



CSC7072: Databases, fall 2015

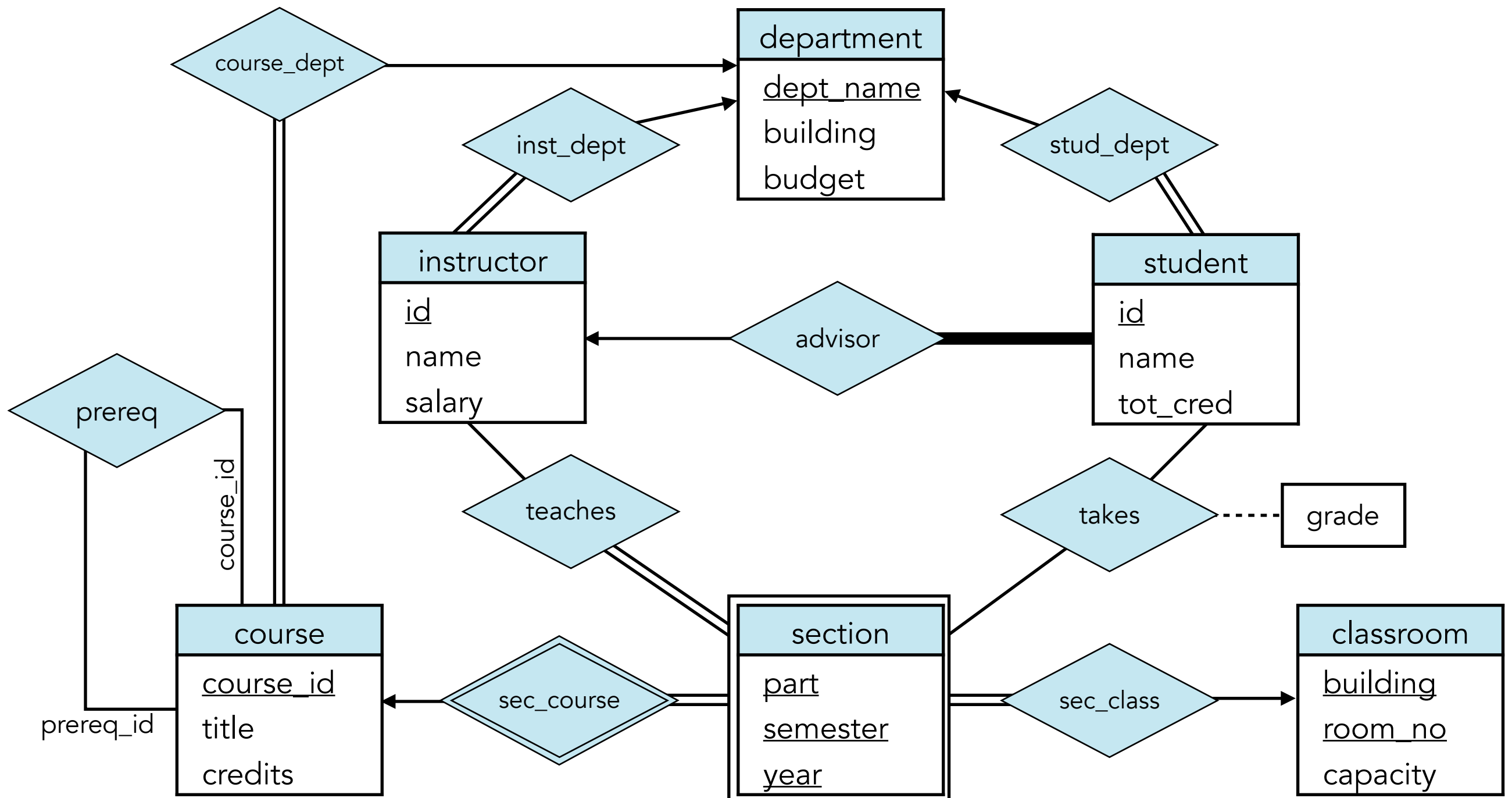
Dr. Kim Bouters



Converting ER to Relation Schemas

Converting ER to Relation Models

example: student ER model



Converting ER to Relation Models



general algorithm for converting an ER model

general approach, convert:

- ① strong entity sets;
- ② weak entity sets;
- ③ relationship sets;
- ④ reduction (of redundancy) of relationship sets;
- ⑤ multivalued attributes;
- ⑥ specialisations.

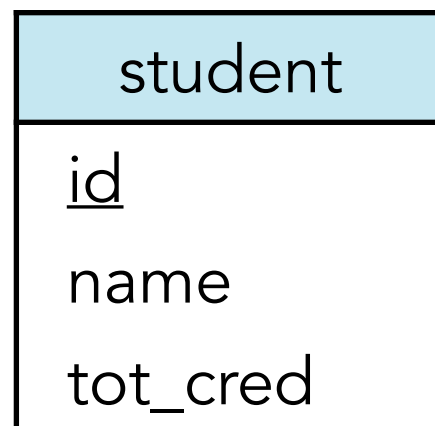
do remember: the quality of the conversion depends (greatly!)
on the quality of the original ER model

Converting ER to Relation Models

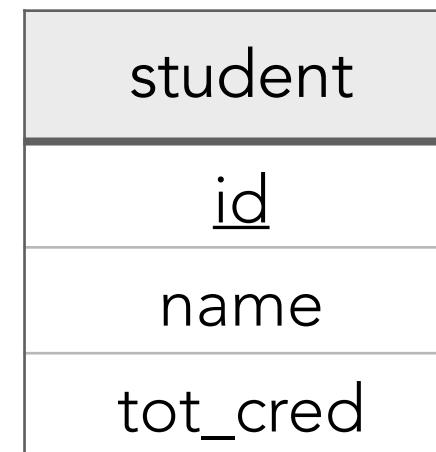
strong entity, simple attributes

strong entity set with simple attributes

trivial conversion!



same attributes
same PK



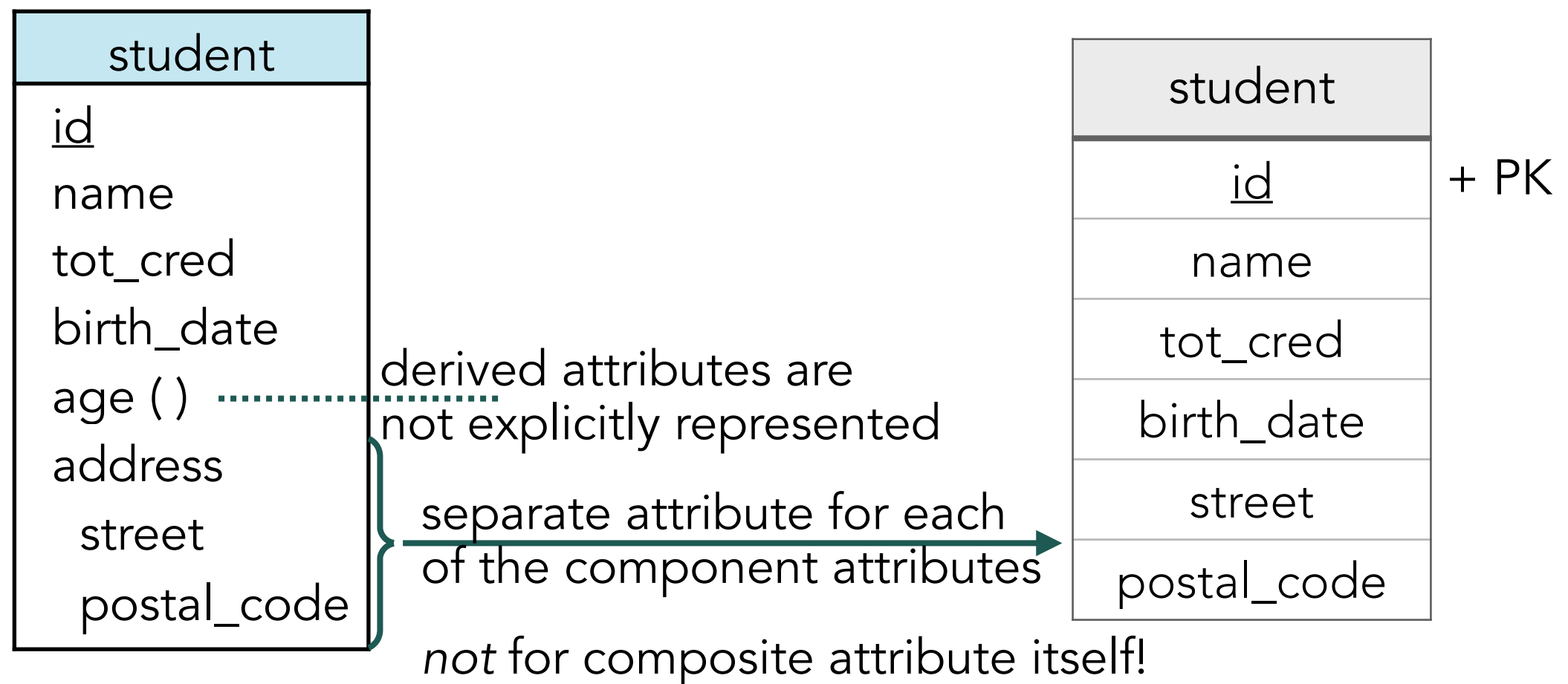
+ PK constraint

Converting ER to Relation Models

strong entity, composite and derived attributes

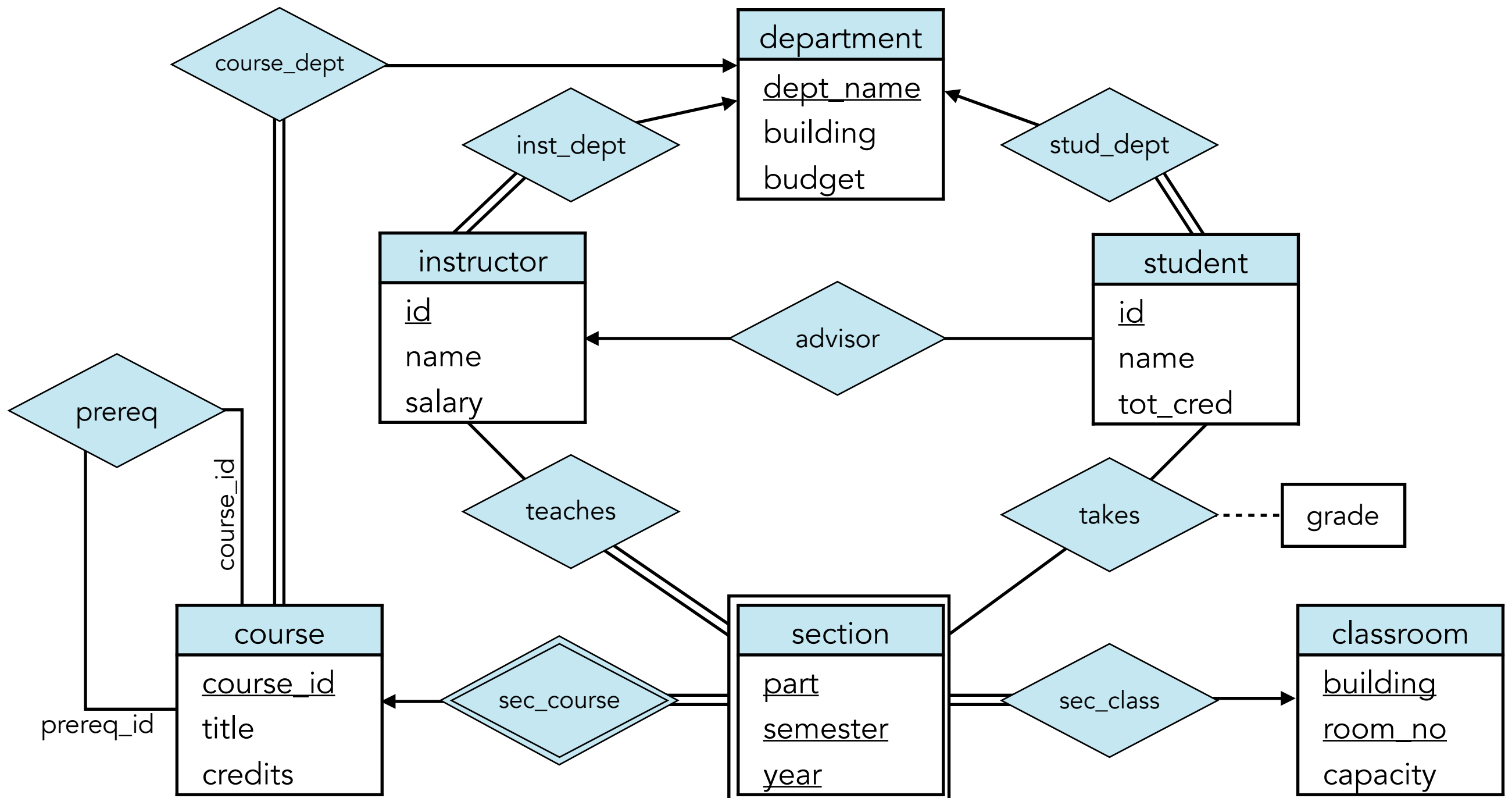
strong entity set with non-simple attributes

composite and derived attributes:



Converting ER to Relation Models

example: student ER model



Converting ER to Relation Models

example: student ER model

schemas for strong entity sets:

classroom(building, room_number, capacity)

instructor(id, name, salary)

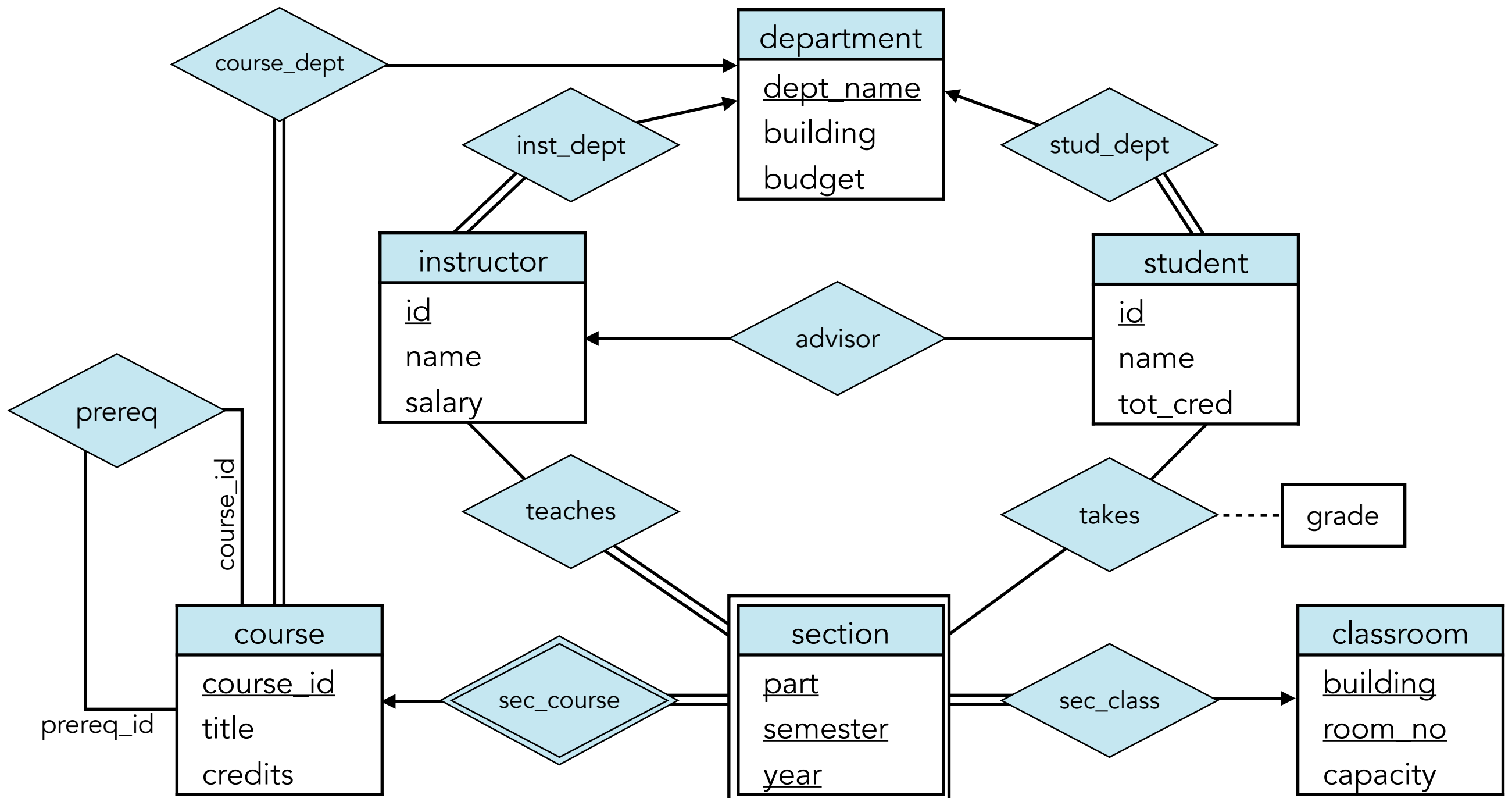
student(id, name, total_cred)

course(course_id, title, credits)

department(dept_name, building, budget)

Converting ER to Relation Models

example: student ER model

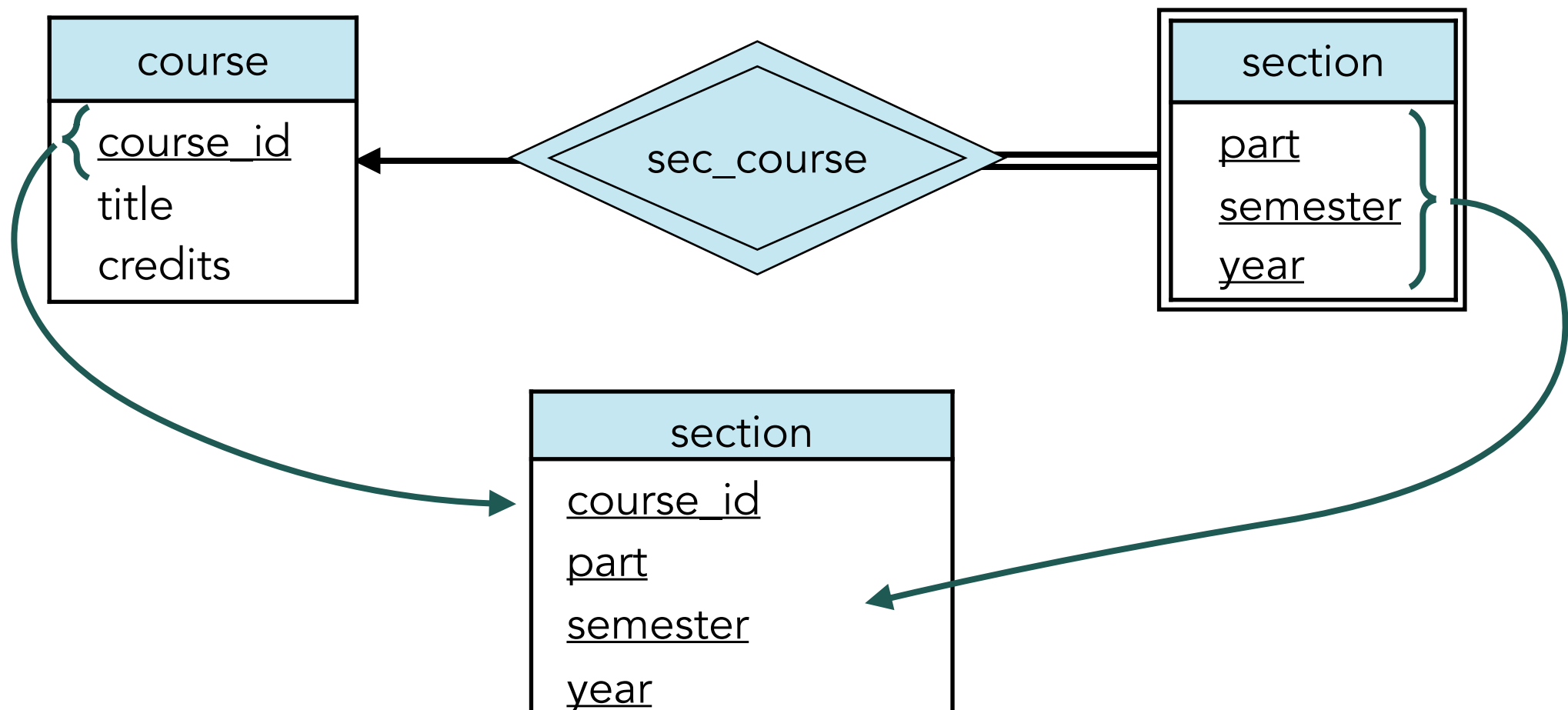


Converting ER to Relation Models

weak entity sets

weak entity sets

becomes a table that includes (a) column(s) for the primary key of the identifying strong entity set



Converting ER to Relation Models

example: student ER model

schemas for weak entity sets:

section(course_id, part, semester, year)

Converting ER to Relation Models



general algorithm for converting an ER model

general approach, convert:

- ① strong entity sets;
- ② weak entity sets;
- ③ relationship sets;
- ④ reduction (of redundancy) of relationship sets;
- ⑤ multivalued attributes;
- ⑥ specialisations.

do remember: the quality of the conversion depends (greatly!)
on the quality of the original ER model

Converting ER to Relation Models

relationships: basics

relationships

follows a basic idea irrespective of cardinality:

- add attributes for the primary keys of both relations
- add any descriptive attributes of the relationship set

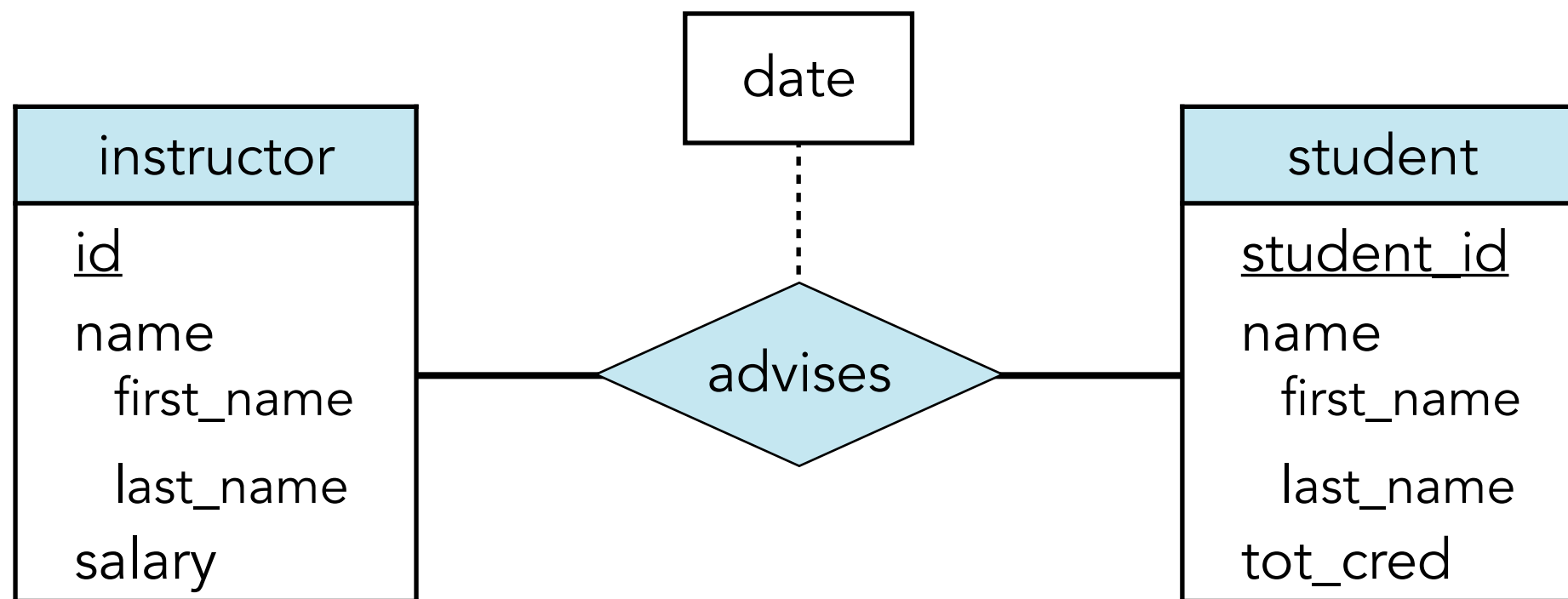
+ special rules depending on cardinality

Converting ER to Relation Models

relationships: many-to-many

relationships: many to many

the combination of the two PKs is the PK of the new table



new table: advisor(instructor_id, student_id, date)

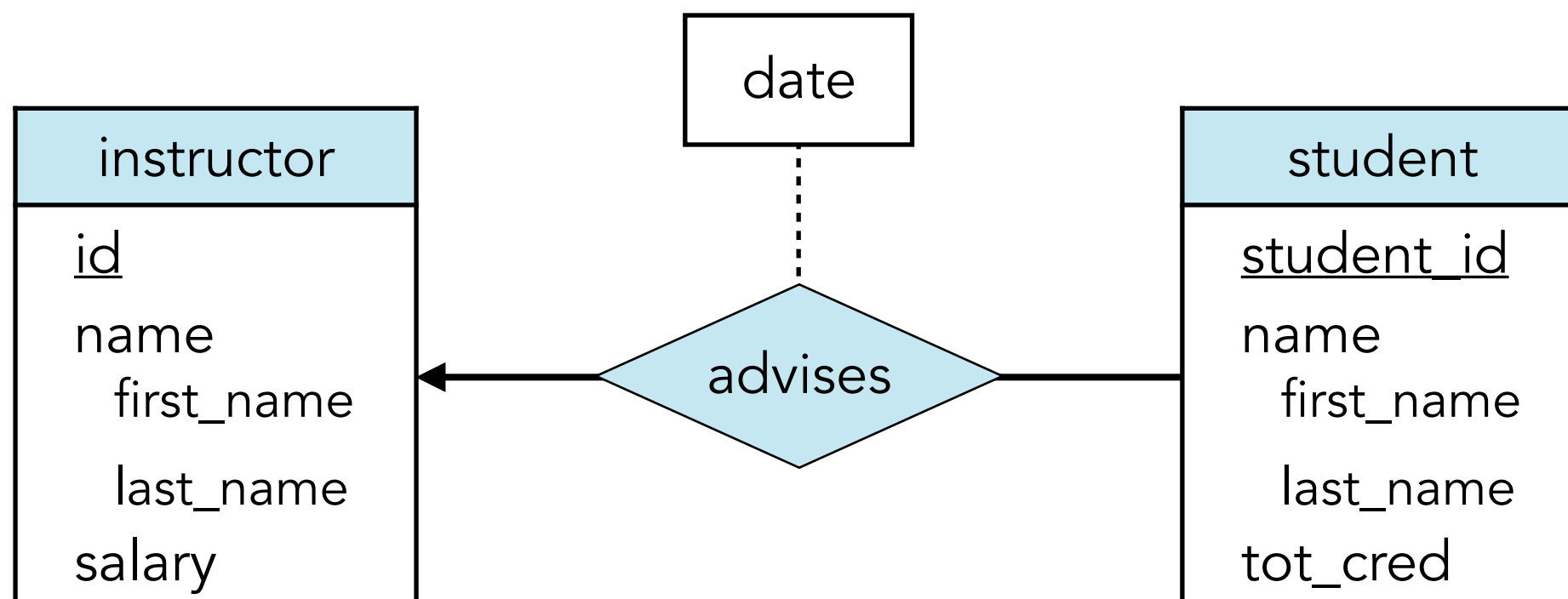
don't forget foreign key constraints!

Converting ER to Relation Models

relationships: many-to-one or vice versa

relationships: many to one, or, one to many

the PK of the new table is PK on the many side



new table: advisor(student_id, instructor_id, date)

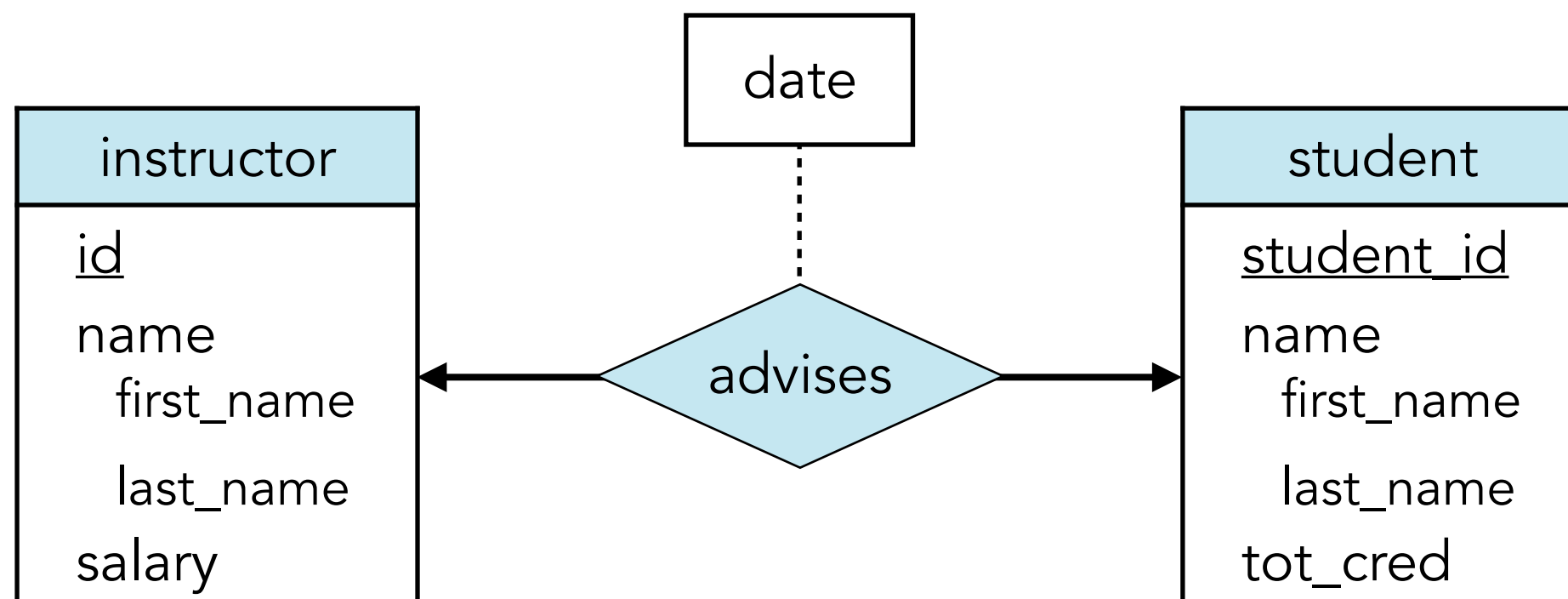
don't forget foreign key constraints!

Converting ER to Relation Models

relationships: one-to-one

relationships: one to one

the PK of the new table is PK of either one of the sides

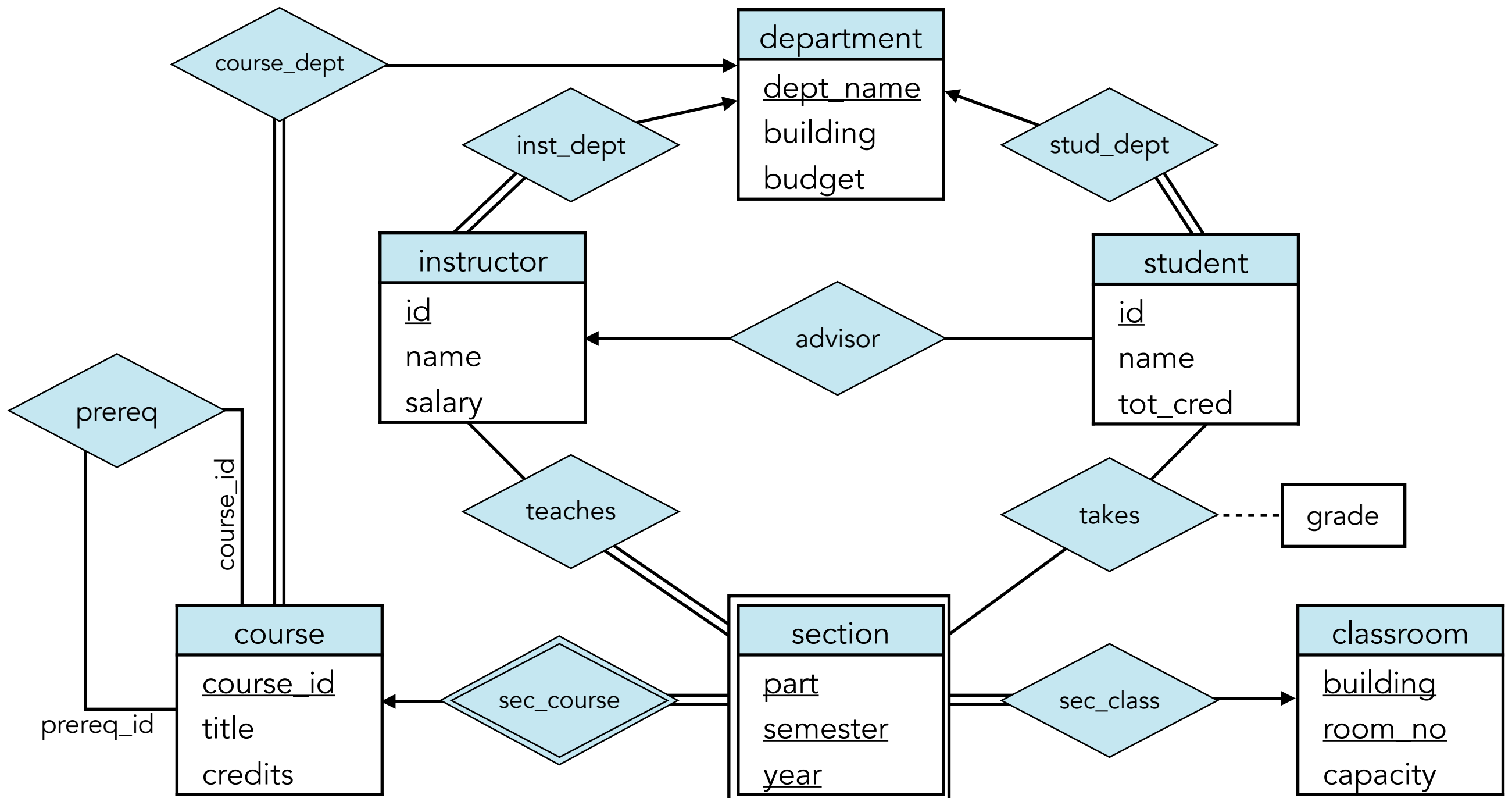


new table: advisor(student_id, instructor_id, date)
or: advisor(instructor_id, student_id, date)

FK!

Converting ER to Relation Models

example: student ER model



Converting ER to Relation Models

example: student ER model

non-redundant schemas for relationship sets:

teaches(instructor_id, course_id, part, semester, year)

takes(student_id, course_id, part, semester, year, grade)

prereq(course_id, prereq_id)

advisor(student_id, instructor_id)

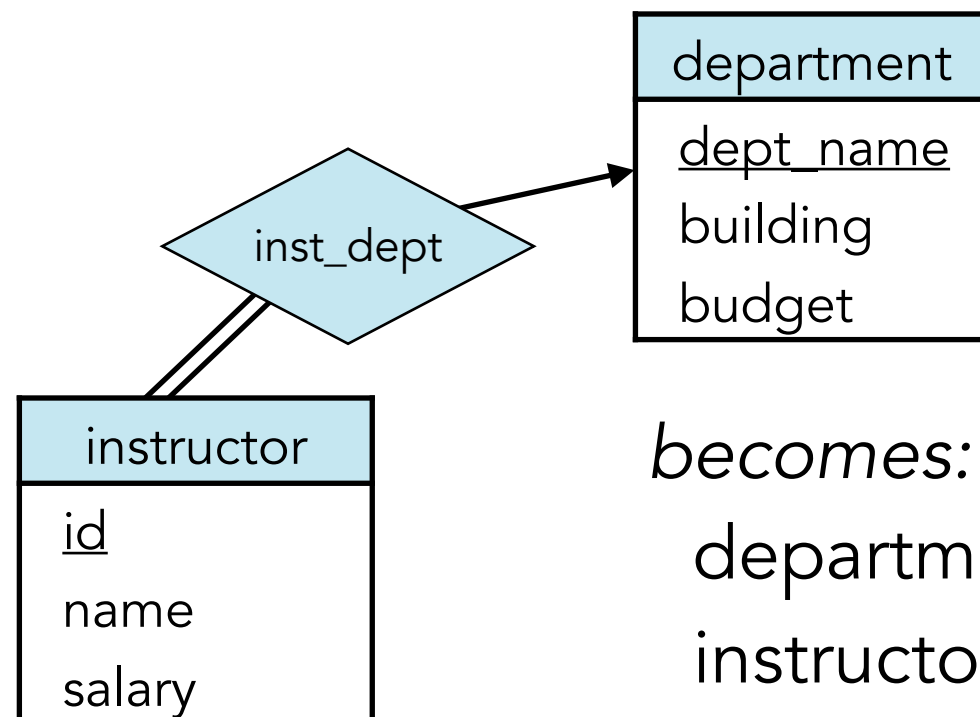
Converting ER to Relation Models

relationships: resolving redundancy

reduction of relationships: many to one, or, one to many

if the participation is total on the *many* side, then redundancy

instead: simply add attribute to the *many* side
consisting of PK on the *one* side



only when total!

otherwise we need undesired *null* values

becomes:

department(dept_name, building, budget)

instructor(id, name, salary, dept_name)

Converting ER to Relation Models

relationships: resolving redundancy cont.

reduction of relationships: one to one

either side can act as the *many* side

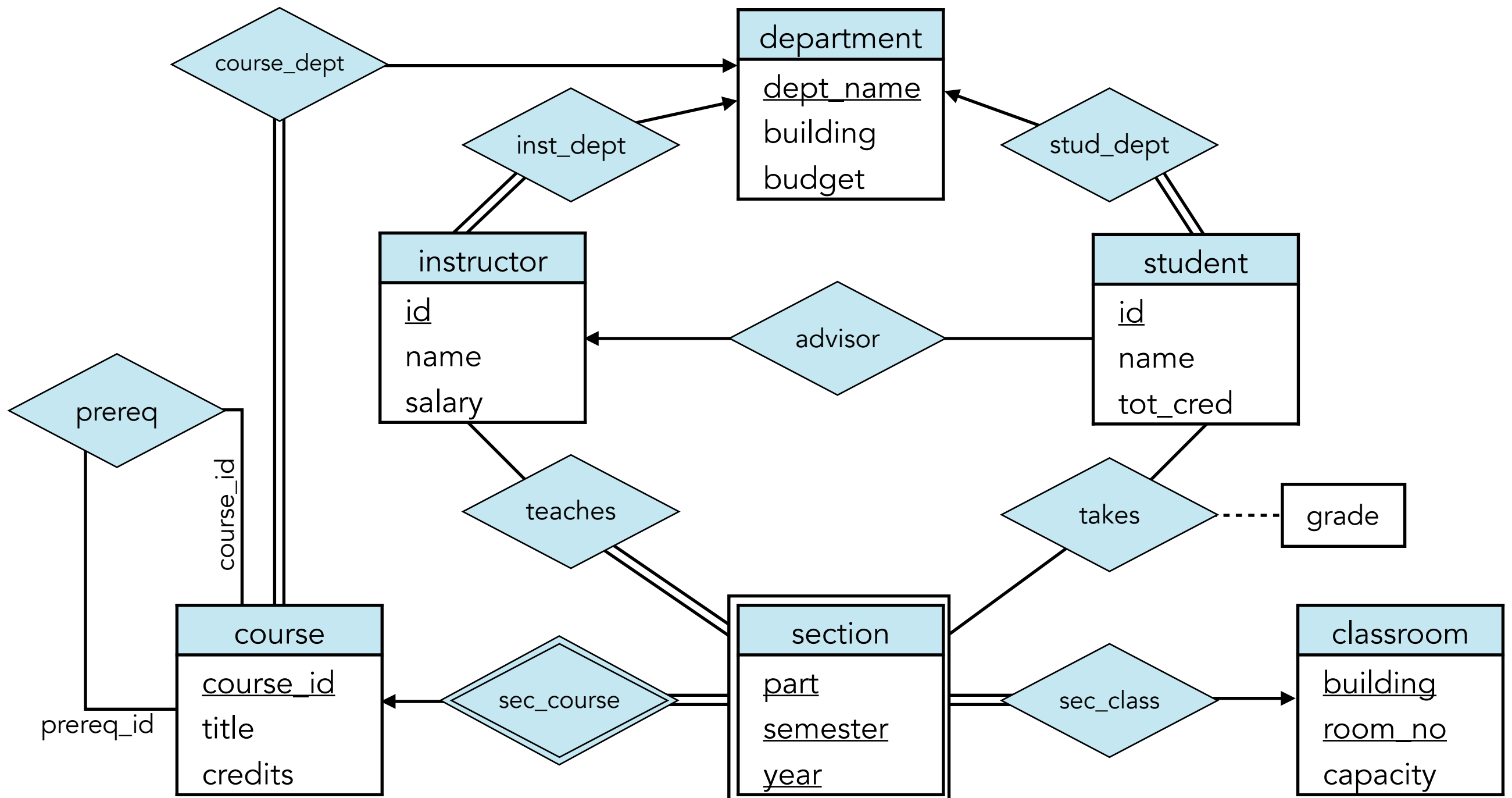
hence: if participation of one side is total, then redundancy

either new table can be chosen to
add PK attribute of other table

finally, note how relationship of weak entity set is also redundant,
so it is never explicitly added

Converting ER to Relation Models

example: student ER model



Converting ER to Relation Models

example: student ER model

redundant schemas for relationship sets:

course_dept(course_id, dept_name)

inst_dept(instructor_id, dept_name)

stud_dept(student_id, dept_name)

sec_course(course_id, part, semester, year)

sec_class(course_id, part, semester, year, grade, building, room_no)

Converting ER to Relation Models

example: student ER model

reducing the *redundant* schemas:

course_dept(course_id, dept_name)

└→ course(course_id, title, credits, dept_name)

inst_dept(instructor_id, dept_name)

└→ instructor(id, name, salary, dept_name)

stud_dept(student_id, dept_name)

└→ student(id, name, credits, dept_name)

Converting ER to Relation Models

example: student ER model

reducing the *redundant* schemas:

~~course_dept(course_id, dept_name)~~

└→ course(course_id, title, credits, dept_name)

~~inst_dept(instructor_id, dept_name)~~

└→ instructor(id, name, salary, dept_name)

~~stud_dept(student_id, dept_name)~~

└→ student(id, name, credits, dept_name)

Converting ER to Relation Models

example: student ER model

reducing the *redundant* schemas:

sec_course(course_id, part, semester, year)

sec_class(course_id, part, semester, year, grade, building, room_no)

└→ section(course_id, part, semester, year, building, room_no)

recall that first one is a weak entity set;
clearly not needed on its own

Converting ER to Relation Models

example: student ER model

reducing the *redundant* schemas:

~~sec_course(course_id, part, semester, year)~~

~~sec_class(course_id, part, semester, year, grade, building, room_no)~~

└→ section(course_id, part, semester, year, building, room_no)

recall that first one is a weak entity set;
clearly not needed on its own

Converting ER to Relation Models



general algorithm for converting an ER model

general approach, convert:

- ➊ strong entity sets;
- ➋ weak entity sets;
- ➌ relationship sets;
- ➍ reduction (of redundancy) of relationship sets;
- ➔ ➎ multivalued attributes;
- ➏ specialisations.

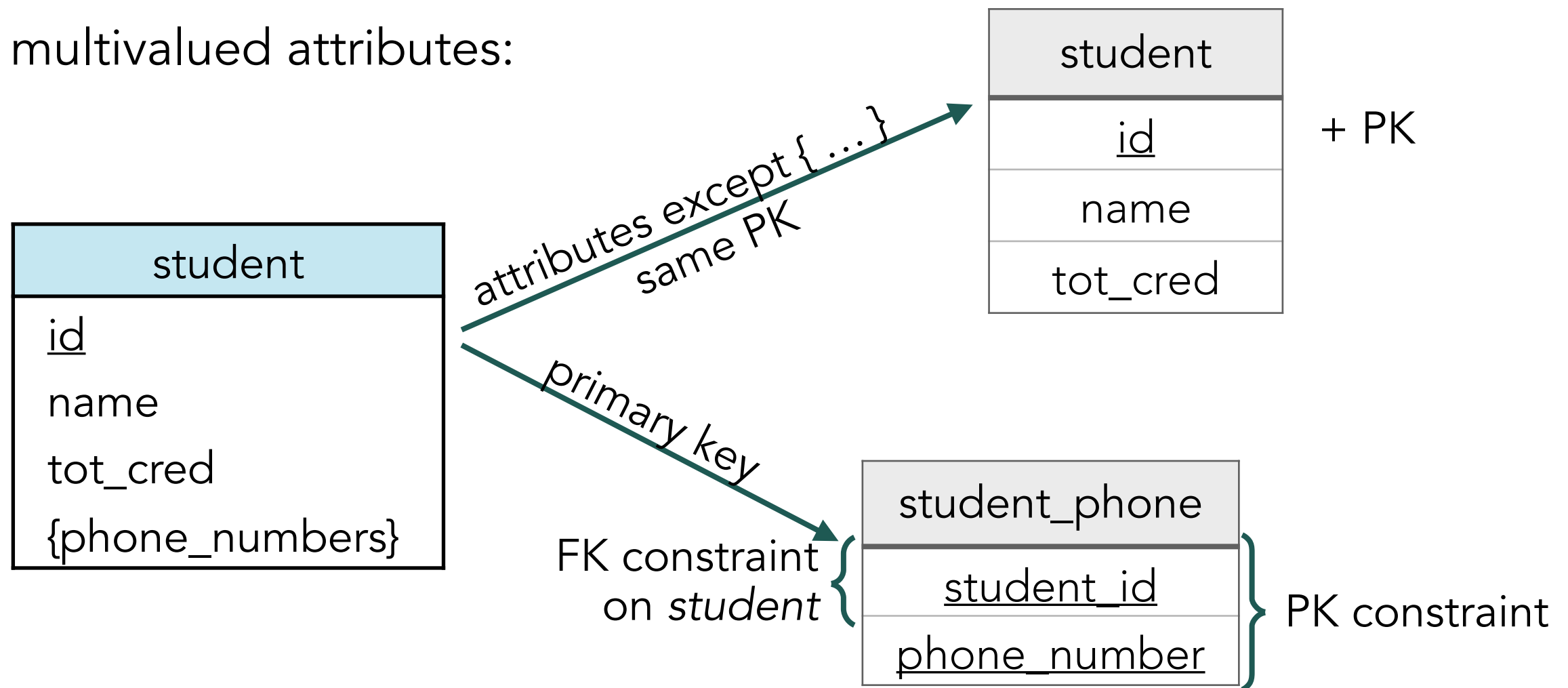
do remember: the quality of the conversion depends (greatly!)
on the quality of the original ER model

Converting ER to Relation Models

strong entity, multi-valued attributes

strong entity set with non-simple attributes

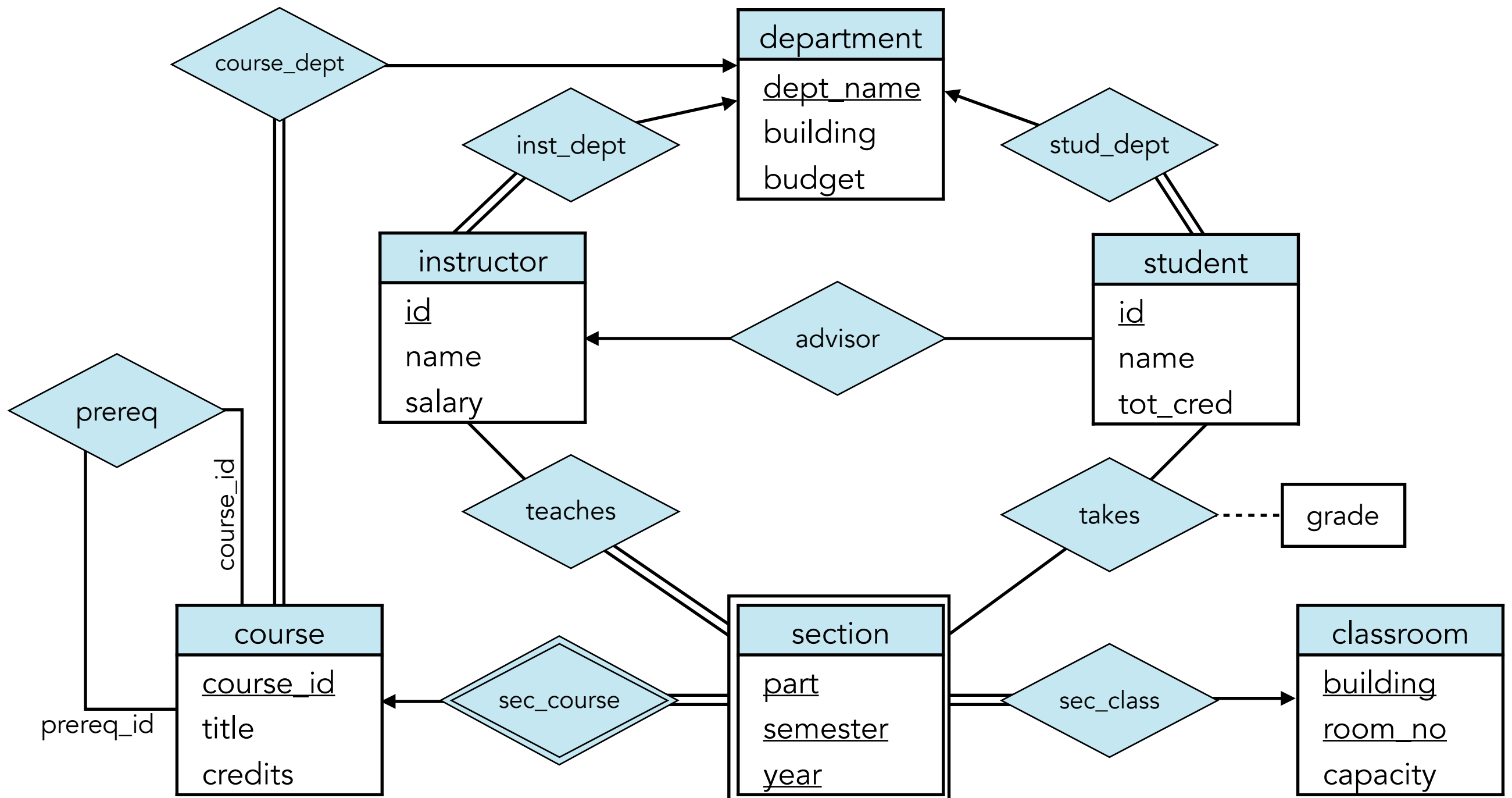
multivalued attributes:



careful: PK may consist of multiple attributes, then all are adopted!

Converting ER to Relation Models

example: student ER model



Converting ER to Relation Models

converting specialisations

converting specialisations

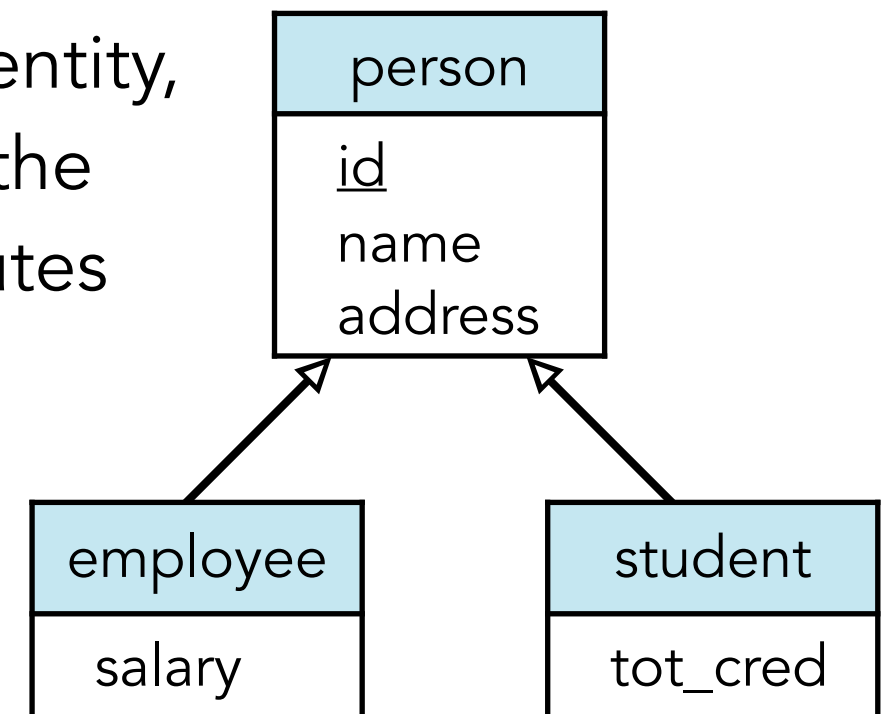
method 1:

- form a schema for the high-level entity (e.g. person)
- form a schema for every lower-level entity, including a FK to the primary key of the high-level entity and any local attributes

person(id, name, address)

employee(person_id, salary)

student(person_id, tot_cred)



drawback: getting information about lower-level entity requires accessing two relations (higher-level and lower-level)

Converting ER to Relation Models

converting specialisations

converting specialisations

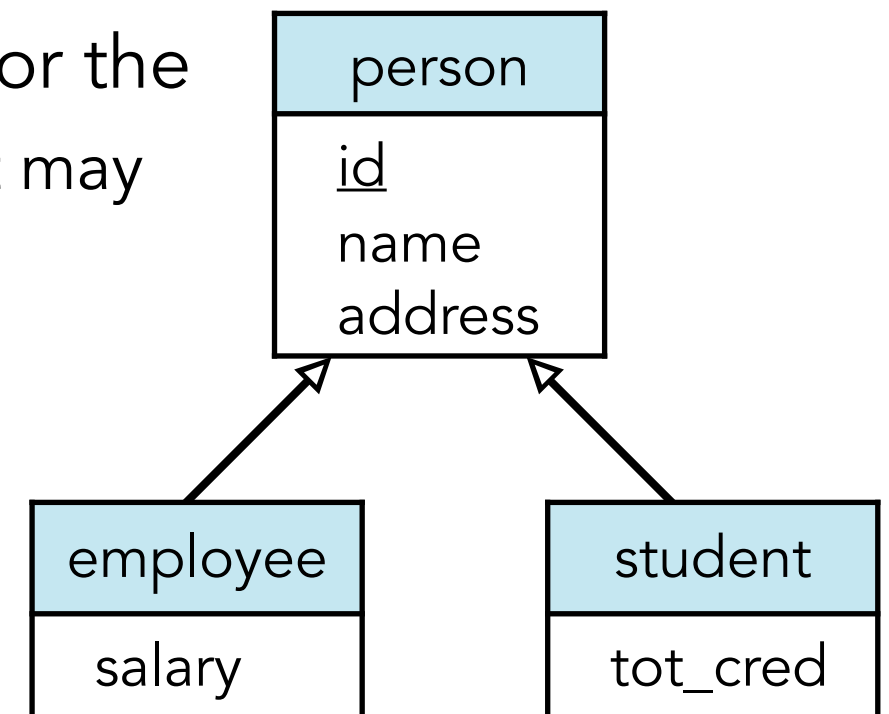
method 2:

- form a schema for each entity including all attributes
- if specialisation is total, the schema for the higher-level entity is not required (but may still be needed for FK constraints)

person(id, name, address)

employee(id, name, address, salary)

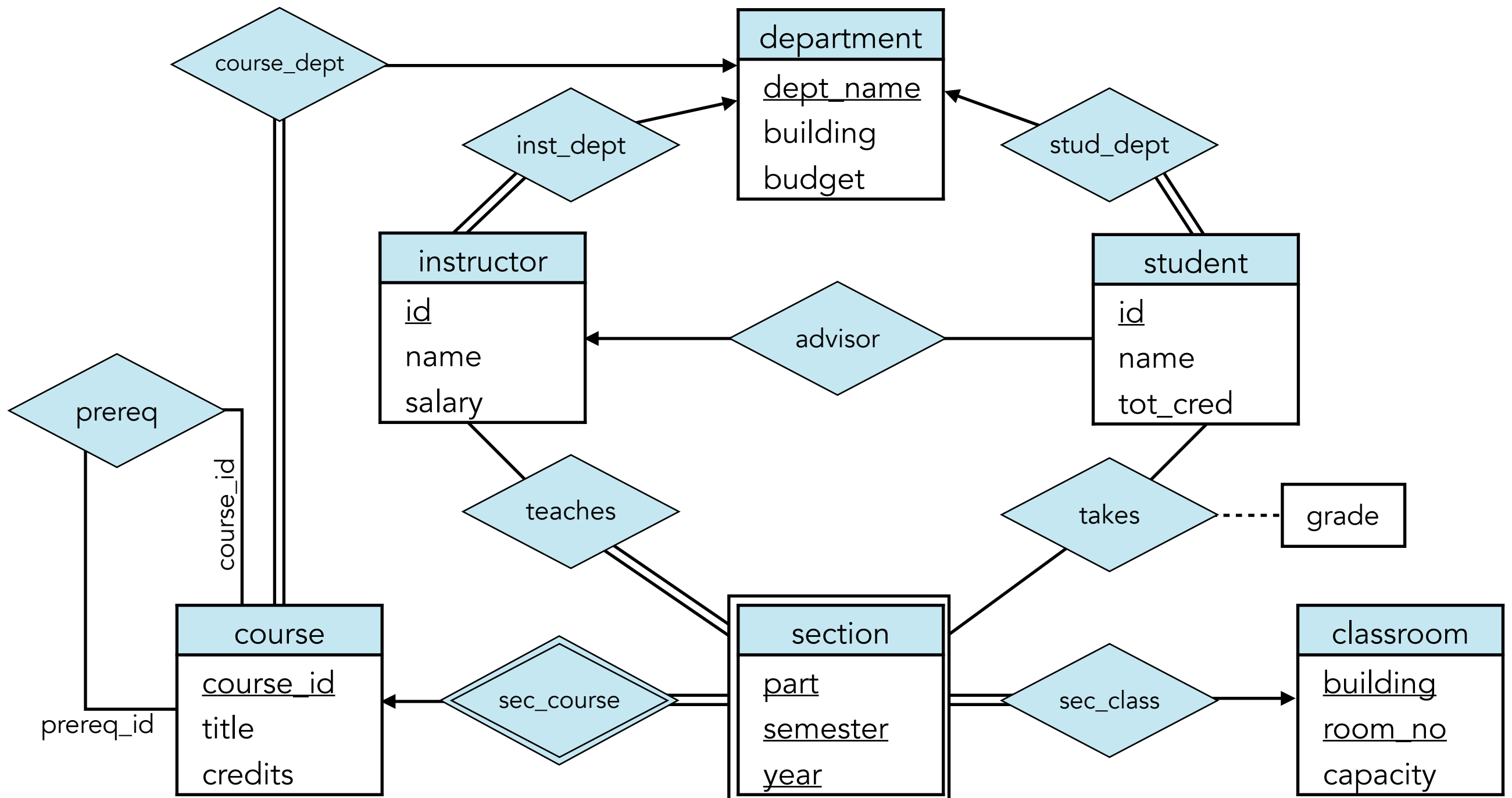
student(id, name, address, tot_cred)



drawback: information may be stored redundantly if not disjoint,
for example: a person is both employee and student

Converting ER to Relation Models

example: student ER model



Converting ER to Relation Models

example: student ER model

classroom(building, room_number, capacity)

department(dept_name, building, budget)

course(course_id, title, credits, dept^{FK}_name)

instructor(id, name, salary, dept^{FK}_name)

section(^{FK}course_id, part, semester, year, building^{FK}, room_no)

teaches(^{FK}instructor_id, course_id, part, semester, year)

student(id, name, credits, dept^{FK}_name)

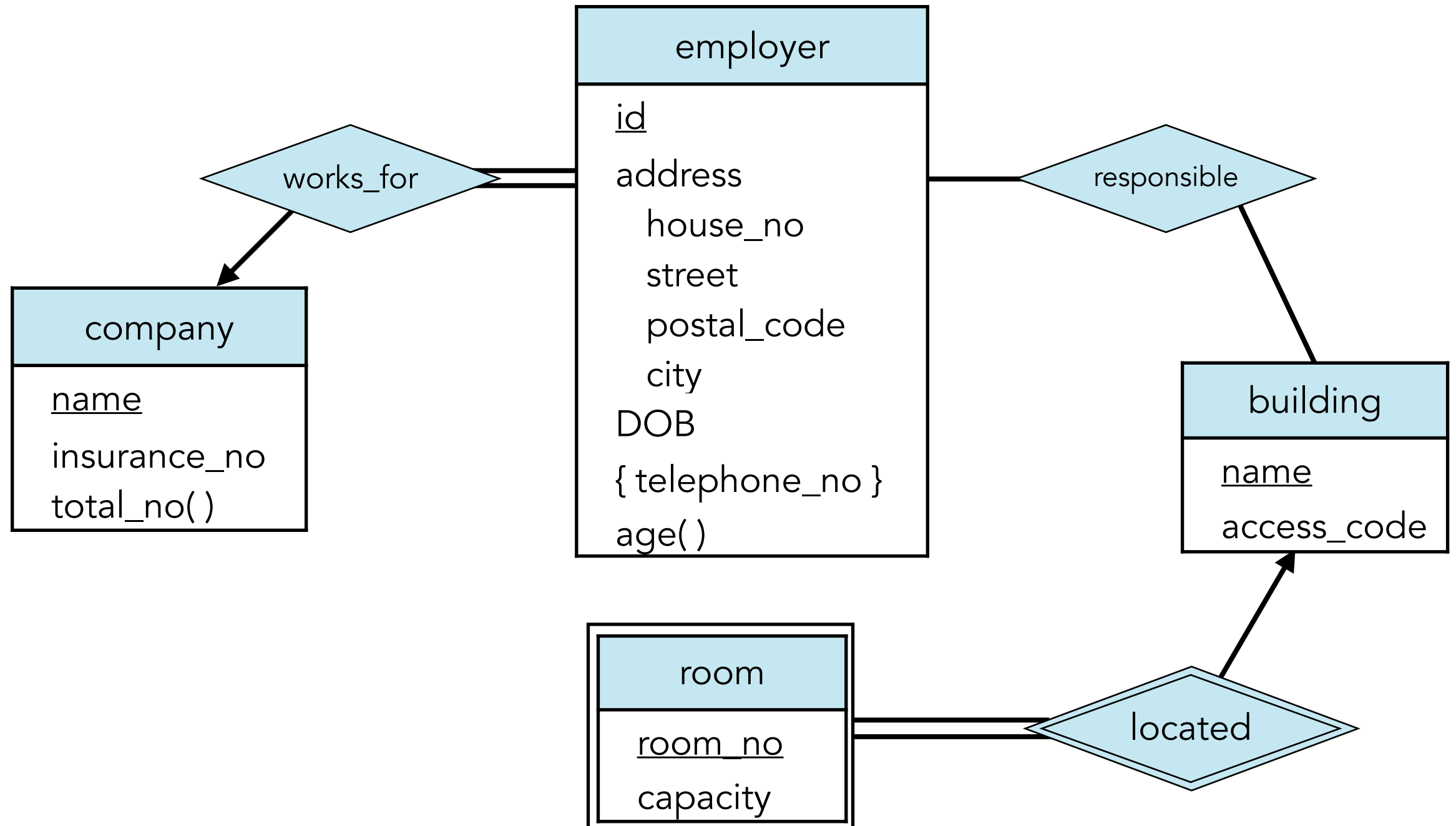
takes(^{FK}student_id, course_id, part, semester, year, grade)

advisor(^{FK}student_id, ^{FK}instructor_id)

prereq(^{FK}course_id, ^{FK}prereq_id)

Converting ER to Relation Models

exercise: ER conversion



Converting ER to Relation Models

exercise: ER conversion

schemas for strong entity sets:

company(name, insurance_no)

employer(id, house_no, street, postal_code,
city, dob, sex, {telephone-no})

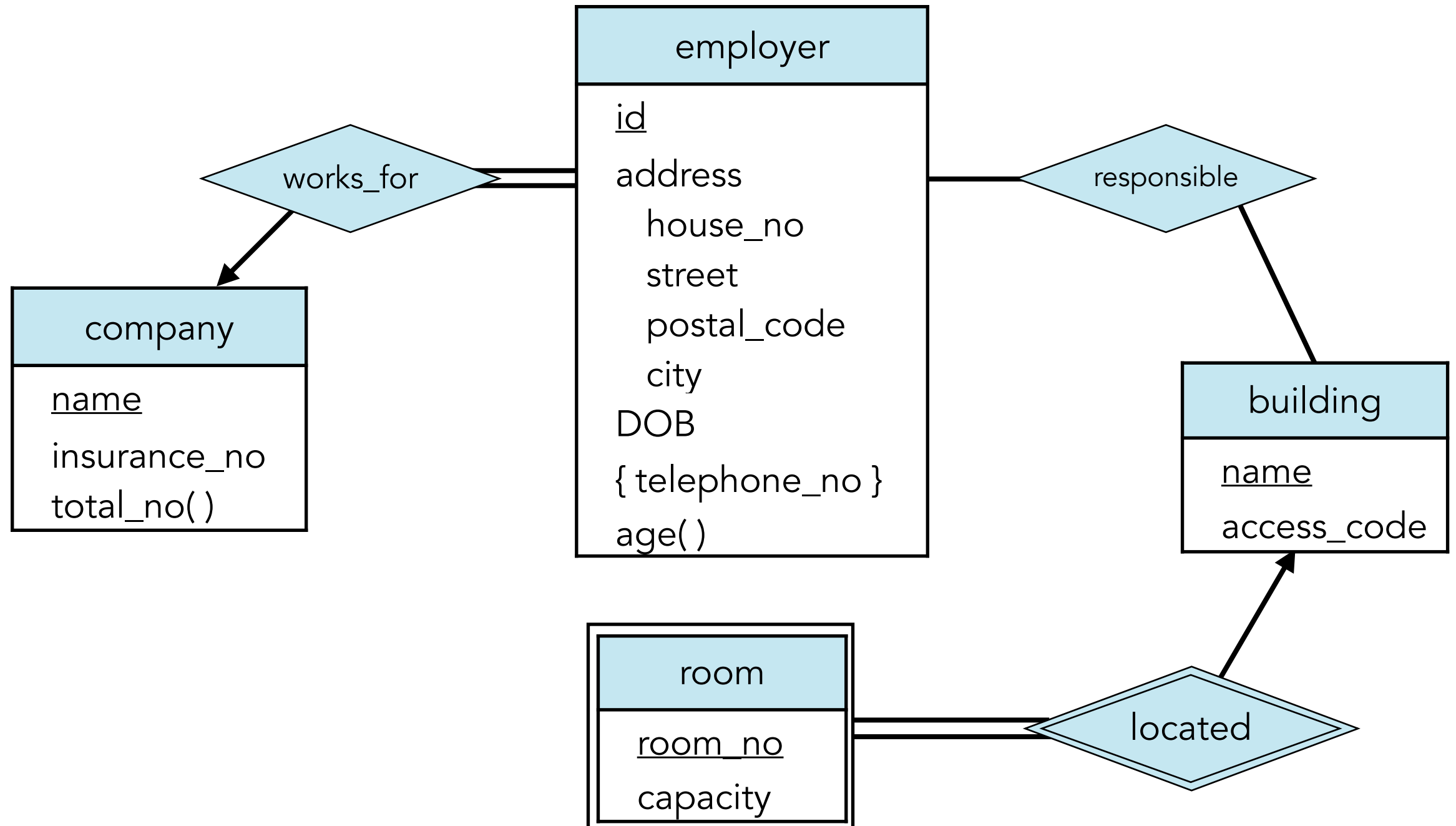
building(name, access_code)

schemas for weak entity sets:

room(building_name, room_no, capacity)

Converting ER to Relation Models

exercise: ER conversion



Converting ER to Relation Models

exercise: ER conversion

non-redundant schemas for relationship sets:

responsible(employer_id, building_name)

redundant schemas for relationship sets:

works_for(employer_id, company_name)

Converting ER to Relation Models

exercise: ER conversion

reducing the *redundant* schemas:

works_for(employer_id, company_name)

└→ employer(id, house_no, ..., company_name)

handling the *multivalued* attributes:

employer(id, house_no, ..., ~~{telephone_no}~~, ...)

employer_tel(employer_id, telephone_no)

Converting ER to Relation Models

example: student ER model

company(name, insurance_no)

employer(id, house_no, street, postal_code,
city, dob, sex, company^{FK}_name)

building(name, access_code)

room(^{FK}building_name, room_no, capacity)

responsible(^{FK}employer_id, ^{FK}building_name)

employer_tel(^{FK}employer_id, telephone_no)