E-R Diagram: Subclass, E-R to relational design

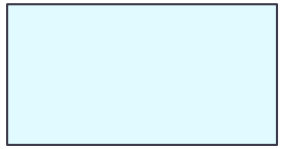
CS 4750 Database Systems

[A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, Ch.6]
[C.M. Ricardo and S.D. Urban, Database Illuminated, Ch.3]

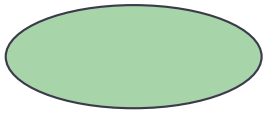
revisit

E-R Diagram: Building Blocks

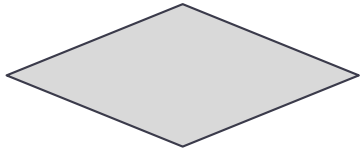
(strong) Entity set



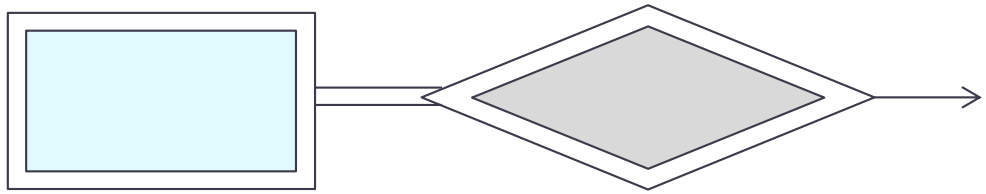
Attribute



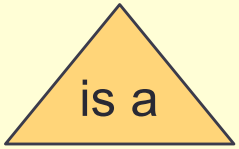
Relationship



Weak entity



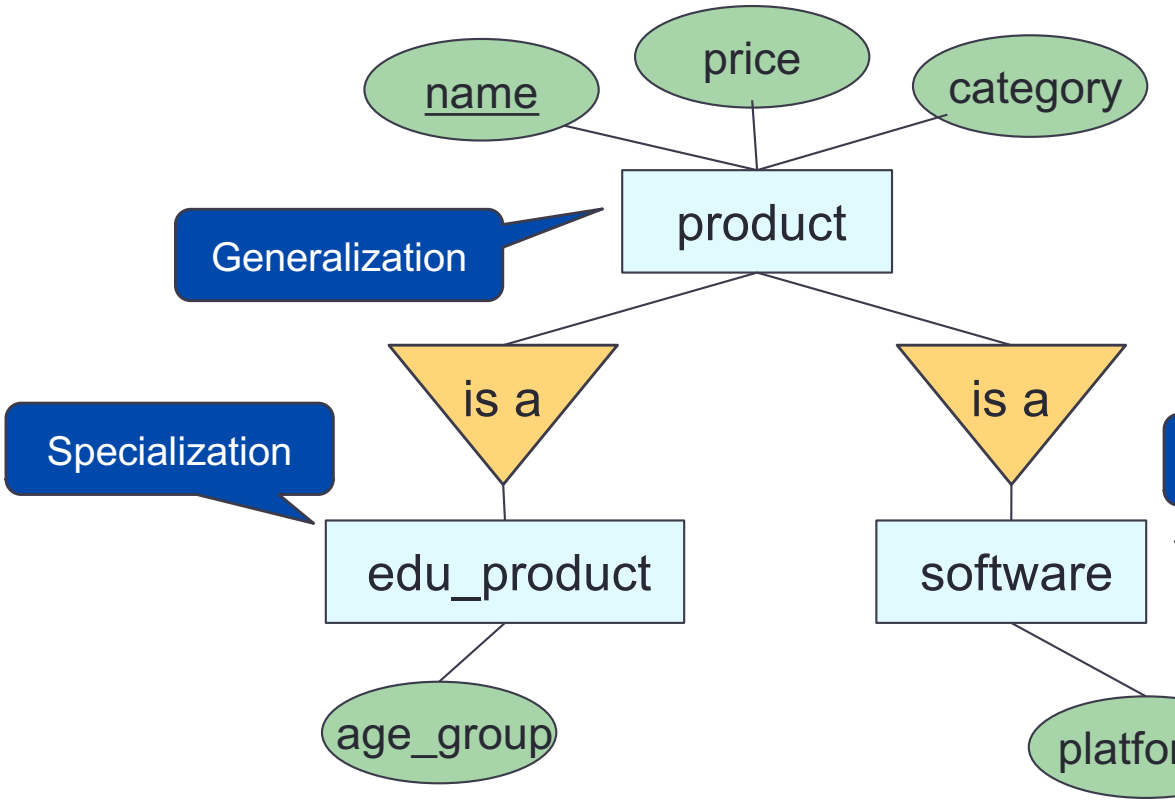
Subclass



Note: colors are not part of E-R Diagram. They simply are used to increase readability.

Subclassing

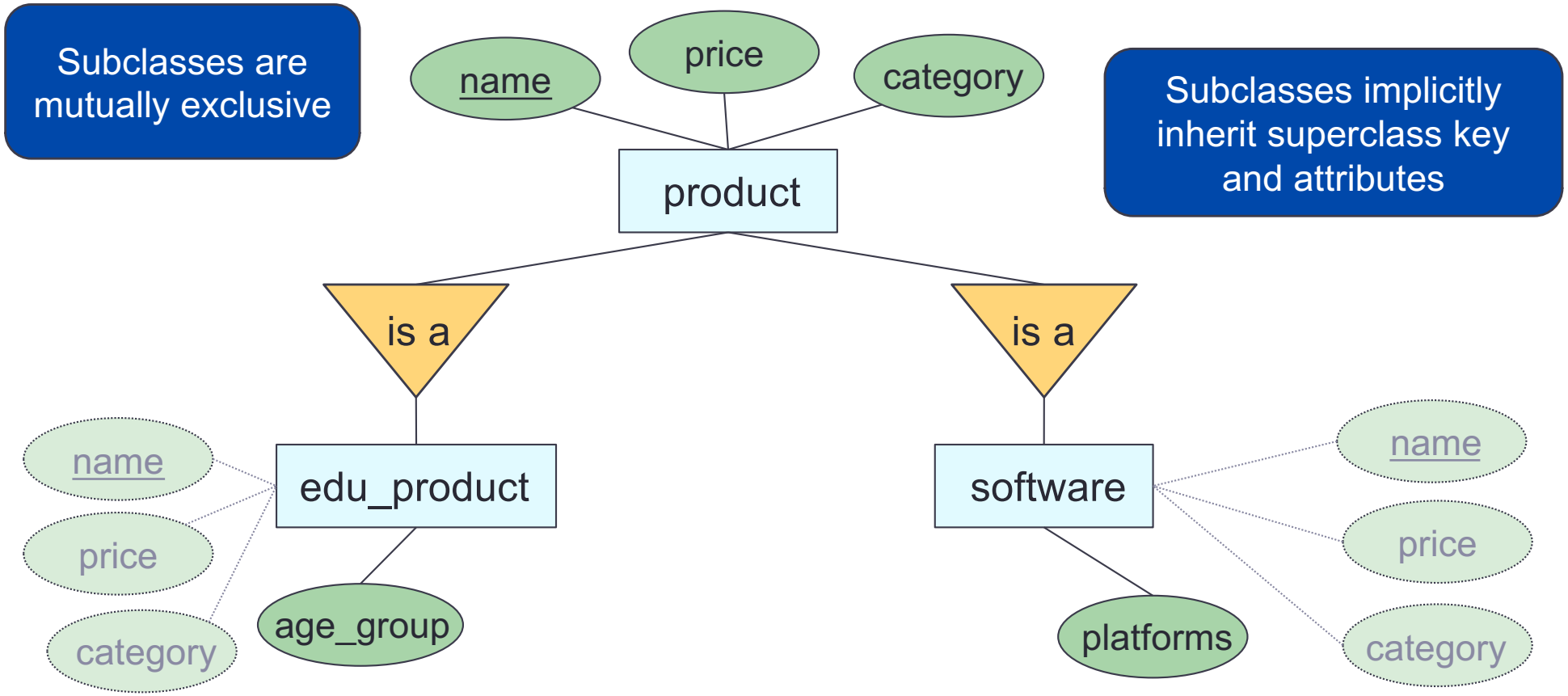
- An entity set may contain entities that have special properties not associated with all members of the set
- Subclasses ~ special-case entity sets
- Isa (or Is-a) ~ special kind of relationship (one-to-one)



Similar to classes in OO (super-classes and sub-classes)

The triangle points to the specialization

Subclassing



Generalization / Specialization
The triangle points to the specialization

Decisions to Make

- Entity set vs. attributes
  - Has more data → entity set
  - Is the data → attribute

- Entity set vs. relationship set
  - Entity set → nouns (students, faculty, loads, …)
  - Relationship → possession verbs (teaches, advises, owns, works for, …)

- Binary vs. n-ary relationship sets
- Specialization / generalization

Rules of Thumb

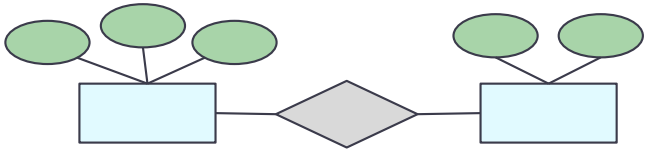
- Pick the right entities
- Keep it simple
- Don’t over complicate things
- Choose the right elements (entities vs. attributes)
- Choose the right relationships
- Follow the specification of the application to be built
- Avoid NULL value
- Avoid redundancy
- Consider small number of tables

Database Design Process

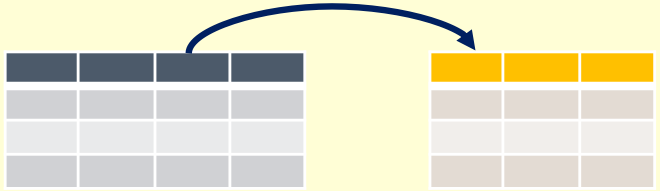
revisit

Interact with users and domain experts to characterize the data

Translate requirements into conceptual model (E-R diagrams)

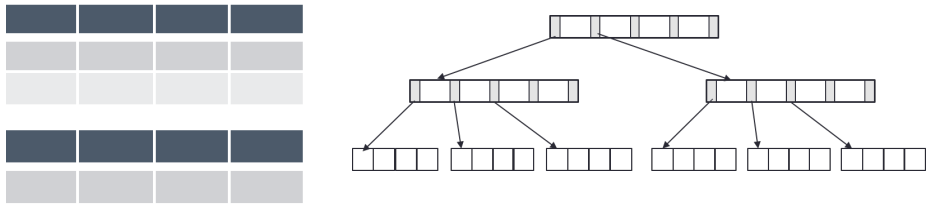


Convert the model to relational model (schema and constraints)



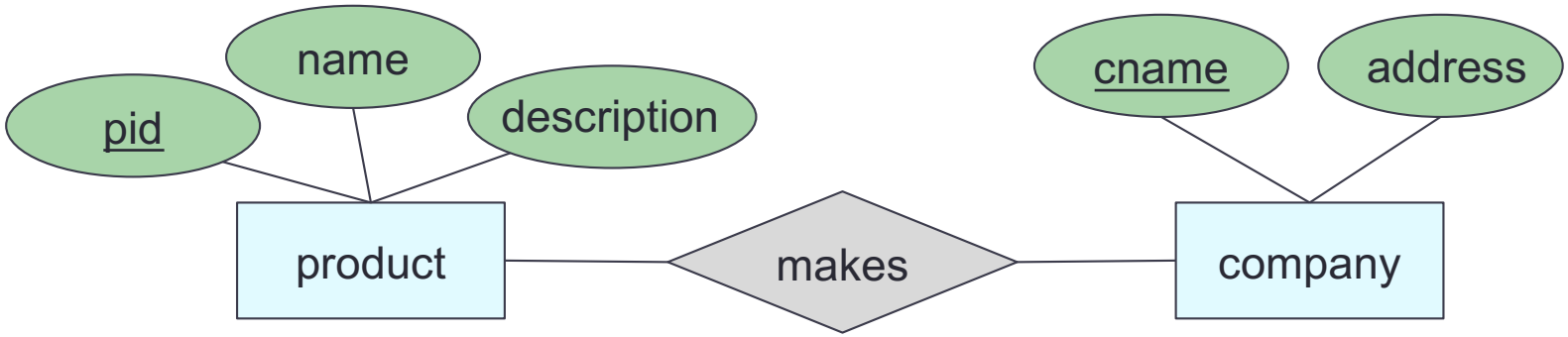
Normalize and develop conceptual (logical) schema of the database

Develop physical schema (partitioning and indexing)



E-R Diagrams to Relations

There is a unique table which is assigned the name of the corresponding entity set or relationship set



product(pid, name, description)

company(cname, address)

makes(cname, pid)

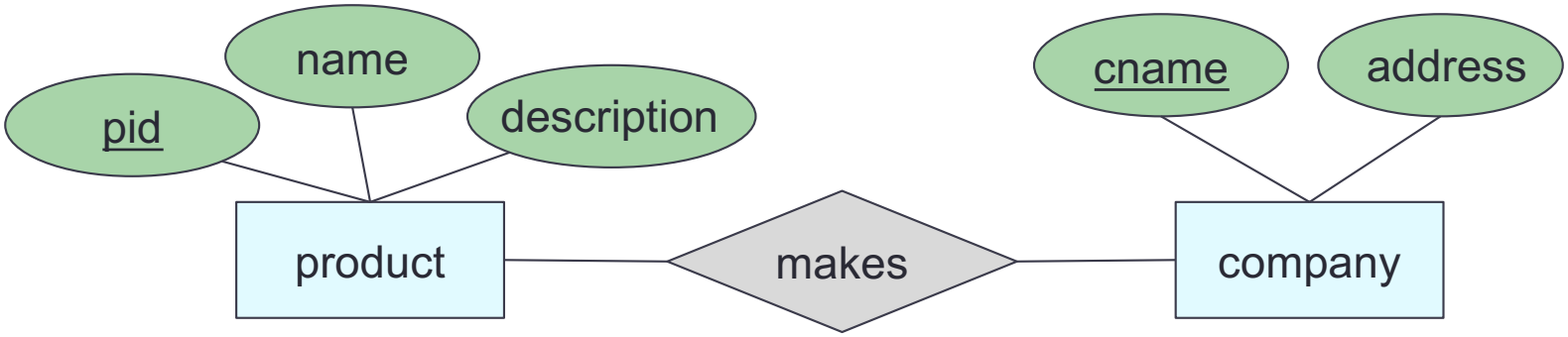
"Schema statement"

[ER to schema worksheet]

Strong Entity Set

Direct map:

- Entity name → relation name
- Attributes → columns
- Primary key: same as entity

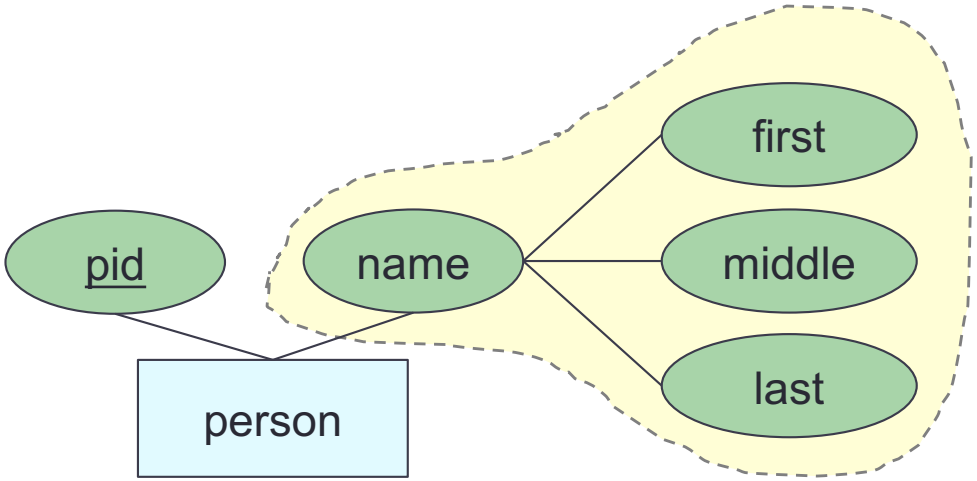


product(<u>pid</u>, name, description)
company(<u>cname</u>, address)
makes(<u>cname</u>, <u>pid</u>)

Strong Entity Set with Composite Attribute

Create separate attributes for each component

Don’t include the higher level attribute



person(pid, first_name, middle_name, last_name)

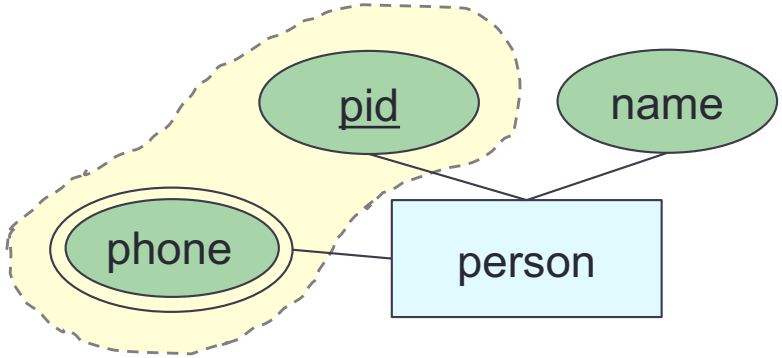
[ER to schema worksheet]

Strong Entity Set with Multivalued Attribute

Create a separate table for the multivalued attribute

Name the table with the concatenation, separated by "_"
entityname_attributename

Primary key: all attributes



person(<u>pid</u>, name)

person_phone(<u>pid</u>, <u>phone_number</u>)

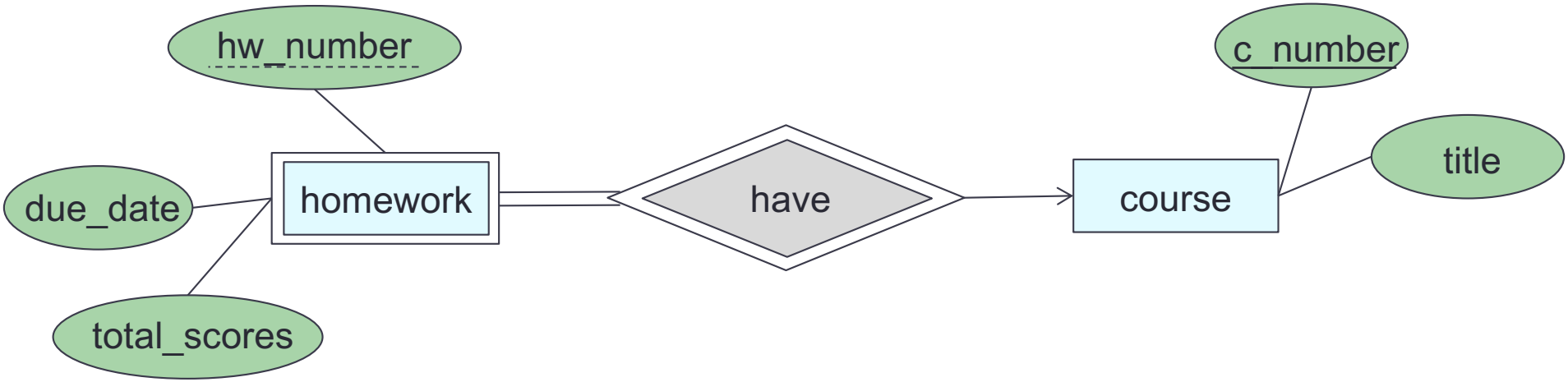
[ER to schema worksheet]

Weak Entity Set

Let *A* be a weak entity set and *B* be the identifying strong entity set on which *A* depends

Create a table with primary key of *B* and all *A*'s attributes

Primary key: primary key of *B* (strong entity) and
discriminator of *A*



course(c_number, title)

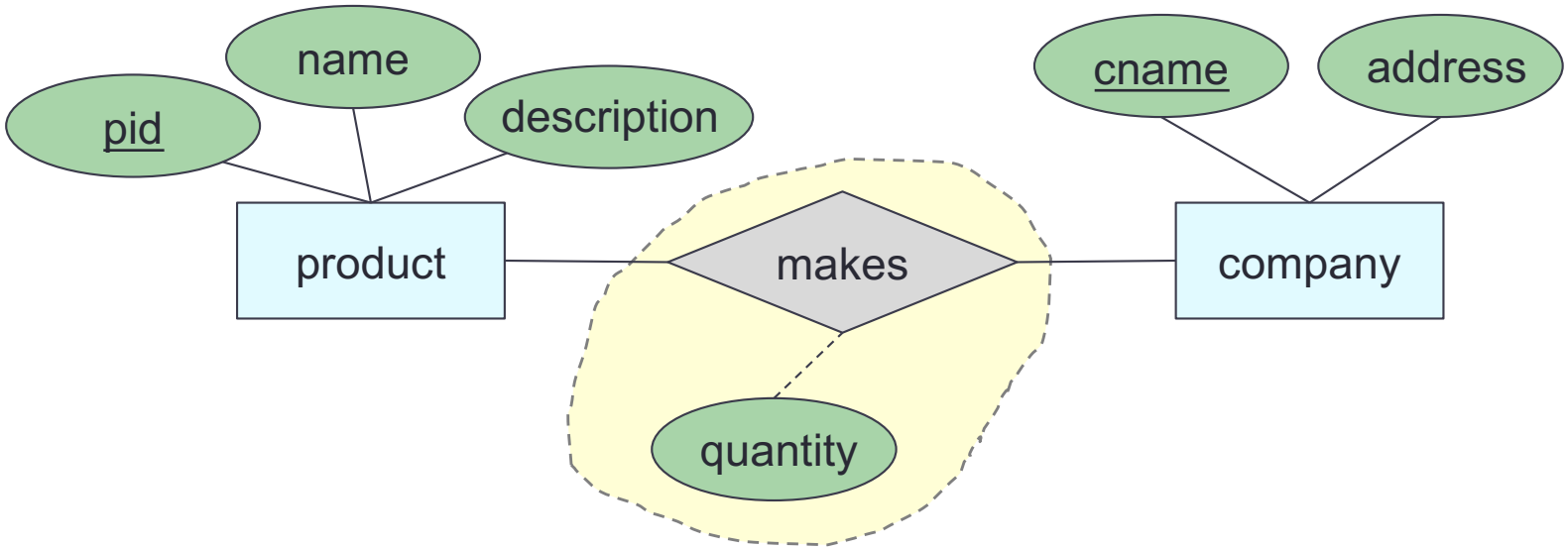
homework(c_number, hw_number, due_date, total_scores)

[ER to schema worksheet]

Relationship Set: Many-to-Many

Table: primary keys of both participating entity sets and any attributes on the relationship itself

Primary key: primary keys of both participating entity sets



product(pid, name, description)
company(cname, address)
makes(pid, cname, quantity)

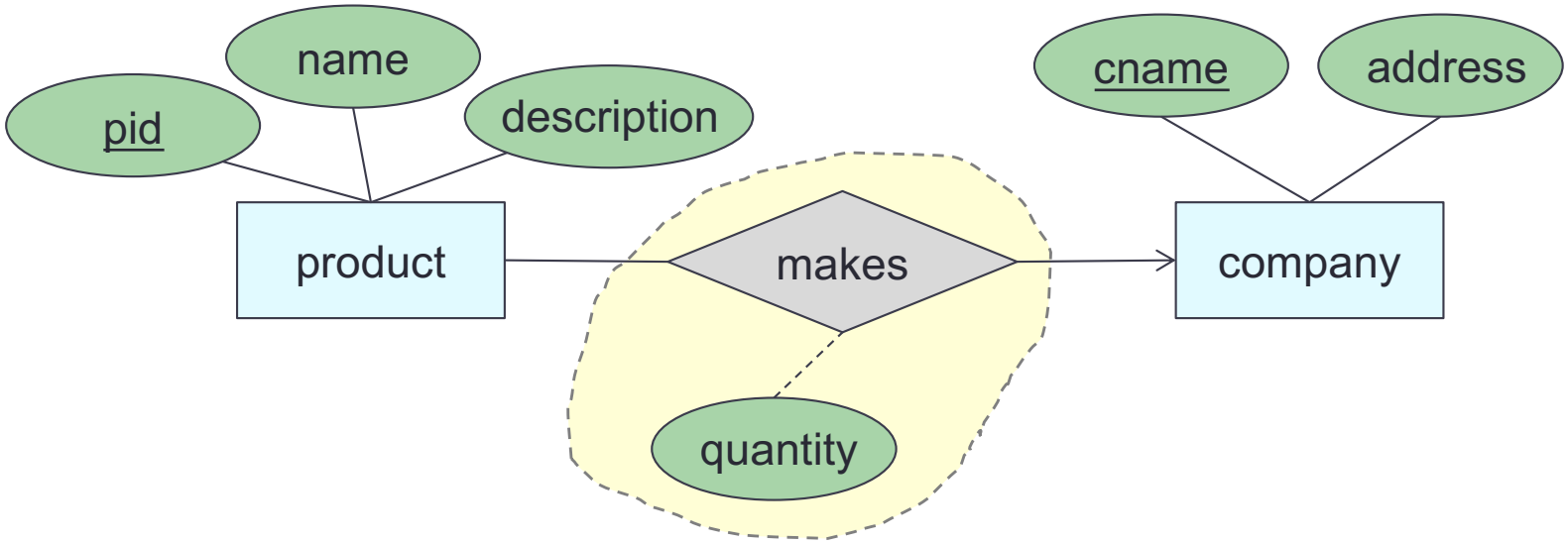
Primary keys of both entities

[ER to schema worksheet]

Relationship Set: Many-to-One / One-to-Many

Table: primary keys of both participating entity sets and any attributes on the relationship itself

Primary key: primary keys of the entity set on the "many" side



product(pid, name, description)
company(cname, address)
makes(pid, cname, quantity)

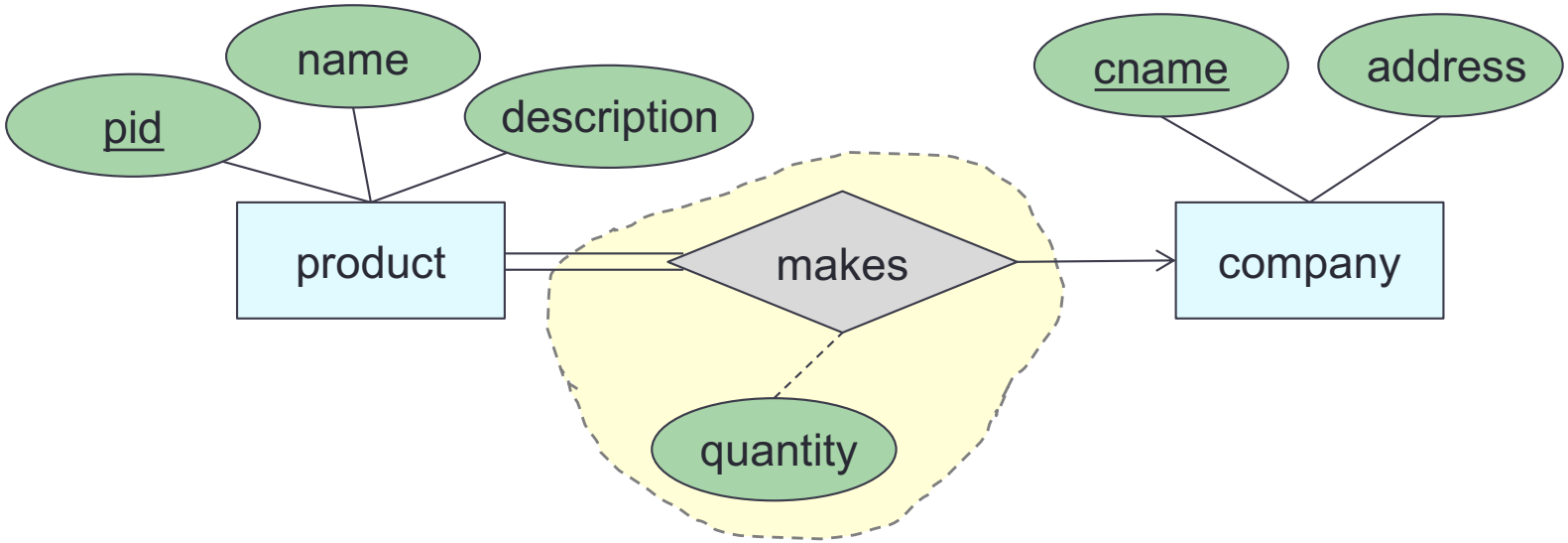
Primary key of the "many" side

[ER to schema worksheet]

Relationship Set: Total Participation

Because the total participation requires all entity to be participated in the relationship

→ add the primary key of the "one" side to the "many" side entity set, no table for relationship needed



product(<u>pid</u>, name, description, cname, quantity)
company(<u>cname</u>, address)

[ER to schema worksheet]

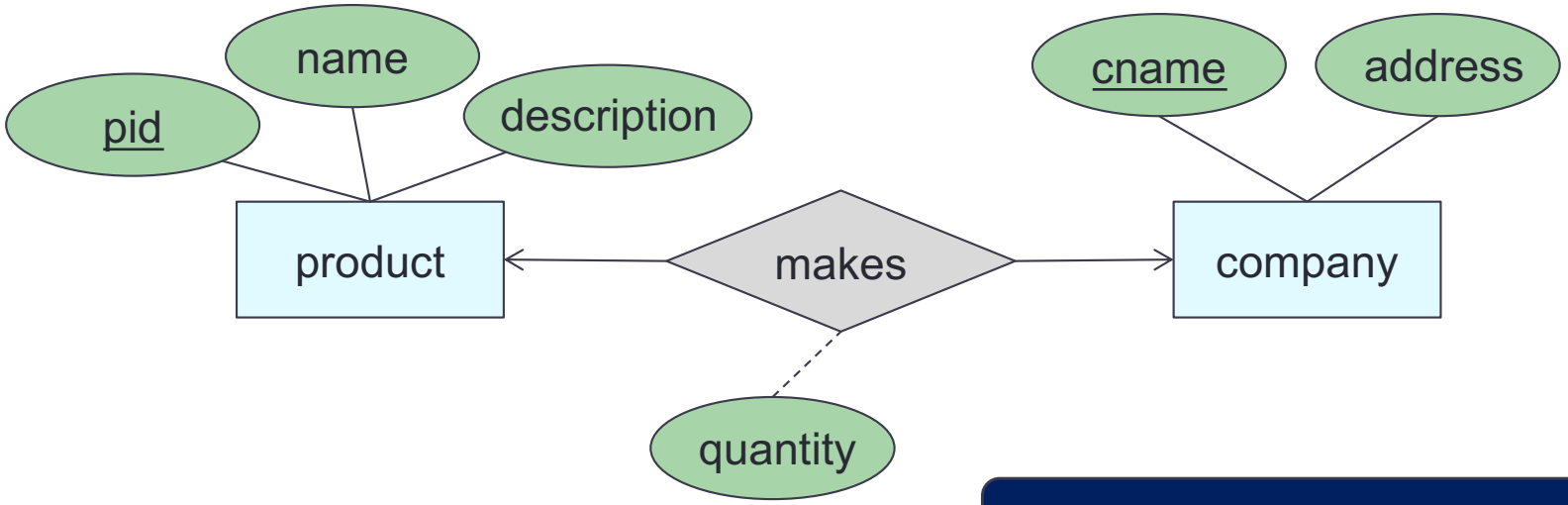
Relationship Set: One-to-One

Table: Either side can be used as the main table

(Which side? doesn't matter. Pick the one that makes the most sense)

Add the other side's primary key to it

Primary key: primary keys of the entity set you pick

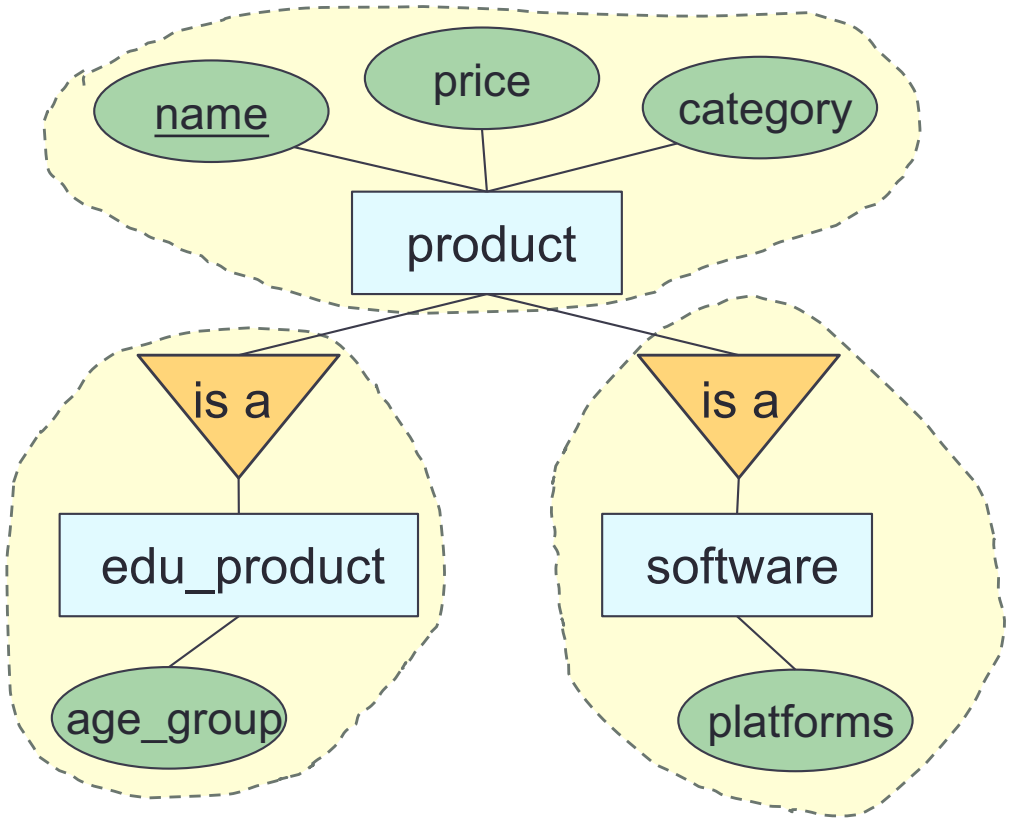


Primary key of the chosen entity

product(pid, name, description)
company(cname, address, pid, quantity)

[ER to schema worksheet]

Subclass (Option 1)



Keep everything

Primary key of the lower level entity set: from the higher level

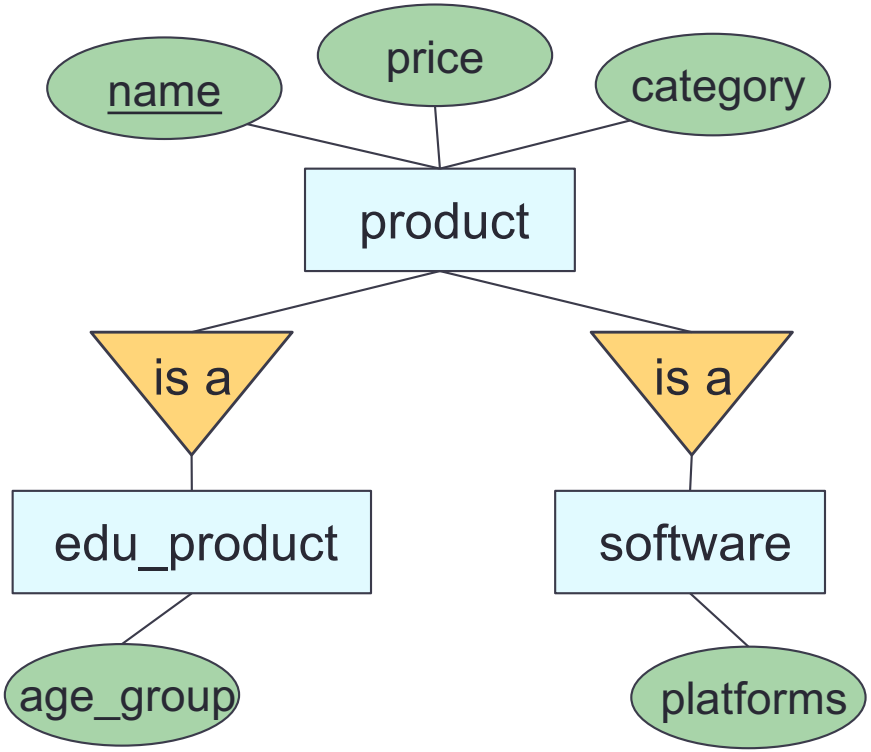
Drawback: need to access more tables to get info about the lower levels

product(<u>name</u>, price, category)
edu_product(<u>name</u>, age_group)
software(<u>name</u>, platforms)

Keep everything

[ER to schema worksheet]

Subclass (Option 2)



Keep specialization entity sets

No table for generalization entity set

Primary key of the lower level entity set: from the higher level

Drawback: redundancy if entities have more than one specialization

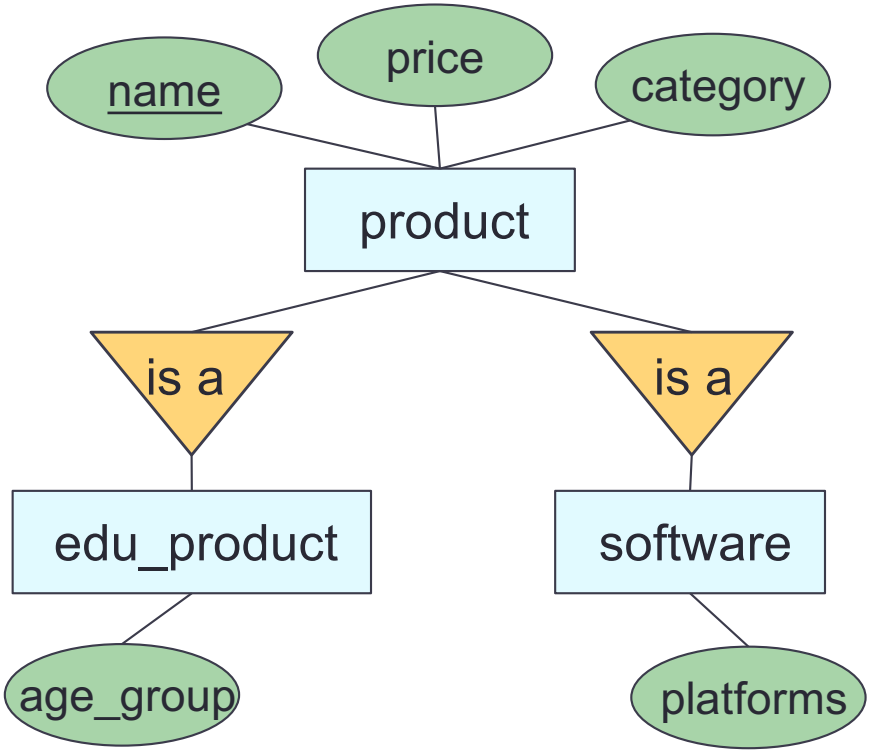
edu_product(name, price, category, age_group)

software(name, price, category, platforms)

Push down

[ER to schema worksheet]

Subclass (Option 3)



Keep generalization entity set

No table for specialization entity sets

Drawback: NULL in attributes from specialization entity sets

Although less duplication of data, need to handle NULL value

product(name, price, category, age_group, platforms)

Push up

[ER to schema worksheet]

Subclass: Design Decision

Depending on the number of attributes of the generalization entity set and specialization entity set

- If balanced → do option 1 (create all)
- If more attributes in specialization → do option 2
- If more attributes in generalization → do option 3

In general, design decision depends on

- The number of attributes
- DB administrator's decision

Overall goal: minimize duplication
(there is no one correct way)

Wrap-Up

- Roles in Relationships
- Relationships: binary, n-ary
- Weak entity
- Subclasses
- Converting from E-R diagrams to relational designs
  - Turn each entity set into a relation with the the same set of attributes
  - Replace a relationship by a relation whose attributes are the keys for the connected entity sets
  - Weak entity sets cannot be translated straightforwardly to relations
  - "Is a" relationships and subclasses require careful treatment

What's next?

- Apply the concept to database scenarios and fine-tuning database structure