COMP9120

Week 7: Schema Refinement and Normalisation

Semester 2, 2016

(Kifer/Bernstein/Lewis – Chapter 6;

Ramakrishnan/Gehrke – Chapter 19;

Ullman/Widom – Chapter 3)

Based on material by Dr. Bryn Jeffries
And Dr. Uwe Röhm

THE UNIVERSITY OF SYDNEY

Copyright warning

Key words

You should understand the following terms/concepts

- Redundancy
  - Insert, update and delete anomalies
- Functional Dependencies
- Normal Forms
  - Candidate keys, super key, attribute closure, minimal key
  - 1NF, 2NF, 3NF, BCNF
- Schema Decomposition
  - Lossless join decompositions
  - Dependency preserving decompositions

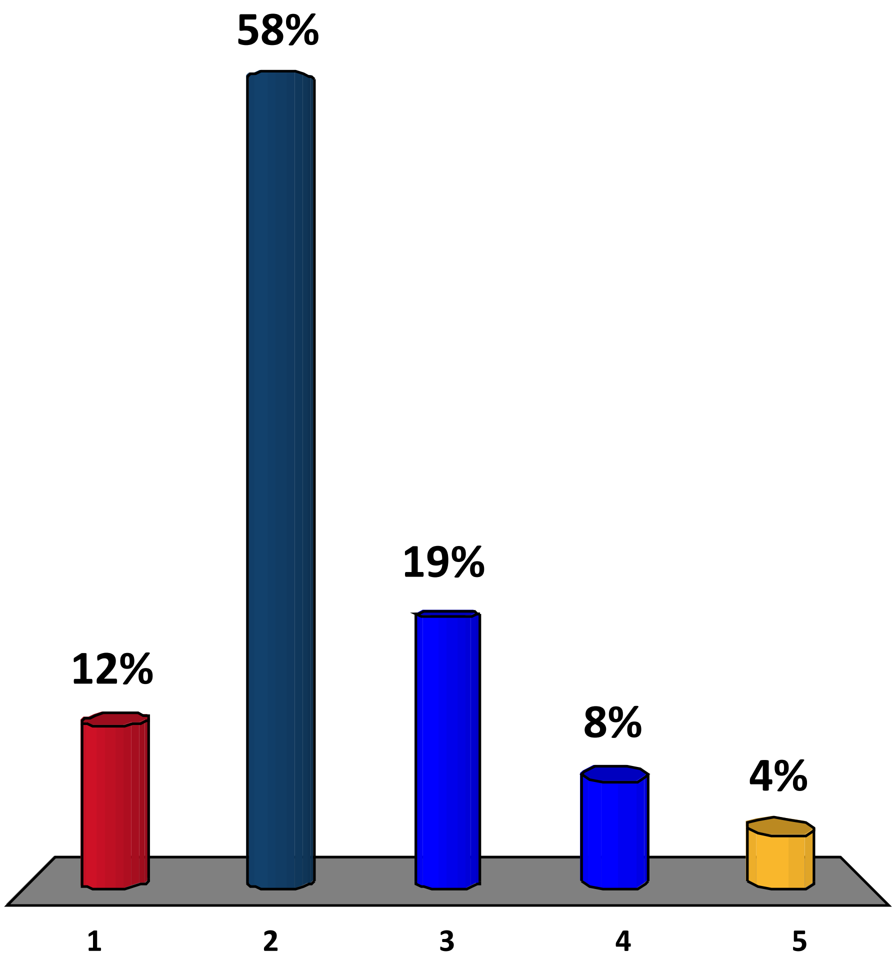
Functional Dependencies

- Integrity constraints, in particular functional dependencies, can be used to identify schemas sources of redundancy and to suggest refinements.
- **Functional Dependency**: The value of one attribute (the determinant) determines the value of another attribute
  - Intuitively: "If two tuples of a relation R agree on values in X, then they must also agree on the Y values."
- We write $X \longrightarrow Y$
  - "X (functionally) determines Y"
  - "Y is functionally dependent on X"

Which FD might be valid?

A	B	C	D
1	1	2	a
1	2	1	b
2	2	3	c
1	2	3	d
3	2	1	e
1	2	1	f

1. AB → C
2. AC → B
3. BC → A
4. A → BC
5. B → AC



Redundant Data and Functional Dependencies

SID	first	last	dept	advisor	award	description
1234	Homer	Simpson	IT	Codd	Rejected	Work not deemed sufficient
3456	Albert	Einstein	IT	Boyce	Conditional	Accepted with minor corrections
3456	Albert	Einstein	Physics	Newton	Accepted	Accepted with no corrections
7546	Alan	Turing	IT	Codd	Accepted	Accepted with no corrections
4879	Brian	Cox	Physics	Newton	Conditional	Accepted with minor corrections
4879	Brian	Cox	Media	Attenborough	Accepted	Accepted with no corrections

What functional dependencies might exist in the PhD relation above? What FDs can you identify?

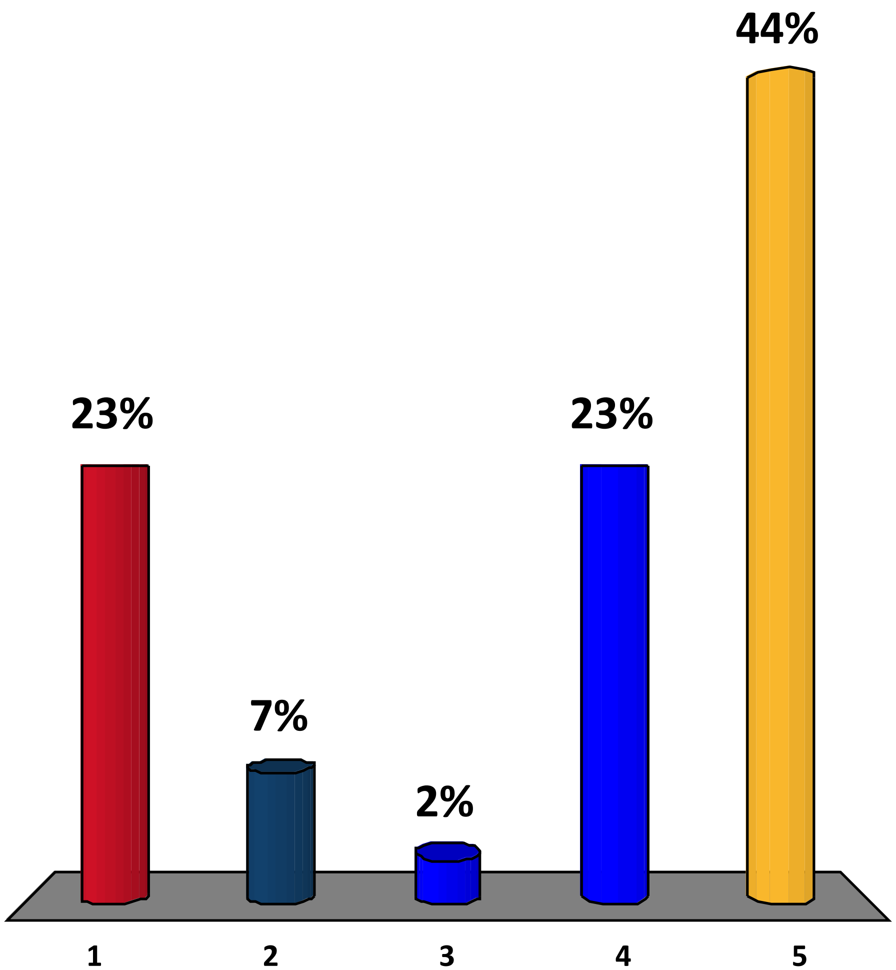
Anomalies

SID	first	last	dept	advisor	award	description
1234	Homer	Simpson	IT	Codd	Rejected	Work not deemed sufficient
3456	Albert	Einstein	IT	Boyce	Conditional	Accepted with minor corrections
3456	Albert	Einstein	Physics	Newton	Accepted	Accepted with no corrections
7546	Alan	Turing	IT	Codd	Accepted	Accepted with no corrections
4879	Brian	Cox	Physics	Newton	Conditional	Accepted with minor corrections
4879	Brian	Cox	Media	Attenborough	Accepted	Accepted with no corrections

- Update “Conditional” description to “Accepted with major corrections”
- Delete Brian Cox’s work in Media dept
- Insert new advisor “Hawking” for Physics dept

Which means “A passenger can book at most one seat on a plane”?

1. plane,seat → passenger
2. passenger,seat → plane
3. seat → passenger,plane
4. passenger → seat,plane
5. passenger,plane → seat



FD closure

- Given some FDs, we can usually infer additional FDs
- **Armstrong's Axioms** (*X, Y, Z* are sets of attributes):

1. Reflexivity rule: If $Y \subseteq X$, then $X \rightarrow Y$

- Example: *cpoints, uos_name* $\rightarrow$ *uos_name*

2. Augmentation rule: If $X \rightarrow Y$, then $XZ \rightarrow YZ$ for any Z

- Example: *cpoints, uos_name, wload* $\rightarrow$ *uos_name*, wload

3. Transitivity rule: If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$

- Example: *uos_code* $\rightarrow$ *cpoints*, *cpoints* $\rightarrow$ *wload* implies *uos_code* $\rightarrow$ *wload*

- **Trivial FDs**: When all RHS attributes appear on LHS
- All FDs logically implied by an initial set of given FDs F, is known as the **closure of F, labelled F^+**

FD Closure (continued)

- Armstrong's Axioms are
  - *Sound*: they generate only FDs in F^+ when applied to a set F of FDs
  - *Complete*: repeated application of these rules will generate all FDs in the closure F^+

- A couple of additional rules (that follow from Armstrong's Axioms.):
  - Union rule: If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$
  - Decomposition rule: If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$

Table Decomposition

Suppose that relation R contains attributes A_1 ... A_n.

- A decomposition of R consists of replacing R by two or more relations such that:
  - Each new relation scheme contains a subset of the attributes of R (and no attributes that do not appear in R), and
  - Every attribute of R appears as an attribute of at least one of the new relations.
  - All new relations differ.
- **Central Idea of Normalisation:** Decompose along a functional dependency.
  - For an FD *X -> Y, decompose R into R-Y and XY.*
  - Benefit that FD can be enforced in *XY* with a key constraint on *X*

Decomposition example

uosCode → uosName

Student	UoSCode	UoSName	Grade
Alice	COMP5138	Database Management Systems	CR
Alice	COMP5338	Advanced Data Models	D
Bob	COMP5138	Database Management Systems	P
Clare	COMP5338	Advanced Data Models	HD
David	COMP5338	Advanced Data Models	CR

$\bowtie$

Student	UoSCode	Grade
Alice	COMP5138	CR
Alice	COMP5338	D
Bob	COMP5138	P
Clare	COMP5338	HD
David	COMP5338	CR

UoSCode	UoSName
COMP5138	Database Management Systems
COMP5338	Advanced Data Models

Properties of Table Decomposition

- A decomposition of R into S and T is **lossless-join** w.r.t. a set of FDs F if, for every instance R that satisfies F:

$$\pi_S(R) \bowtie \pi_T(R) = R$$

  - i.e., the common attributes of S and T contain a key of either S or T

- A decomposition of R into S and T is **dependency preserving** if all FDs that were given to hold on R must hold on S and/or T.
  - All attributes of an FD must appear in a single relation

- Dependency preserving does not imply lossless join, or vice versa
- It is essential that all decompositions used to deal with redundancy be lossless

Lossless Join Property

uosCode → uosName

Student	UoSCode
Alice	COMP5138
Alice	COMP5338
Bob	COMP5138
Clare	COMP5338
David	COMP5338

UoSCode	UoSName	Grade
COMP5138	Database Management Systems	CR
COMP5138	Database Management Systems	P
COMP5338	Advanced Data Models	CR
COMP5338	Advanced Data Models	D
COMP5338	Advanced Data Models	HD

$\bowtie$

Student	UoSCode	UoSName	Grade
Alice	COMP5138	Database Management Systems	CR
Alice	COMP5138	Database Management Systems	P
Alice	COMP5338	Advanced Data Models	CR
Alice	COMP5338	Advanced Data Models	D
Alice	COMP5338	Advanced Data Models	HD
Bob	COMP5138	Database Management Systems	CR
Bob	COMP5138	Database Management Systems	P
Clare	COMP5338	Advanced Data Models	CR
Clare	COMP5338	Advanced Data Models	D
Clare	COMP5338	Advanced Data Models	HD
David	COMP5338	Advanced Data Models	CR
David	COMP5338	Advanced Data Models	D
David	COMP5338	Advanced Data Models	HD

Dependency Preserving Property

~~Title, City → Theatre~~

Theatre → City

Theatre	City	Title
Guild	Menlo Park	Antz
Park	Menlo Park	Antz

Theatre → City

Theatre	City
Guild	Menlo Park
Park	Menlo Park

$\bowtie$

Dependency not preserved: Title, City→ Theatre

Theatre	Title
Guild	Antz
Park	Antz

Keys and Functional Dependencies

- A **candidate key** (or just key) is the minimal set of columns that can be used to uniquely identify each tuple in a relation.
- There may be many candidate keys for a relation, but only 1 primary key.
- If you know the functional dependencies, then you can check whether a column (or set of columns) is a key for the relation

- A **superkey** is a column or set of columns that includes a candidate key
  - A superkey's **attribute closure** is the whole relation
  - A *candidate key*, plus perhaps extra columns

- A candidate key is a superkey that is **minimal**
  - No subset can also be a superkey

Attribute Closure

Starting with the given set of attributes, one repeatedly expands the set by adding the right side of an FD as soon as the left side is present:

1. Initialise *result with the given set of attributes:*
   $X=\{A_1, \ldots, A_n\}$
2. Repeatedly search for some FD $A_1\ A_2 \ldots A_m \rightarrow C$ such that all $A_1, \ldots, A_m$ *are already in the set of attributes result, but C is not.*
3. Add *C to the set result.*
4. Repeat step 2 until no more attributes can be added to *result*
5. The set *result is the correct value of* X^+

Example: Test for Candidate keys

$R = (A, B, C, G, H, I)$

$F = \{$

$A \rightarrow B$

$A \rightarrow C$

$CG \rightarrow H$

$CG \rightarrow I$

$B \rightarrow H$

$\}$

- Check $(AG)^+$
  1. $result = AG$
  2. $result = ABG\ (A \rightarrow B)$
  3. $result = ABCG\ (A \rightarrow C)$
  4. $result = ABCGH\ (CG \rightarrow H)$
  5. $result = ABCGHI\ (CG \rightarrow I)$
- Is (*AG*) *a candidate key?*
  1. Is (AG) a super key? YES!: $(AG)^+ = R$
  2. Is (AG) minimal? (Is any subset of (AG) a superkey?)

  - $(A)^+ = ABCH \neq R$
  - $(G)^+ = G \neq R$

THE UNIVERSITY OF SYDNEY

Exercise 2: Candidate Keys

What are the keys of our relation?

PhD(SID, first, last, dept, advisor, award, description)

Given FDs

a) sid → first, last

b) advisor → dept

c) description → award

d) sid,dept → advisor,description

1. (sid, dept)
2. (sid, advisor)

Normal Forms

- Database theory identifies several normal forms for relational schemas.
- Each normal form is characterized by a set of restrictions.
- For a relation to be in a normal form it must satisfy the restrictions associated with that form.
- A relation *R* is in **first normal form (1NF)** if the domains of all attributes of *R* are *atomic*.
- Domain is atomic if its elements are considered to be indivisible units
  - Examples of non-atomic domains:
    - multivalued attributes, composite attributes
  - Non-atomic values complicate storage and encourage redundant (repeated) storage of data

Second Normal Form

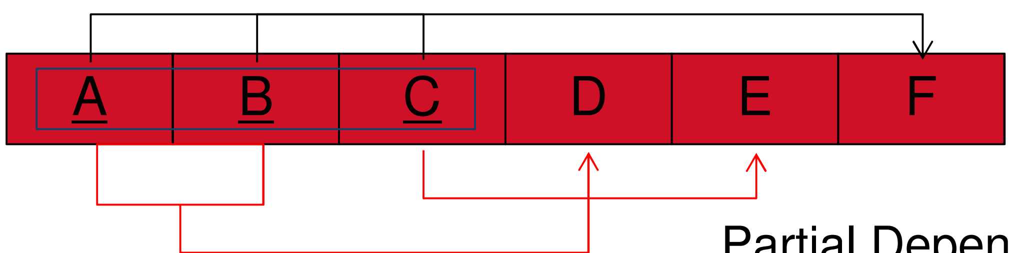
Second Normal Form (2NF)

- 1NF + prohibition of non-trivial dependencies $X \longrightarrow Y$ for a relation R where X is a strict (proper) subset of a key in R, and Y are non-key attributes
- This means: **No partial dependencies**

  (no non-trivial FD $X \longrightarrow Y$ for R where X is a strict (proper) subset of some key for R and Y are non-key attributes)

Full Dependency: ABC → F

Key: ABC



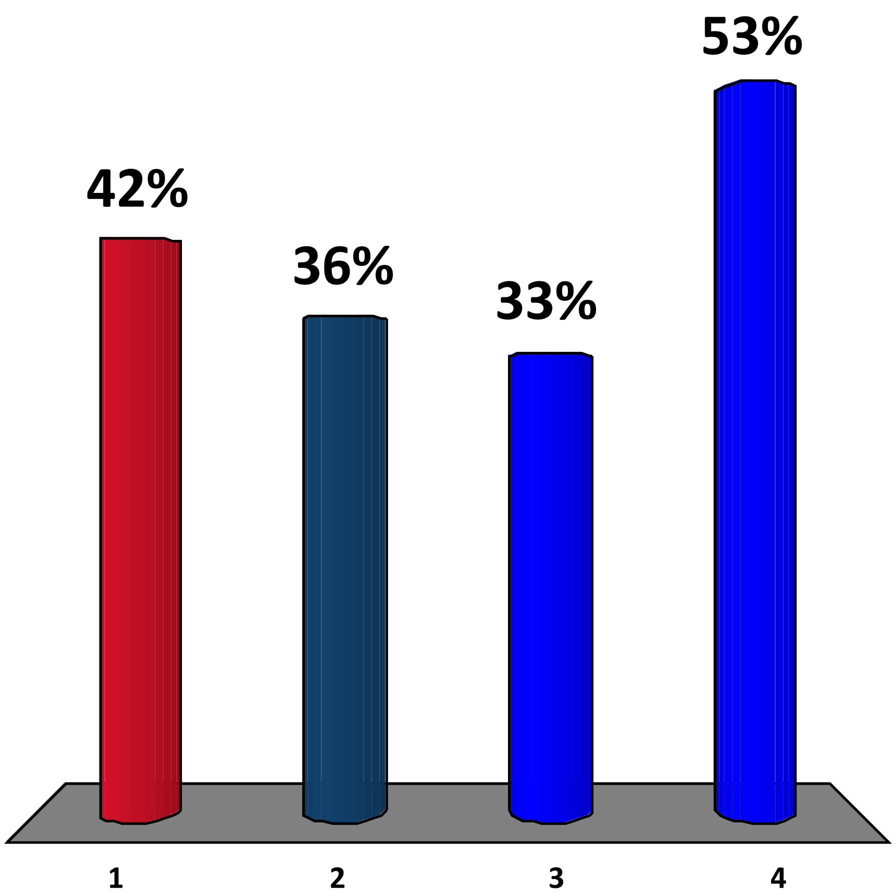
Partial Dependency: C → E

Partial Dependency: AB → D

Which are the partial dependencies?

Invoice(<u>orderId</u>, date, custD, name, addr, <u>ProdID</u>, desc, price, qty)

1. orderID → date, custID, name, addr
2. orderID,prodID → qty
3. custID → name, addr
4. prodID → desc, price



THE UNIVERSITY OF SYDNEY

Exercise 3: Second Normal Form

SID	first	last	dept	advisor	award	description
1234	Homer	Simpson	IT	Codd	Rejected	Work not deemed sufficient
3456	Albert	Einstein	IT	Boyce	Conditional	Accepted with minor corrections
3456	Albert	Einstein	Physics	Newton	Accepted	Accepted with no corrections
7546	Alan	Turing	IT	Codd	Accepted	Accepted with no corrections
4879	Brian	Cox	Physics	Newton	Conditional	Accepted with minor corrections
4879	Brian	Cox	Media	Attenborough	Accepted	Accepted with no corrections

Decompose the relation along FD (sid → first, last) to get it into 2NF

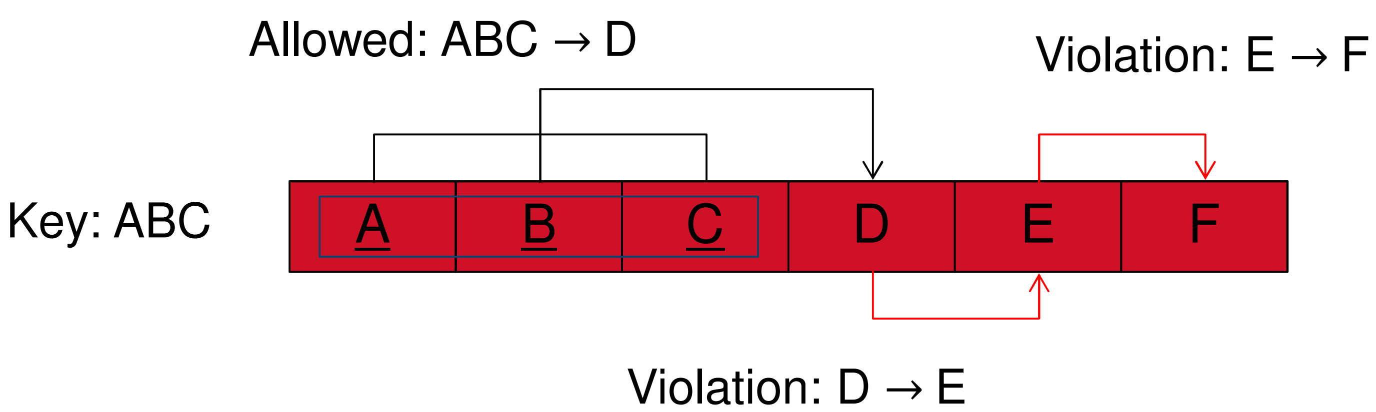
SID	dept	advisor	award	description
1234	IT	Codd	Rejected	Work not deemed sufficient
3456	IT	Boyce	Conditional	Accepted with minor corrections
3456	Physics	Newton	Accepted	Accepted with no corrections
7546	IT	Codd	Accepted	Accepted with no corrections
4879	Physics	Newton	Conditional	Accepted with minor corrections
4879	Media	Attenborough	Accepted	Accepted with no corrections

<u>SID</u>	first	last
1234	Homer	Simpson
3456	Albert	Einstein
7546	Alan	Turing
4879	Brian	Cox

Third Normal Form

› Third Normal Form (3NF)

- For all non-trivial $X \rightarrow Y$ for R:
  - (X is either a superkey of R) OR (Y is part of a key in R)



Which of the following is a violation of 3NF?

Enrol(uosCode, title, credits, hours, lecturer, sid, grade)

1. uosCode → title, credits, lecturer
2. credits → hours
3. uosCode, sid → grade

Exercise 4: Third Normal Form

SID	dept	advisor	award	description
1234	IT	Codd	Rejected	Work not deemed sufficient
3456	IT	Boyce	Conditional	Accepted with minor corrections
3456	Physics	Newton	Accepted	Accepted with no corrections
7546	IT	Codd	Accepted	Accepted with no corrections
4879	Physics	Newton	Conditional	Accepted with minor corrections
4879	Media	Attenborough	Accepted	Accepted with no corrections

Decompose the relation along FD (description → award) to get it into 3NF

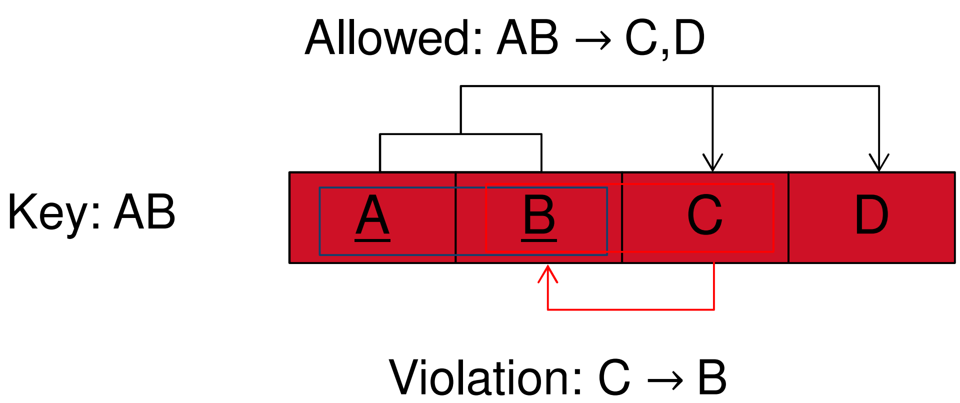
SID	dept	advisor	description
1234	IT	Codd	Work not deemed sufficient
3456	IT	Boyce	Accepted with minor corrections
3456	Physics	Newton	Accepted with no corrections
7546	IT	Codd	Accepted with no corrections
4879	Physics	Newton	Accepted with minor corrections
4879	Media	Attenborough	Accepted with no corrections

award	<u>description</u>
Rejected	Work not deemed sufficient
Conditional	Accepted with minor corrections
Accepted	Accepted with no corrections

Boyce-Codd Normal Form

- A relation *R* is in **BCNF** if the only non-trivial FDs that hold over *R* have a superkey of R on the LHS.
  - Formal:

For all *non-trivial* $X \rightarrow Y$ *for R*: *X is a superkey for R*



Exercise 5: Boyce-Codd Normal Form

SID	dept	advisor	description
1234	IT	Codd	Work not deemed sufficient
3456	IT	Boyce	Accepted with minor corrections
3456	Physics	Newton	Accepted with no corrections
7546	IT	Codd	Accepted with no corrections
4879	Physics	Newton	Accepted with minor corrections
4879	Media	Attenborough	Accepted with no corrections

Decompose the relation to get it into BCNF. What compromise did you have to make?

SID	advisor	description
1234	Codd	Work not deemed sufficient
3456	Boyce	Accepted with minor corrections
3456	Newton	Accepted with no corrections
7546	Codd	Accepted with no corrections
4879	Newton	Accepted with minor corrections
4879	Attenborough	Accepted with no corrections

1234	Boyce	Accepted with minor corrections

dept	advisor
IT	Codd
IT	Boyce
Physics	Newton
Media	Attenborough

This would not be allowed by FD sid,dept → advisor,description

Central Theorem of Schema Refinement

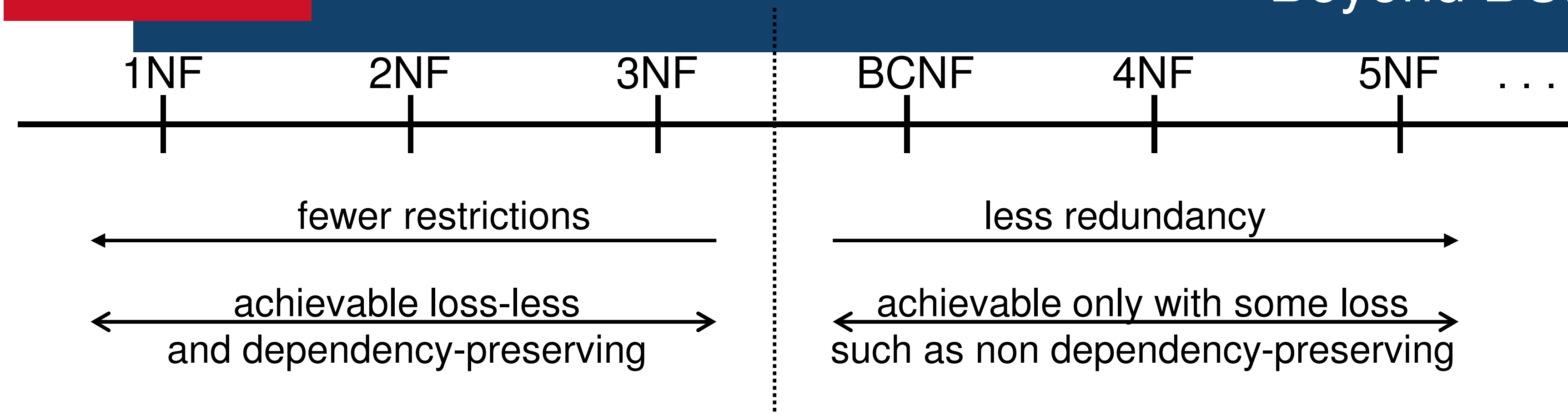
- Theorem:
  A relation R with schema ***R*** and a set of FDs *F*. Let

  X $\longrightarrow$ Y a functional dependency with X $\cap$ Y = $\varnothing$. Then is the decomposition of R into XY and ***R*** -Y a *lossless-join decomposition*.

- Every relation R with functional dependencies F can be decomposed into **3NF** relations, through a decomposition that is both *lossless* and *dependency-preserving*.

- For every relation R with set of FDs *F* exists a *lossless-join* decomposition into **BCNF** relations.

Beyond BCNF



- Other types of dependencies exist
  - Multi-valued dependencies
  - Join dependencies
  - Inclusion dependencies
- And higher normal forms
  - 4NF, 5NF, 6NF/DKNF
- May also *choose* to have more redundancy for performance reasons: **denormalisation**

Key Skills

You should be able to perform the following tasks

- Identify and interpret functional dependencies for a database schema
- Identify the candidate keys and normal form (up to BCNF) of a relation schema using its functional dependencies
  - Identify whether a set of attributes forms is a minimal superkey
  - Determine all possible candidate keys
- Decompose a relational instance into a set of 3NF or BCNF relational instances
  - Determine a dependency-preserving, lossless join decomposition of a relational schema
  - Correctly determine the decomposed relation instances

Next Lecture

- Database Application Development
  - Embedded SQL in Client Code
  - Call-level Database APIs
  - Server-Side Application Development with Stored Procedures

- Readings:
  - Kifer/Bernstein/Lewis book, Chapter 8
  - Ramakrishnan/Gehrke (Cow book), Chapter 6
  - Ullman/Widom, Chapter 9