Updated 11/30/2022

Topic 2: Database design
L22: Normalization

Wolfgang Gatterbauer

CS3200 Database design (fa22)

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

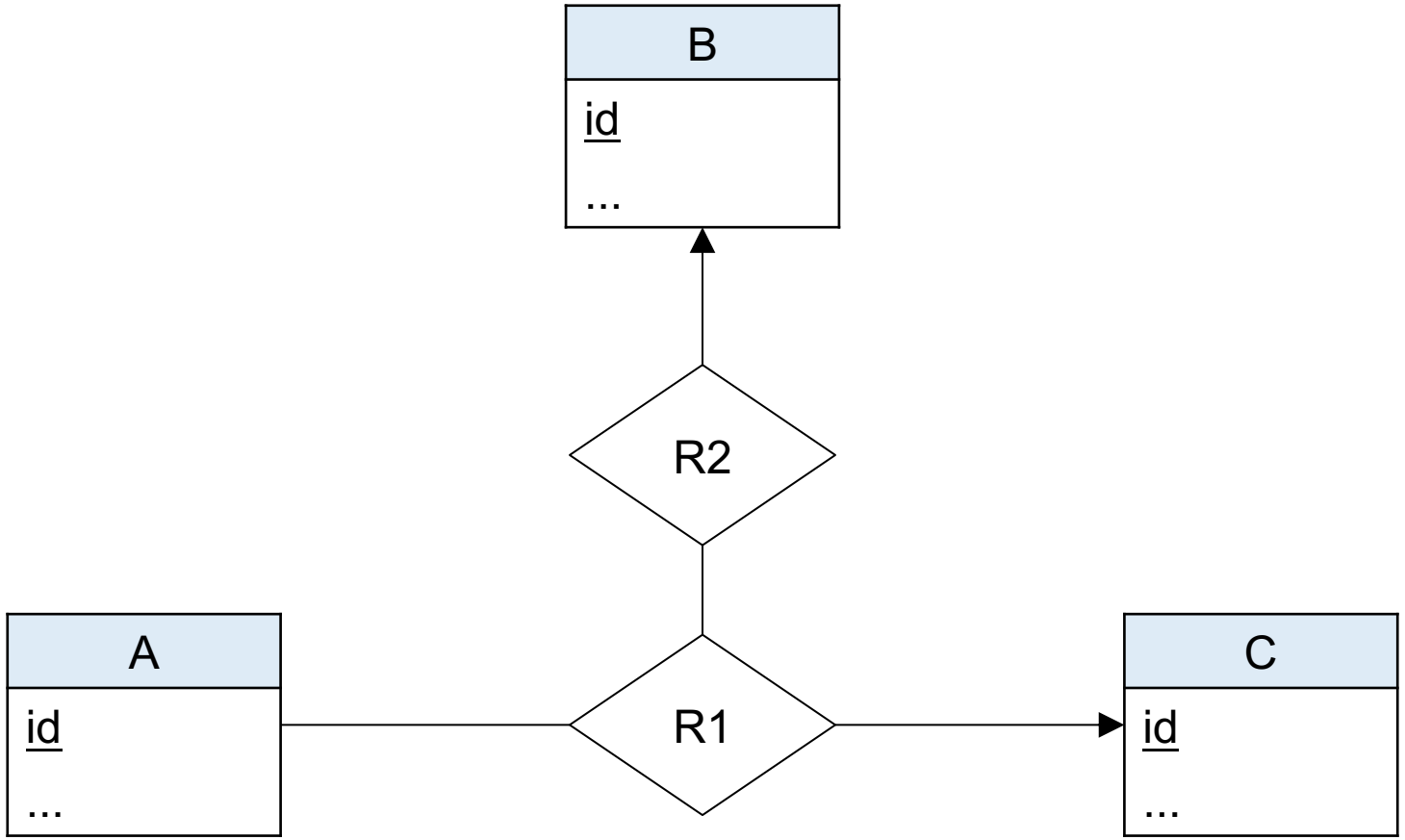
11/30/2022

Class warm-up

- Exchange of best practice ideas on drawing ER diagrams: What works for you?
  - text analysis (with bullet points): entities -> relationships -> attributes
- Project reports: data requirements easiest to list in bullet points
  - see comments on Canvas
- Normalization from last tim

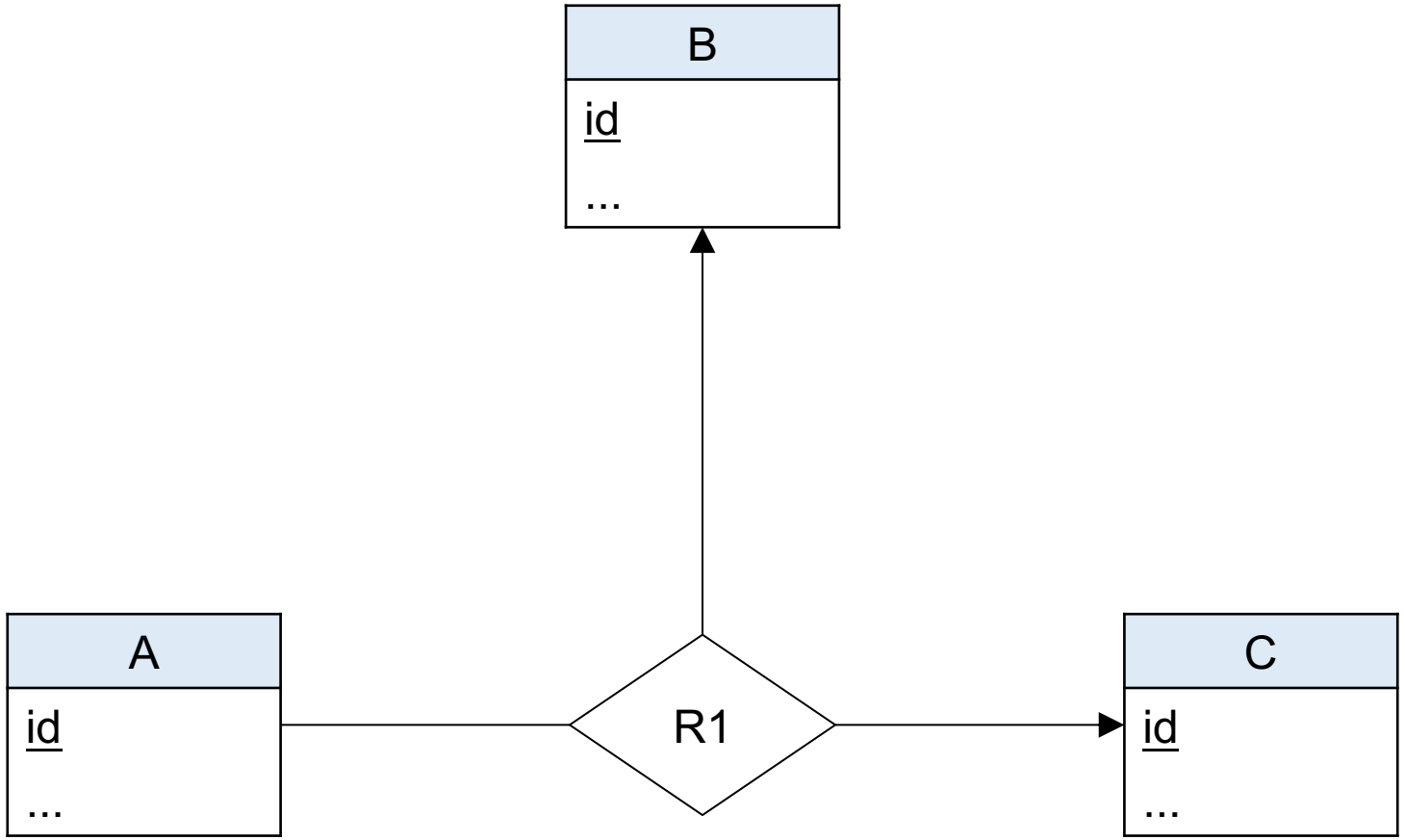
- Today: normalization
- Then: transactions

ERD best practice



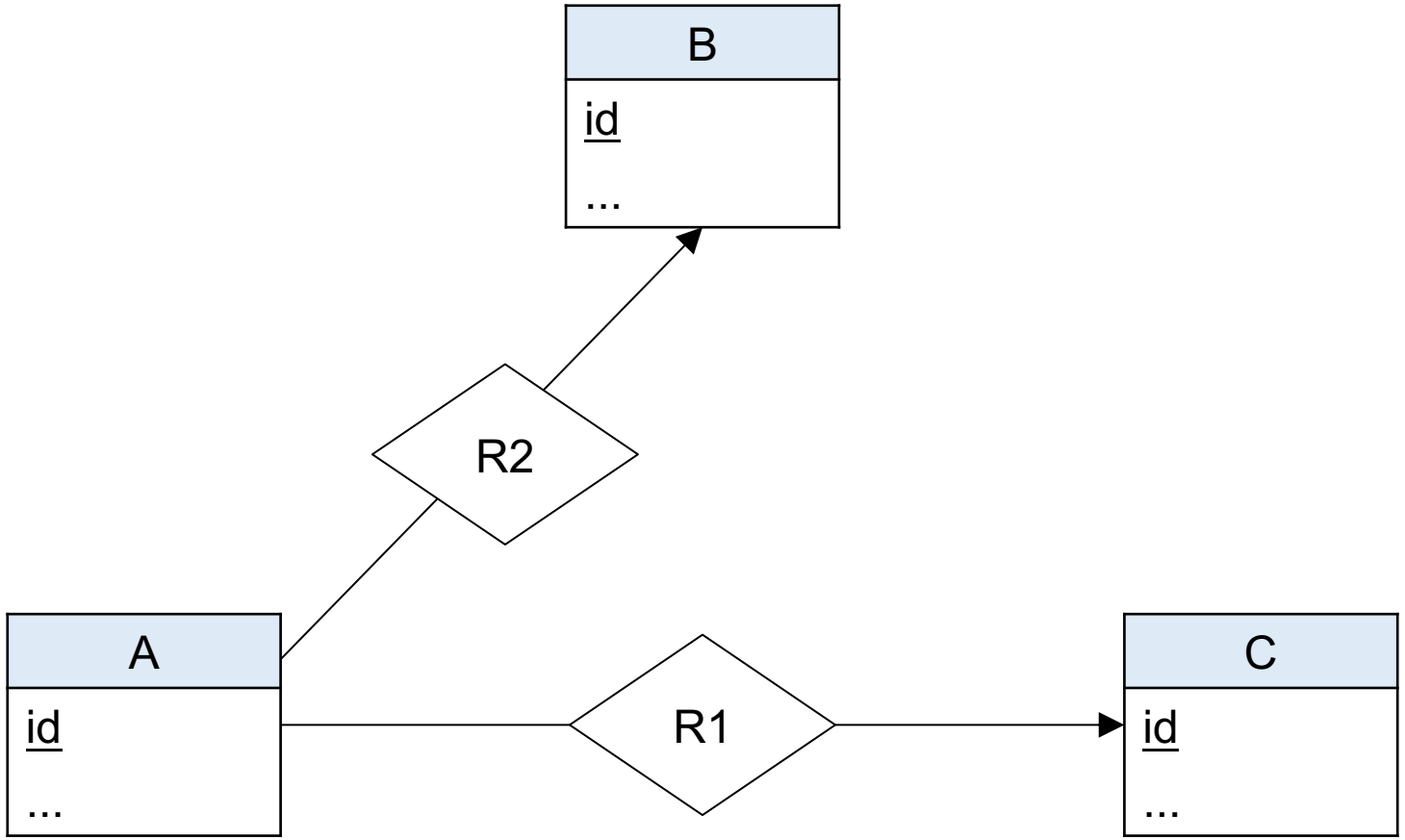
?

ERD best practice



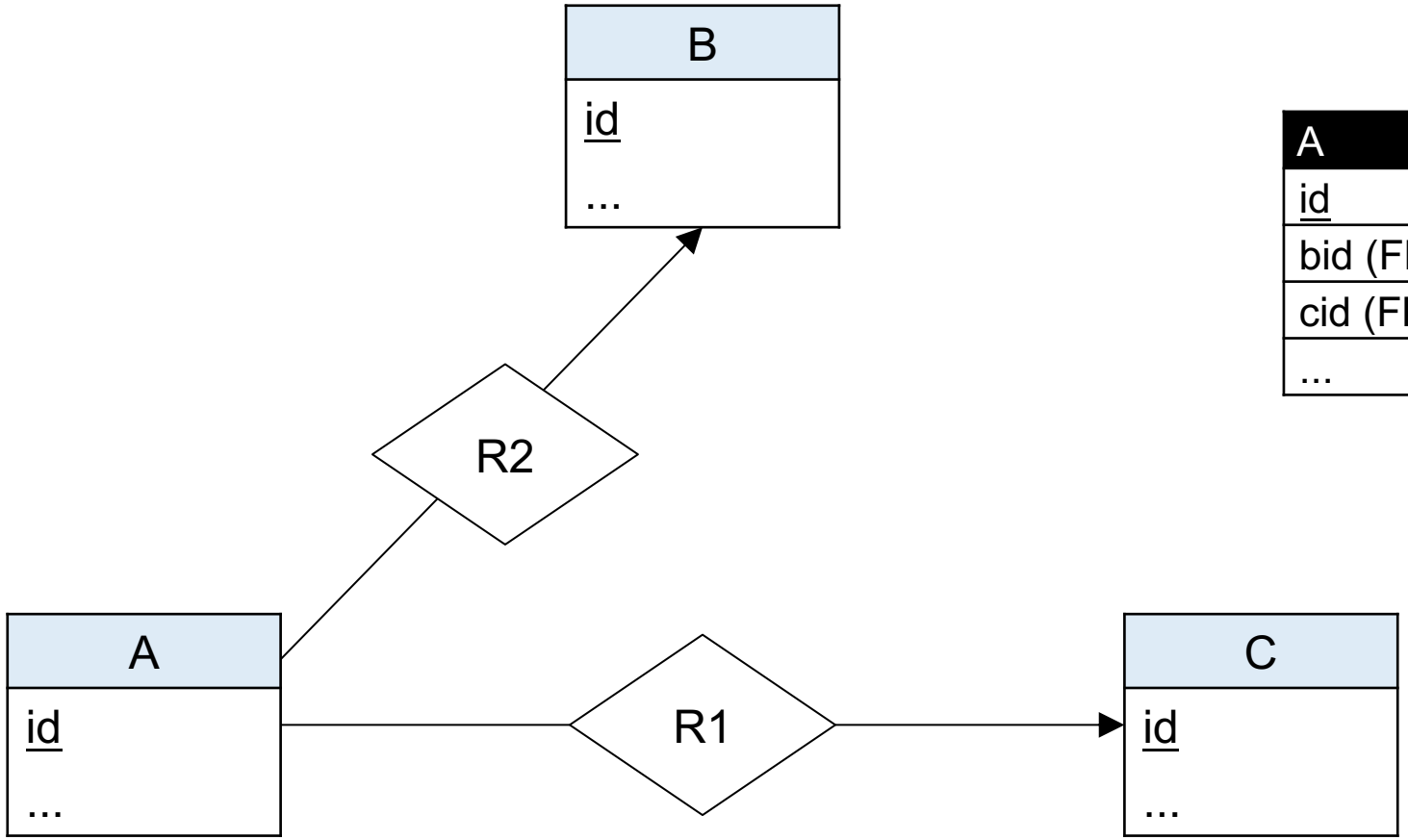
Assume no additional attributes on R1

ERD best practice

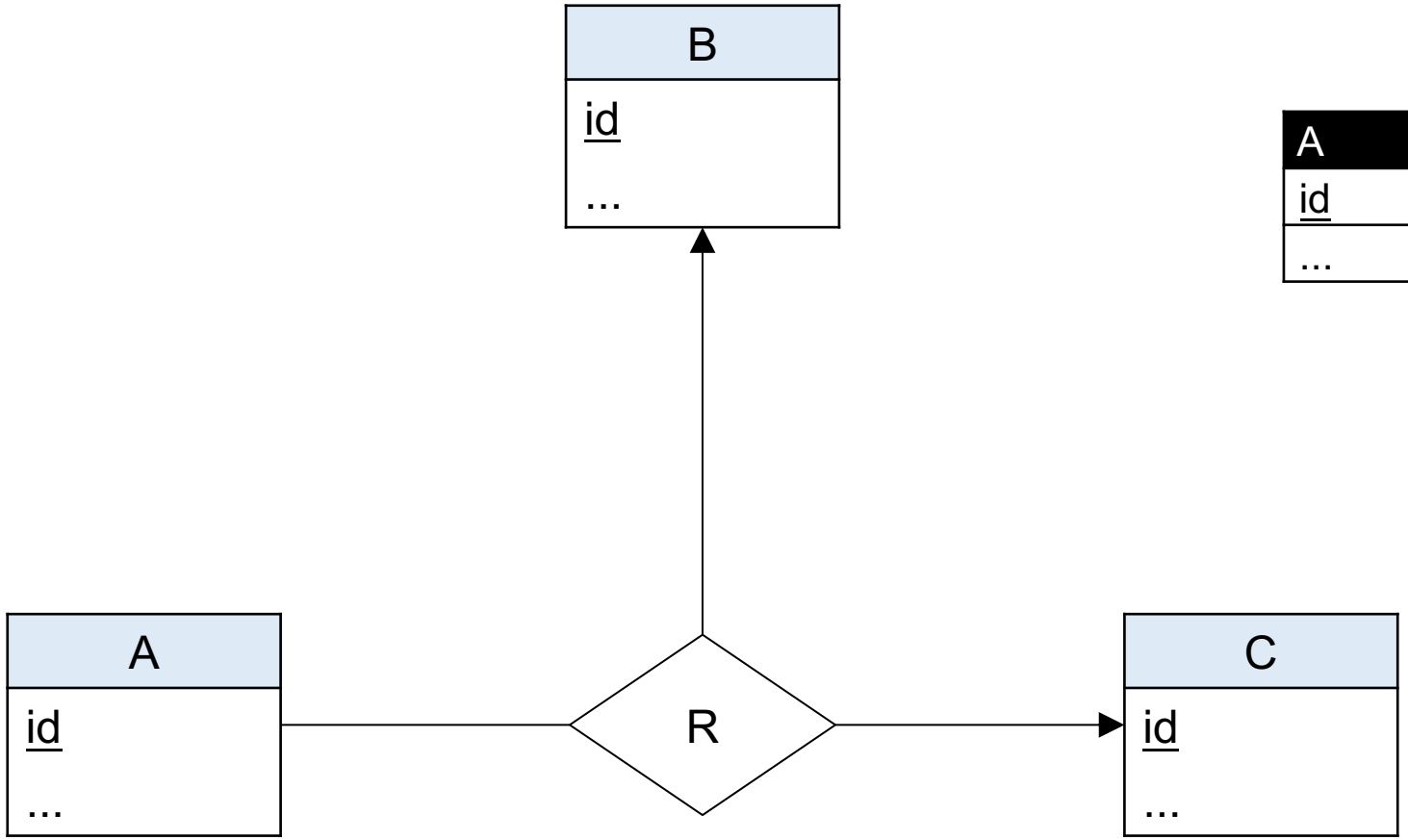


As tables ?

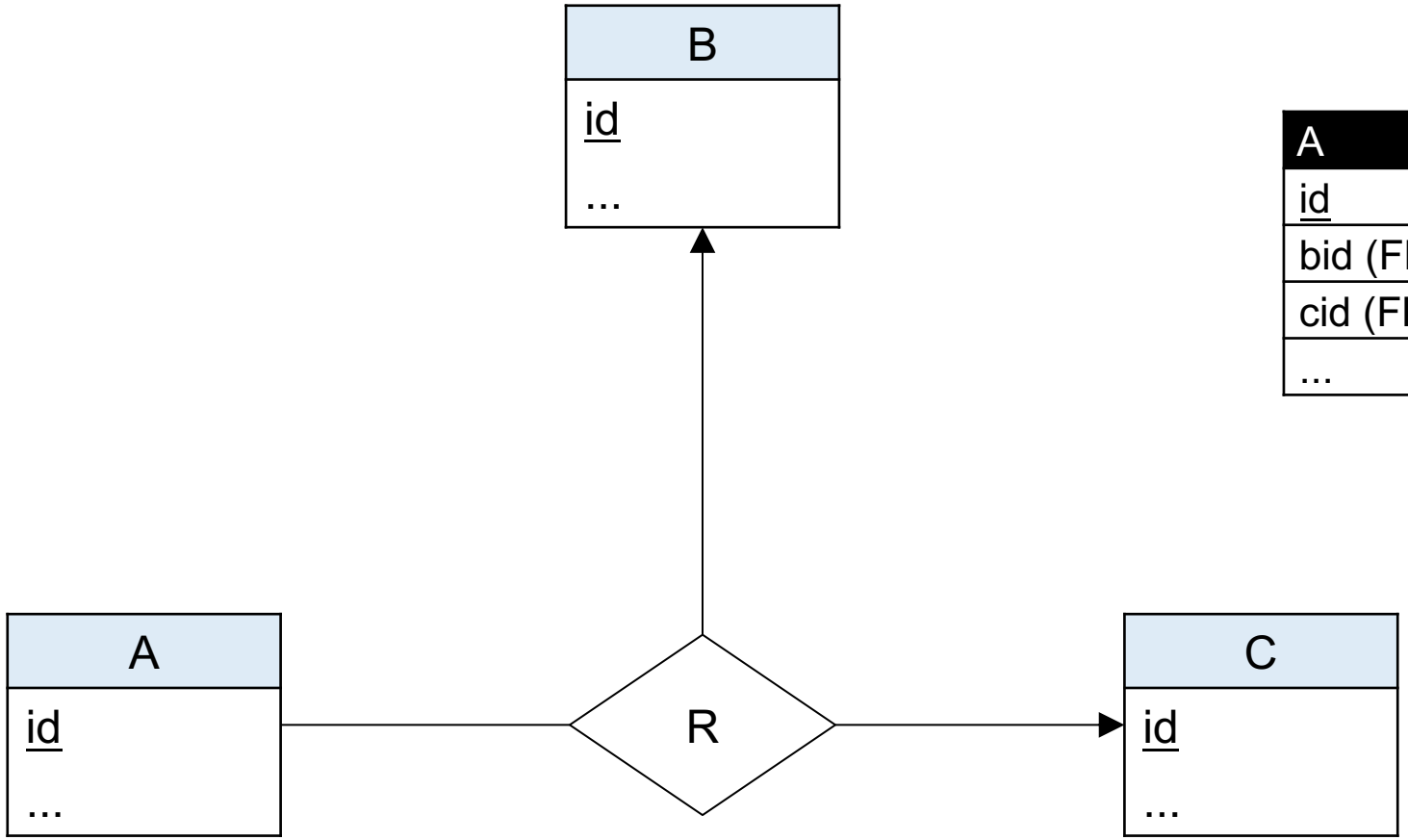
ERD best practice



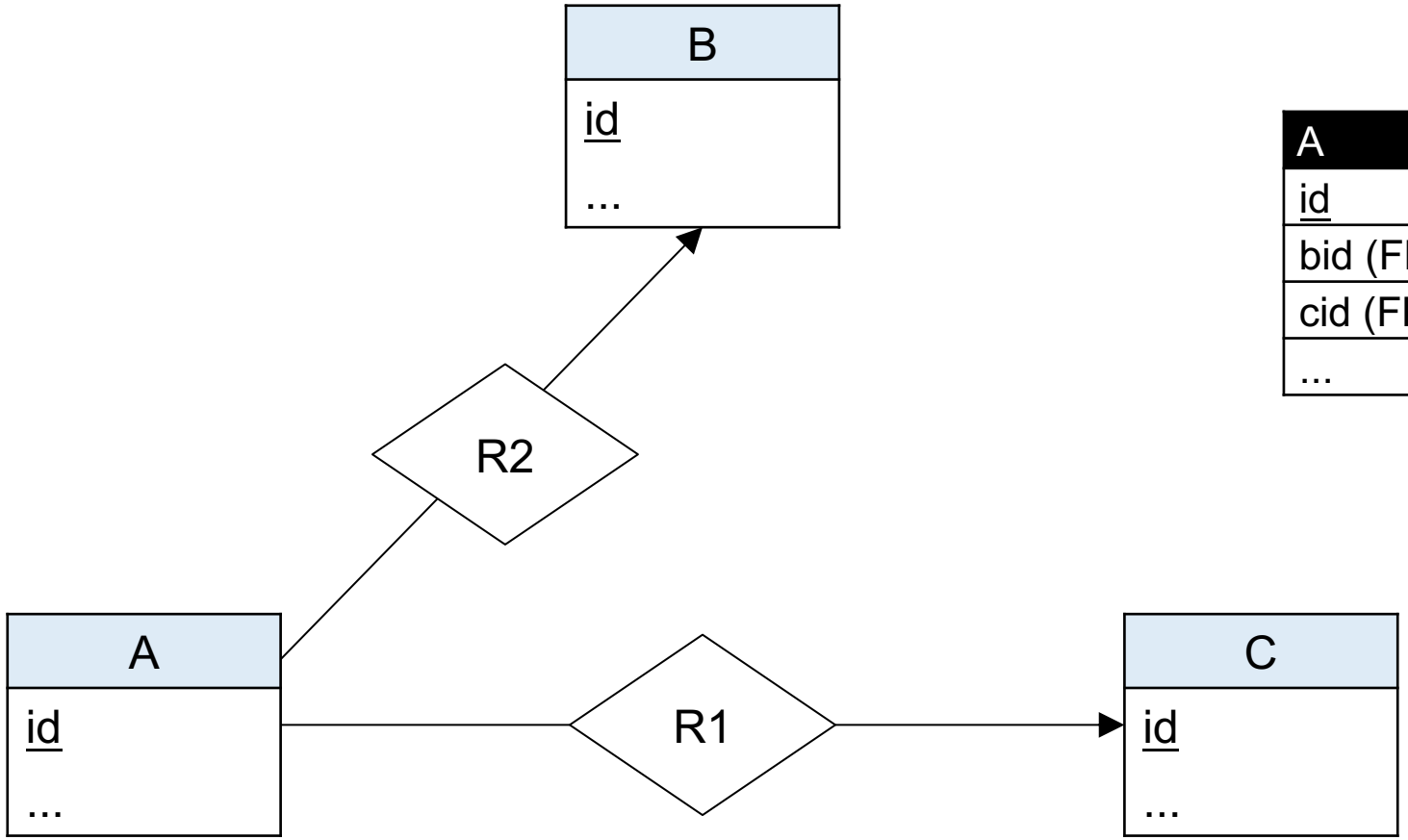
ERD best practice



ERD best practice

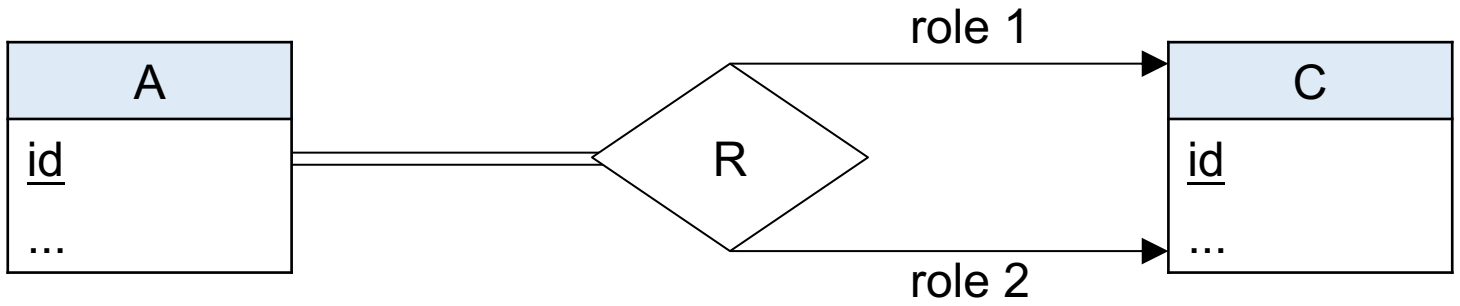


ERD best practice



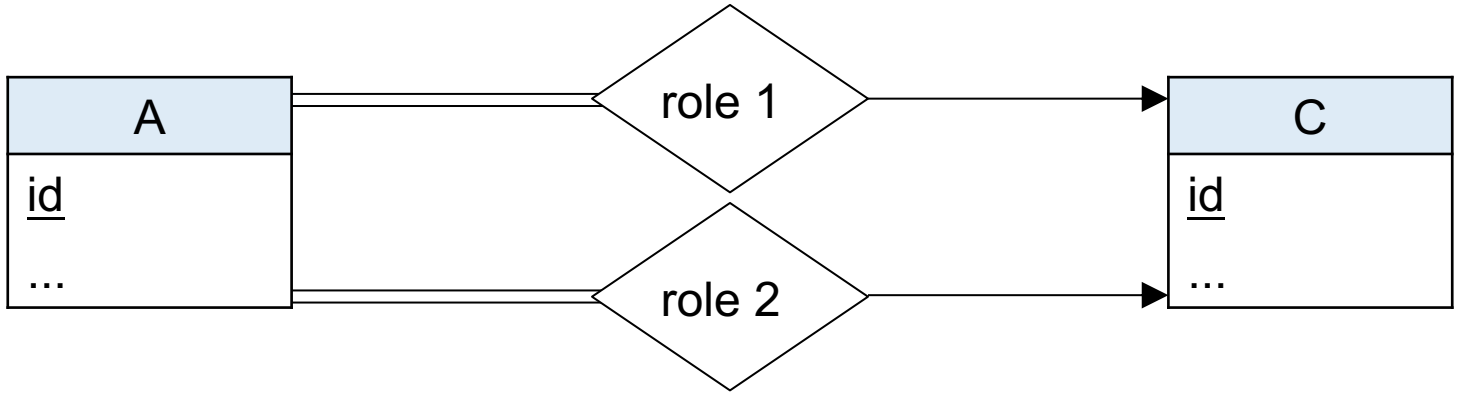
ERD best practice

?

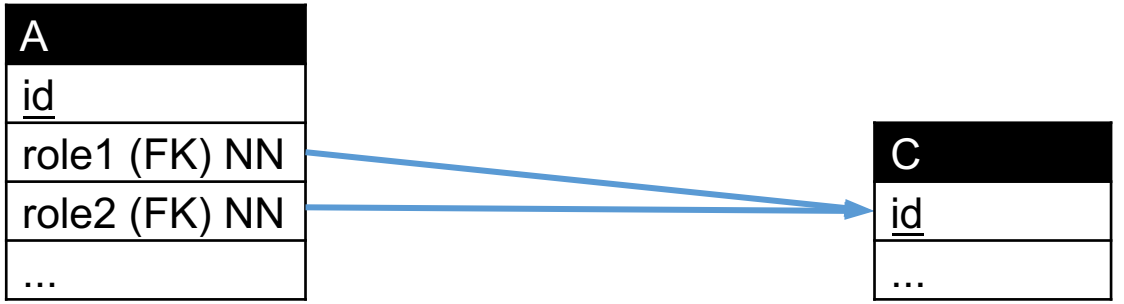


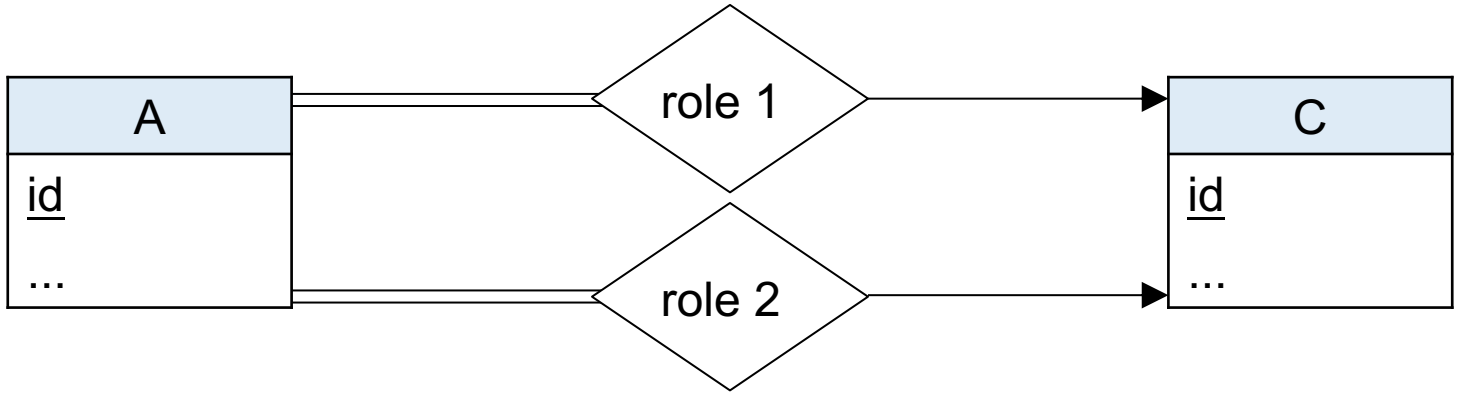
ERD best practice

As tables ?

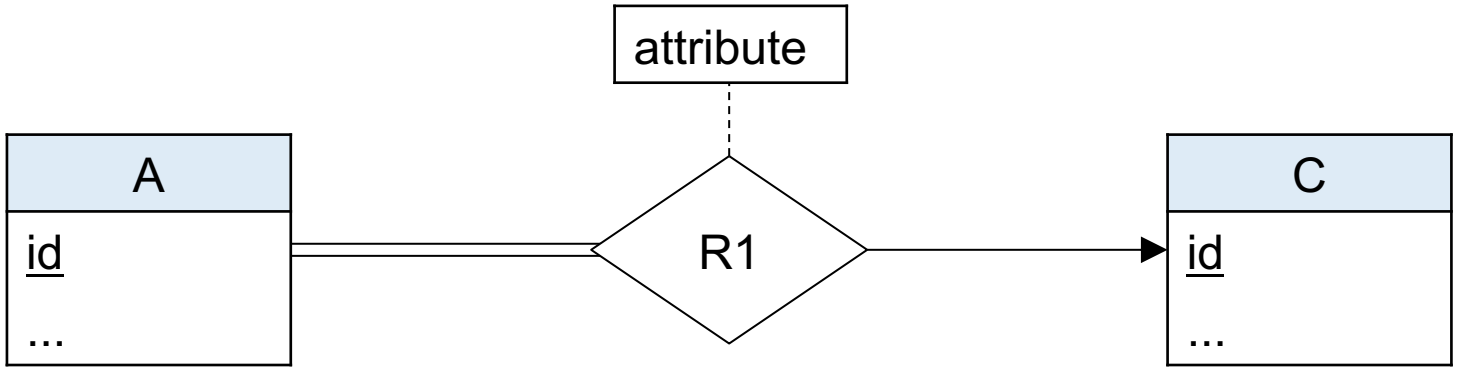


ERD best practice



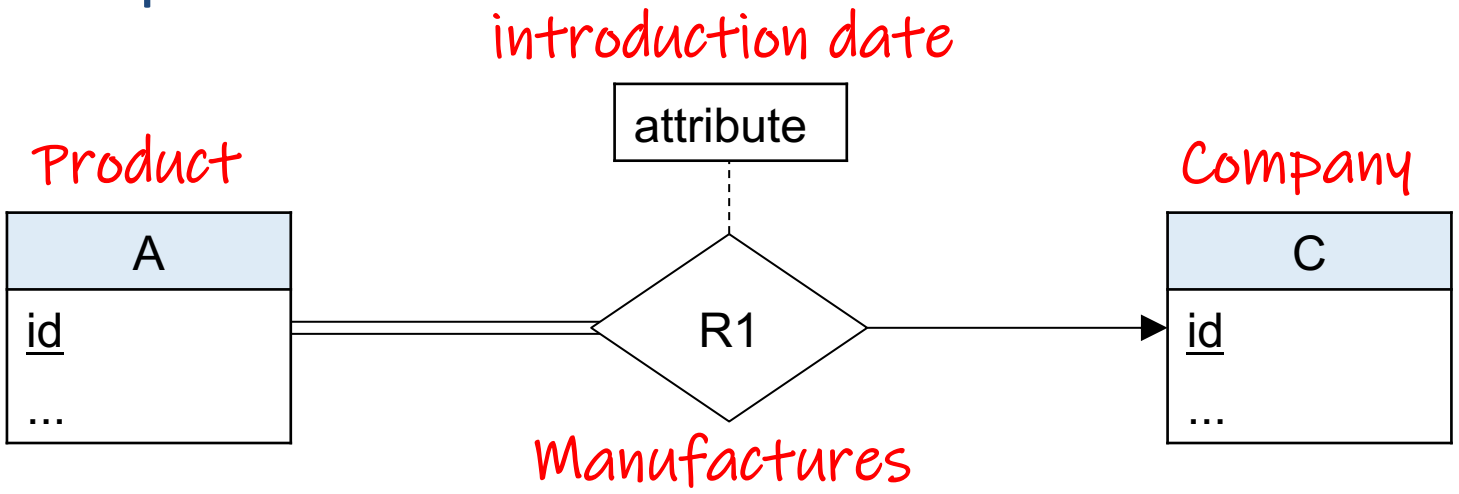


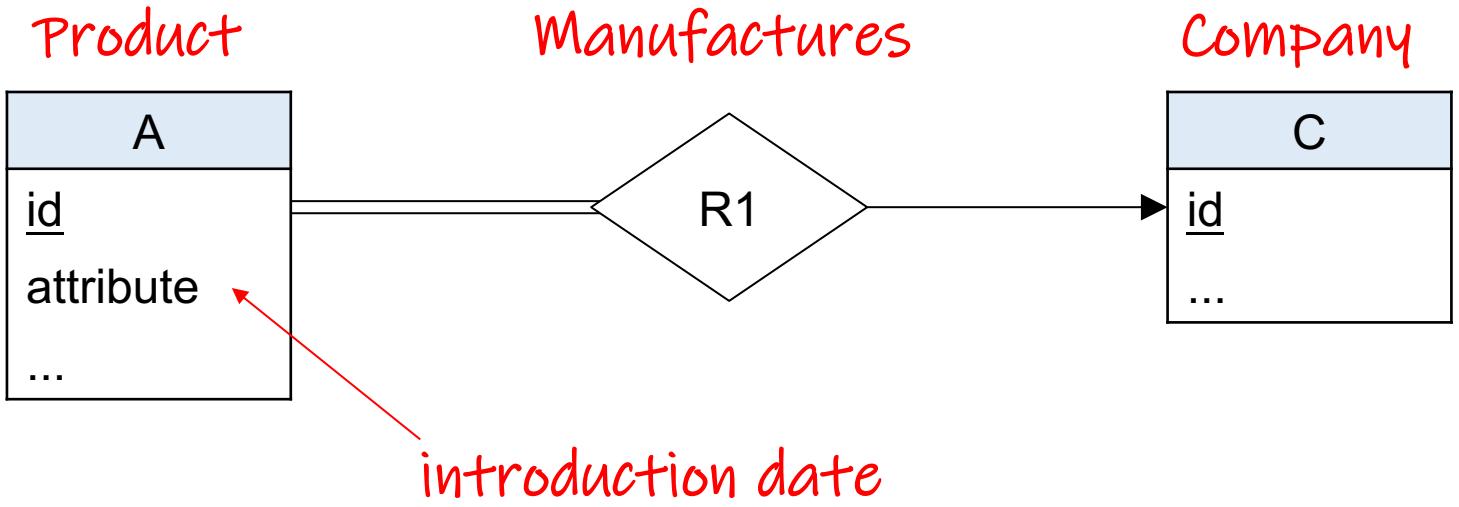
ERD best practice



?

ERD best practice





Boyce-Codd Normal Form (BCNF)

Quick recap FDs

- **Functional Dependency (FD)**: The value of one set of attributes (the **determinant**) uniquely determines the value of another set of attributes (the **dependents**)
- A **superkey (SK)** is as a set of attributes of a relation schema upon which all attributes of the schema are functionally dependent.
- A **candidate key (CK)** is a non-redundant (minimal) SK (sometimes called just "**a key**")
  - **Prime attribute**: belonging to some candidate key (the opposite is sometimes called a "**nonkey attribute**")
- **Partial FD**: FD in which some non-prime attributes are functionally dependent on part (but not all) of any CK
- **Transitive FD**: An FD between *two (or more) nonkey attributes* (important for distinction 3NF vs BCNF!)
- **3NF**: no partial nor transitive FD

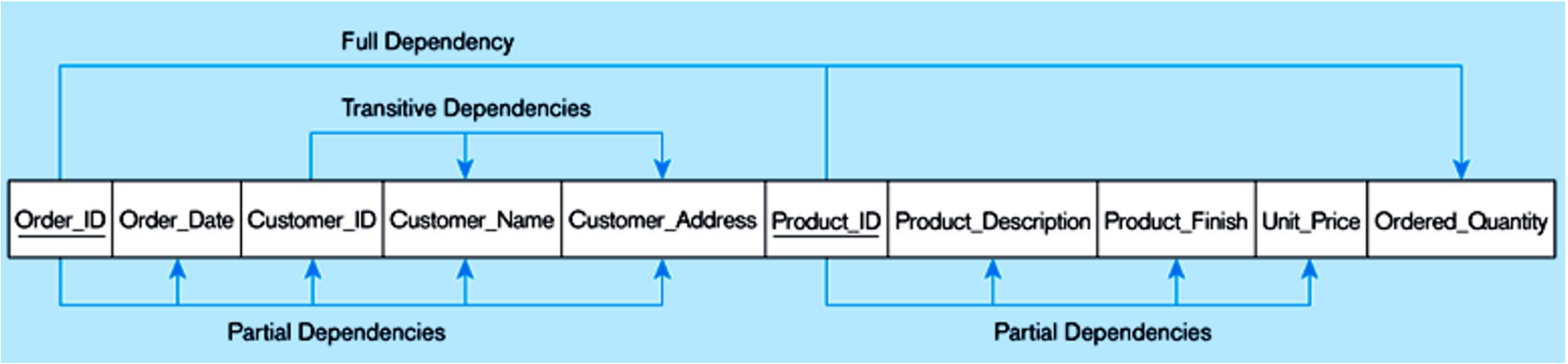


Fig Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010.

3NF to BCNF

Assume this FD holds

STUDENT ADVISOR

SID	Major	Advisor	MajGPA
123	Physics	Hawking	4.0
123	Music	Mahler	3.3
456	Literature	Michener	3.2
789	Music	Bach	3.7
678	Physics	Hawking	3.5

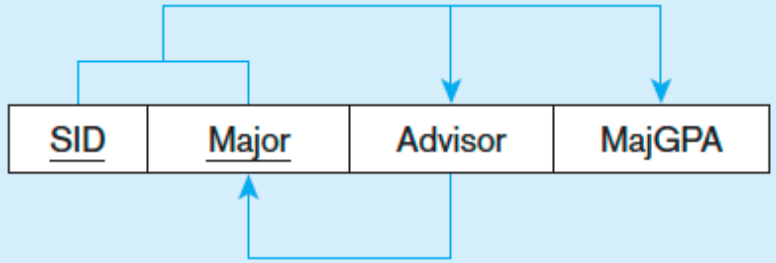
Fig Source: Hoffer, Ramesh, Topi, "Modern database management", 10th ed, Appendix B, 2010.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

3NF to BCNF

Assume this FD holds

STUDENT ADVISOR

SID	Major	Advisor	MajGPA
123	Physics	Hawking	4.0
123	Music	Mahler	3.3
456	Literature	Michener	3.2
789	Music	Bach	3.7
678	Physics	Hawking	3.5



Recall:

- partial FD: non-prime attributes are functionally dependent on part (but not all) of any CK
- transitive FD: FD between two (or more) nonkey attributes
- Prime attribute: belonging to some CK

Is this in 3NF?

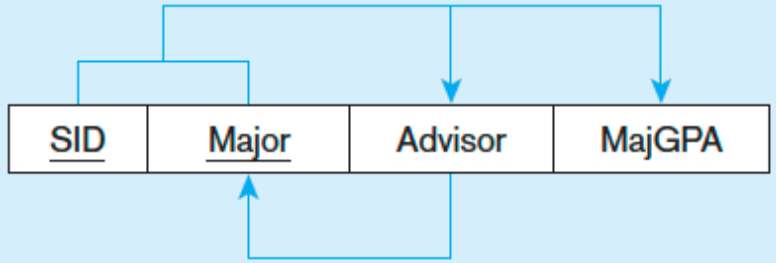
?

Fig Source: Hoffer, Ramesh, Topi, "Modern database management", 10th ed, Appendix B, 2010.

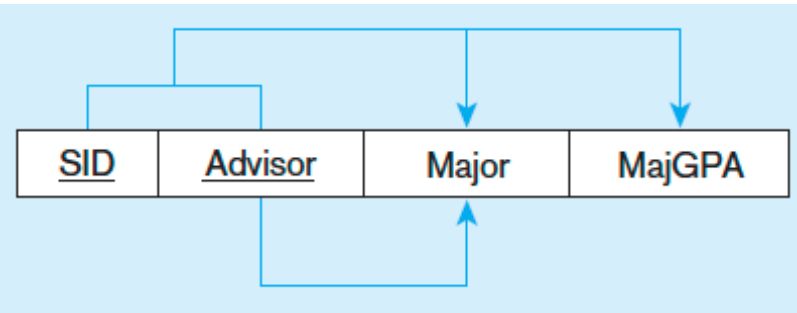
3NF to BCNF

STUDENT ADVISOR

SID	Major	Advisor	MajGPA
123	Physics	Hawking	4.0
123	Music	Mahler	3.3
456	Literature	Michener	3.2
789	Music	Bach	3.7
678	Physics	Hawking	3.5



in 3NF



not in 3NF (so we know how to decompose it)

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

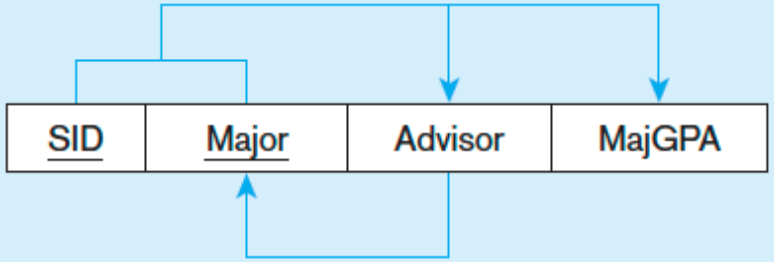
3NF to BCNF

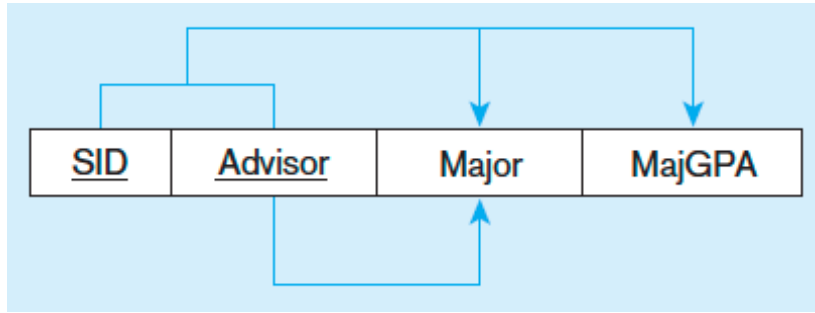
in 3NF, but not BCNF

in 3NF and BCNF

STUDENT ADVISOR

SID	Major	Advisor	MajGPA
123	Physics	Hawking	4.0
123	Music	Mahler	3.3
456	Literature	Michener	3.2
789	Music	Bach	3.7
678	Physics	Hawking	3.5





STUDENT

SID	Advisor	MajGPA
123	Hawking	4.0
123	Mahler	3.3
456	Michener	3.2
789	Bach	3.7
678	Hawking	3.5

ADVISOR

Advisor	Major
Hawking	Physics
Mahler	Music
Michener	Literature
Bach	Music

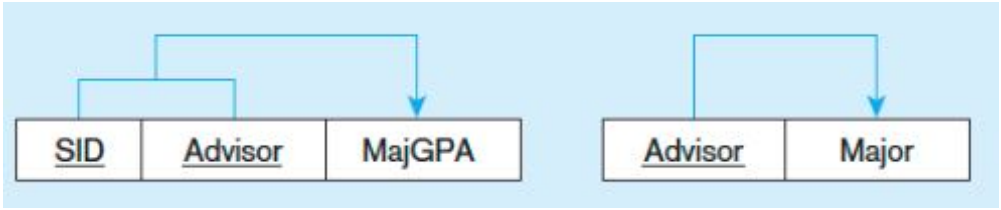


Fig Source: Hoffer, Ramesh, Topi, "Modern database management", 10th ed, Appendix B, 2010.

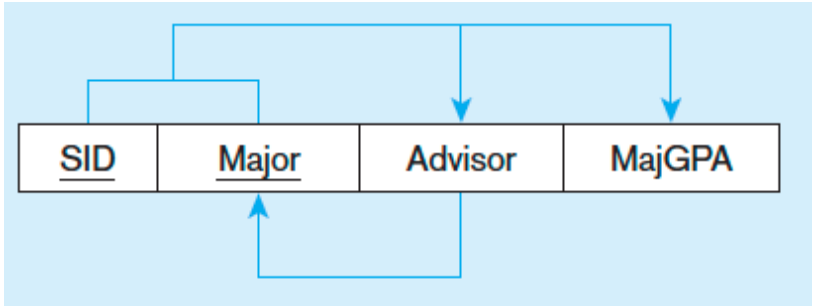
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

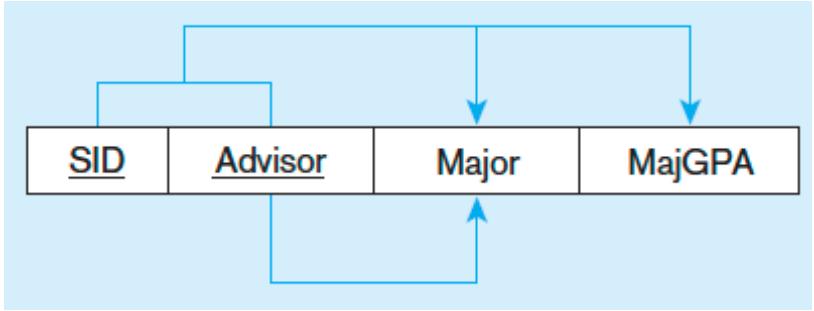
3NF to BCNF

in 3NF, but not BCNF

STUDENT ADVISOR

SID	Major	Advisor	MajGPA
123	Physics	Hawking	4.0
123	Music	Mahler	3.3
456	Literature	Michener	3.2
789	Music	Bach	3.7
678	Physics	Hawking	3.5





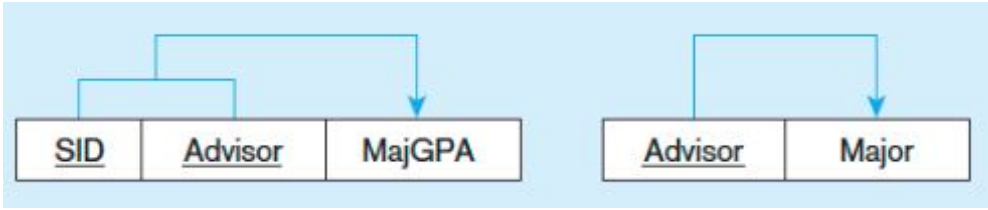
in 3NF and BCNF

STUDENT

SID	Advisor	MajGPA
123	Hawking	4.0
123	Mahler	3.3
456	Michener	3.2
789	Bach	3.7
678	Hawking	3.5

ADVISOR

Advisor	Major
Hawking	Physics
Mahler	Music
Michener	Literature
Bach	Music



BCNF instead of 3NF:

- less redundancy

Fig Source: Hoffer, Ramesh, Topi, "Modern database management", 10th ed, Appendix B, 2010.

3NF to BCNF

unpreserved FD

STUDENT ADVISOR

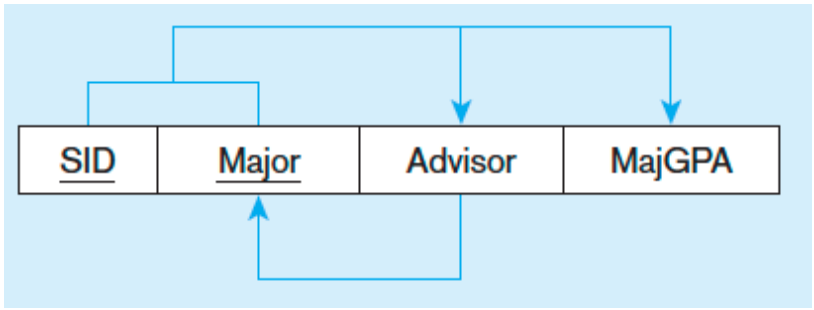
SID	Major	Advisor	MajGPA
123	Physics	Hawking	4.0
123	Music	Mahler	3.3
456	Literature	Michener	3.2
789	Music	Bach	3.7
678	Physics	Hawking	3.5

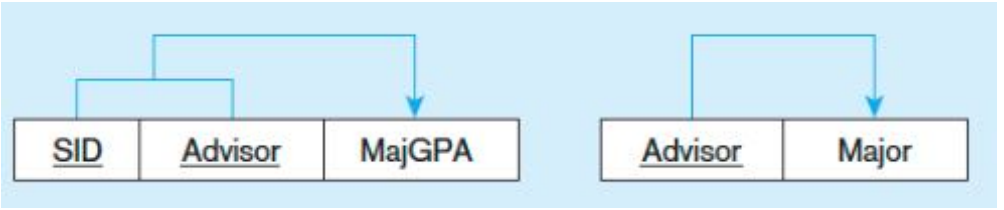
STUDENT

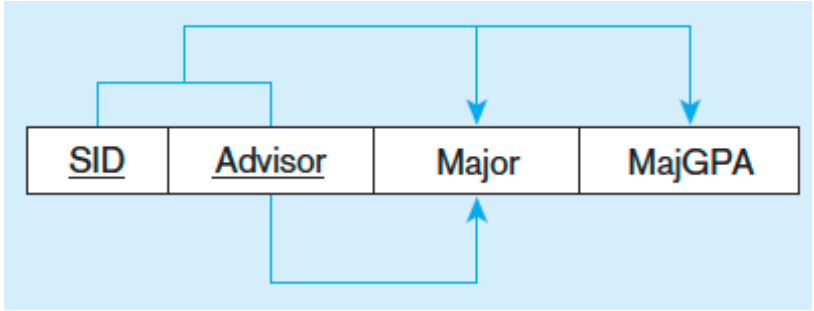
SID	Advisor	MajGPA
123	Hawking	4.0
123	Mahler	3.3
456	Michener	3.2
789	Bach	3.7
678	Hawking	3.5

ADVISOR

Advisor	Major
Hawking	Physics
Mahler	Music
Michener	Literature
Bach	Music





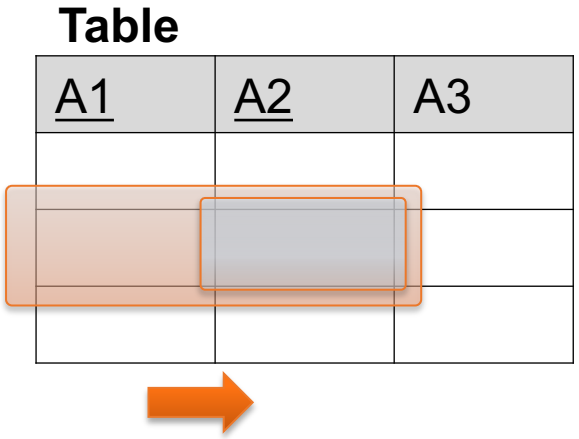


BCNF instead of 3NF:

- less redundancy
- but some FDs may not be preserved (split across tables) and thus hard to enforce: insert Student(789, "Mahler", 3.8) ?

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

Definition Trivial FDs



$A_1, A_2 \rightarrow A_2$

More general:

$A_1, \ldots, A_m \rightarrow A_j$ for any $j=1,\ldots,m$

Boyce-Codd Normal Form (BCNF)

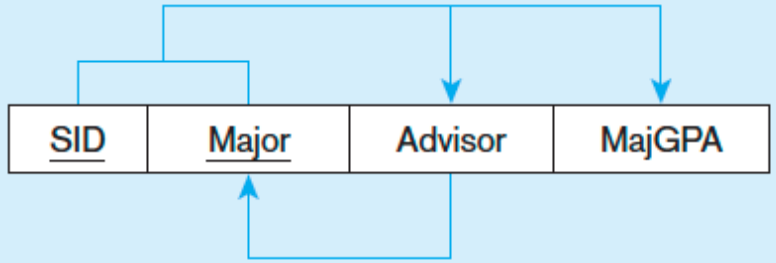
- Boyce-Codd normal form (BCNF)
  - A relation is in BCNF, if and only if, every **(non-trival) determinant** is a superkey (SK), i.e. it determines all other attributes in a relation.

- The difference between 3NF and BCNF is that for a FD A→B,
  - 3NF allows this dependency in a relation if B is a prime attribute
    - (even if A is not a candidate key (CK))
  - whereas BCNF insists that for this dependency to remain in a relation, A must be a SK (contain a CK).

3NF to BCNF

STUDENT ADVISOR

SID	Major	Advisor	MajGPA
123	Physics	Hawking	4.0
123	Music	Mahler	3.3
456	Literature	Michener	3.2
789	Music	Bach	3.7
678	Physics	Hawking	3.5



Is this in BCNF?

?

Recall:

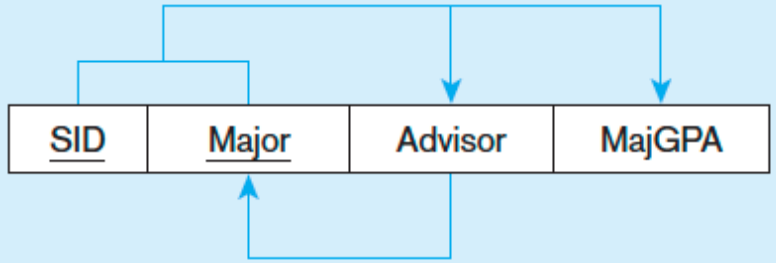
- partial FD: non-prime attributes are functionally dependent on part (but not all) of any CK
- transitive FD: FD between two (or more) nonkey attributes
- Prime attribute: belonging to some CK
- A relation is in BCNF, if and only if, every (non-trival) determinant is a superkey (SK).

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

3NF to BCNF

STUDENT ADVISOR

SID	Major	Advisor	MajGPA
123	Physics	Hawking	4.0
123	Music	Mahler	3.3
456	Literature	Michener	3.2
789	Music	Bach	3.7
678	Physics	Hawking	3.5



Recall:

- partial FD: non-prime attributes are functionally dependent on part (but not all) of any CK
- transitive FD: FD between two (or more) nonkey attributes
- Prime attribute: belonging to some CK
- A relation is in BCNF, if and only if, every (non-trival) determinant is a superkey (SK).

Is this in BCNF?

No

!

Difference 3NF / BCNF: For a FD A→B:

- 3NF allows this FD in a relation if B is a prime attribute
- BCNF insists that for this FD to remain in a relation, A must be a SK (contain a CK).

Fig Source: Hoffer, Ramesh, Topi, Modern database management, 10th ed, Appendix B, 2010.

BCNF vs 3NF

- **BCNF**: For every nontrival FD X→Y over relation *R*:
  - X is a superkey of R

- **3NF**: For every nontrival FD X→Y over relation *R*, either:
  - X is a superkey of R
  - or Y is prime (i.e. it is part of some CK)

Recall: a FD X→Y is "trivial" iff Y⊆X

Recall: no subset of a CK is a CK

Why do we have both BCNF and 3NF?

There is a trade-off

- BCNF is stricter than 3NF (if data is in BCNF, then it is also in 3NF)
- **Advantages** of **3NF** over BCNF:
  - It is always possible to obtain a 3NF design and **preserving all functional dependencies** ("dependency preservation": all FDs hold within one table)
  - BCNF may not preserve all FDs (they become split across tables)
- **Disadvantages** of **3NF** over BCNF:
  - There is the problem of repetition of information (**more redundancy**).
  - We may have to use null values to represent some of the possible meaningful relationships among data items.

Decompositions

Recap: We decompose relations to remove redundancies

- We saw that **redundancies** in the data (caused by “bad FDs”) can lead to data **anomalies** (insert, update, deletion anomalies)

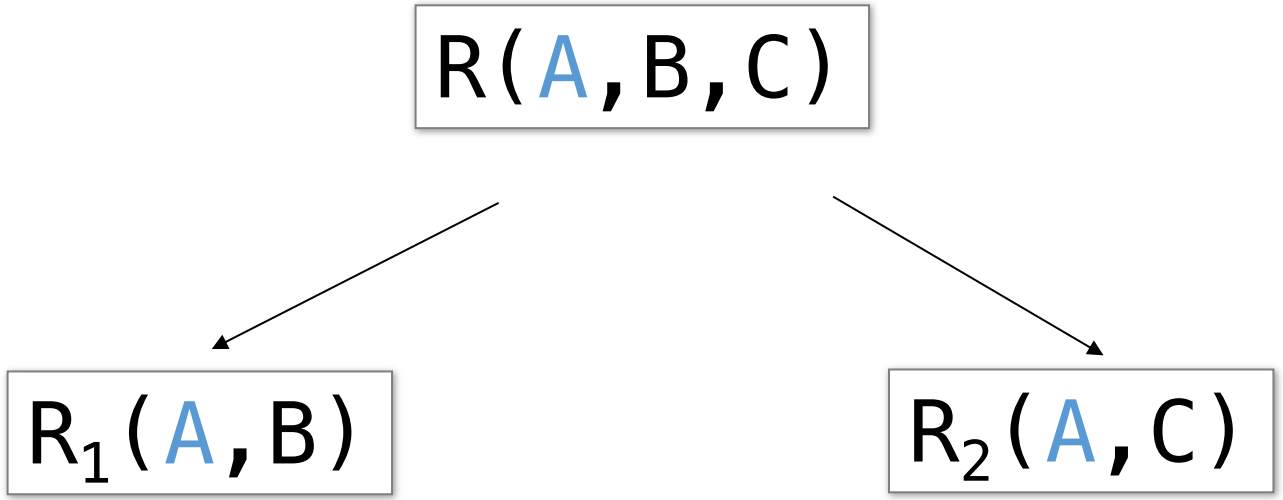
- We developed mechanisms to detect and remove redundancies by decomposing tables into 3NF or BCNF ("**normalization**")
  - Both 3NF and BCNF decomposition are standard practice and widely used!

- However, sometimes "**decompositions**" can lead to more subtle unwanted effects…

Next let's try to understand when this can happen ☺

Decompositions in General

Attributes A, B, C can also stand for a set of attributes $A_1, \ldots A_n$, $B_1, \ldots B_m$, $C_1, \ldots C_p$, and everything still holds in an obvious way. But it is far easier to follow along this simplified presentation.



R_1 = the *projection* of R on A, B

R_2 = the *projection* of R on A, C

Lossless Decomposition

A B C

Name	Price	Category
Gizmo	19.99	Gadget
OneClick	24.99	Camera
Gizmo	19.99	Camera

A B

Name	Price
Gizmo	19.99
OneClick	24.99
~~Gizmo~~	~~19.99~~

A C

Name	Category
Gizmo	Gadget
OneClick	Camera
Gizmo	Camera

Sometimes a decomposition is "correct", i.e. it is lossless (i.e. we don't "loose" information)

Lossless Decomposition

C B A

Name	Price	Category
Gizmo	19.99	Gadget
OneClick	24.99	Camera
Gizmo	19.99	Camera

C A

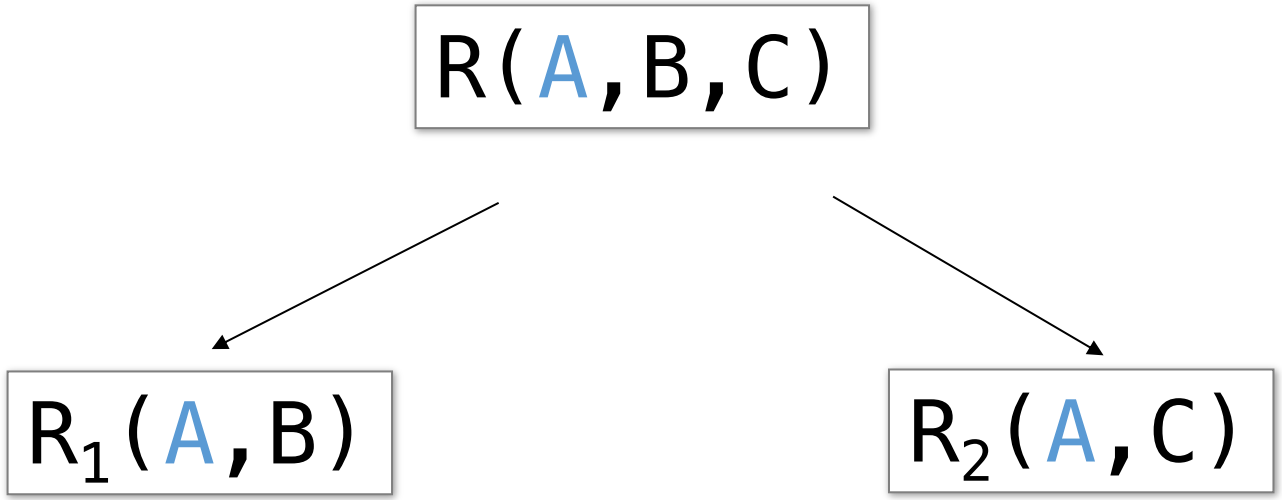
Name	Category
Gizmo	Gadget
OneClick	Camera
Gizmo	Camera

B A

Price	Category
19.99	Gadget
24.99	Camera
19.99	Camera

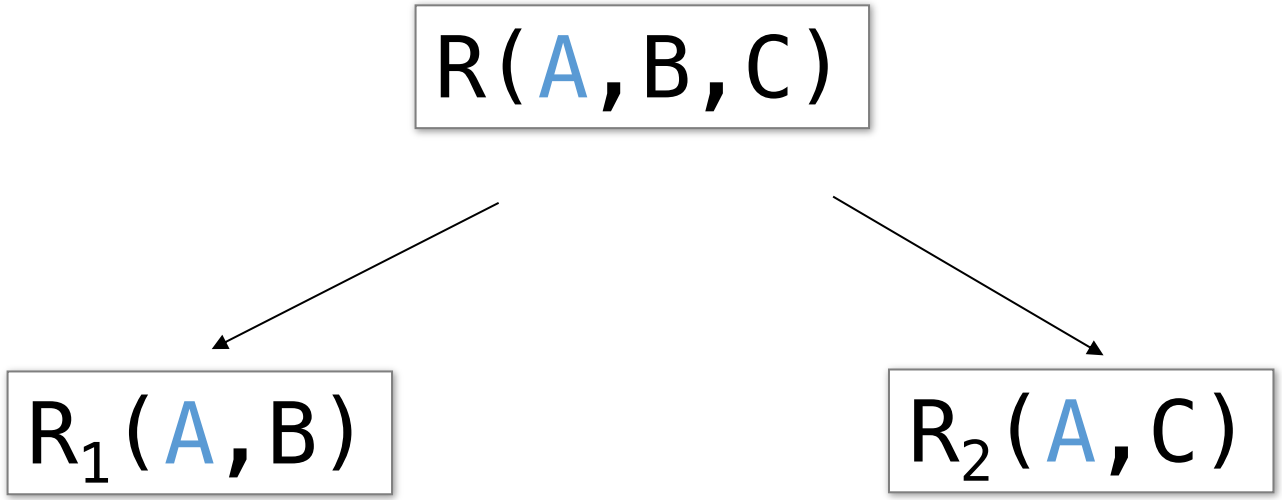
Sometimes such a decomposition is wrong. Here we lost information (which names appear with which prices?). Why does this happen?

Lossless Decompositions



A decomposition R to (R1, R2) is **lossless** if R = R1 ⋈ R2

Lossless Decompositions



If A → B then the decomposition is lossless

Note: we don't need A → C at the same time

BCNF decomposition is always lossless. Why?

A familiar example

Item

PName	Price	Category	Manufacturer	StockPrice	Country
Gizmo	$19.99	Gadgets	GizmoWorks	25	USA
Powergizmo	$29.99	Gadgets	GizmoWorks	25	USA
SingleTouch	$149.99	Photography	Canon	65	Japan
MultiTouch	$203.99	Household	Hitachi	15	Japan

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Company

Manufacturer	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

A problem with BCNF

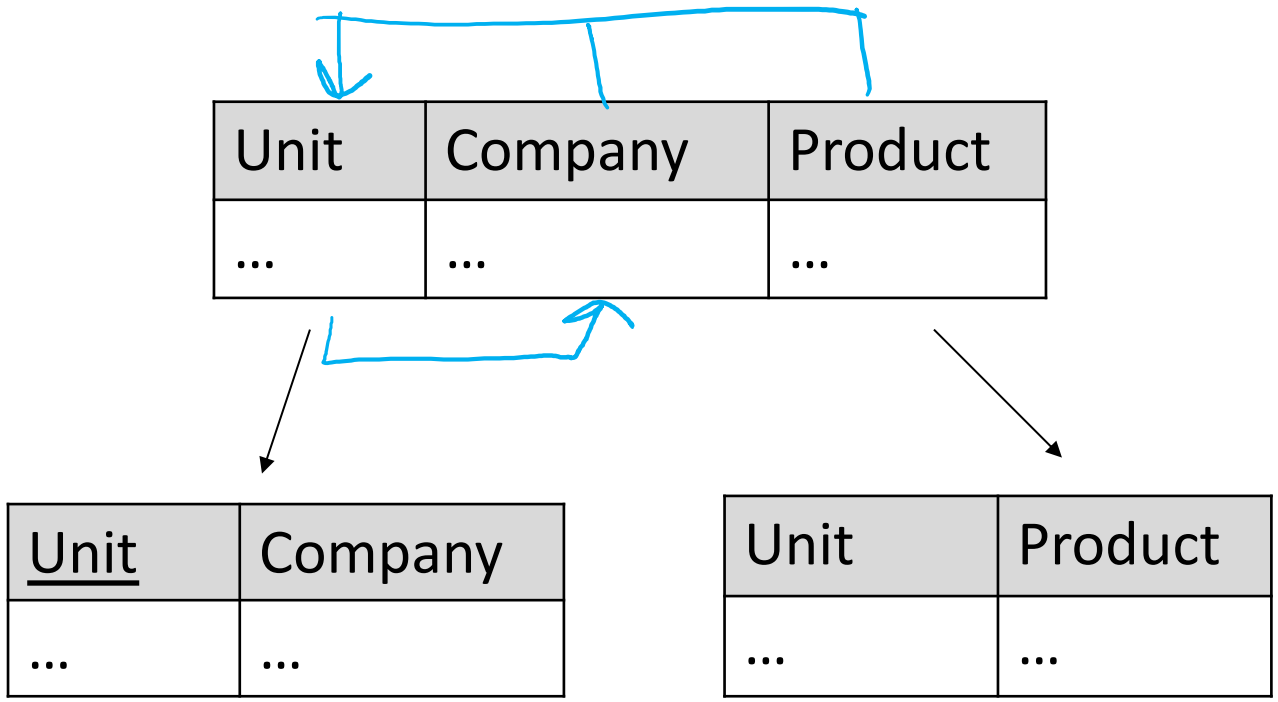
Problem: To enforce a FD, we may have to reconstruct original relation—*on each insert!*

Note: This is historically inaccurate, but it makes it easier to explain

A problem with BCNF

Assume following FDs hold:

Unit → Company
Company, Product → Unit



We do a BCNF decomposition on FD: Unit → Company

We have not preserved the FD Company, Product → Unit !

So Why is that a Problem?

Assume following FDs hold:

Unit → Company
Company, Product → Unit

Unit	Company
Toys	GizmoWorks
Future	GizmoWorks

Unit	Product
Toys	Gizmo
Future	Gizmo

No problem so far. All local FD's are satisfied:
Unit → Company

Unit	Company	Product
Toys	GizmoWorks	Gizmo
Future	GizmoWorks	Gizmo

Let's put all the data back into a single table again:

Violates the FD Company, Product → Unit

The Problem

- We started with a table **R** and FDs **F**
- We decomposed R into BCNF tables **R1**, **R2**, …
  with their own FDs **F1**, **F2**, …
- We insert some tuples into each of the relations which satisfy their local FDs but when reconstruct it violates some FD across tables!

Practical Problem: To enforce FD, must reconstruct R, *on each insert!*

Possible Solutions

- Various ways to handle so that decompositions are dependency preserving
  - For example 3NF: stop short of full BCNF decompositions.

- Usually a tradeoff between redundancy / data anomalies and FD preservation…

BCNF still more common- with additional steps to keep track of lost FDs…

Summary and Quiz

- A decomposition of relation R into relation R1 and R2 is **lossless** if

?

Summary and Quiz

- A decomposition of relation R into relation R1 and R2 is **lossless** if $R1 \bowtie R2 = R$
- A decomposition is **dependency-preserving** if

?

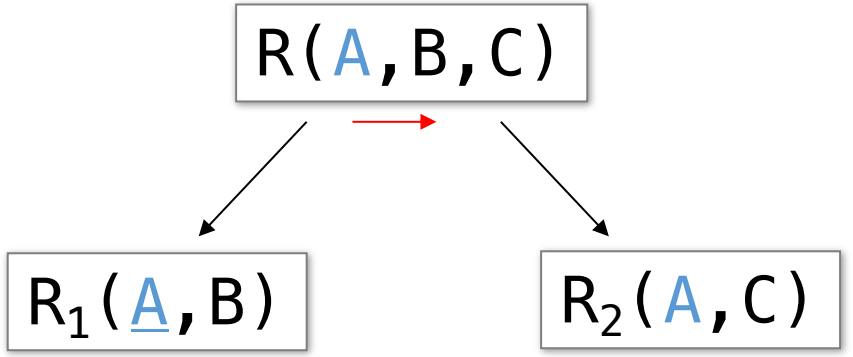
Summary and Quiz

- A decomposition of relation R into relation R1 and R2 is **lossless** if R1 ⋈ R2 = R
- A decomposition is **dependency-preserving** if
  - all FDs (functional dependencies) from R are preserved in either R1 or R2 (or both or derivable from a combination of the FDs in R1 and R2).
    - thus the FDs of R can be obtained by taking the union of the FDs of all the decomposed relation
  - The **dependency preservation decomposition** is another property of decomposed relational database schema D in which each FD X → Y either appeared directly in one of the relation schemas R_i in the decomposed D or could be inferred from the dependencies that appear in some Ri.
- A decomposition of Relation R into R1 and R2 is lossless if and only if at least one of following dependencies hold:

?

Summary and Quiz

- A decomposition of relation R into relation R1 and R2 is **lossless** if R1 ⋈ R2 = R
- A decomposition is **dependency-preserving** if
  - all FDs (functional dependencies) from R are preserved in either R1 or R2 (or both or derivable from a combination of the FDs in R1 and R2).
    - thus the FDs of R can be obtained by taking the union of the FDs of all the decomposed relation
  - The **dependency preservation decomposition** is another property of decomposed relational database schema D in which each FD X → Y either appeared directly in one of the relation schemas R_i in the decomposed D or could be inferred from the dependencies that appear in some Ri.
- A decomposition of Relation R into R1 and R2 is lossless if and only if at least one of following dependencies hold:
  - 1. R1 ∩ R2 → R1
  - 2. R1 ∩ R2 → R2



4NF and higher

3NF Motivation

A relation R is in 3rd normal form if :
Whenever there is a nontrivial dep. $A_1, A_2, ..., A_n \rightarrow B$ for R,
then $\{A_1, A_2, ..., A_n\}$ is a super-key for R,
or B is part of a key.

Tradeoffs:

BCNF: no anomalies, but may not preserve some FDs

3NF: keeps all FDs, but may have some anomalies

Motivation of 4NF and higher

Assume for each course, we can independently choose a lecturer and a book. What is the problem?

Classes

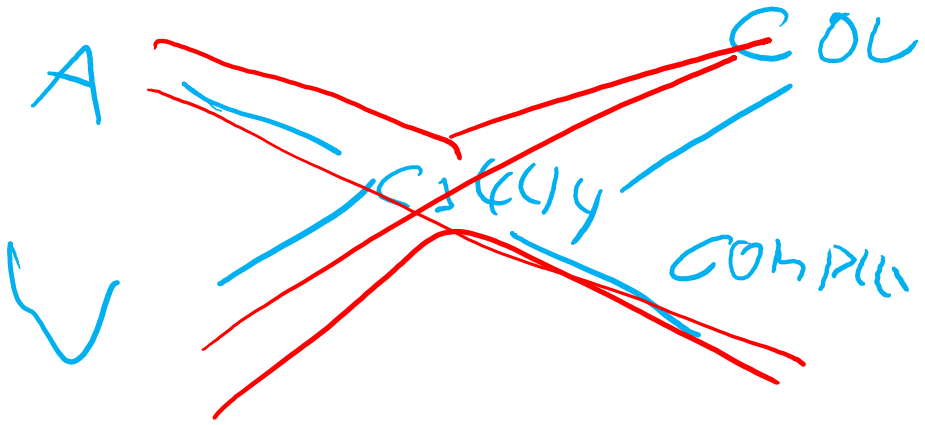
Course	Lecturer	Book
cse444	Alexandra	Complete book
cse444	Wolfgang	Complete book
cse444	Alexandra	Cow book

Motivation of 4NF and higher

Assume for each course, we can independently choose a lecturer and a book. What is the problem?

Classes

Course	Lecturer	Book
cse444	Alexandra	Complete book
cse444	Wolfgang	Complete book
cse444	Alexandra	Cow book
cse444	Wolfgang	Cow book



Multi-valued dependency (MVD) Course $\rightarrow\!\!\rightarrow$ Lecturer:

In every legal instance, each Course value is associated with a set of Lecturer values and this set is independent of the values in the other attributes (here Book).

Multi-valued dependency (MVD) generalized Functional dependencies (FDs)

Normalization Practice!

Parking Tickets: Original List

TABLE 4-6 Parking Tickets at Millennium College

Parking Ticket Table

St ID	L Name	F Name	Phone No	St Lic	Lic No	Ticket #	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
						16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
						16293	11/18/10	1	$15
						17892	12/13/10	2	$25

In what normal form is this data ?

Figure source: Hoffer, Ramesh, Topi, Modern database management, 10th ed, 2010.

Parking Tickets: Original List

TABLE 4-6 Parking Tickets at Millennium College

Parking Ticket Table

St ID	L Name	F Name	Phone No	St Lic	Lic No	Ticket #	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
						16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
						16293	11/18/10	1	$15
						17892	12/13/10	2	$25

Above none! Not even 1NF. Below now yes. Except there are still two problems ?

ST ID	L Name	F Name	Phone No	St Lic	Lic No	Ticket #	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
38249	Brown	Thomas	111-7804	FL	BRY 123	16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
82453	Green	Sally	391-1689	AL	TRE 141	16293	11/18/10	1	$15
82453	Green	Sally	391-1689	AL	TRE-141	17892	12/13/10	2	$25

Figure source: Hoffer, Ramesh, Topi, Modern database management, 10th ed, 2010.

Parking Tickets: Original List

TABLE 4-6 Parking Tickets at Millennium College

Parking Ticket Table

St ID	L Name	F Name	Phone No	St Lic	Lic No	Ticket #	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
						16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
						16293	11/18/10	1	$15
						17892	12/13/10	2	$25

ST ID	L Name	F Name	Phone No	St Lic	Lic No	Ticket #	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
38249	Brown	Thomas	111-7804	FL	BRY 123	16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
82453	Green	Sally	391-1689	AL	TRE 141	16293	11/18/10	1	$15
82453	Green	Sally	391-1689	AL	TRE-141	17892	12/13/10	2	$25

Figure source: Hoffer, Ramesh, Topi, Modern database management, 10th ed, 2010.
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Parking Tickets: Relation in 1NF

TABLE 4-6 Parking Tickets at Millennium College

Parking Ticket Table

St ID	L Name	F Name	Phone No	St Lic	Lic No	Ticket #	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
						16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
						16293	11/18/10	1	$15
						17892	12/13/10	2	$25

1. No empty spaces in attribut names. 2. We still need a PK!

?

ParkingTickets

STID	LName	FName	PhoneNo	StLic	LicNo	Ticketnr	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
38249	Brown	Thomas	111-7804	FL	BRY 123	16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
82453	Green	Sally	391-1689	AL	TRE 141	16293	11/18/10	1	$15
82453	Green	Sally	391-1689	AL	TRE-141	17892	12/13/10	2	$25

Figure source: Hoffer, Ramesh, Topi, Modern database management, 10th ed, 2010.

Parking Tickets: Relation in 1NF

TABLE 4-6 Parking Tickets at Millennium College

Parking Ticket Table

St ID	L Name	F Name	Phone No	St Lic	Lic No	Ticket #	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
						16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
						16293	11/18/10	1	$15
						17892	12/13/10	2	$25

ParkingTickets

STID	LName	FName	PhoneNo	StLic	LicNo	Ticketnr	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
38249	Brown	Thomas	111-7804	FL	BRY 123	16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
82453	Green	Sally	391-1689	AL	TRE 141	16293	11/18/10	1	$15
82453	Green	Sally	391-1689	AL	TRE-141	17892	12/13/10	2	$25

Figure source: Hoffer, Ramesh, Topi, Modern database management, 10th ed, 2010.

Parking Tickets: Dependency diagram

Draw all FDs!

1. Functional Dependencies (FDs) resulting from Primary Key (PK).

?

STID	LName	FName	PhoneNo	StLic	LicNo	Ticketnr	Date	Code	Fine

ParkingTickets

STID	LName	FName	PhoneNo	StLic	LicNo	Ticketnr	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
38249	Brown	Thomas	111-7804	FL	BRY 123	16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
82453	Green	Sally	391-1689	AL	TRE 141	16293	11/18/10	1	$15
82453	Green	Sally	391-1689	AL	TRE-141	17892	12/13/10	2	$25

Parking Tickets: Dependency diagram

1. Functional Dependencies (FDs) resulting from Primary Key (PK).

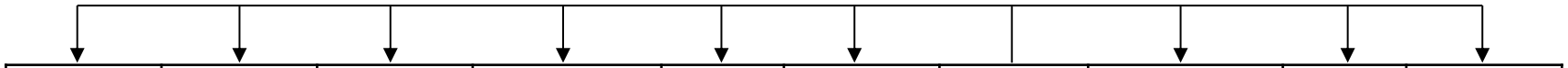
STID	LName	FName	PhoneNo	StLic	LicNo	Ticketnr	Date	Code	Fine

ParkingTickets

STID	LName	FName	PhoneNo	StLic	LicNo	Ticketnr	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
38249	Brown	Thomas	111-7804	FL	BRY 123	16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
82453	Green	Sally	391-1689	AL	TRE 141	16293	11/18/10	1	$15
82453	Green	Sally	391-1689	AL	TRE-141	17892	12/13/10	2	$25

Parking Tickets: Dependency diagram

1. Functional Dependencies (FDs) resulting from Primary Key (PK).
2. Code encodes the type of violation and thus determines the fine.



STID	LName	FName	PhoneNo	StLic	LicNo	Ticketnr	Date	Code	Fine

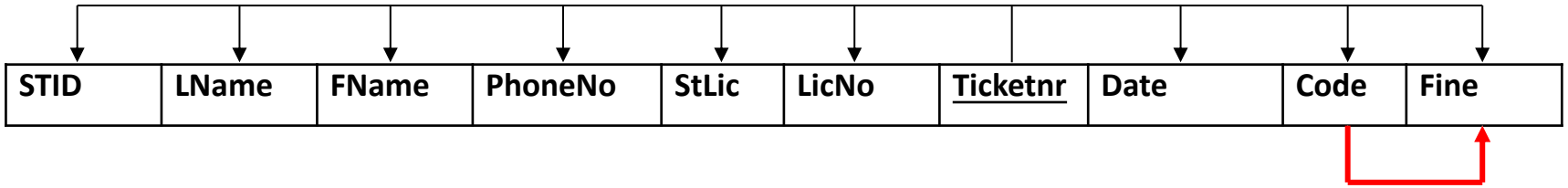
?

ParkingTickets

STID	LName	FName	PhoneNo	StLic	LicNo	Ticketnr	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
38249	Brown	Thomas	111-7804	FL	BRY 123	16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
82453	Green	Sally	391-1689	AL	TRE 141	16293	11/18/10	1	$15
82453	Green	Sally	391-1689	AL	TRE-141	17892	12/13/10	2	$25

Parking Tickets: Dependency diagram

1. Functional Dependencies (FDs) resulting from Primary Key (PK).
2. Code encodes the type of violation and thus determines the fine.

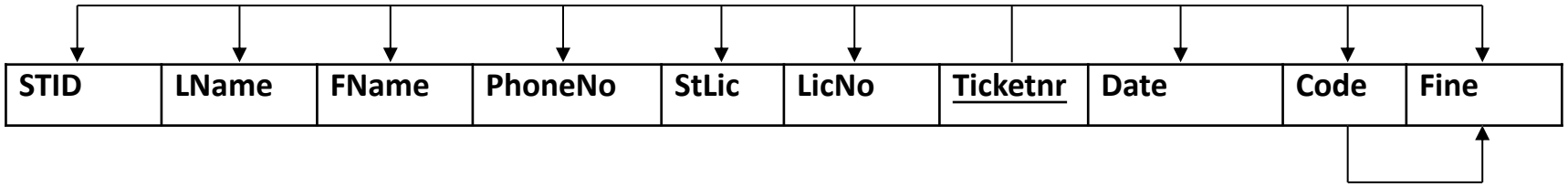


ParkingTickets

STID	LName	FName	PhoneNo	StLic	LicNo	Ticketnr	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
38249	Brown	Thomas	111-7804	FL	BRY 123	16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
82453	Green	Sally	391-1689	AL	TRE 141	16293	11/18/10	1	$15
82453	Green	Sally	391-1689	AL	TRE-141	17892	12/13/10	2	$25

Parking Tickets: Dependency diagram

3. Assume that each student can have maximal one car



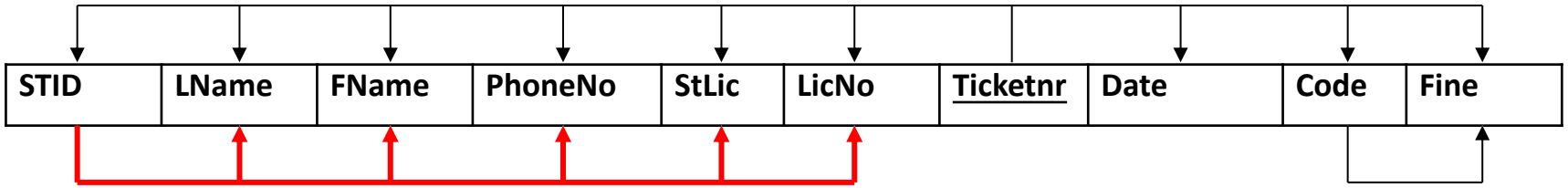
?

ParkingTickets

STID	LName	FName	PhoneNo	StLic	LicNo	Ticketnr	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
38249	Brown	Thomas	111-7804	FL	BRY 123	16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
82453	Green	Sally	391-1689	AL	TRE 141	16293	11/18/10	1	$15
82453	Green	Sally	391-1689	AL	TRE-141	17892	12/13/10	2	$25

Parking Tickets: Dependency diagram

3. Assume that each student can have maximal one car

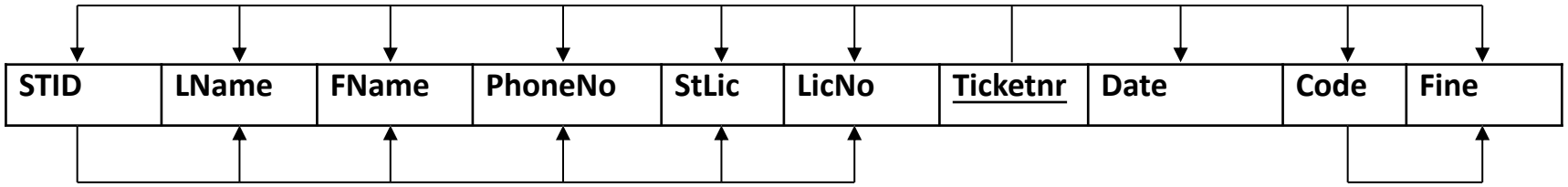


ParkingTickets

STID	LName	FName	PhoneNo	StLic	LicNo	Ticketnr	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
38249	Brown	Thomas	111-7804	FL	BRY 123	16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
82453	Green	Sally	391-1689	AL	TRE 141	16293	11/18/10	1	$15
82453	Green	Sally	391-1689	AL	TRE-141	17892	12/13/10	2	$25

Parking Tickets: Dependency diagram

3. Assume that each student can have maximal one car



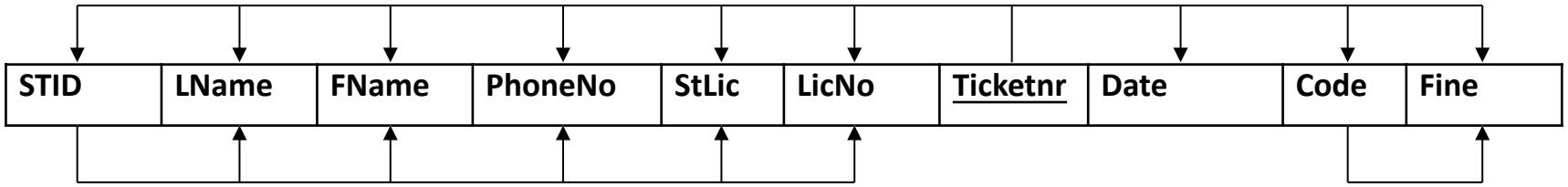
3'. Assume instead that each student can have more than one car ?

ParkingTickets

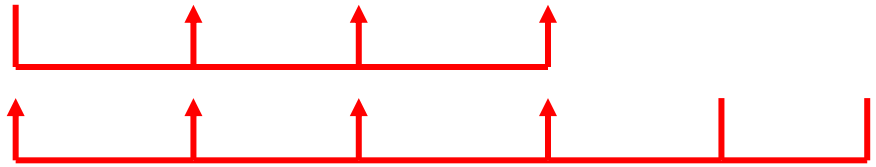
STID	LName	FName	PhoneNo	StLic	LicNo	Ticketnr	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
38249	Brown	Thomas	111-7804	FL	BRY 123	16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
82453	Green	Sally	391-1689	AL	TRE 141	16293	11/18/10	1	$15
82453	Green	Sally	391-1689	AL	TRE-141	17892	12/13/10	2	$25

Parking Tickets: Dependency diagram

3. Assume that each student can have maximal one car



3'. Assume instead that each student can have more than one car

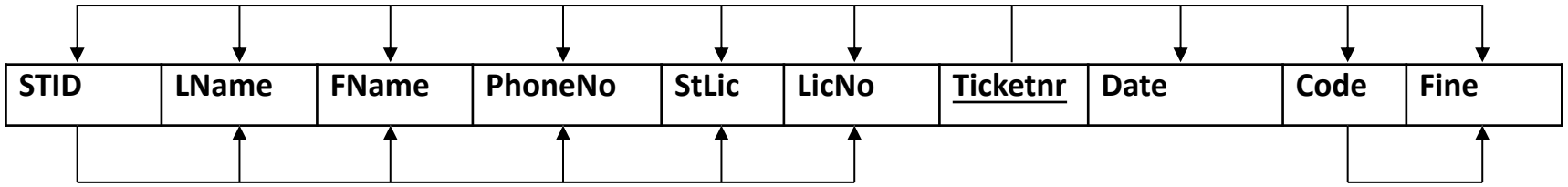


ParkingTickets

STID	LName	FName	PhoneNo	StLic	LicNo	Ticketnr	Date	Code	Fine
38249	Brown	Thomas	111-7804	FL	BRY 123	15634	10/17/10	2	$25
38249	Brown	Thomas	111-7804	FL	BRY 123	16017	11/13/10	1	$15
82453	Green	Sally	391-1689	AL	TRE 141	14987	10/05/10	3	$100
82453	Green	Sally	391-1689	AL	TRE 141	16293	11/18/10	1	$15
82453	Green	Sally	391-1689	AL	TRE-141	17892	12/13/10	2	$25

Parking Tickets: Relations in 3NF

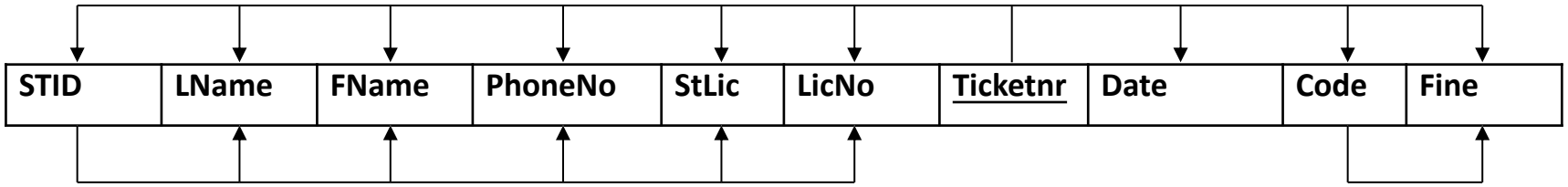
3. Assume that each student can have maximal one car



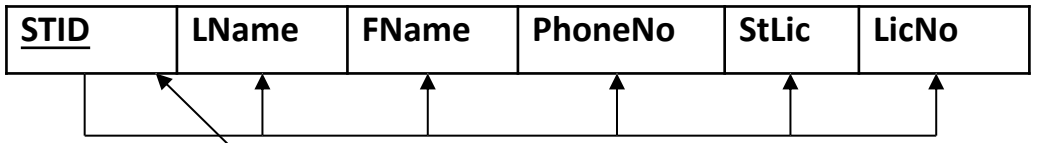
How do we normalize ?

Parking Tickets: Relations in 3NF

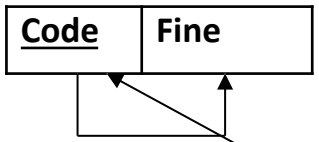
3. Assume that each student can have maximal one car



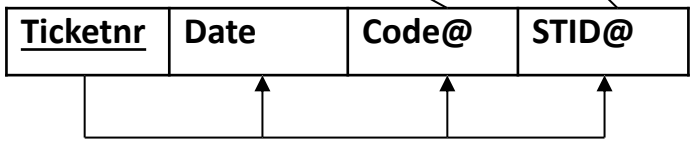
Student



TicketCode



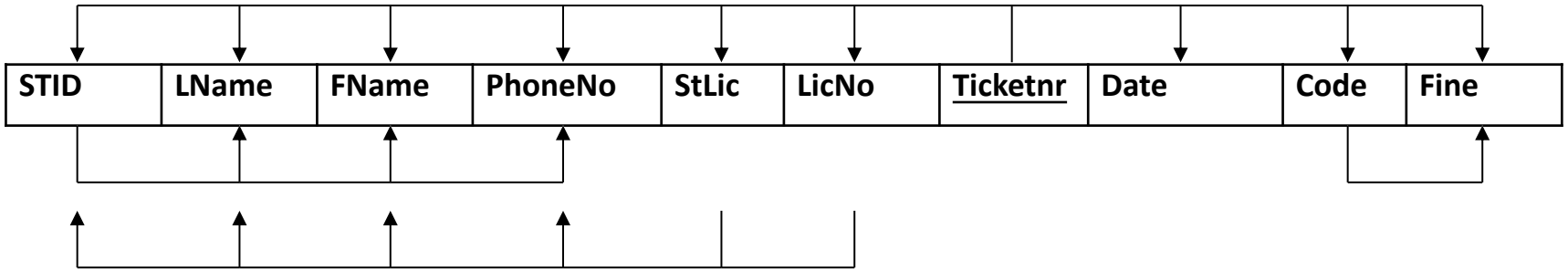
Ticket



No more partial nor transitive FDs!

Parking Tickets: Relations in 3NF

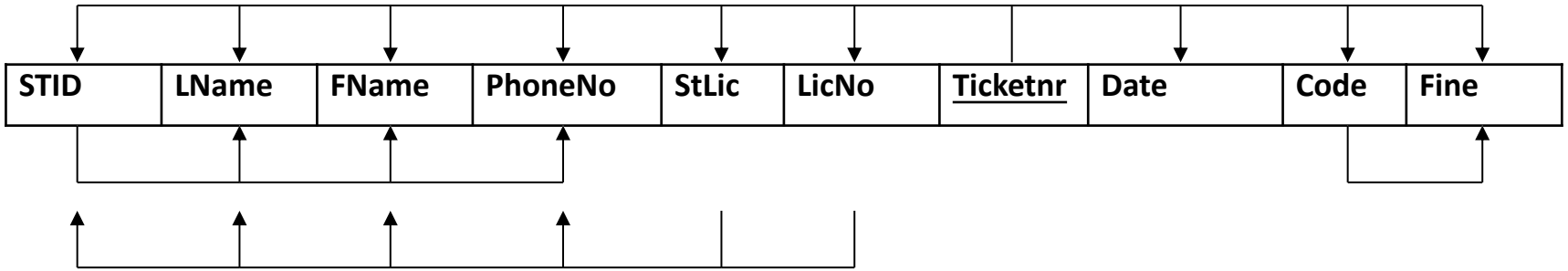
3'. Assume instead that each student can have more than one car



How do we normalize ?

Parking Tickets: Relations in 3NF

Next assume, students can have more than one car:



Student

STID	LName	FName	PhoneNo

Car

StLic	LicNo	STID@

TicketCode

Code	Fine

Ticket

Ticketnr	Date	Code@	StLic@	LicNo@