Updated 12/7/2022

Topics 1-2: Design & SQL
L24: Practice

Wolfgang Gatterbauer

CS3200 Database design (fa22)

https://northeastern-datalab.github.io/cs3200/fa22s3/

12/7/2022

Pre-class conversations

- Recall exam 3 is next week MON Dec 12:
  - Possible design question: here is a design, improve it
  - tips: submit to the correct place on Gradescope, restart the laptop and open a new database to make sure no problems during the exam, scrap paper ok
- Project:
  - The value of active exercise vs passive consumption

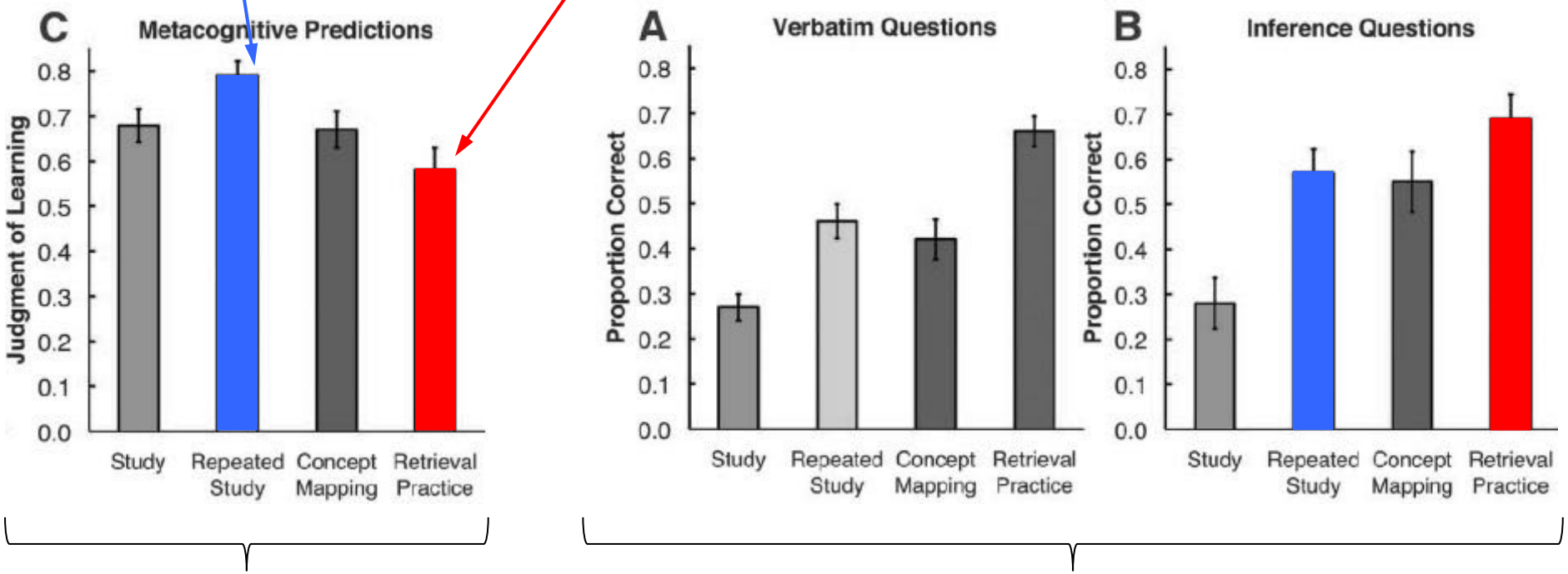
  - Bad practice: A creates, B and C "approve"
  - Good practice: A,B,C come to meeting with worked-out solutions, then they compare
  - Stop early today and available for project questions

- Practice today:
  - Normalization
  - ERD
  - SQL

Studying new material: "Under which study condition do you think you learn better?"

passive reading

active Q&A



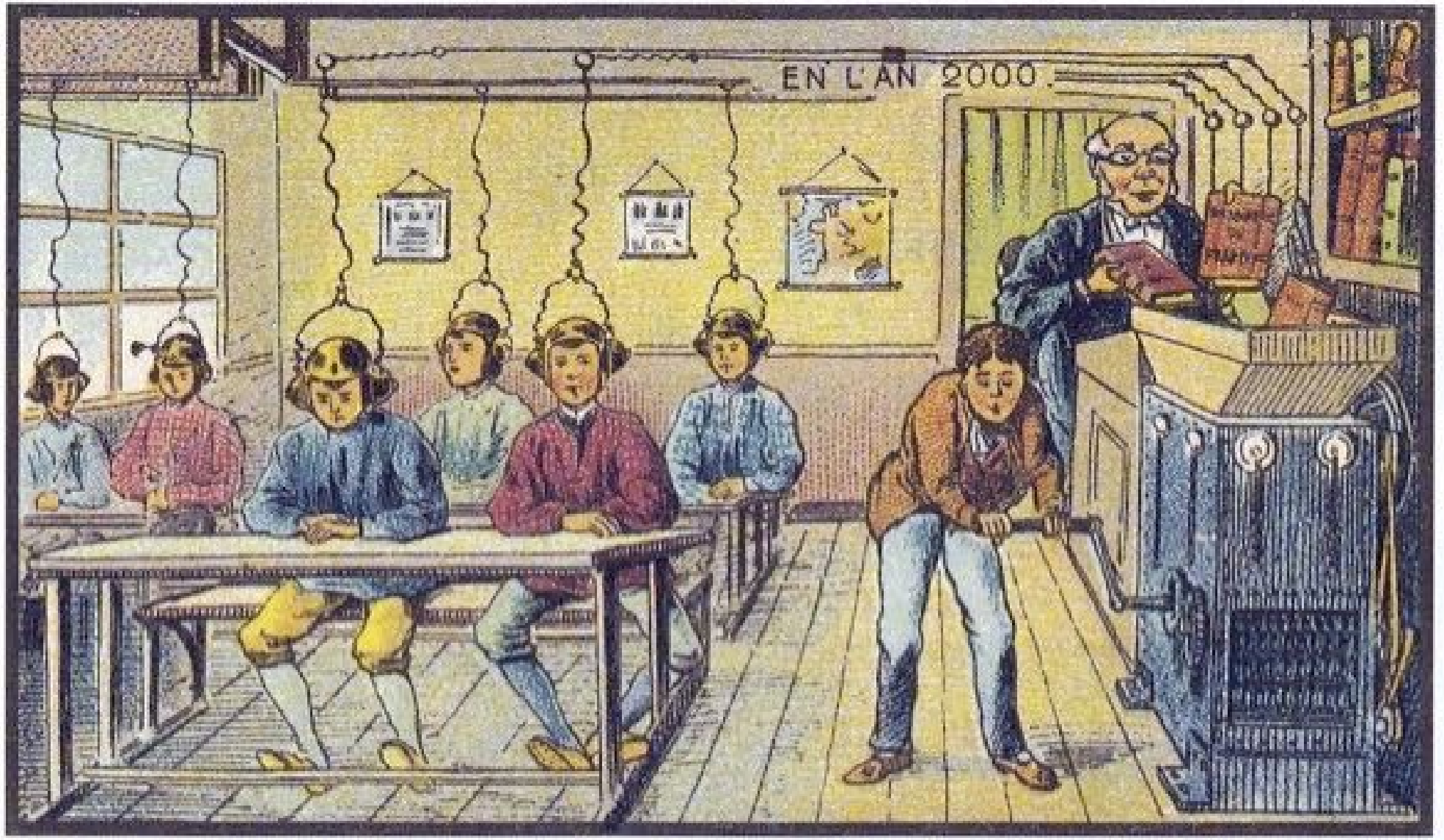
Judged performance
(=what people think)

Actual performance
(=what is actually working)

Data from: Karpicke & Blunt, "Retrieval Practice Produces More Learning than Elaborative Studying with Concept Mapping," Science, 2011. https://doi.org/10.1126/science.1199327

Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

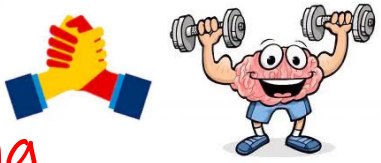
The year 2000 imagined in 1900

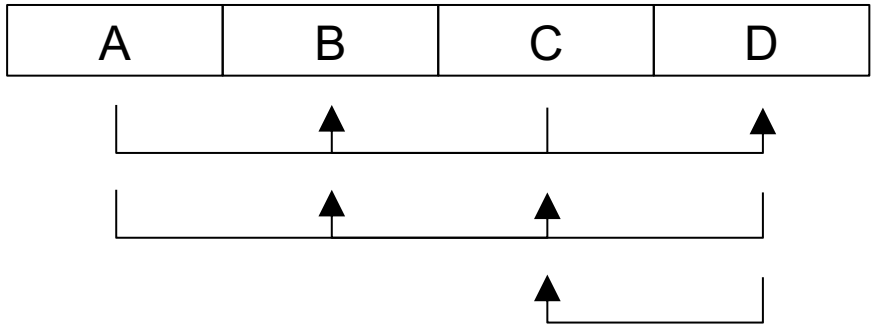


At School

Source: https://publicdomainreview.org/collection/a-19th-century-vision-of-the-year-2000

Normalization exercise





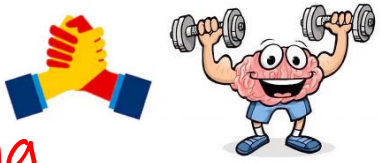
Goals: Create all of the following normalizations if possible:

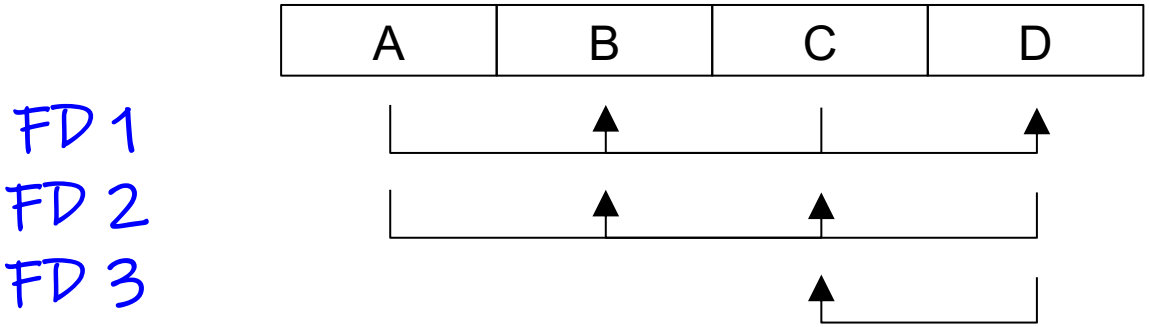
1. neither 3NF, nor BCNF
2. only 3NF, not BCNF
3. not 3NF, but BCNF
4. in both 3NF and BCNF

Suggested Approach:

1. try all possible candidate keys as PKs
2. then normalize until you hit 3NF or BCNF

Normalization exercise

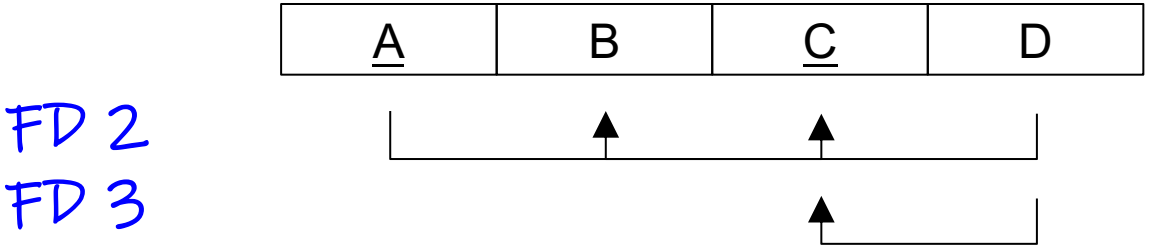




Goals: Create all of the following normalizations if possible:

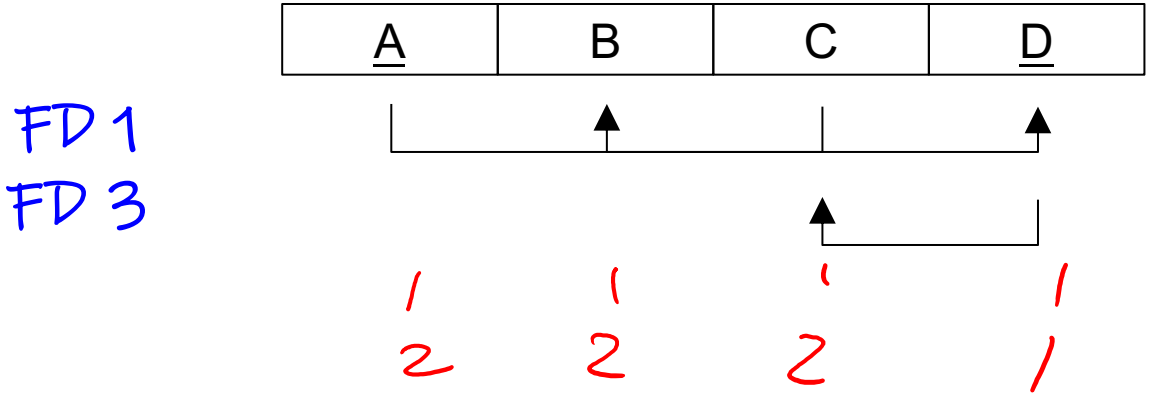
1. neither 3NF, nor BCNF
2. only 3NF, not BCNF
3. ~~not 3NF, but BCNF~~
4. in both 3NF and BCNF

FD 1 is full and used as PK



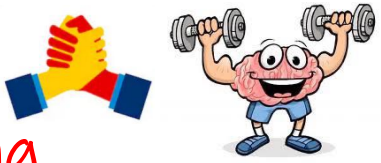
in 3NF (no more partial nor transitive FDs), not in BCNF (determinant of FD3 is not candidate key)

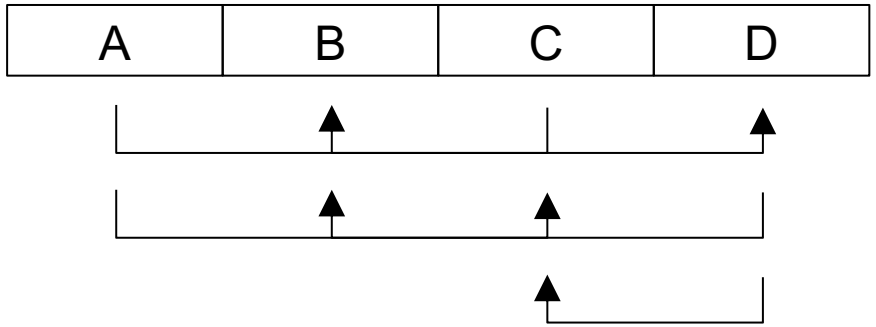
FD 2 is full and used as PK



not in 3NF (partial FD3), thus neither in BCNF

Normalization exercise

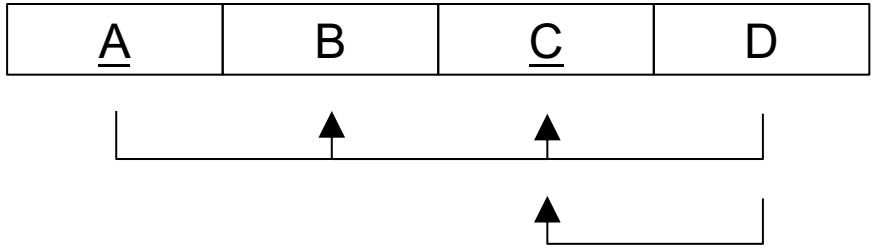




Goals: Create all of the following normalizations if possible:

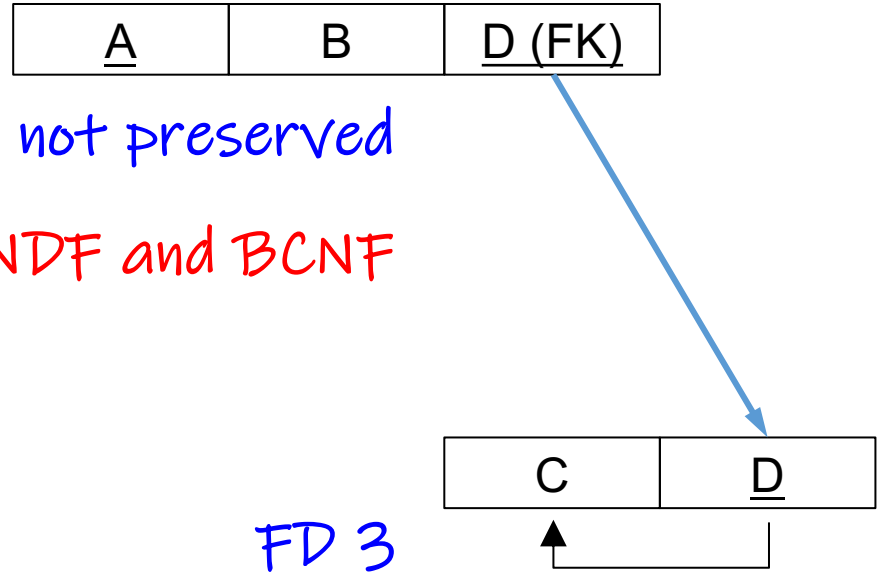
1. neither 3NF, nor BCNF
2. only 3NF, not BCNF
3. not 3NF, but BCNF
4. in both 3NF and BCNF

FD 1 is full and used as PK

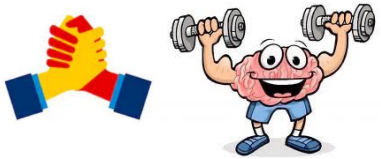


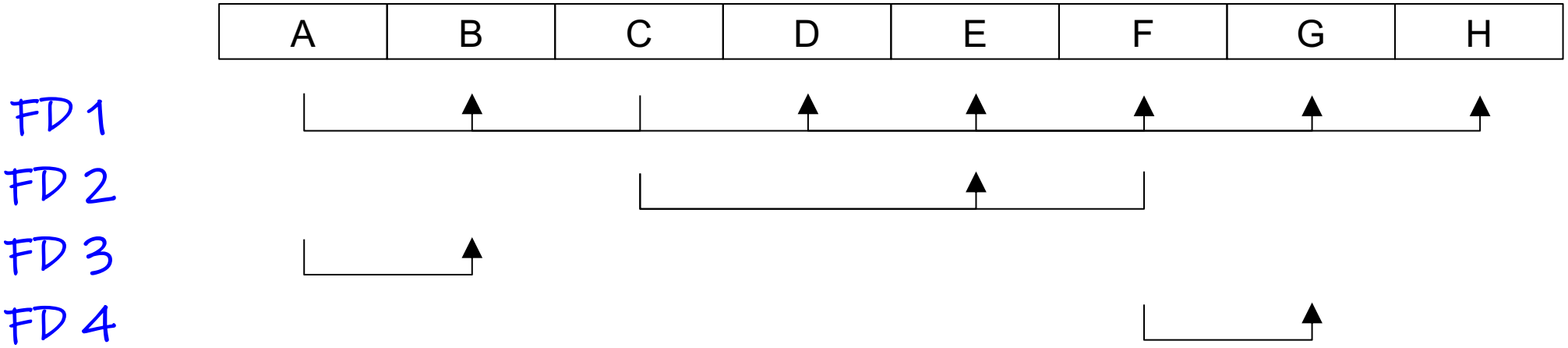
in 3NF (no more partial nor transitive FDs), not in BCNF (determinant of FD3 is not candidate key)

FD 2 is full and used as PK



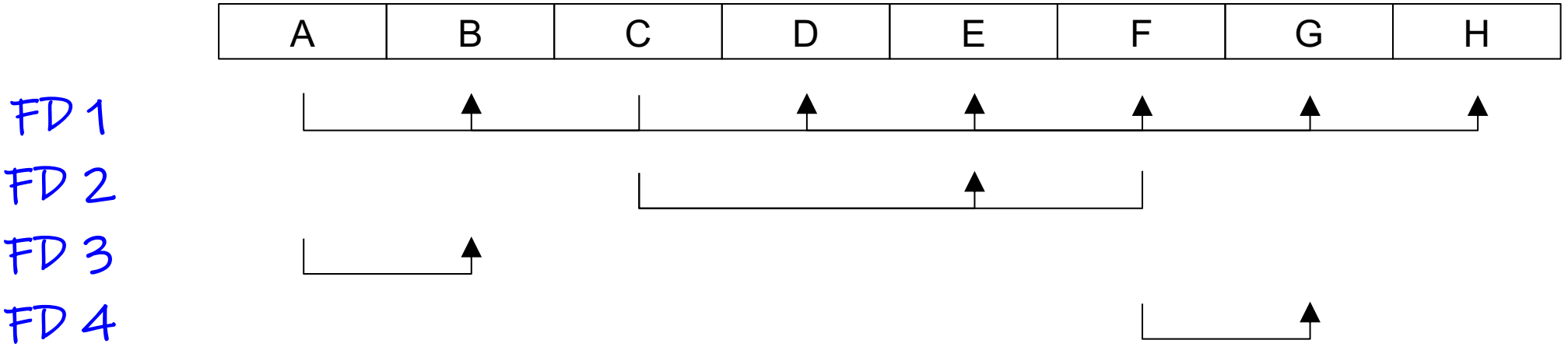
Normalization exercise





Goal: Normalize into 3NF, then BCNF ?

Normalization exercise



First: what is a possible PK?

?

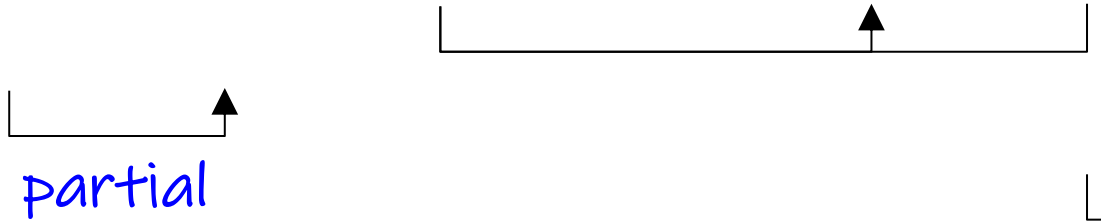
Normalization exercise

A	B	C	D	E	F	G	H

FD 2

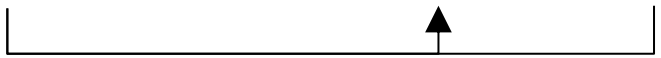
FD 3

FD 4

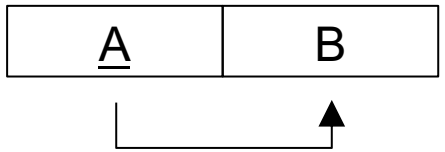


Normalization exercise

FD 2

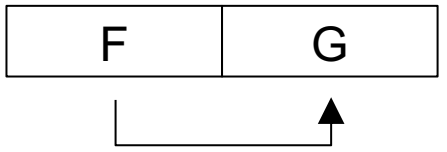


neither partial (determinant is not part of key), nor transitive (not between non-prime attributes only

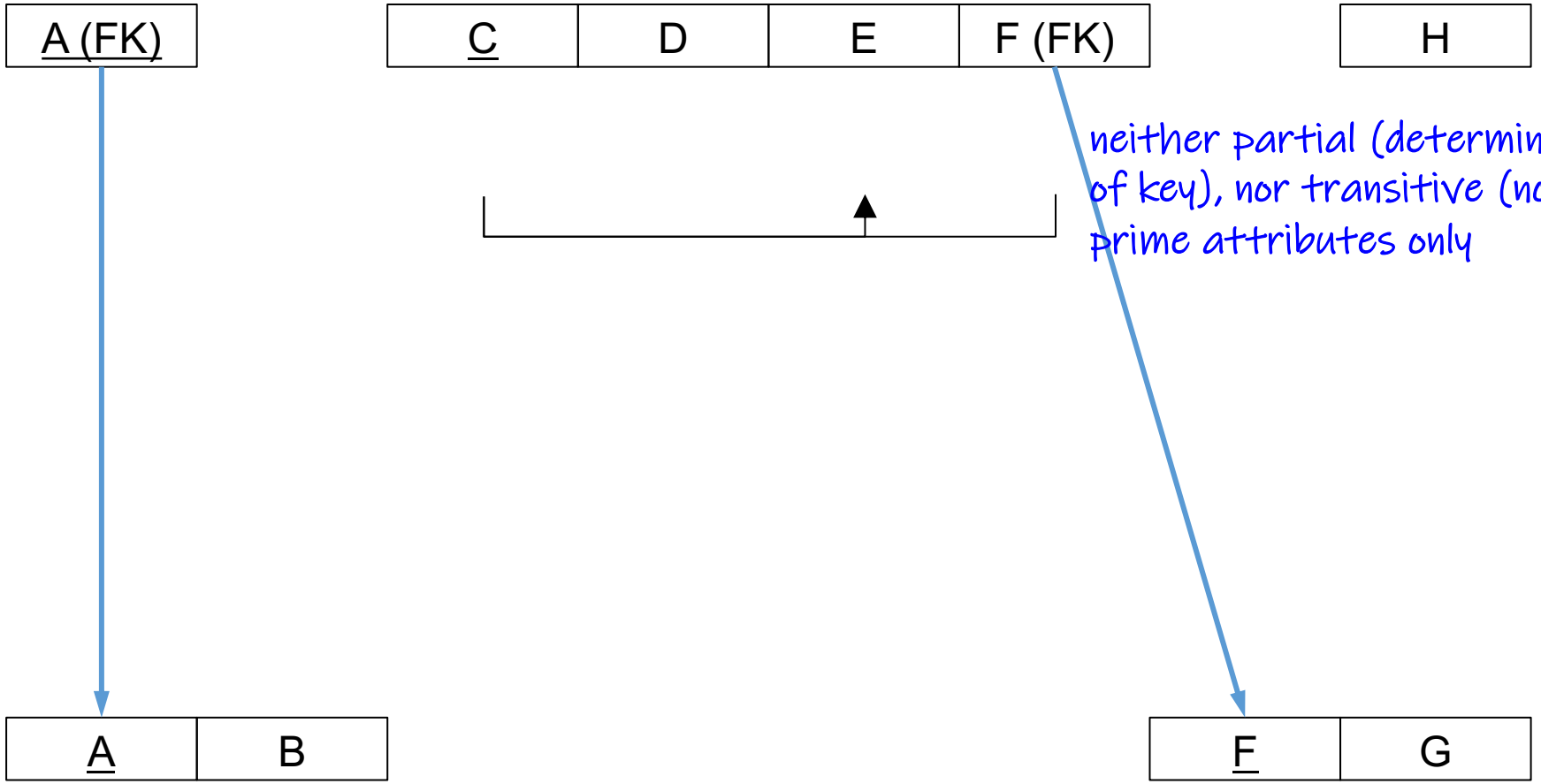


FD 3

FD 4

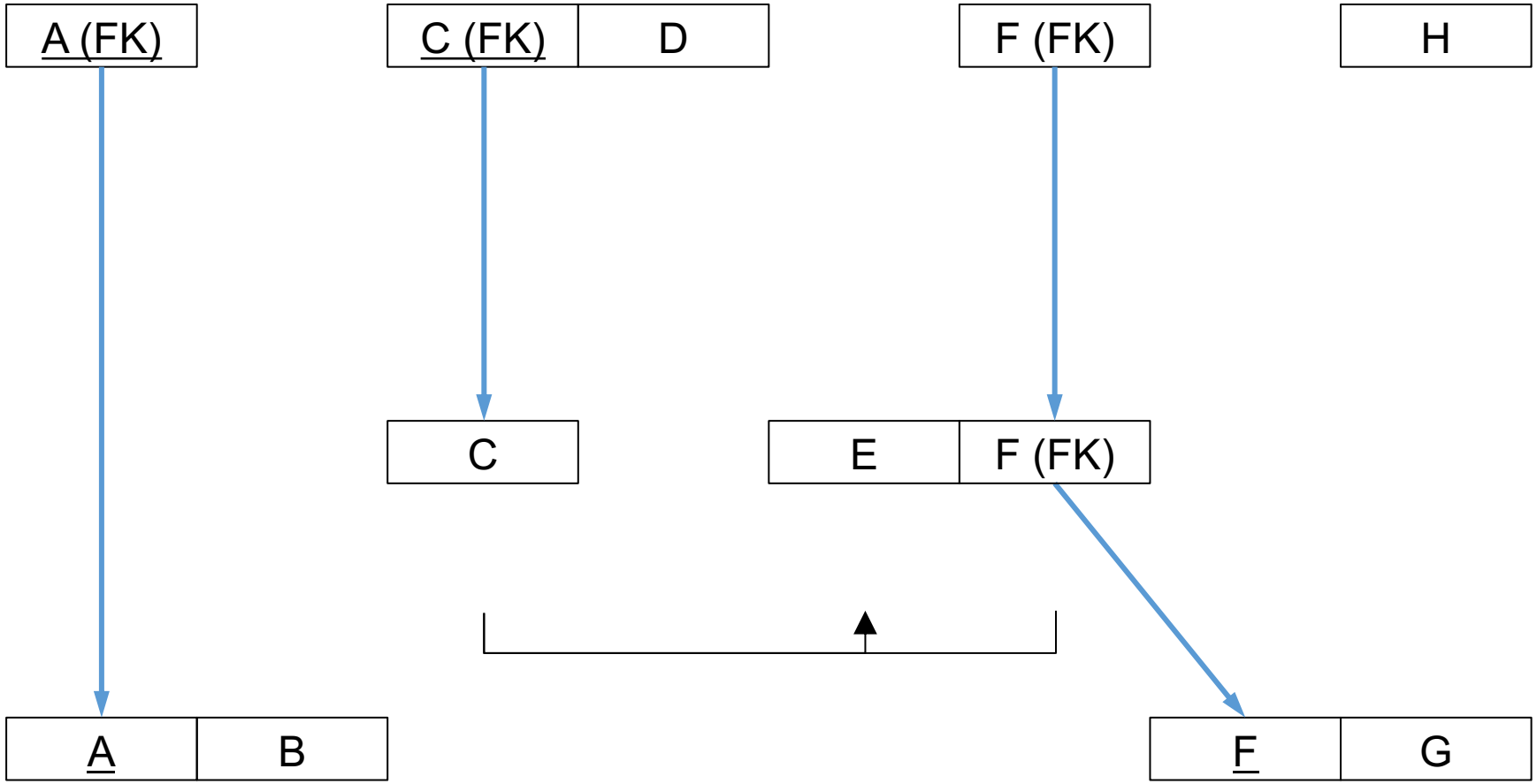


Normalization exercise

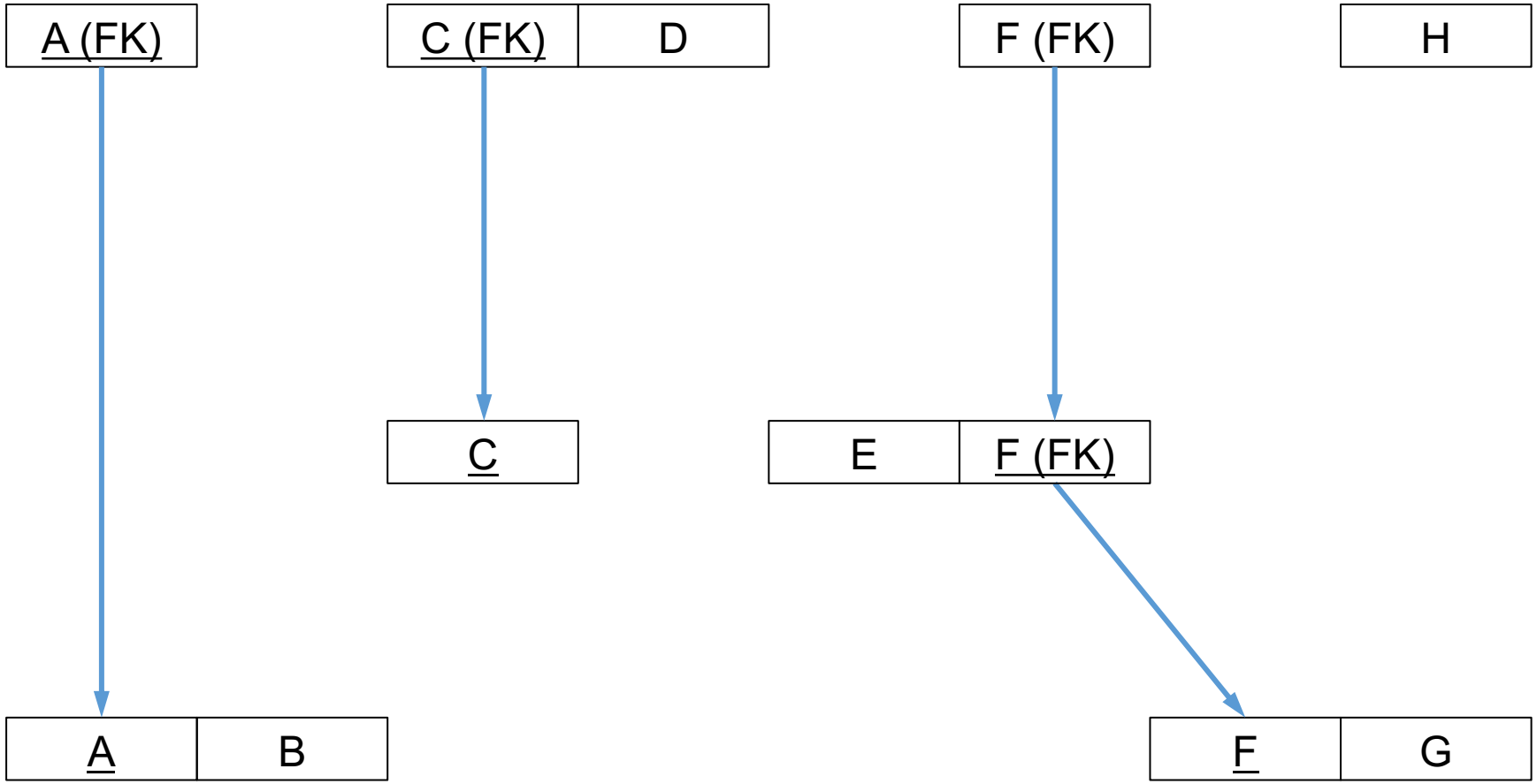


That is 3NF
but not in BCNF

Normalization exercise



Normalization exercise



That is in BCNF
(and thus also in 3NF)

Example: multiple relationships

For this exercise, ignore attributes:

- Each employee is assigned to one department
- Each employee has one supervisor
- Each department is manged by one manager

?

Example: multiple relationships

For this exercise, ignore attributes:

- Each employee is assigned to one department
- Each employee has one supervisor
- Each department is manged by one manager

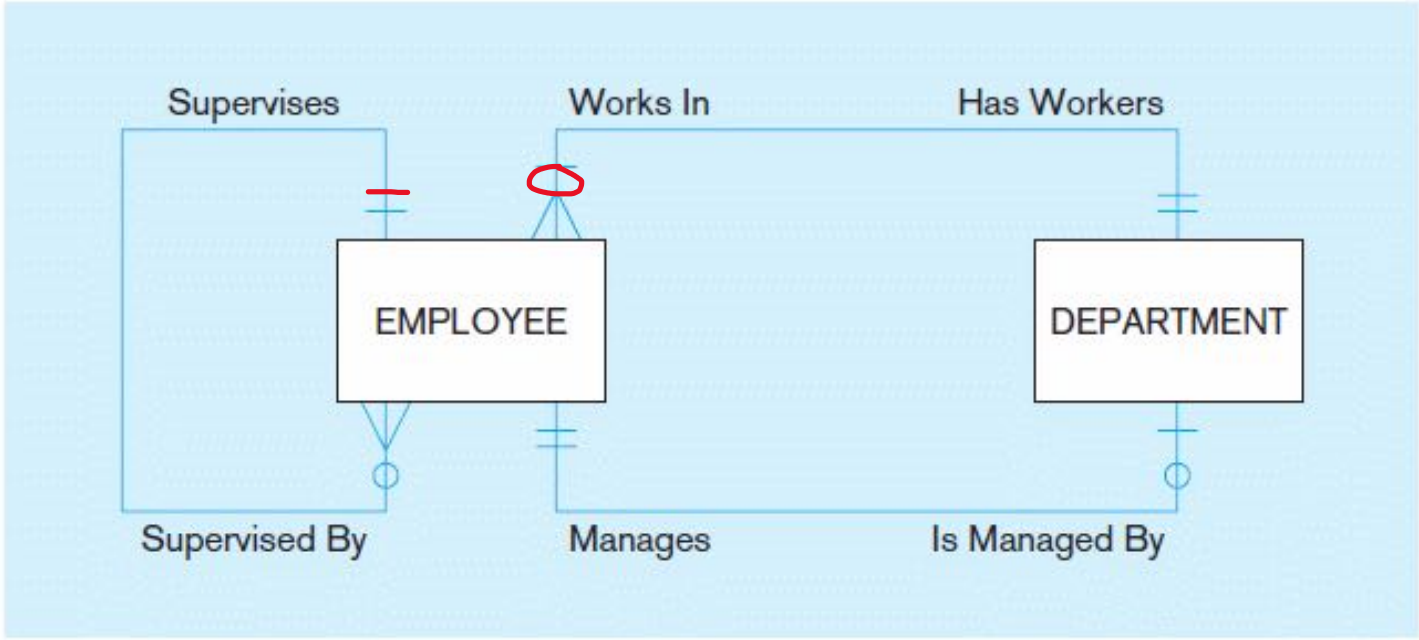
?

All 3 appear mandatory!

Example: multiple relationships

For this exercise, ignore attributes:

- Each employee is assigned to one department
- Each employee has one supervisor
- Each department is manged by one manager



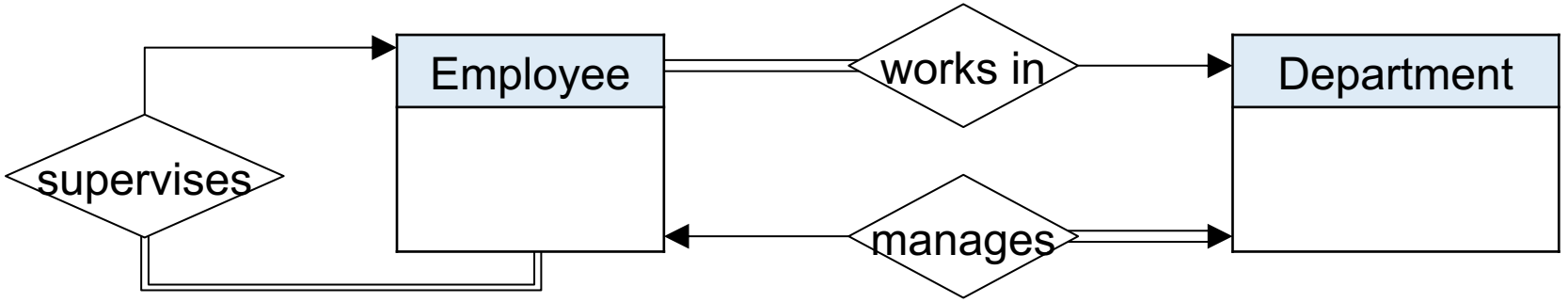
Recall: Entities can be related to one another in more than one way

Source: Fig. 2-21, Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

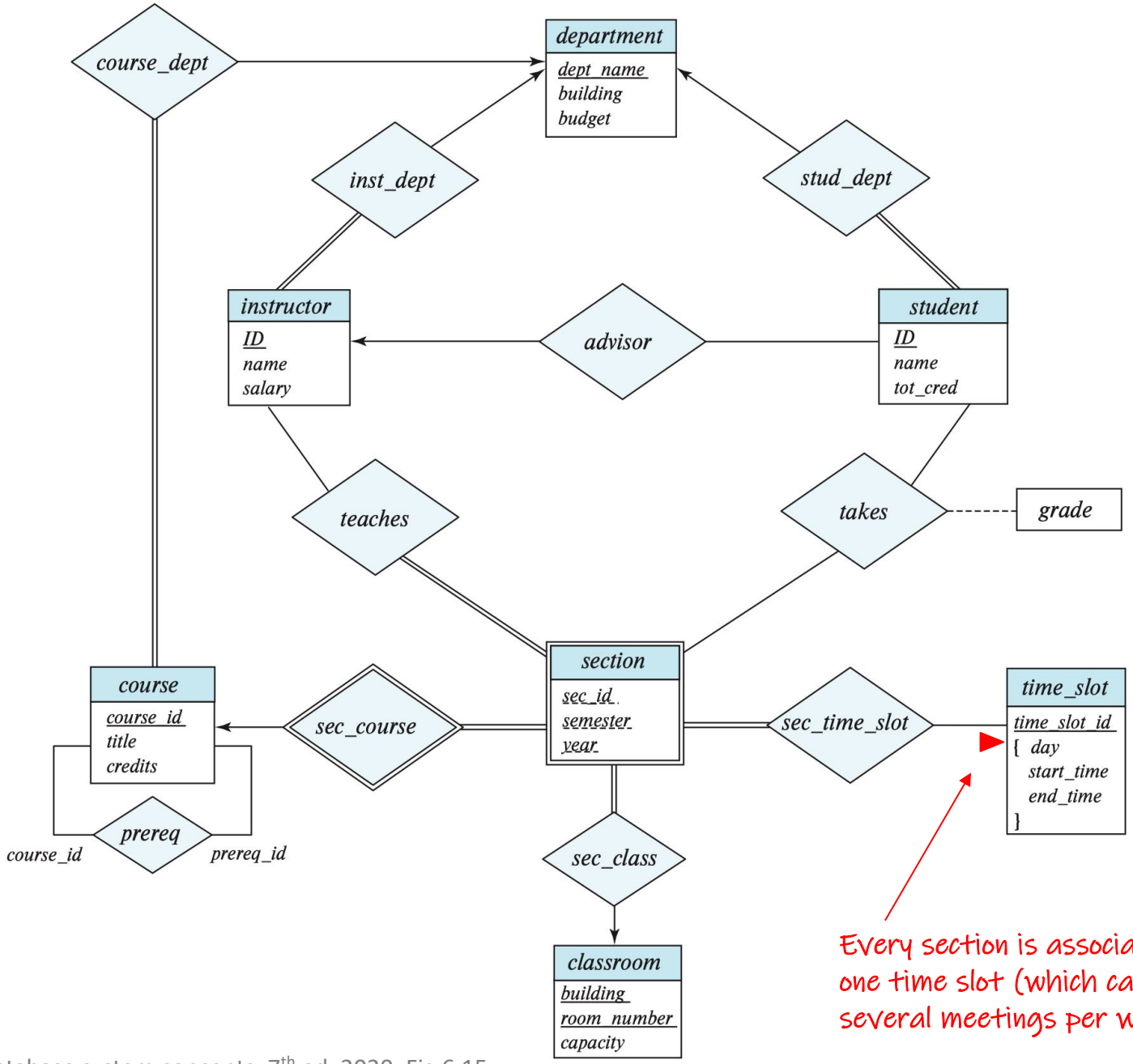
Example: multiple relationships

For this exercise, ignore attributes:

- Each employee is assigned to one department
- Each employee has one supervisor
- Each department is manged by one manager

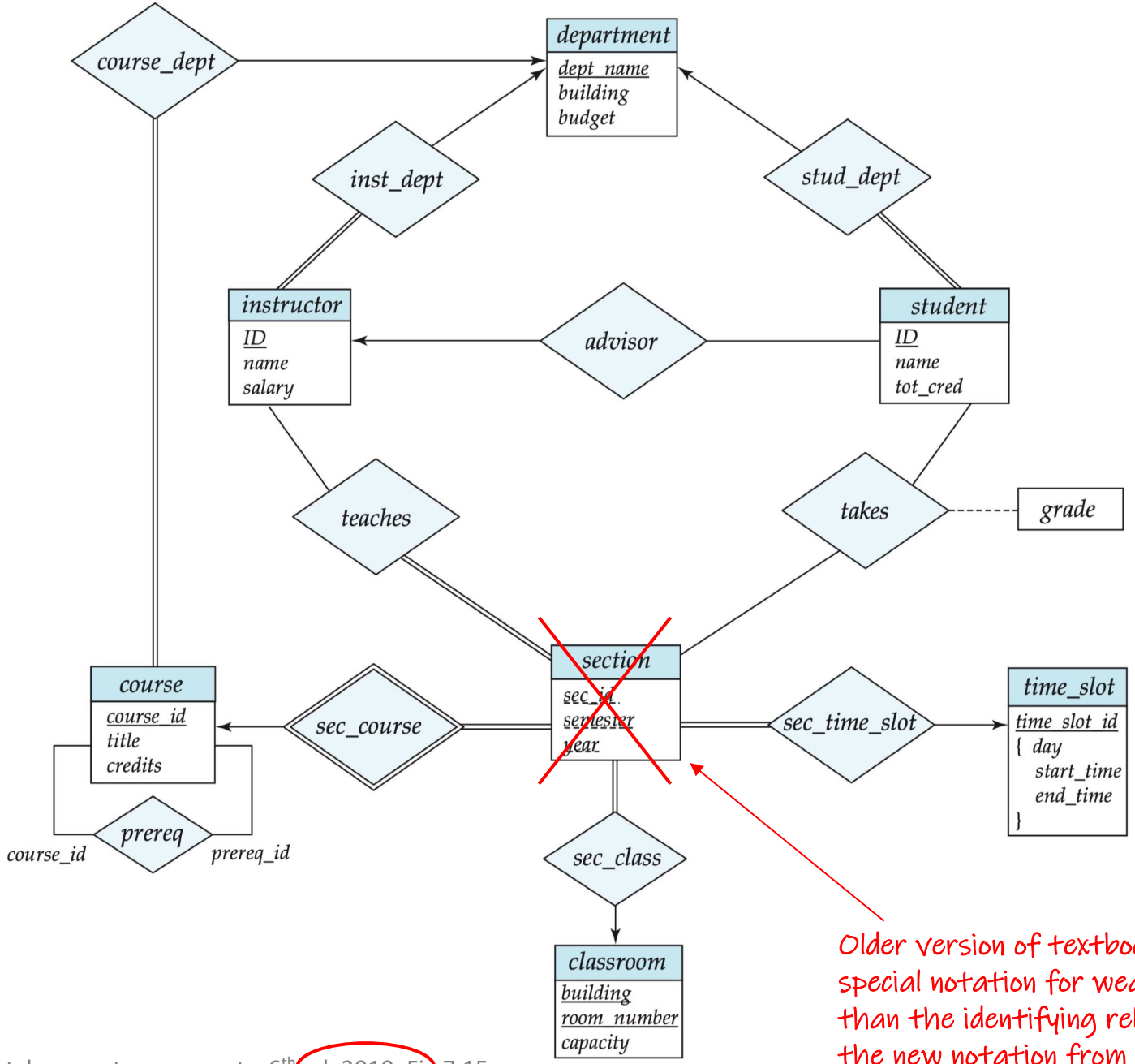


University ERD



Source: "SDK", Silberschatz, Korth, Sudarshan, Database system concepts, 7th ed, 2020. Fig 6.15
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

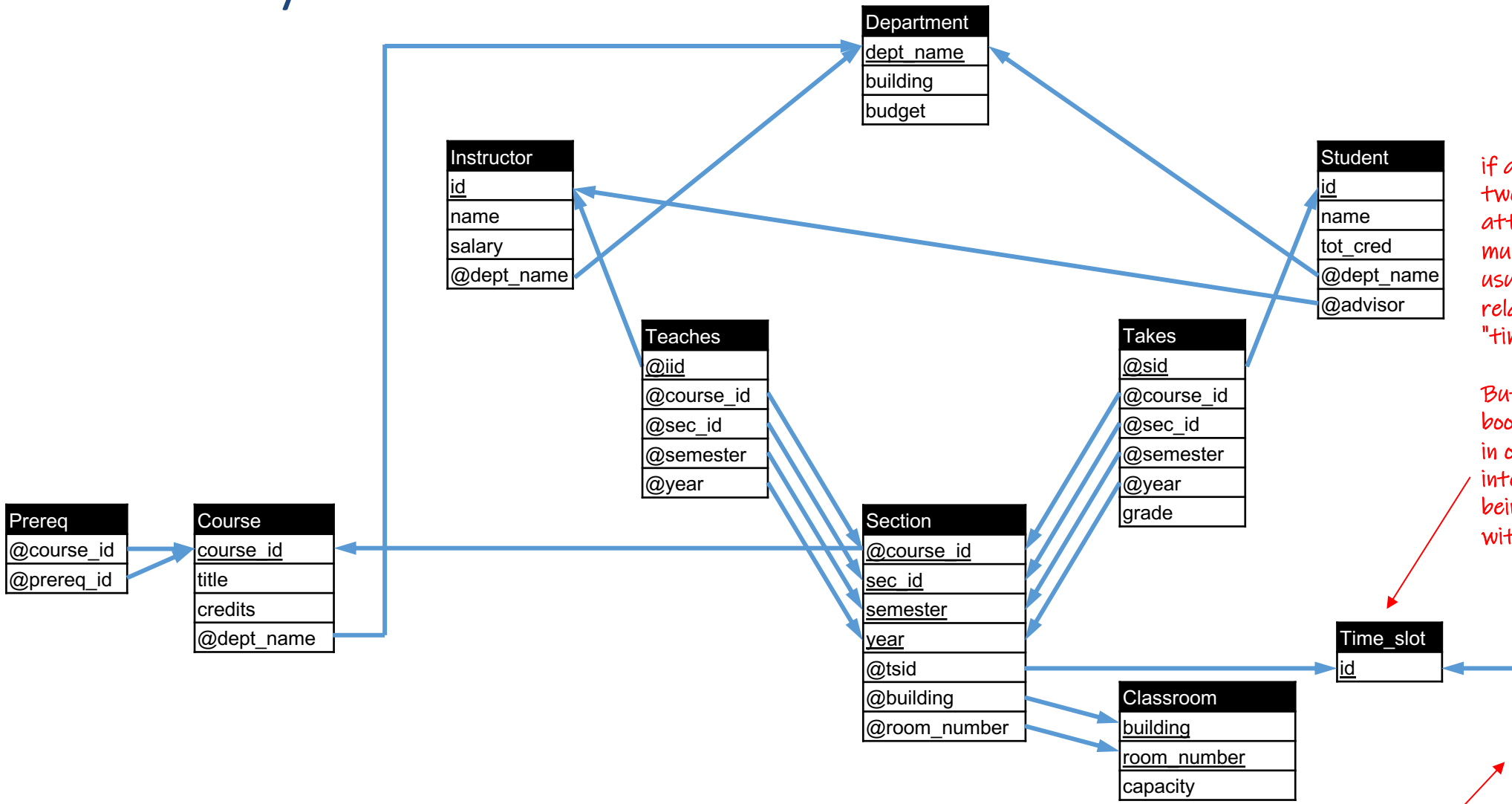
University ERD



Older version of textbook did not have a special notation for weak entities other than the identifying relationship. We use the new notation from the previous slide!

Source: "SDK", Silberschatz, Korth, Sudarshan, Database system concepts, 6th ed, 2010. Fig 7.15
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

University Schema



if an entity set consists of only two attributes (a single PK attribute B and a single multivalued attribute M), we can usually just represent it with one relation (that would be only "time_slot_entry" here).

But here I disagree with the book. We should have "Time_slot" in order to enforce the referential integrity constraint of sections being able to be associated only with predefined time slots.

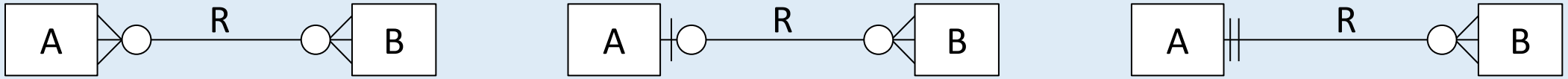
"Although not represented as a constraint on the E-R diagram, we know that there cannot be two meetings of a class that start at the same time of the same day of the week but end at different times; based on this constraint, end time has been omitted from the primary key of the time slot schema."

Relational schema corresponding to slightly fixed ERD from "SDK", Silberschatz, Korth, Sudarshan, Database system concepts, 7th ed, 2020. Fig 6.15
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

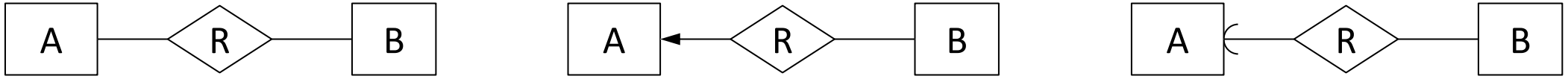
Limits of enforcing cardinality & participation constraints w/ FKs (1/2)

SDK
[Silberschatz+'20]

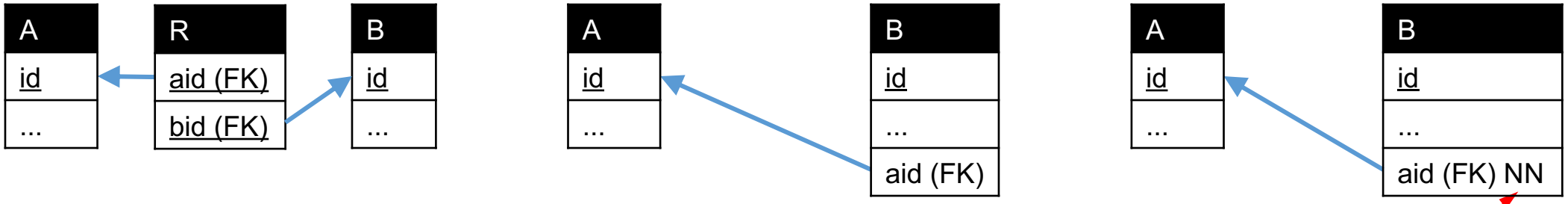
Crow's foot
[Hoffer+'10]



Stanford
[Complete book'03]
Gradiance



Relational Translation

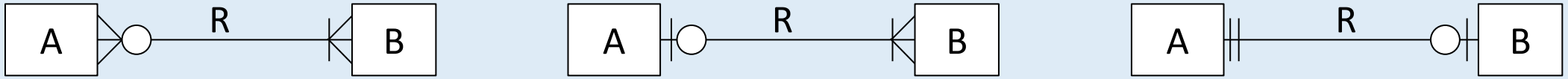


not null (NN) constraints enforce mandatory participation constraints

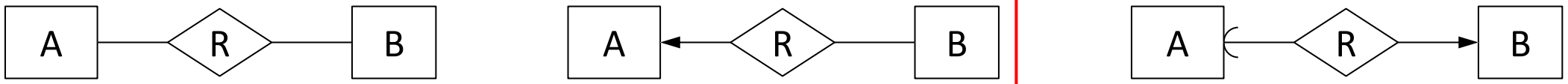
Limits of enforcing cardinality & participation constraints w/ FKs (2/2)

SDK
[Silberschatz+'20]

Crow's foot
[Hoffer+'10]



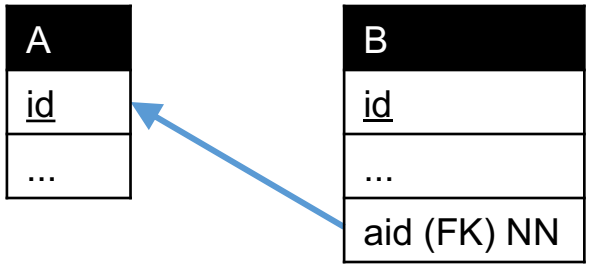
Stanford
[Complete book'03]
Gradiance



The Stanford arrow notation does not allow to indicate mandatory participation on the "to many" side. An intuitive justification: we anyway can't enforce those later in the relational translation, thus why even represent. This is the closest we can get.

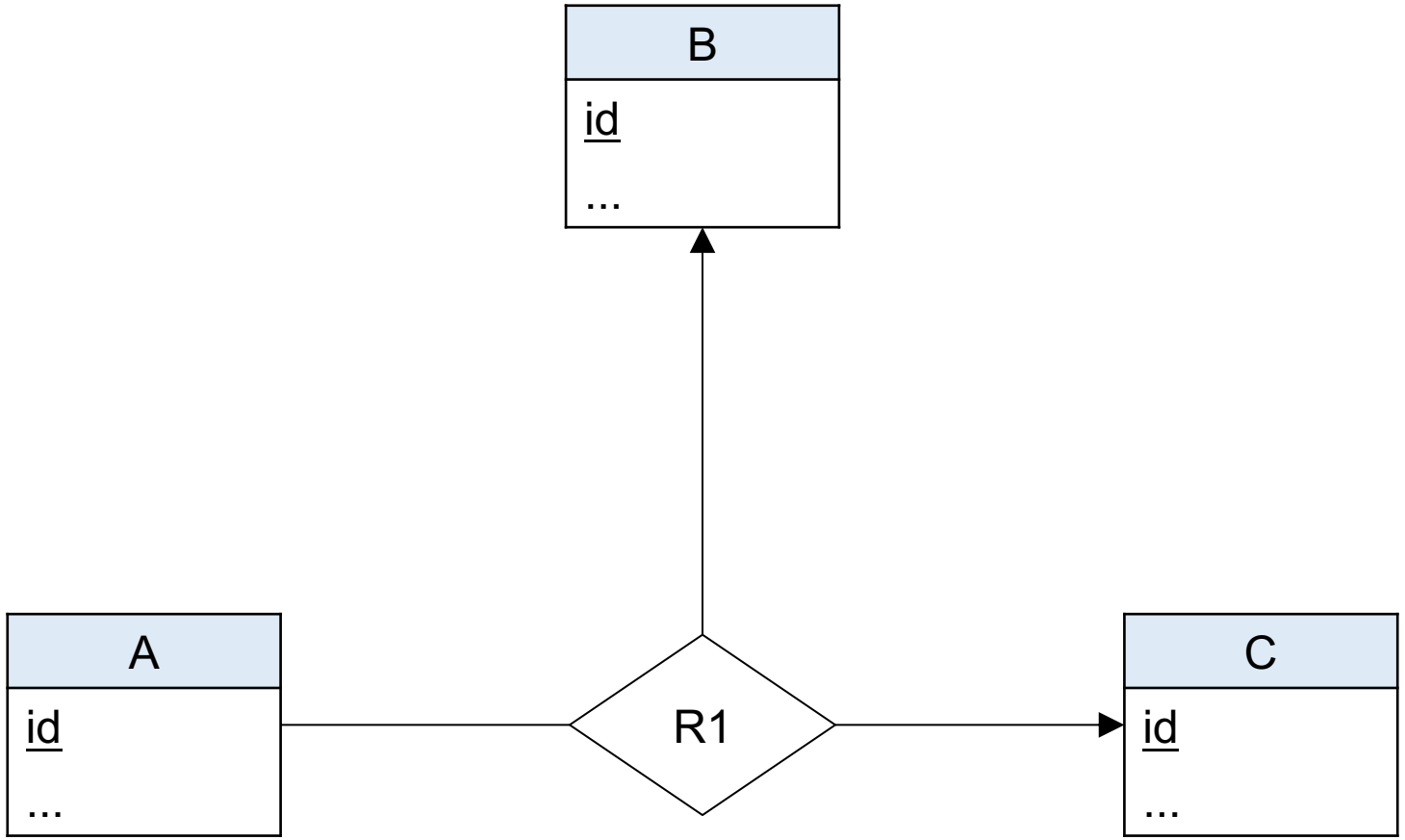
Relational Translation

We cannot enforce mandatory participation on the "to many" side with FKs alone. We thus, in practice, just ignore them. This translation is the closest we can get.



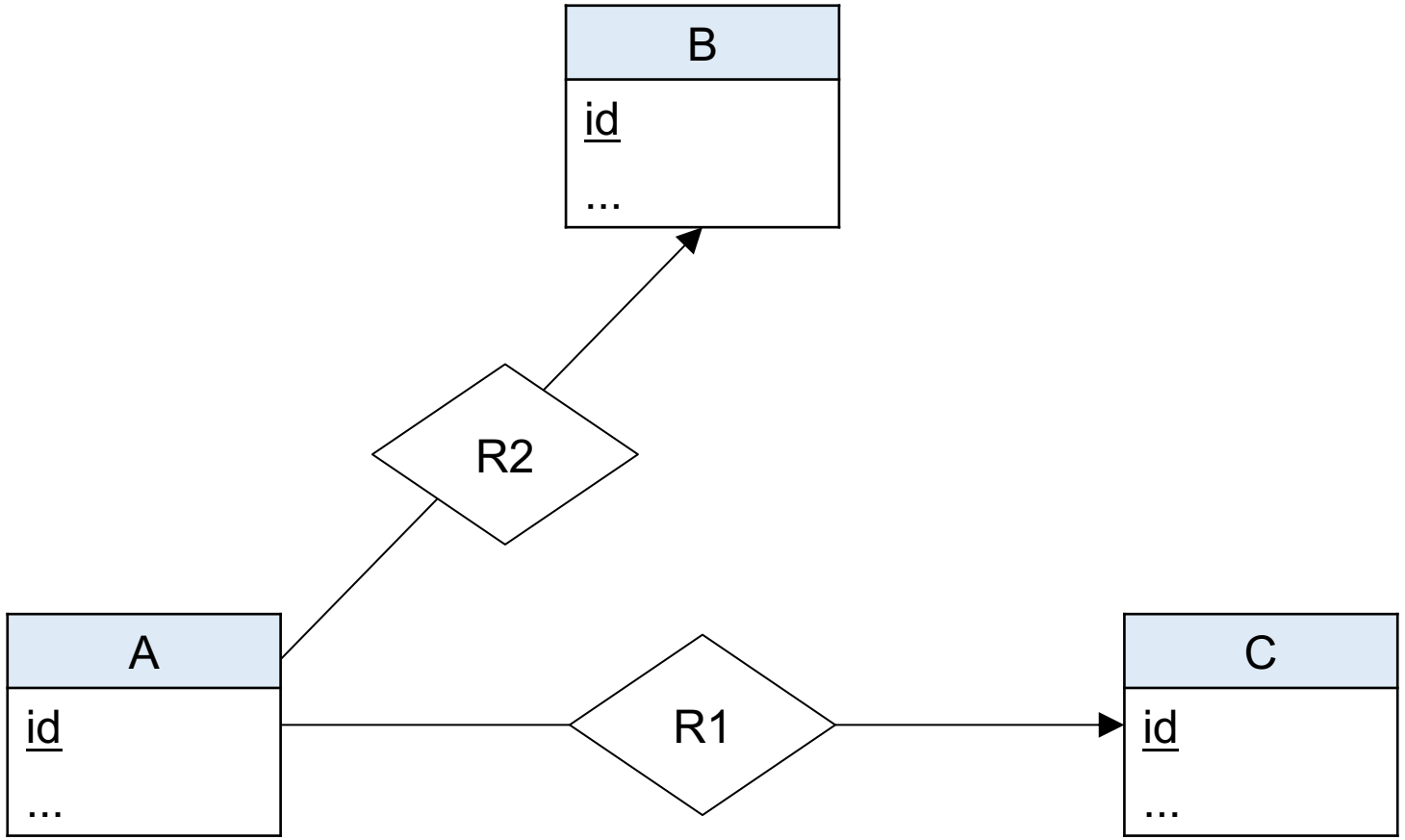
We cannot distinguish 1:1 from 1:many with FKs only. This is the closest we can get.

ERD best practice



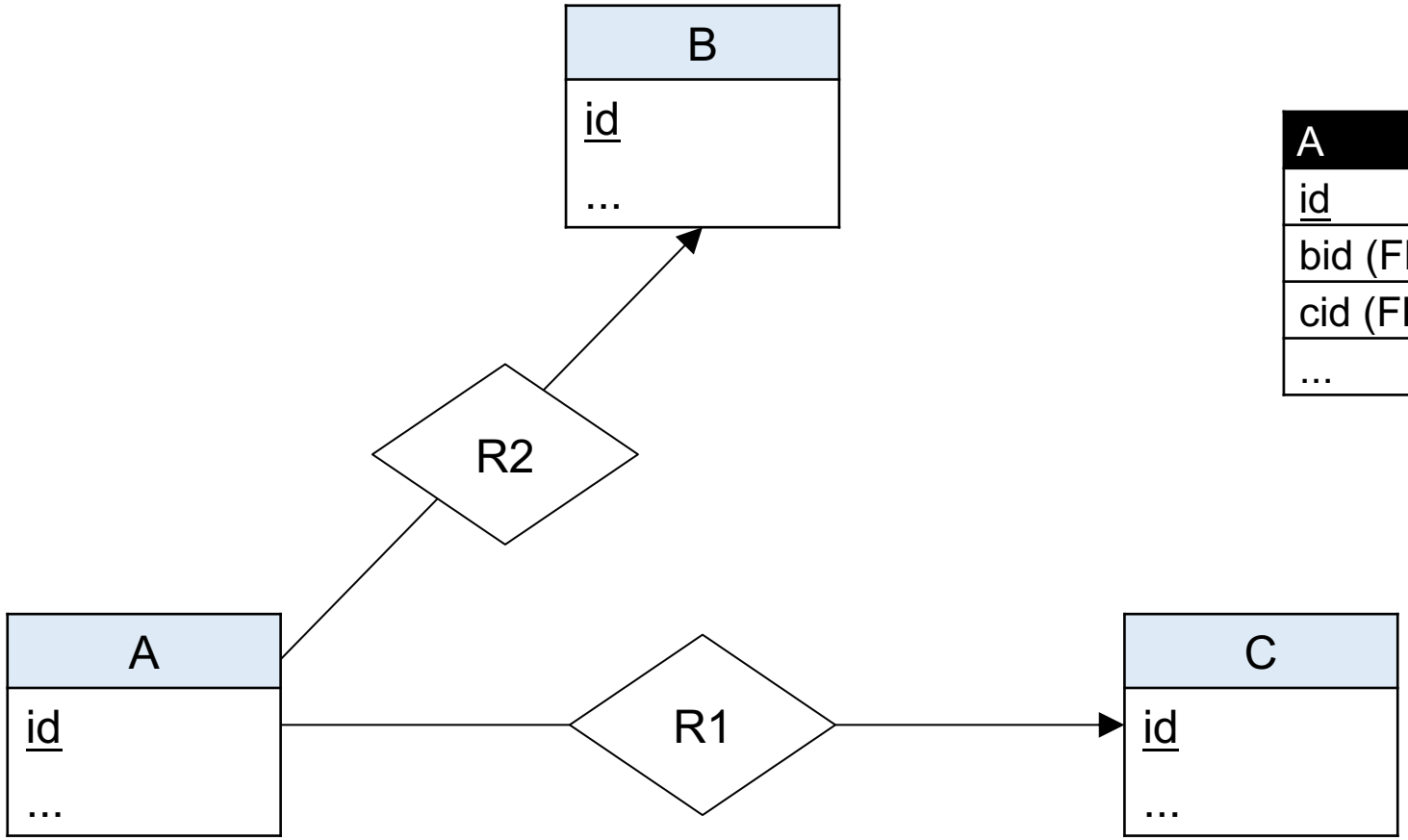
Assume no additional attributes on R1

ERD best practice

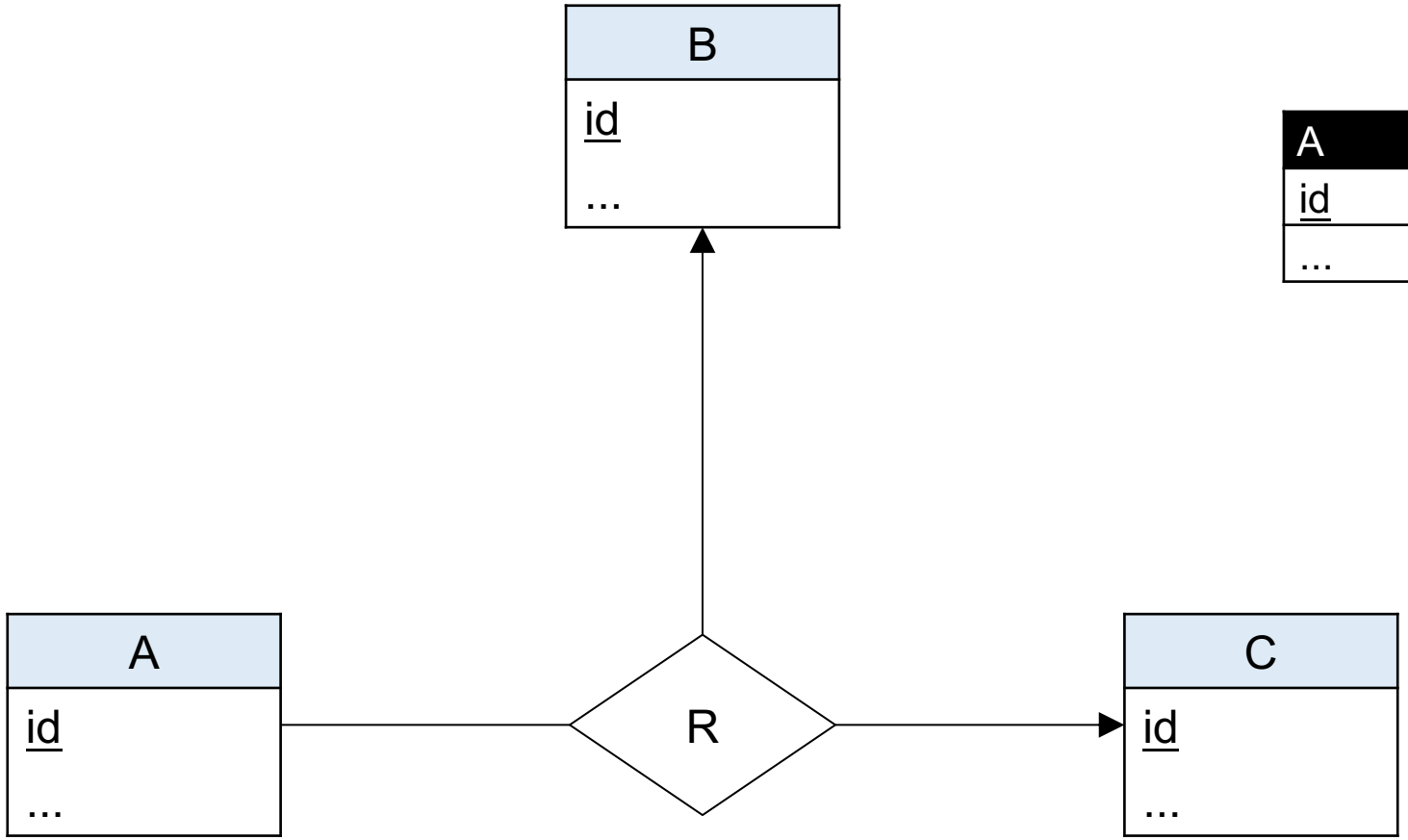


As tables ?

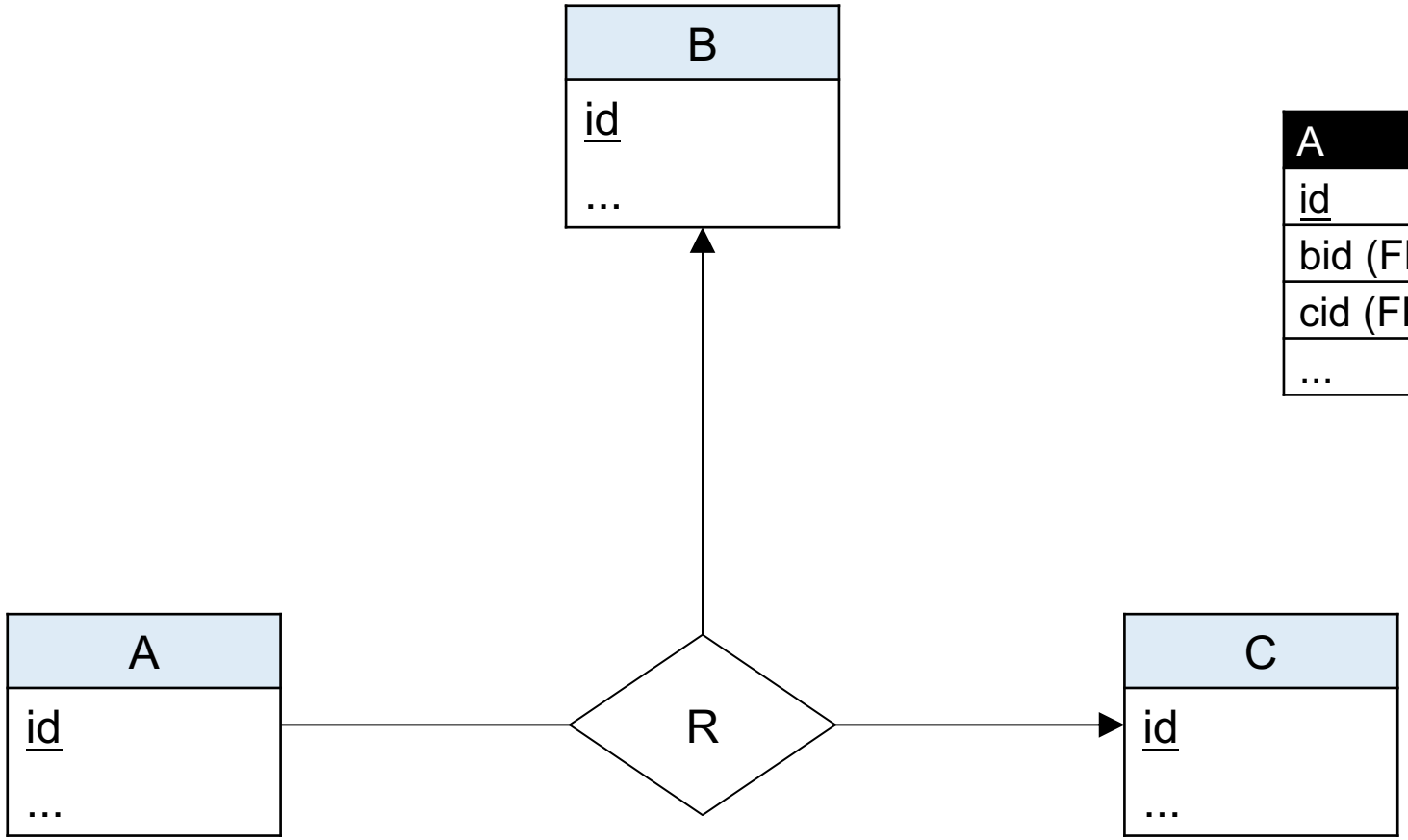
ERD best practice



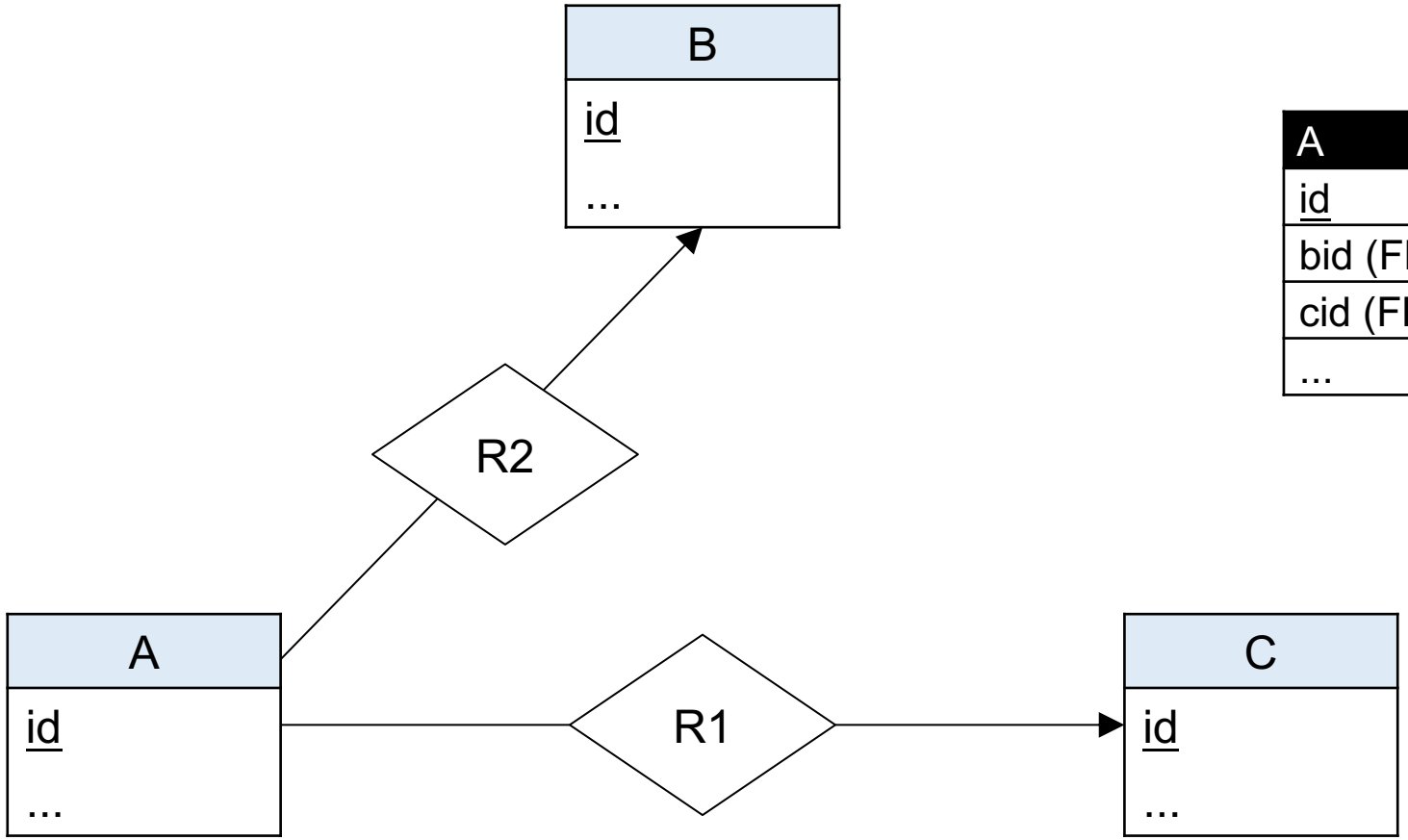
ERD best practice



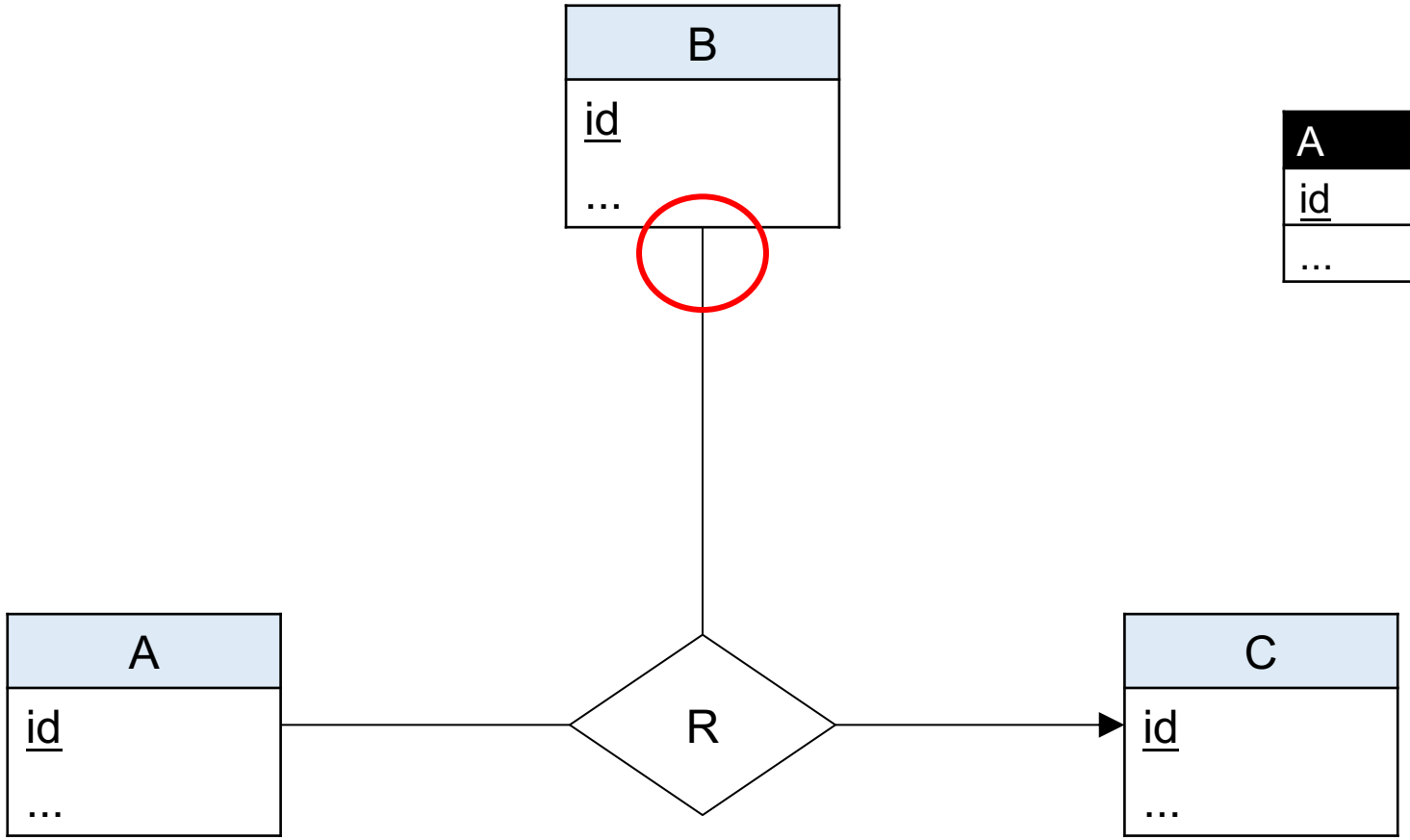
ERD best practice



ERD best practice

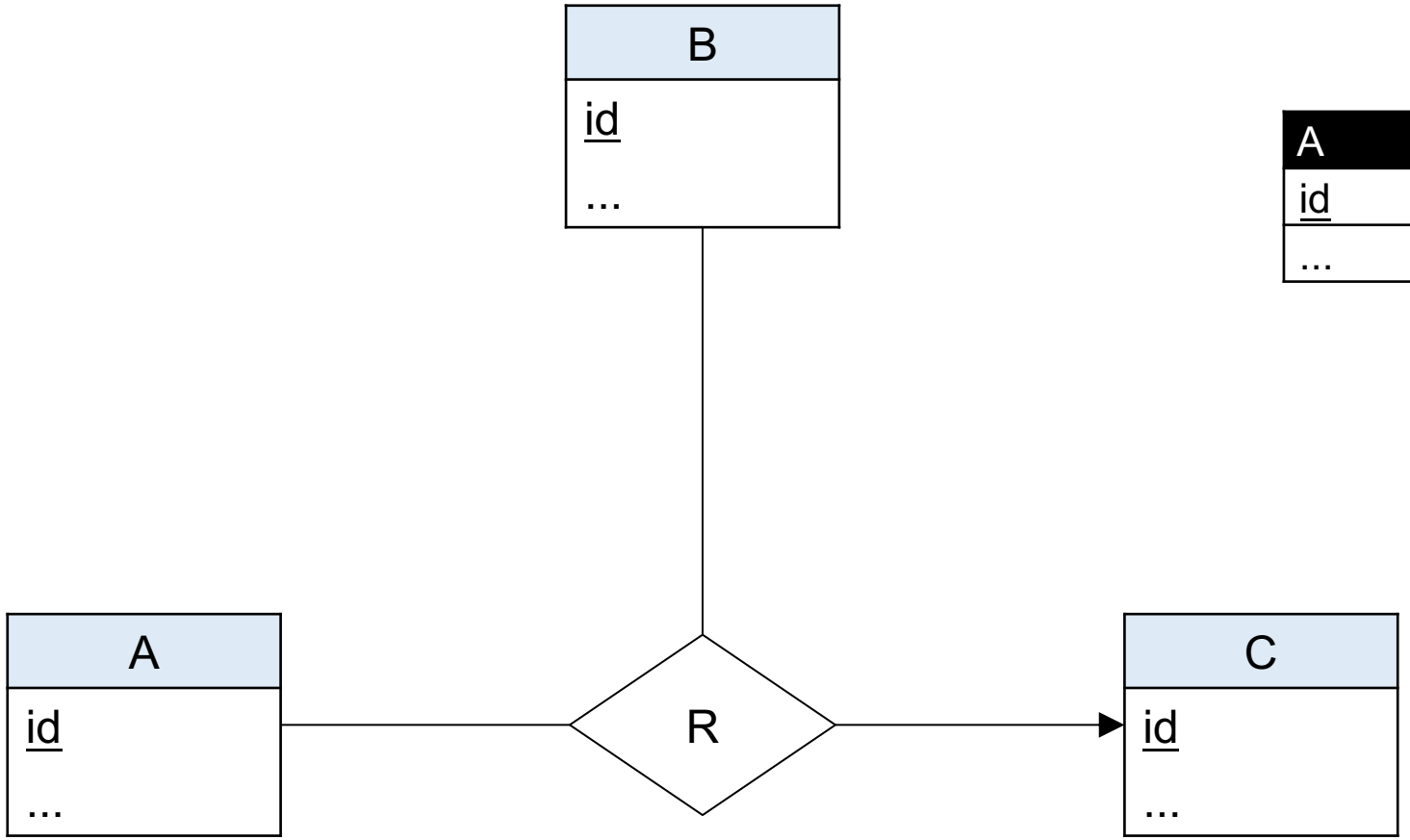


ERD best practice



What needs to change in the relational schema ?

ERD best practice



Now we *need*
the 4th relation!