

ER Model

- Entity-Relationship Data Modelling
- Entity-Relationship (ER) Diagrams
- Entity Sets
- Keys
- Example: Identifying Keys
- Relationship Sets
- Example: Relationship Semantics
- Weak Entity Sets
- Subclasses and Inheritance
- Design Using the ER Model
- Large ER Diagrams
- Summary of ER

❖ Entity-Relationship Data Modelling

The world is viewed as a collection of **inter-related entities**.

ER has three major modelling constructs:

- **attribute**: **data item** describing a property of interest
- **entity**: collection of attributes describing **object** of interest
- **relationship**: **association** between entities (objects)

The ER model is not a standard, so notational variations exist

Lecture notes use notation from SKS and GUW books (simple)

❖ Entity-Relationship (ER) Diagrams

ER diagrams are a graphical tool for data modelling.

An ER diagram consists of:

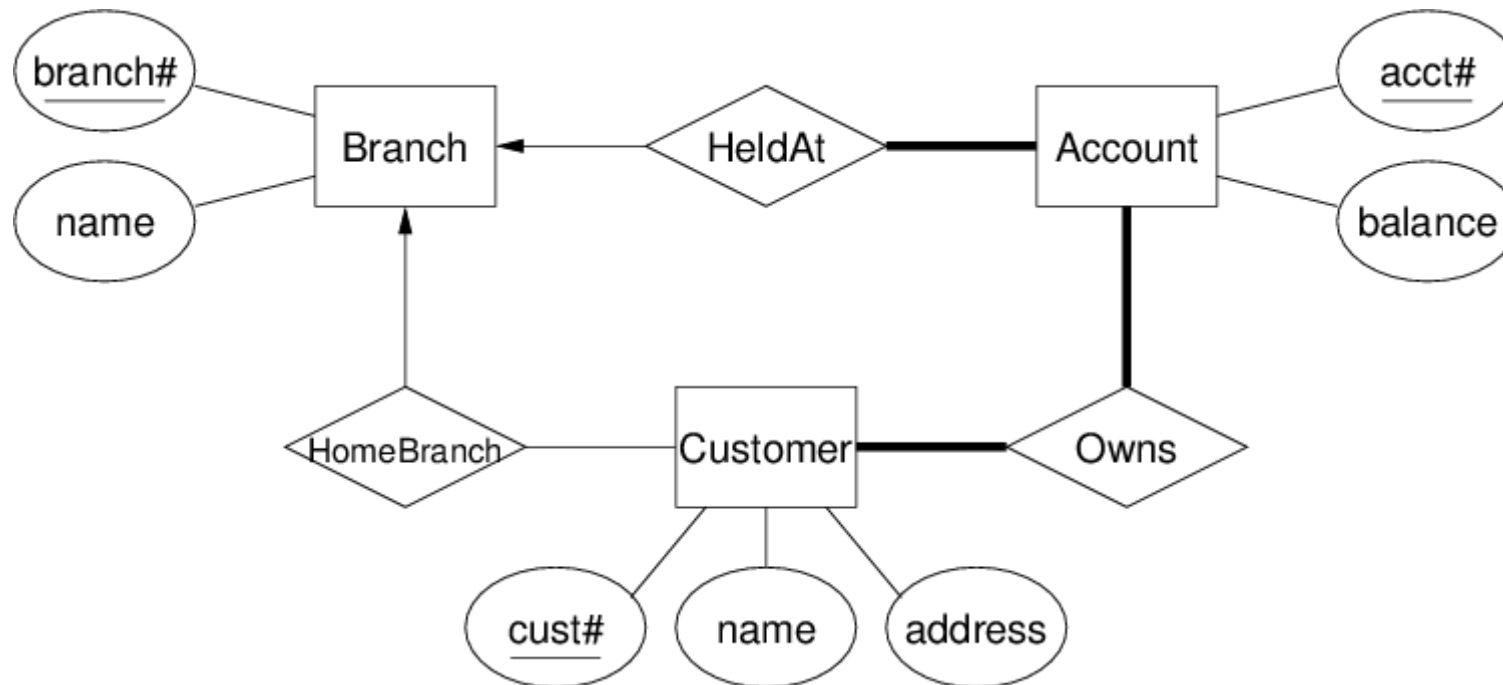
- a collection of **entity set** definitions
- a collection of **relationship set** definitions
- **attributes** associated with entity and relationship sets
- connections between entity and relationship sets

Terminology abuse:

- we say "entity" when we mean "entity set"
- we say "relationship" when we mean "relationship sets"
- we say "entity instance" to refer to a particular entity

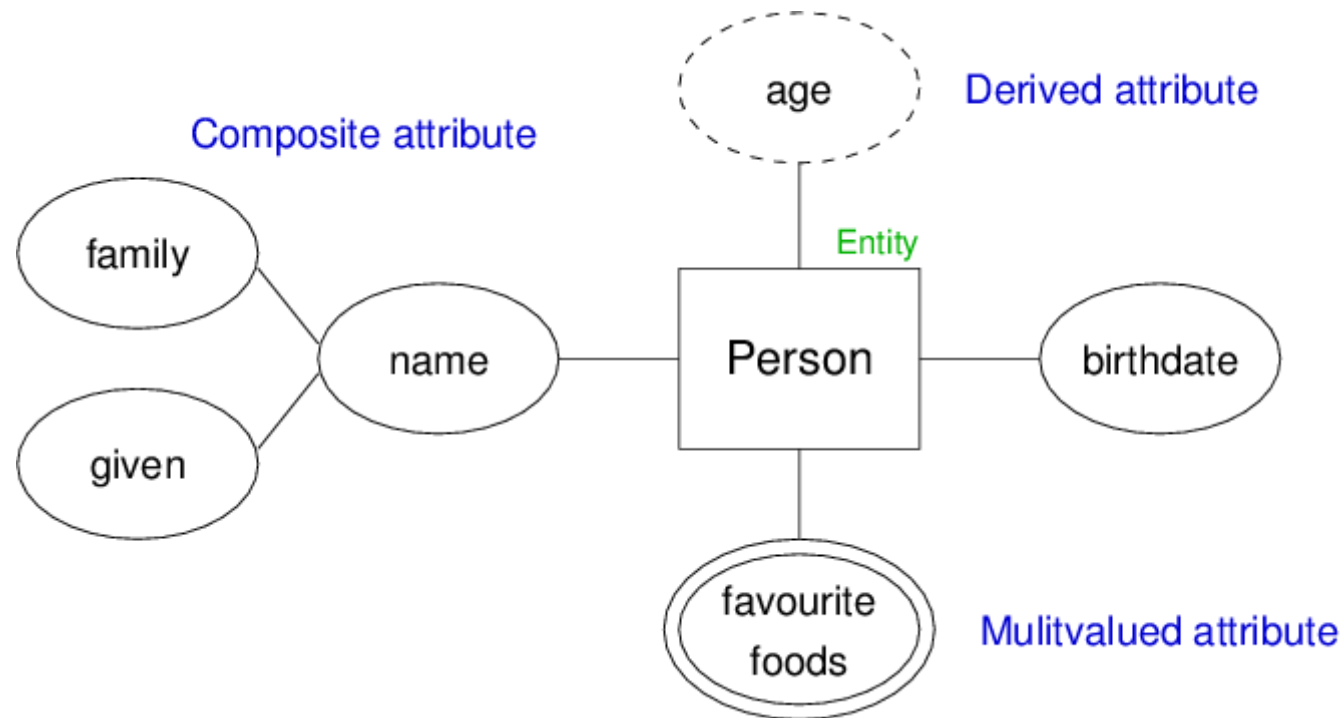
❖ Entity-Relationship (ER) Diagrams (cont)

Example ER diagram:



❖ Entity-Relationship (ER) Diagrams (cont)

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)

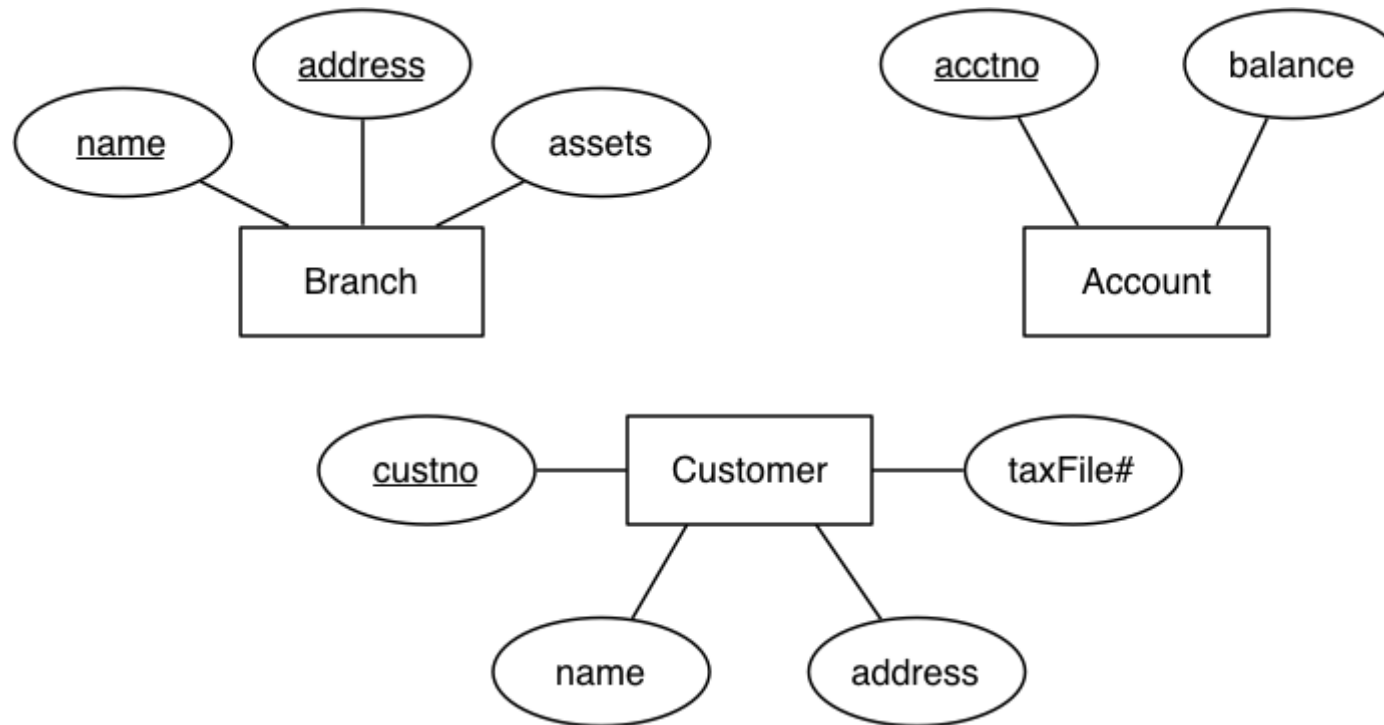
Candidate key = minimal superkey (no subset is a key)

Primary key = candidate key chosen by DB designer

Keys are indicated in ER diagrams by underlining

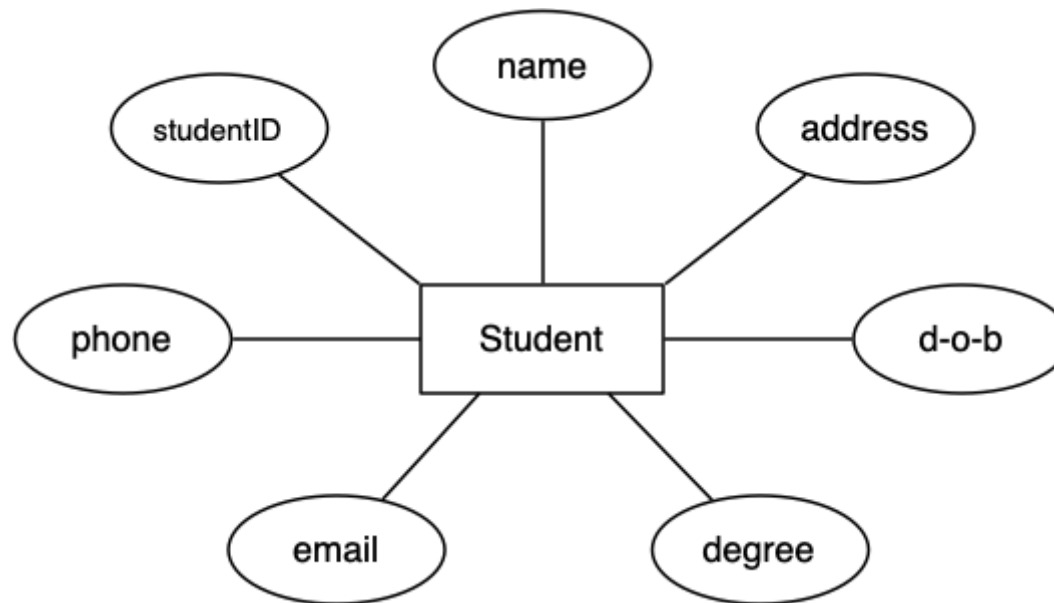
❖ Keys

Sometimes primary keys are obvious ...



❖ Example: Identifying Keys

Candidate keys in the following ER diagram ...



Possibilities: {studentID}, {phone}, {email}, {name,address,d-o-b}?

❖ Relationship Sets

Relationship: an association among several entities

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

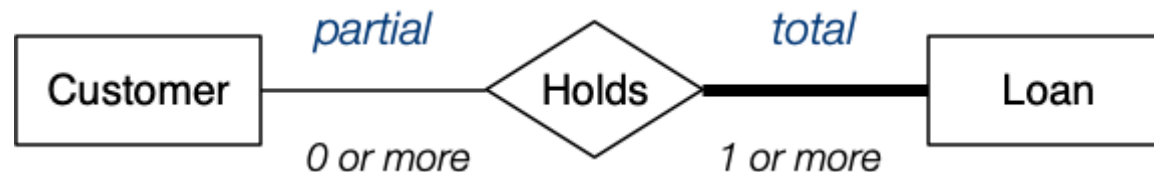
Relationship set: collection of relationships of the same type

Degree = # entities involved in reln (in ER model, ≥ 2)

Cardinality = # associated entities on each side of reln

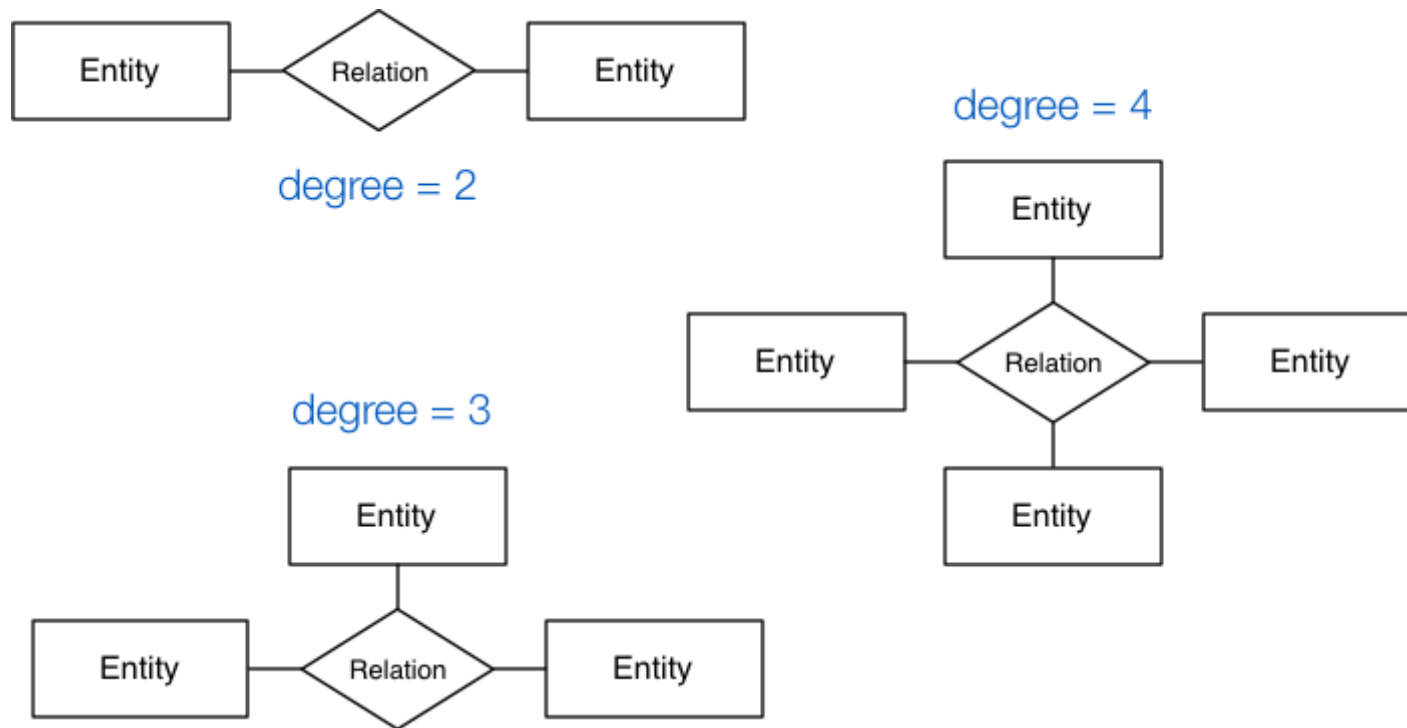
Participation = must every entity be in the relationship

Example: relationship participation



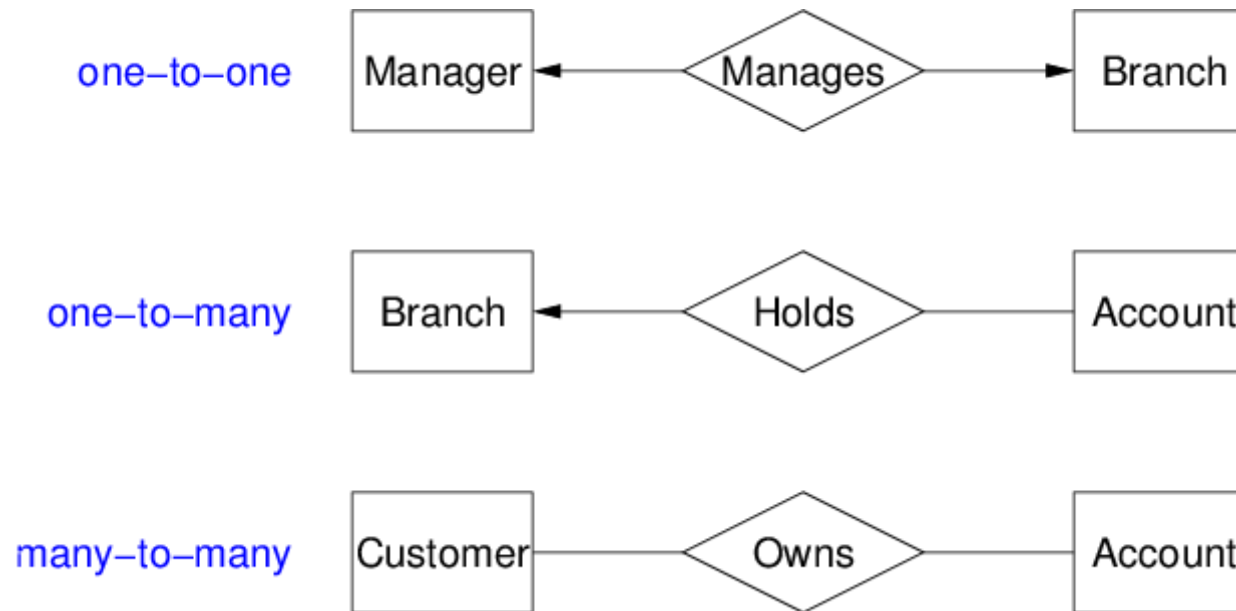
❖ Relationship Sets (cont)

Examples: relationship degree



❖ Relationship Sets (cont)

Examples: relationship cardinality



❖ Example: Relationship Semantics

Semantics of the following relationships ...



*A lecturer may teach
one course
Every course is taught
by 1 or more lecturers*



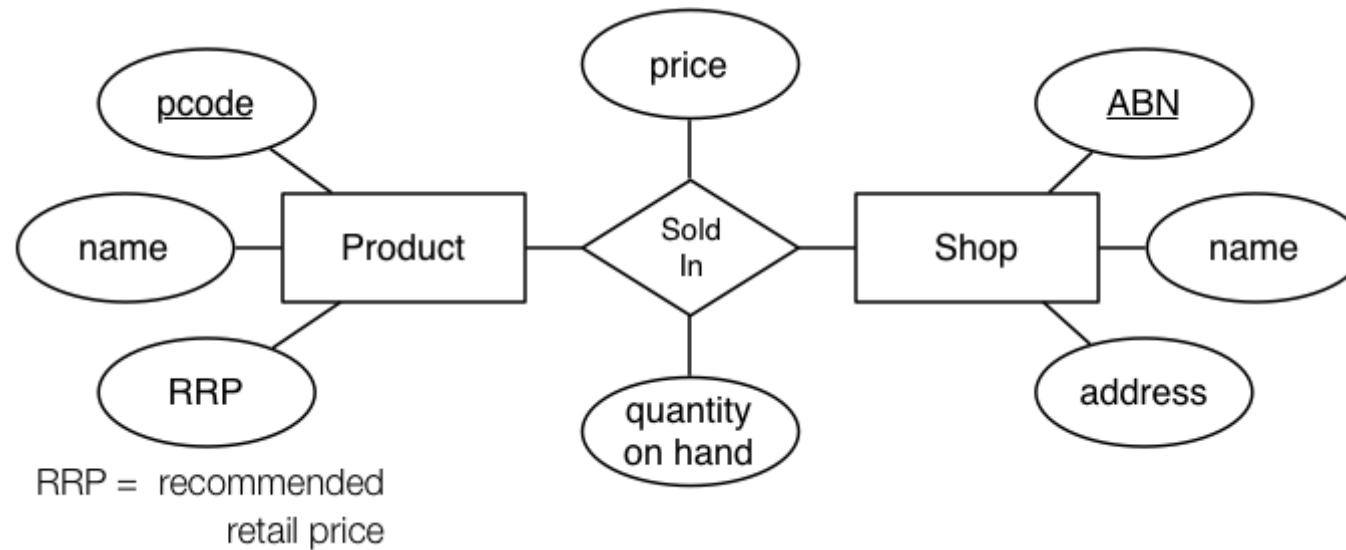
*Every lecturer teaches
one or more courses
Every course is taught
by 1 or more lecturers*



*Every lecturer may teach
one course
Every course is taught
by one lecturer*

❖ Example: Relationship Semantics (cont)

In some cases, a relationship needs associated attributes.



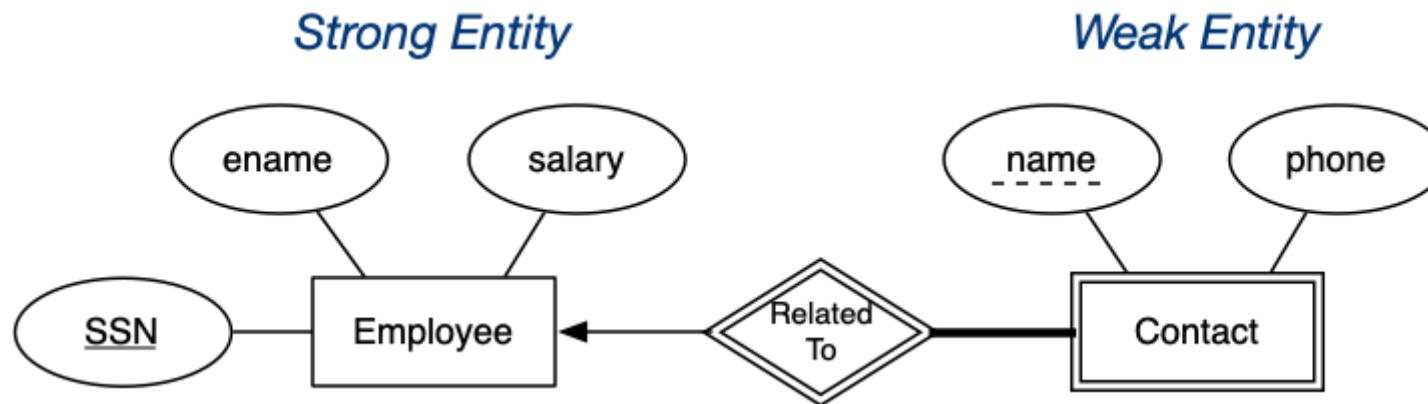
(Price and quantity are related to products in a particular shop)

❖ Weak Entity Sets

Weak entities

- exist only because of association with strong entities.
- have no key of their own; have a **discriminator**

Example:



❖ Subclasses and Inheritance

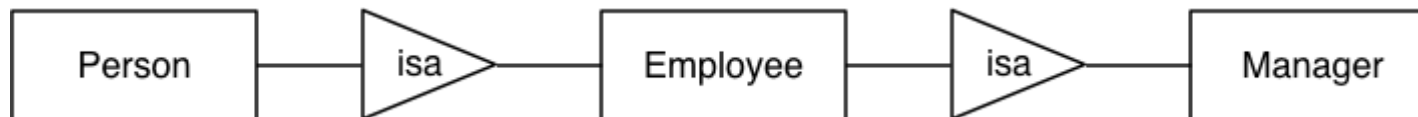
A **subclass** of an entity set A is a set of entities:

- 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?)

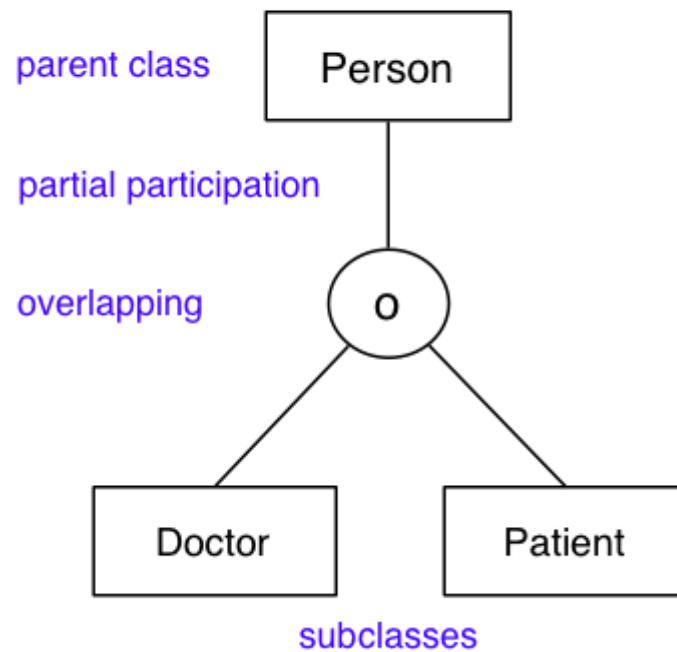
Special case: entity has one subclass ("B **is-a** A" specialisation)



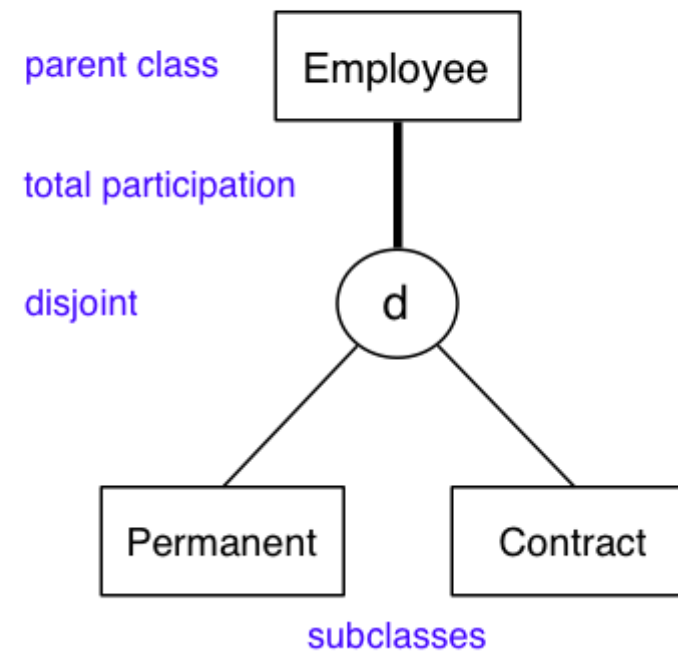
❖ Subclasses and Inheritance (cont)

Example:

*A person may be a doctor and/or
may be a patient or may be neither*



*Every employee is either a permanent
employee or works under a contract*



❖ 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.

❖ Large ER Diagrams

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

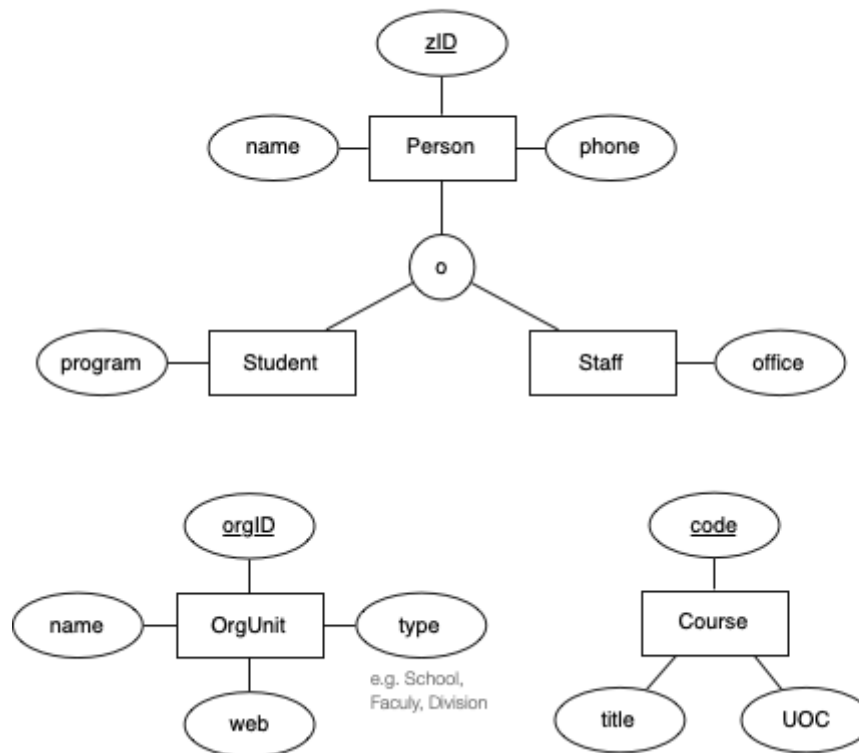
One commonly used strategy:

- 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

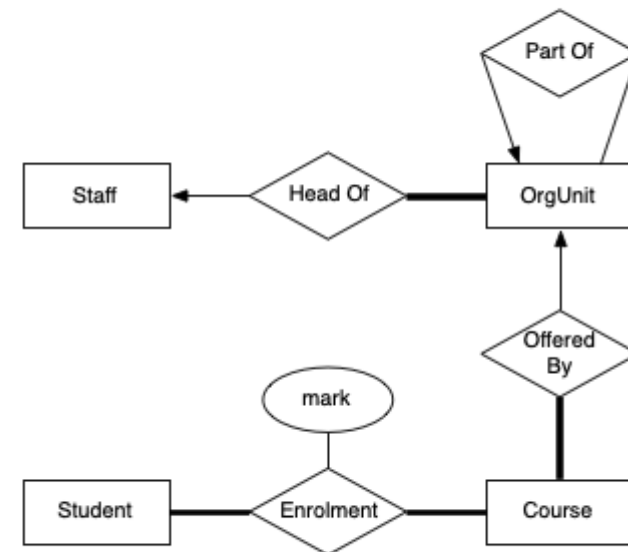
❖ Large ER Diagrams (cont)

Example of drawing large ER diagram:

Entity/Attribute Details



Relationship Details



❖ Summary of ER

ER model is popular for doing conceptual design

- high-level, models relatively easy to understand
- good expressive power, can capture many details

Basic constructs: entities, relationships, attributes

Relationship constraints: total / partial, n:m / 1:n / 1:1

Other constructs: inheritance hierarchies, weak entities

Many notational variants of ER exist
(especially in the expression of constraints on relationships)

Produced: 13 Sep 2020