on your requirements



Placing 'constraints' on the relationships

Participation (think is it 'every (or total)' or 'some' entity instances ?) – this should depend on your requirements



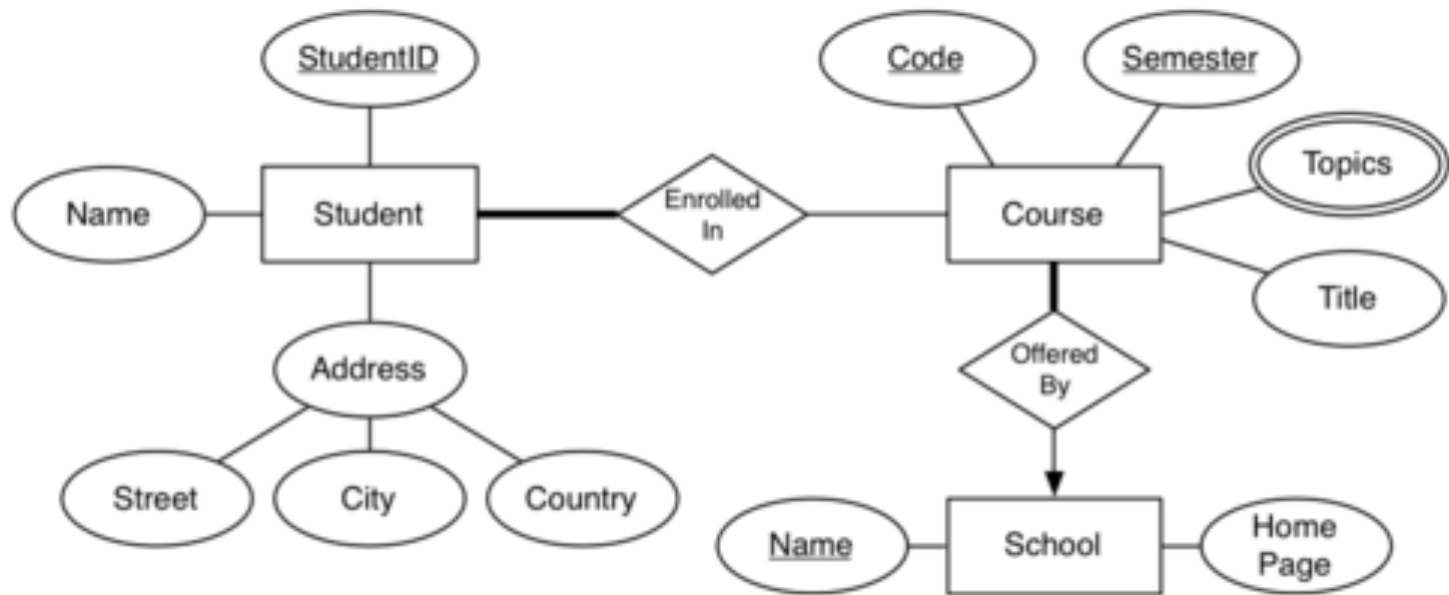
Exercise 2: Relationship Semantics

Describe precisely the scenarios implied by the following relationships:



ER: the story so far

Entities, relationships, attributes, keys, cardinality, participation, ...



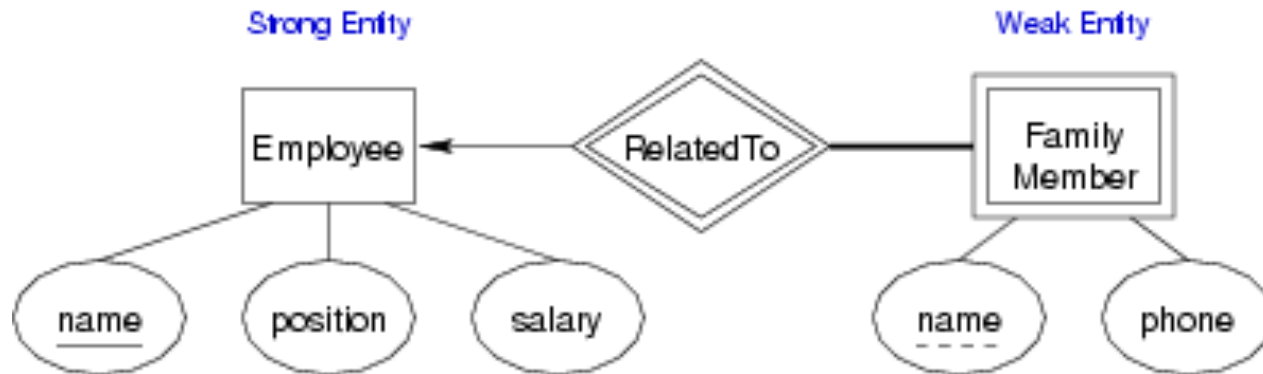
Weak Entity Sets

Weak entities

exist only because of association with strong entities.

have no key of their own; have a discriminator

Example:

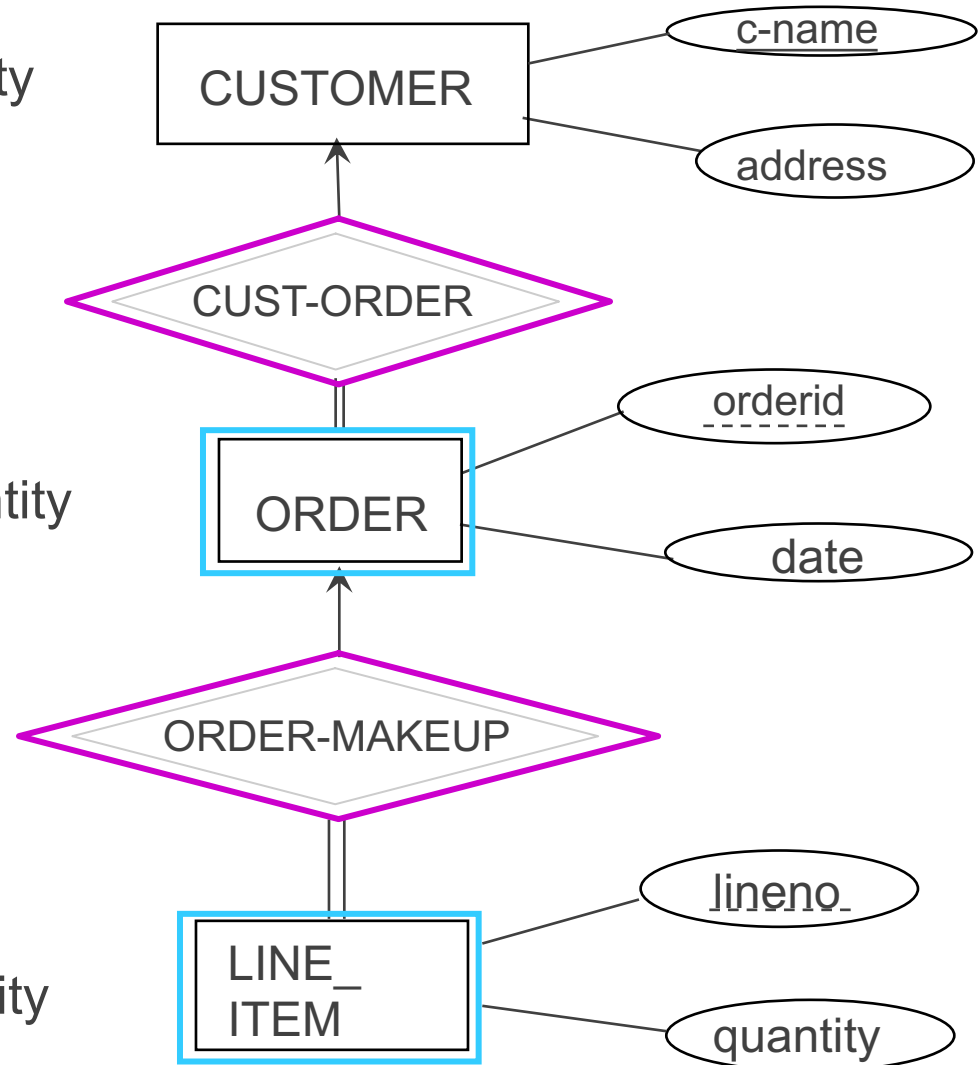


Weak Entities

Strong entity

Weak entity

Weak entity



Subclasses and Inheritance

A subclass of an entity set A is a set of entities:

- with all attributes of A, plus (usually) its own attributes
- that is involved in all of A's relationships, plus its own

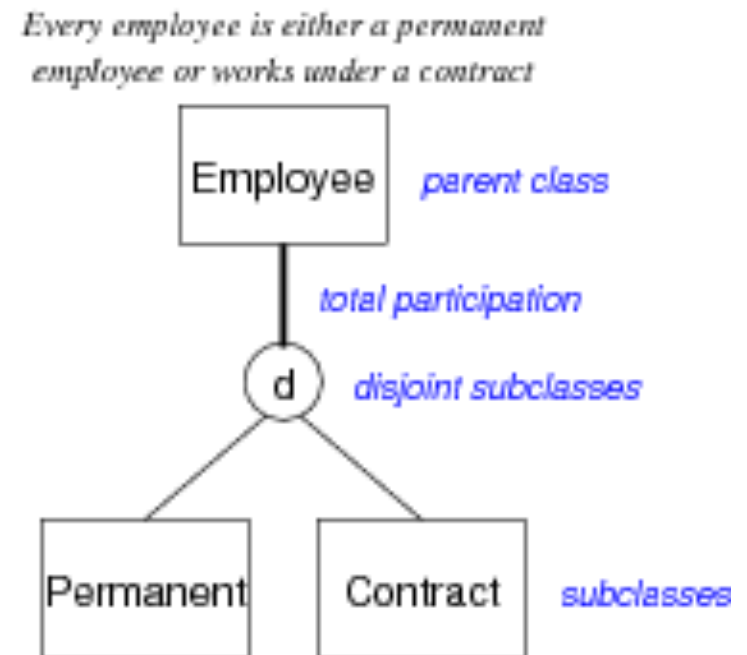
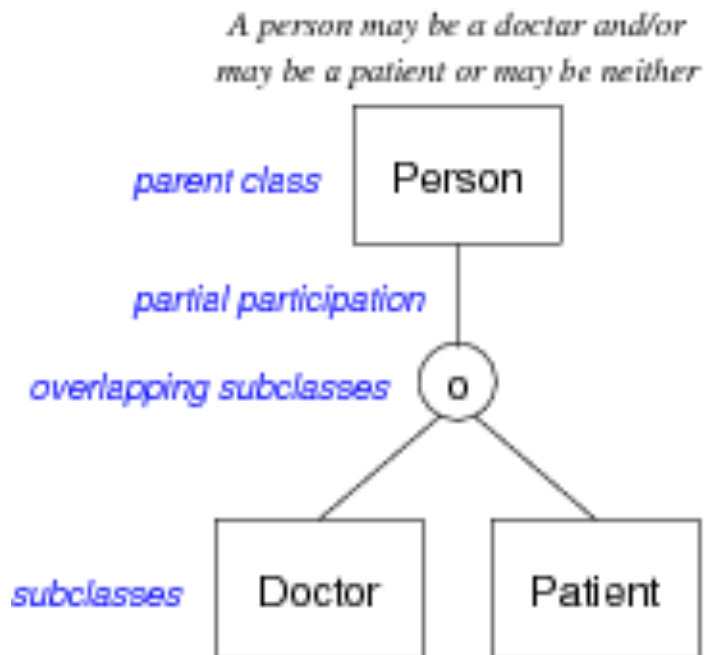
Properties of subclasses:

- overlapping or disjoint (can an entity be in multiple subclasses?)
- total or partial (does every entity have to also be in a subclass?)

Special case: entity has one subclass ("B is-a A" specialisation)

Subclasses and Inheritance

Example:



Exercise 3: Subclasses in the University

- Every person has a name and an address. A person is uniquely identified by their name.
- At a university, there are two groups of persons, employees and students. **Every employee receives a salary, while every student pays a fee.**
- Among the employees, there is research and teaching staff. An employee can belong to both groups.
- Among the teaching staff, there are lecturers and tutors. **A tutor works for several courses.**
- Every student is either a postgraduate student or an undergraduate student.
- A postgraduate student has **a thesis title, on which he/she is working.**
- Every undergraduate student **is working on a project.**
- Every undergraduate student is supervised by a member of the teaching staff.

Design Using the ER Model

ER model: simple, powerful set of data modelling tools

Some considerations in designing ER models:

- should an "object" be represented by an attribute or entity?
- is a "concept" best expressed as an entity or relationship?
- should we use n-way relationship or several 2-way relationships?
- is an "object" a strong or weak entity? (usually strong)
- are there subclasses/superclasses within the entities?

Answers to above are worked out by thinking about the application domain.

Exercise 4: ER Design Choices

The following two diagrams both represent
a person has some types of food that they like



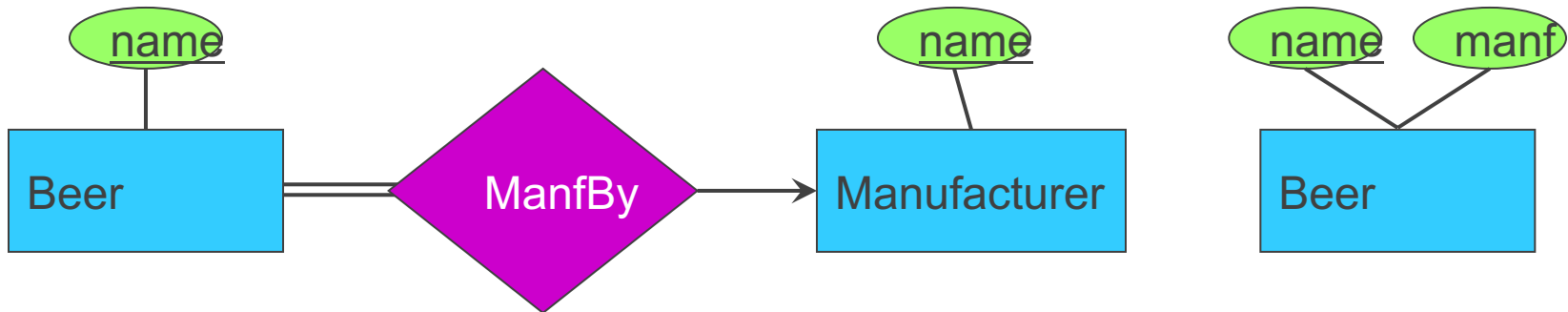
Why might we favour one over the other?

Entities Vs. Attributes

Sometime it is not clear which concepts are worthy of being entities, and which are handled more simply as attributes ...

Example:

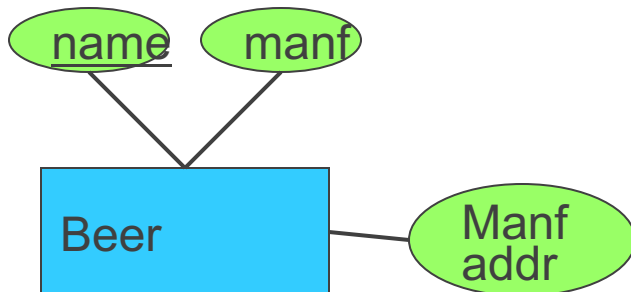
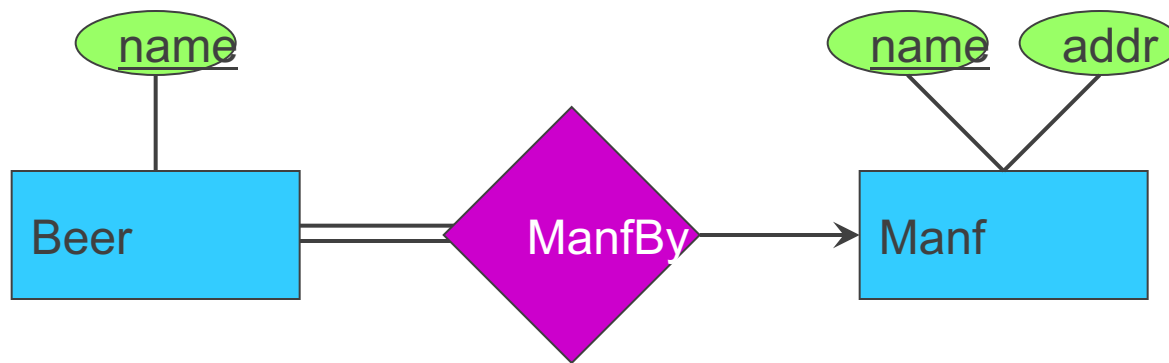
Which are the pros and cons of each of the two designs below?



“Don’t Say the Same Thing More Than Once”

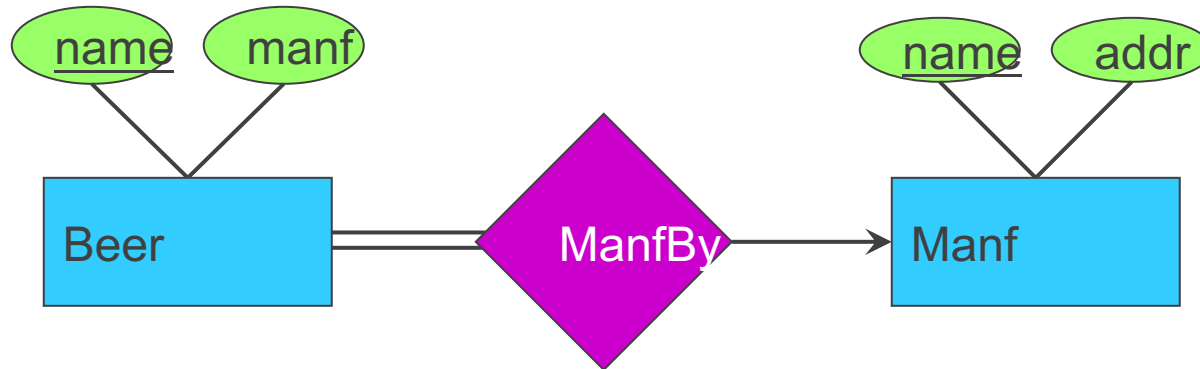
Redundancy wastes space and encourages inconsistency

Example:



*Which is good design,
and which is bad?
Why?*

And What About This?

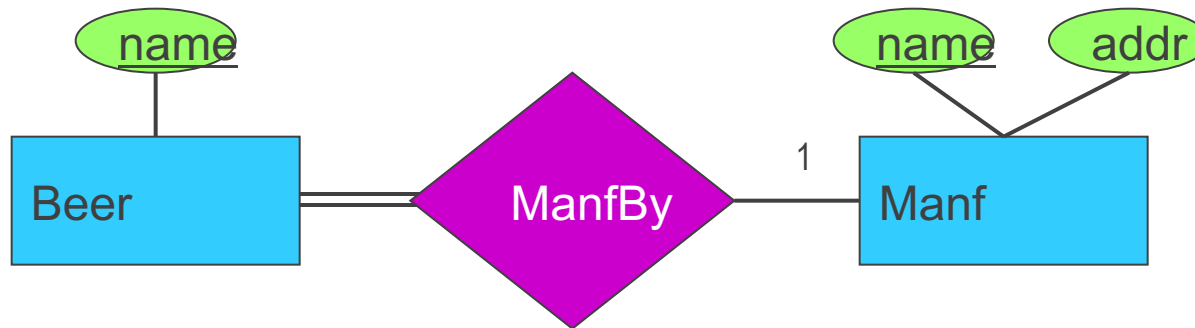


Entity Vs. Attribute: Rules of Thumb

Only make an entity if either:

- It is more than a name of something; i.e., it has non-key attributes or relationships with a number of different entities, or
- It is the “many” in a many-one relationship

Entity Vs. Attribute: Example



The following design illustrates both points:

- Manfs deserves to be an entity because we record addr, a non-key attribute
- Beers deserves to be an entity because it is at the “many” end
- If not, we would have to make “set of beers” an attribute of Manfs

Design Using the ER Model

ER diagrams are typically too large to fit on a single screen
(or a single sheet of paper, if printing)

One commonly used strategy:

- define entity sets separately, showing attributes
- combine entities and relationships on a single diagram
(but without showing entity attributes)
- if very large design, may use several linked diagrams

Exercise 5: Medical Information

Develop an ER design for the following scenario:

- Patients are identified by an SSN, and their names, addresses and ages must be recorded.
- Doctors are identified by an SSN. For each doctor, the name, specialty and years of experience must be recorded.
- Each pharmacy has a name, address and phone number. A pharmacy must have a manager.
- A pharmacist is identified by an SSN, he/she can only work for one pharmacy. For each pharmacist, the name, qualification must be recorded.
- For each drug, the trade name and formula must be recorded.
- Every patient has a primary physician. Every doctor has at least one patient.
- Each pharmacy sells several drugs, and has a price for each. A drug could be sold at several pharmacies, and the price could vary between pharmacies.
- Doctors prescribe drugs for patients. A doctor could prescribe one or more drugs for several patients, and a patient could obtain prescriptions from several doctors. Each prescription has a date and quantity associated with it.