

CZ2007 Introduction to Database Systems (Week 3)

Topic 3: Boyce-Codd Normal Form (1)



Copyright Notice

All course materials (including and not limited to lecture slides, handouts, recordings, assessments and assignments), are solely for your own educational purposes at NTU only. All course materials are protected by copyright, trademarks or other rights.

All rights, title and interest in the course materials are owned by, licensed to or controlled by the University, unless otherwise expressly stated. The course materials shall not be uploaded, reproduced, distributed, republished or transmitted in any form or by any means, in whole or in part, without written approval from the University.

You are also not allowed to take any photograph, video recording, audio recording or other means of capturing images and/or voice of any of the course materials (including and not limited to lectures, tutorials, seminars and workshops) and reproduce, distribute and/or transmit in any form or by any means, in whole or in part, without written permission from the University.

Appropriate action(s) will be taken against you (including and not limited to disciplinary proceedings and/or legal action) if you are found to have committed any of the above or infringed copyright.

This Lecture

- Boyce-Codd Normal Form 
- BCNF Decomposition

Normal Forms

- Various normal forms (in increasing order of strictness)
 - First normal form
 - Second normal form
 - Third normal form (3NF)
 - Boyce-Codd normal form (BCNF)
 - Fourth normal form
 - Fifth normal form
 - Sixth normal form
- 3NF and BCNF are most commonly used

Boyce-Codd Normal Form (BCNF)

- A table R is in BCNF, if and only if
 - The left hand side of **every** non-trivial FD contains a key of R
- Non-trivial FD:
 - An FD that is not implied by the axiom of reflexivity
 - Example:
 - $A \rightarrow B$ -- Non-trivial
 - $AC \rightarrow BC$ -- Non-trivial
 - $AC \rightarrow A$ -- Trivial
 - $AC \rightarrow C$ -- Trivial

Boyce-Codd Normal Form (BCNF)

- A table R is in BCNF, if and only if
 - The left hand side of **every** non-trivial FD contains a key of R
- R(A, B)
- Given FDs: $A \rightarrow B$
- Key: **A**
- All FDs on R: $A \rightarrow B$, $AB \rightarrow A$, $AB \rightarrow B$, $AB \rightarrow AB$
 - $AB \rightarrow A$, $AB \rightarrow B$, $AB \rightarrow AB$: trivial
 - $A \rightarrow B$: The left hand side contains a key
- Therefore, R is in BCNF

Boyce-Codd Normal Form (BCNF)

- A table R is in BCNF, if and only if
 - The left hand side of **every** non-trivial FD contains a key of R
- R(A, B, C)
- Given FDs: $A \rightarrow B$
- Key: AC
- All FDs on R: $A \rightarrow B$, $AB \rightarrow B$, $AC \rightarrow C$, ...
 - The left hand side of $A \rightarrow B$ does not contain a key
- Therefore, R is NOT in BCNF

BCNF: Example

Name	{ <u>NRIC</u>	<u>Phone</u> }	Address
Alice	1234	67899876	Jurong East
Alice	1234	83848384	Jurong East
Bob	5678	98765432	Pasir Ris

- NRIC → Name, Address
- Key: {NRIC, Phone} (key is a composite of two attributes)
- Not in BCNF

BCNF: Straightforward Checking

- Given: A table R , a set of FDs on R
- Step 1: Derive the keys of R
- Step 2: Derive all non-trivial FDs on R
- Step 3: For each non-trivial FD, check if its left hand side contains a key
- Step 4: If all FDs pass the check, then R is in BCNF; otherwise, R is not in BCNF

BCNF: Straightforward Checking

- Given: A table R, A set of FDs on R
- Step 1: Derive the keys of R
- Step 2: Derive all non-trivial FDs on R
 - This is too time-consuming
 - Trick: Only check the FDs **given** on R instead of all FDs
- Step 3: For each non-trivial FD, check if its left hand side contains a key
- Step 4: If all FDs pass the check, then R is in BCNF; otherwise, R is not in BCNF

BCNF Checking: Example

- $R(A, B, C, D)$
- Given: $A \rightarrow B, A \rightarrow C, A \rightarrow D$
- Keys: A
- Check: For each **given** non-trivial FD, check if its left hand side contains a key
- $A \rightarrow B$: Non-trivial, LHS contains a key
- $A \rightarrow C$: Non-trivial, LHS contains a key
- $A \rightarrow D$: Non-trivial, LHS contains a key
- Therefore, R is in BCNF

BCNF Checking: Example

- $R(A, B, C, D)$
- Given: $A \rightarrow B, B \rightarrow C, C \rightarrow D$
- Keys: A
- Check: For each **given** non-trivial FD, check if its left hand side contains a key
- $A \rightarrow B$: Non-trivial, LHS contains a key
- $B \rightarrow C$: Non-trivial, LHS does NOT contain a key
- Therefore, R is NOT in BCNF

BCNF Checking: Rationale

- A table R is in BCNF, if and only if for **every** FD on R,
 - Either the FD is trivial
 - Or the left hand side of the FD contains a key
- The definition says “**every**”, but we only check the ones “**given**” on R
- This leaves the “**hidden**” ones unchecked
- Why does it work?
- Rationale: If the “**given**” FDs pass the check, then all “**hidden**” ones will pass the check because all derived FDs will come from the given FDs.

BCNF Checking: Multiple-Key Cases

- $R(A, B, C, D)$
- Given: $A \rightarrow B$, $B \rightarrow A$, $B \rightarrow C$, $B \rightarrow D$, $A \rightarrow D$
- Keys: A, B (i.e., R has two keys)
- Check: For each given non-trivial FD, check if its left hand side contains a key
- $A \rightarrow B$, $A \rightarrow D$: Non-trivial, LHS contains a key
- $B \rightarrow A$, $B \rightarrow C$, $B \rightarrow D$: Non-trivial, LHS contains a key
- Therefore, R is in BCNF

BCNF: Intuition

- Basically, BCNF requires that there forbids any non-trivial $X \rightarrow Y$ where X does not contain a key
- Why does this make sense?
- Intuition: Such $X \rightarrow Y$ indicates that the table has some redundancies

BCNF Intuition: Example

Name	{ <u>NRIC</u> }	<u>Phone</u> }	Address
Alice	1234	67899876	Jurong East
Alice	1234	83848384	Jurong East
Bob	5678	98765432	Pasir Ris

- $\text{NRIC} \rightarrow \text{Name, Address}$
- Key: {NRIC, Phone} (single composite key)
- $\text{NRIC} \text{ decides Name and Address}$
- Therefore, every time NRIC is repeated, Name and Address would also be repeated
- Since NRIC is not a key, the same NRIC can appear multiple times in the table
- This leads to redundancy
- BCNF prevents this

This Lecture

- Boyce-Codd Normal Form
- BCNF Decomposition ←

BCNF Decomposition

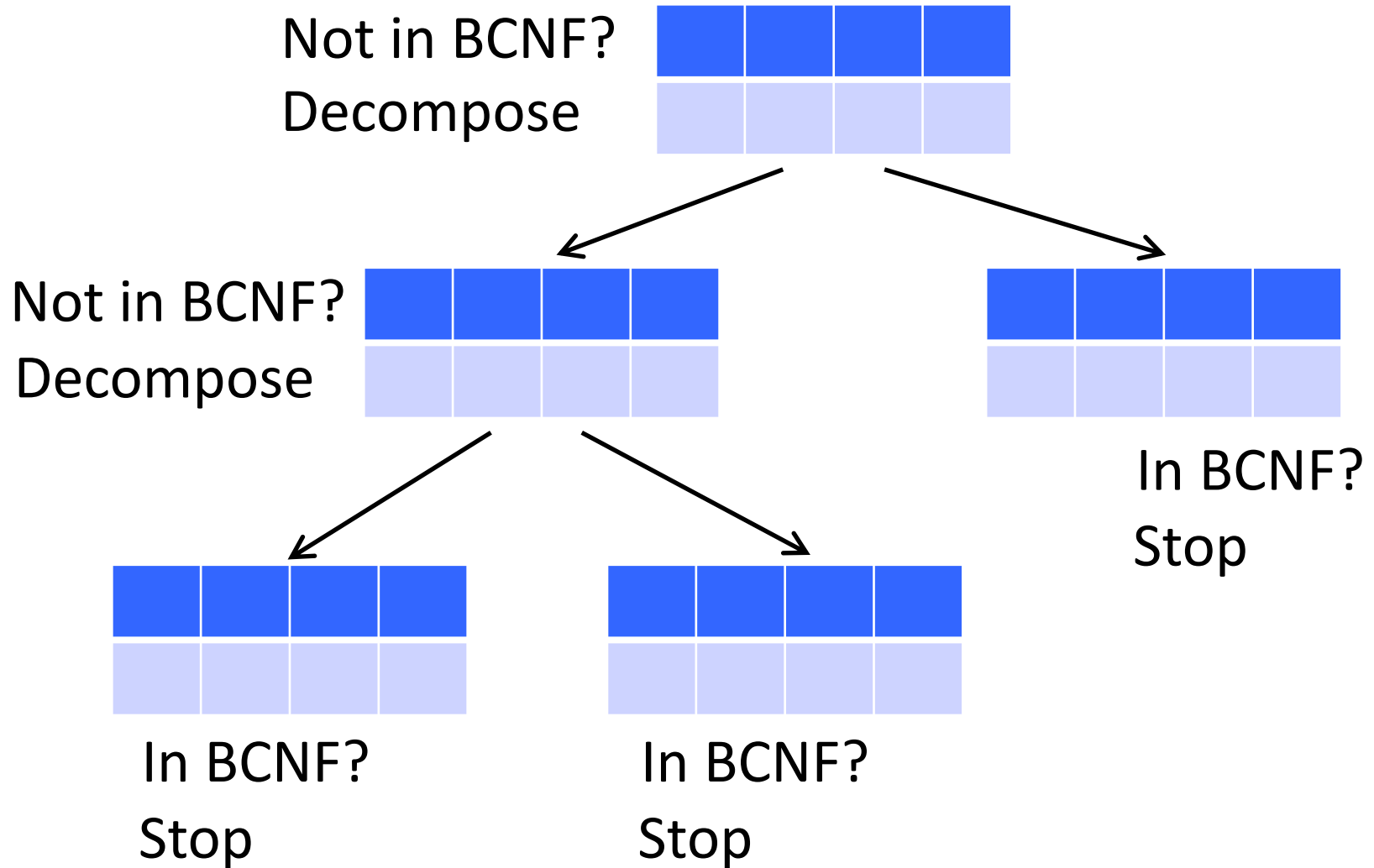
Name	{ <u>NRIC</u> }	<u>Phone</u> }	Address
Alice	1234	67899876	Jurong East
Alice	1234	83848384	Jurong East
Bob	5678	98765432	Pasir Ris

- What can we do if a table violates BCNF?
- Answer: Decompose it (i.e., normalize it)

Name	<u>NRIC</u>	Address
Alice	1234	Jurong East
Bob	5678	Pasir Ris

{ <u>NRIC</u> }	<u>Phone</u> }
1234	67899876
1234	83848384
5678	98765432

Decompose, until all are in BCNF



BCNF Decomposition: Example

- $R(A, B, C, D)$
- Given: $A \rightarrow B, B \rightarrow C, C \rightarrow D$
- Key of R: A
- Step 1: Identify a FD that violates BCNF
- $B \rightarrow C$ is a violation, since
 - It is non-trivial
 - Its left hand side does not contain a key

BCNF Decomposition: Example

- $R(A, B, C, D)$
- Given: $A \rightarrow B, B \rightarrow C, C \rightarrow D$
- Key of R : A
- Step 1: $B \rightarrow C$ is a violation
- Step 2: Compute the closure of the left hand side of the FD
- $\{B\}^+ = \{BCD\}$

BCNF Decomposition: Example

- $R(A, B, C, D)$
- Given: $A \rightarrow B, B \rightarrow C, C \rightarrow D$
- Key of R : A
- Step 1: $B \rightarrow C$ is a violation
- Step 2: $\{B\}^+ = \{BCD\}$
- Step 3: Decompose R into two tables R_1 and R_2
 - $R_1(B, C, D)$ contains all attributes in the closure
 - $R_2(A, B)$ contains all attributes NOT in the closure plus B , LHS of violating FD
- Step 4: Check R_1 and R_2 , decompose if necessary

BCNF Decomposition Algorithm

- Input: A table R BCNF
- Step 1: Find a FD $X \rightarrow Y$ on R that violates BCNF
 - If cannot find, stop, R in BCNF
- Step 2: Compute the closure $\{X\}^+$
- Step 3: Break R into two tables R_1 and R_2
 - R_1 contains all attributes in $\{X\}^+$
 - R_2 contains all attributes NOT in $\{X\}^+$ plus X
- Repeat Steps 1-3 on R_1 and R_2

BCNF Decomposition: Example

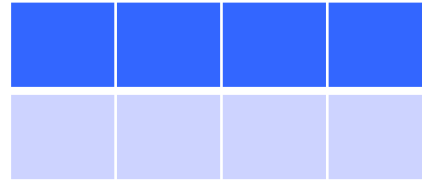
- $R(A, B, C, D)$
- Given: $A \rightarrow B, B \rightarrow C, C \rightarrow D$
- Key of R : A
- Previous results: $R_1(B, C, D), R_2(A, B)$
- Is R_2 in BCNF?
 - Yes. So R_2 is done
- Is R_1 in BCNF?
 - No. Key of R_1 is B and $C \rightarrow D$ is a violation.
- Decompose R_1 into R_3 and R_4
 - $\{C\}^+ = \{CD\}$
 - $R_3(C, D)$ contains all attributes in $\{C\}^+$
 - $R_4(B, C)$ contains C and all attribute NOT in $\{C\}^+$
- Are R_3 and R_4 in BCNF?
 - Yes. So we stop here.

BCNF Decomposition: Example

- $R(A, B, C, D)$
- Given: $A \rightarrow B, B \rightarrow C, C \rightarrow D$
- Key of R : A
- Final BCNF Decomposition
 - $R_2(A, B)$
 - $R_3(C, D)$
 - $R_4(B, C)$

Decompose, until all are in BCNF

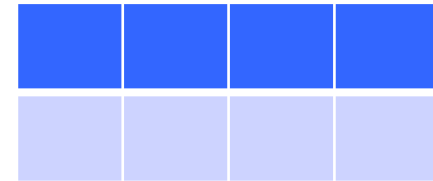
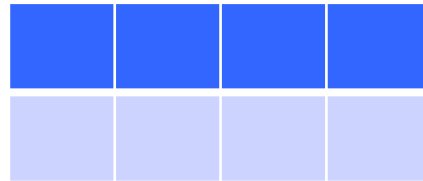
$X \rightarrow Y$ violation?
Decompose



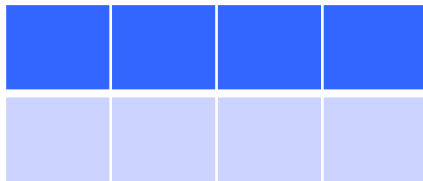
Attributes in $\{X\}^+$

X + Attributes
NOT in $\{X\}^+$

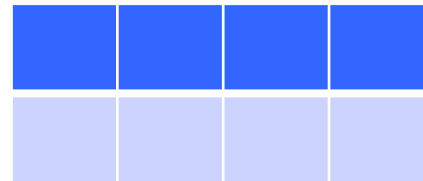
$X' \rightarrow Y'$ violation?
Decompose



Attributes in $\{X'\}^+$



In BCNF then
Stop



In BCNF then
Stop

Notes

- The BCNF decomposition of a table may not be unique
- If a table has only two attributes, then it must be in BCNF
 - Therefore, you do not need to check tables with only two attributes

Exercise: BCNF Decomposition

- $R(A, B, C, D, E)$
- Given: $AB \rightarrow C, C \rightarrow D, D \rightarrow E$
- Key of R: AB (one single composite key)
- $D \rightarrow E$ is a violation
- Decompose $R(A, B, C, D, E)$
 - $\{D\}^+ = \{D, E\}$
 - $R_1(D, E), R_2(A, B, C, D)$
 - R_1 has only two attributes. Must be in BCNF.
 - $R_2 \dots$ Key of R_2 : AB. Therefore, $C \rightarrow D$ is a violation
- Decompose $R_2(A, B, C, D)$
 - $\{C\}^+ = \{C, D, E\}$. E is omitted since it is not in R_2
 - $R_3(C, D), R_4(A, B, C)$
 - R_3 has only two attributes. Must be in BCNF.
 - $R_4 \dots$ Key of R_4 : AB. No violation of BCNF. Done

To continue in

Topic 3: Boyce-Codd Normal Form (2)