CS2102
Database Systems

Last Lecture

R

A	B	C

A→B, B→C

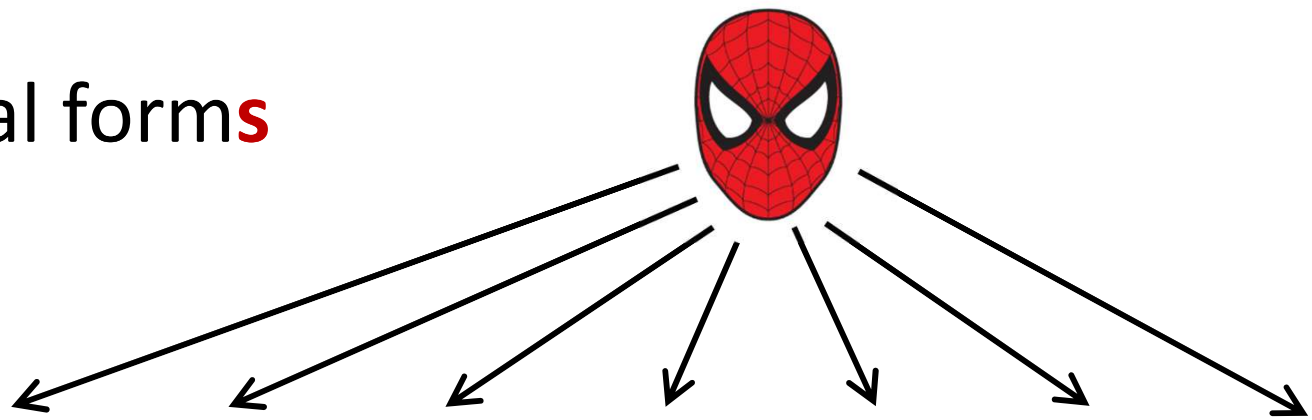
- Functional dependencies (FD)
  - Example above: A→B, B→C
- Superkeys of a table R
  - A set of attribute that can decide all other attributes in R
  - Example above: A, AB, AC, ABC are all superkeys of R
- Keys
  - A superkey that is minimal
  - Example above: A is the only key of R
- Finding superkeys/keys from R
  - Using closures, based on the given FDs

Coming Next

- Normal forms

Coming Next

- Normal form**s**



1NF 2NF 3NF BCNF 4NF 5NF …

…

Normal Forms

- Conditions that a "good" table should satisfy
- Various normal forms
  (in increasing order of strictness)
  - 1st NF
  - 2nd NF

  → Easy to satisfy
  May have high redundancy

  - 3rd NF (3NF)
  - Boyce-Codd NF (BCNF)
  - 4th NF
  - 5th NF
  - 6th NF

  → Very little redundancy
  Not always possible to satisfy

Normal Forms

- Conditions that a "good" table should satisfy
- Various normal forms
  (in increasing order of strictness)
  - 1st NF
  - 2nd NF
  - 3rd NF (3NF)
  - Boyce-Codd NF (BCNF)
  - 4th NF
  - 5th NF
  - 6th NF

Get rid of most redundancies
Always possible to satisfy

Roadmap

- We will focus on 3NF and BCNF since they are the most commonly used NF
- We will start from BCNF since it is conceptually simpler

Non-trivial and Decomposed FD

- To simplify our discussions of BCNF and 3NF, we will focus on non-trivial and decomposed FDs
- Decomposed FD: an FD whose right hand side has only one attribute
  - E.g., $A \rightarrow C$, $BC \rightarrow D$, $DEF \rightarrow E$
- Note: a non-decomposed FD can always be transformed into an equivalent set of decomposed FDs
  - E.g., $BC \rightarrow DE$ <==> $BC \rightarrow D$ and $BC \rightarrow E$

Non-trivial and Decomposed FD

- To simplify our discussions of BCNF and 3NF, we will focus on non-trivial and decomposed FDs
- Non-trivial and decomposed FD: a decomposed FD whose right hand side does not appear in the left hand side
  - E.g., A→C, BC→D
- We will check normal forms based on the non-trivial and decomposed FDs on a table
- How do we derive such FDs?

Non-trivial and Decomposed FD

- Non-trivial and decomposed FDs can be found using closures, in a way similar to finding keys
- Example: R(A, B, C), with A→B, B→A, B→C given
- Step 1: Consider all attribute subsets in R
  - {A} {B} {C}
  - {AB} {AC} {BC}

Non-trivial and Decomposed FD

- Non-trivial and decomposed FDs can be found using closures, in a way similar to finding keys
- Example: R(A, B, C), with A→B, B→A, B→C given
- Step 2: Compute the closure of each subset
  - {A} {B} {C}
  - {AB} {AC} {BC}

Non-trivial and Decomposed FD

- Non-trivial and decomposed FDs can be found using closures, in a way similar to finding keys
- Example: R(A, B, C), with $A \rightarrow B$, $B \rightarrow A$, $B \rightarrow C$ given
- Step 2: Compute the closure of each subset

  - $\{A\}^+ =$ $\{B\}^+ =$ $\{C\}^+ =$
  - $\{AB\}^+ =$ $\{AC\}^+ =$ $\{BC\}^+ =$

Non-trivial and Decomposed FD

- Non-trivial and decomposed FDs can be found using closures, in a way similar to finding keys
- Example: R(A, B, C), with $A \rightarrow B$, $B \rightarrow A$, $B \rightarrow C$ given
- Step 2: Compute the closure of each subset

  - $\{A\}^+ = \{ABC\}$, $\{B\}^+ = \{ABC\}$, $\{C\}^+ = \{C\}$
  - $\{AB\}^+ = \{ABC\}$, $\{AC\}^+ = \{ABC\}$, $\{BC\}^+ = \{ABC\}$

Non-trivial and Decomposed FD

- Non-trivial and decomposed FDs can be found using closures, in a way similar to finding keys
- Example: R(A, B, C), with $A \rightarrow B$, $B \rightarrow A$, $B \rightarrow C$ given
- Step 3: From each closure, remove the "trivial" attributes
  - $\{A\}^+ = \{ABC\}$, $\{B\}^+ = \{ABC\}$, $\{C\}^+ = \{C\}$
  - $\{AB\}^+ = \{ABC\}$, $\{AC\}^+ = \{ABC\}$, $\{BC\}^+ = \{ABC\}$

Non-trivial and Decomposed FD

- Non-trivial and decomposed FDs can be found using closures, in a way similar to finding keys
- Example: R(A, B, C), with A→B, B→A, B→C given
- Step 3: From each closure, remove the "trivial" attributes
  - $\{A\}^+ = \{ABC\}$, $\{B\}^+ = \{ABC\}$, $\{C\}^+ = \{C\}$
  - $\{AB\}^+ = \{ABC\}$, $\{AC\}^+ = \{ABC\}$, $\{BC\}^+ = \{ABC\}$

Non-trivial and Decomposed FD

- Non-trivial and decomposed FDs can be found using closures, in a way similar to finding keys
- Example: R(A, B, C), with A→B, B→A, B→C given
- Step 4: Derive non-trivial and decomposed FDs from each closure
  - $\{A\}^+ = \{ABC\}$, $\{B\}^+ = \{ABC\}$, $\{C\}^+ = \{C\}$
  - $\{AB\}^+ = \{ABC\}$, $\{AC\}^+ = \{ABC\}$, $\{BC\}^+ = \{ABC\}$

Non-trivial and Decomposed FD

- Non-trivial and decomposed FDs can be found using closures, in a way similar to finding keys
- Example: R(A, B, C), with A→B, B→A, B→C given
- Step 4: Derive non-trivial and decomposed FDs from each closure
  - A→B, A→C, B→A, B→C
  - AB→C, AC→B, BC→A

BCNF: Definition

- A table R is in BCNF, if every non-trivial and decomposed FD has a superkey on its left hand side
- Example: R(A, B, C), with $A \rightarrow B$, $B \rightarrow A$, $B \rightarrow C$ given
- Non-trivial and decomposed FDs on R:
  - $A \rightarrow B$, $A \rightarrow C$, $B \rightarrow A$, $B \rightarrow C$
  - $AB \rightarrow C$, $AC \rightarrow B$, $BC \rightarrow A$
- Keys: A, B
- For each of the above FD, the left hand side is a superkey
- So R satisfies BCNF

BCNF: Definition

- A table R is in BCNF, if every non-trivial and decomposed FD has a superkey on its left hand side
- Example: R(A, B, C), with A→B, B→C given
- Key: A
- Observe that
  - B->C is a non-trivial and decomposed FD
  - The left hand side of B→C is not a superkey
- So R does not satisfy BCNF

BCNF: Intuition

- BCNF requires that if there is any non-trivial and decomposed FD $A_1A_2 \ldots A_n \rightarrow B$, then $A_1A_2 \ldots A_n$ must be a superkey
- In other words, all attributes B can depend only on superkeys
- Any dependency on non-superkeys is prohibited by BCNF

BCNF: Intuition

- In other words, any attribute B can depend only on superkeys
- Why does this make sense?
- Suppose that B depends on a non-superkey $C_1C_2 \ldots C_n$
- Since $C_1C_2 \ldots C_n$ is not a superkey, the same $C_1C_2 \ldots C_n$ may appear multiple times in the table
- Whenever this happens, the same B would appear multiple times in the table
- This leads to redundancy
- BCNF prevents this from happening

BCNF: Intuition

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

- Key: {NRIC, Phone}
- We have NRIC → Name, which violates BCNF
- Since NRIC is not a superkey, the same NRIC can appear multiple times in the table
- Every time the same NRIC is repeated, the corresponding Name and Address are also be repeated
- This leads to redundancy
- BCNF prevents this

Coming Next

- How do we check whether a table is in BCNF?

BCNF Check

- A table R is in BCNF, if every non-trivial and decomposed FD has a superkey on its left hand side
- Algorithm for checking BCNF
  - Compute the closure of each attribute subset
  - Derive the keys of R (using closures)
  - Derive all non-trivial and decomposed FDs on R (again, using closures)
  - Check the non-trivial and decomposed FDs to see if they satisfy the BCNF requirement
  - If all of them satisfy the requirement, then R is in BCNF

BCNF Check: Example

- R(A, B, C, D) with FDs AB → C, C → D, and D→A

BCNF Check: Example

- R(A, B, C, D) with FDs AB → C, C → D, and D → A

  1. Compute the closure for each subset of the attributes in R

     - $\{A\}^+ = \{A\}$, $\{B\}^+ = \{B\}$, $\{C\}^+ = \{ACD\}$, $\{D\}^+ = \{AD\}$
     - $\{AB\}^+ = \{ABCD\}$, $\{AC\}^+ = \{ACD\}$, $\{AD\}^+ = \{AD\}$
     - $\{BC\}^+ = \{ABCD\}$, $\{BD\}^+ = \{ABCD\}$, $\{CD\}^+ = \{ACD\}$
     - $\{ABC\}^+ = \{ABD\}^+ = \{BCD\}^+ = \{ABCD\}$
     - $\{ACD\}^+ = \{ACD\}$
     - $\{ABCD\}^+ = \{ABCD\}$

BCNF Check: Example

- R(A, B, C, D) with FDs AB → C, C → D, and D→A

  2. Derive the keys of R

  - $\{A\}^+ = \{A\}$, $\{B\}^+ = \{B\}$, $\{C\}^+ = \{ACD\}$, $\{D\}^+ = \{AD\}$
  - $\{AB\}^+ = \{ABCD\}$, $\{AC\}^+ = \{ACD\}$, $\{AD\}^+ = \{AD\}$
  - $\{BC\}^+ = \{ABCD\}$, $\{BD\}^+ = \{ABCD\}$, $\{CD\}^+ = \{ACD\}$
  - $\{ABC\}^+ = \{ABD\}^+ = \{BCD\}^+ = \{ABCD\}$
  - $\{ACD\}^+ = \{ACD\}$
  - $\{ABCD\}^+ = \{ABCD\}$

BCNF Check: Example

- R(A, B, C, D) with FDs AB → C, C → D, and D→A

  2. Derive the keys of R: AB, BC, BD

  - $\{A\}^+ = \{A\}$, $\{B\}^+ = \{B\}$, $\{C\}^+ = \{ACD\}$, $\{D\}^+ = \{AD\}$
  - $\{AB\}^+ = \{ABCD\}$, $\{AC\}^+ = \{ACD\}$, $\{AD\}^+ = \{AD\}$
  - $\{BC\}^+ = \{ABCD\}$, $\{BD\}^+ = \{ABCD\}$, $\{CD\}^+ = \{ACD\}$
  - $\{ABC\}^+ = \{ABD\}^+ = \{BCD\}^+ = \{ABCD\}$
  - $\{ACD\}^+ = \{ACD\}$
  - $\{ABCD\}^+ = \{ABCD\}$

BCNF Check: Example

- R(A, B, C, D) with FDs AB → C, C → D, and D→A
  2. Derive the keys of R: AB, BC, BD
  3. Derive the non-trivial and decomposed FDs on R
  - $\{A\}^+ = \{A\}$, $\{B\}^+ = \{B\}$, $\{C\}^+ = \{ACD\}$, $\{D\}^+ = \{AD\}$
  - $\{AB\}^+ = \{ABCD\}$, $\{AC\}^+ = \{ACD\}$, $\{AD\}^+ = \{AD\}$
  - $\{BC\}^+ = \{ABCD\}$, $\{BD\}^+ = \{ABCD\}$, $\{CD\}^+ = \{ACD\}$
  - $\{ABC\}^+ = \{ABD\}^+ = \{BCD\}^+ = \{ABCD\}$
  - $\{ACD\}^+ = \{ACD\}$
  - $\{ABCD\}^+ = \{ABCD\}$

BCNF Check: Example

- R(A, B, C, D) with FDs AB $\rightarrow$ C, C $\rightarrow$ D, and D$\rightarrow$A
  2. Derive the keys of R: AB, BC, BD
  3. Derive the non-trivial and decomposed FDs on R
     - C$\rightarrow$A, C$\rightarrow$D, D$\rightarrow$A
     - $\{AB\}^+ = \{ABCD\}$, $\{AC\}^+ = \{ACD\}$, $\{AD\}^+ = \{AD\}$
     - $\{BC\}^+ = \{ABCD\}$, $\{BD\}^+ = \{ABCD\}$, $\{CD\}^+ = \{ACD\}$
     - $\{ABC\}^+ = \{ABD\}^+ = \{BCD\}^+ = \{ABCD\}$
     - $\{ACD\}^+ = \{ACD\}$
     - $\{ABCD\}^+ = \{ABCD\}$

BCNF Check: Example

- R(A, B, C, D) with FDs AB $\rightarrow$ C, C $\rightarrow$ D, and D$\rightarrow$A

  2. Derive the keys of R: AB, BC, BD

  3. Derive the non-trivial and decomposed FDs on R

  - C$\rightarrow$A, C$\rightarrow$D, D$\rightarrow$A
  - AB$\rightarrow$C, AB$\rightarrow$D, AC$\rightarrow$D
  - $\{BC\}^+ = \{ABCD\}$, $\{BD\}^+ = \{ABCD\}$, $\{CD\}^+ = \{ACD\}$
  - $\{ABC\}^+ = \{ABD\}^+ = \{BCD\}^+ = \{ABCD\}$
  - $\{ACD\}^+ = \{ACD\}$
  - $\{ABCD\}^+ = \{ABCD\}$

BCNF Check: Example

- R(A, B, C, D) with FDs AB → C, C → D, and D→A

  2. Derive the keys of R: AB, BC, BD
  3. Derive the non-trivial and decomposed FDs on R

  - C→A, C→D, D→A
  - AB→C, AB→D, AC→D
  - BC→A, BC→D, BD→A, BD→C, CD→A
  - $\{ABC\}^+ = \{ABD\}^+ = \{BCD\}^+ = \{ABCD\}$
  - $\{ACD\}^+ = \{ACD\}$
  - $\{ABCD\}^+ = \{ABCD\}$

BCNF Check: Example

- R(A, B, C, D) with FDs AB → C, C → D, and D→A

  2. Derive the keys of R: AB, BC, BD
  3. Derive the non-trivial and decomposed FDs on R

  - C→A, C→D, D→A
  - AB→C, AB→D, AC→D
  - BC→A, BC→D, BD→A, BD→C, CD→A
  - ABC→D, ABD→C, BCD→A

BCNF Check: Example

- R(A, B, C, D) with FDs AB → C, C → D, and D→A

  2. Derive the keys of R: AB, BC, BD

  3. Derive the non-trivial and decomposed FDs on R

     - C→A, C→D, D→A
     - AB→C, AB→D, AC→D
     - BC→A, BC→D, BD→A, BD→C, CD→A
     - ABC→D, ABD→C, BCD→A

  4. For each non-trivial and decomposed FD, check whether its left hand side is a super-key

BCNF Check: Example

- R(A, B, C, D) with FDs AB → C, C → D, and D→A

  2. Derive the keys of R: AB, BC, BD

  3. Derive the non-trivial and decomposed FDs on R

  - C→A, C→D, D→A
  - AB→C, AB→D, AC→D
  - BC→A, BC→D, BD→A, BD→C, CD→A
  - ABC→D, ABD→C, BCD→A

  4. For each non-trivial and decomposed FD, check whether its left hand side is a super-key

Not in BCNF

BCNF Check

- The previous algorithm
  - Compute the closure of each attribute subset
  - Derive the keys of R (using closures)
  - Derive all non-trivial and decomposed FDs on R (again, using closures)
  - Check the non-trivial and decomposed FDs to see if they satisfy BCNF requirement
  - If all of them satisfy the requirement, then R is in BCNF
- Observation:
  - The three steps in blue are quite tedious
  - We will simplify them by combing them together, again using closures

Simplified BCNF Check: How?

- What we need: check if there is a non-trivial and decomposed FD $A_1A_2 \ldots A_k \rightarrow B_1$, such that $A_1A_2 \ldots A_k$ is not a superkey
- Question: if $A_1A_2 \ldots A_k$ is not a superkey, what would its closure $\{A_1A_2 \ldots A_k\}^+$ look like?
- First, the closure should contain B_1, since $A_1A_2 \ldots A_k \rightarrow B_1$
  - i.e., the closure contains **more** attributes than $\{A_1A_2 \ldots A_k\}$ does
- Second, the closure should not contain all attributes in the table, since $A_1A_2 \ldots A_k$ is not a superkey
  - i.e., the closure contains **not all** attributes in the table
- Conclusion: we have a violation of BCNF, iff we have a closure that satisfies the "**more but not all**" condition

Simplified BCNF Check: Algorithm

- Conclusion: we have a violation of BCNF, iff we have a closure that satisfies the "more but not all" condition
- Simplified Algorithm for BCNF check:
  - Compute the closure of each attribute subset
  - Check if there is a closure $\{A_1A_2\ldots A_k\}^+$, such that
    - The closure contains some attribute not in $\{A_1A_2\ldots A_k\}$
    - The closure does not contain all attributes in the table
    - i.e., a "**more but not all**" closure
  - If such a closure exists, then R is NOT in BCNF

Simplified BCNF Check: Example

- R(A, B, C, D) with FDs AB → C, C → D, and D→A

Simplified BCNF Check: Example

- R(A, B, C, D) with FDs AB → C, C → D, and D→A
  1. Compute the closure of each attribute subset
     - $\{A\}^+ = \{A\}$, $\{B\}^+ = \{B\}$, $\{C\}^+ = \{ACD\}$, $\{D\}^+ = \{AD\}$
- Stop right there…
- Take a look at $\{C\}^+ = \{ACD\}$
  - $\{C\}^+$ contains more attributes than $\{C\}$ does
  - $\{C\}^+$ does not contain all attributes in R
- "More but not all", which is a violation of BCNF
- So R is not in BCNF

Exercise

- R(A, B, C, D) with FDs B → C, B → D
- Is R in BCNF?

Exercise

Not in BCNF

- R(A, B, C, D) with FDs B → C, B → D
  - Compute the closure of each attribute subset
    - $\{A\}^+ = \{A\}$, $\{B\}^+ = \{BCD\}$, $\{C\}^+ = \{C\}$, $\{D\}^+ = \{D\}$
  - $\{B\}^+ = \{BCD\}$ stratifies the "more but not all" property
  - So it indicates a violation of BCNF

Exercise

- R(A, B, C, D) with FDs A → B, B → C, C→D, and D→A
- Is R in BCNF?

Exercise

In BCNF

- R(A, B, C, D) with FDs $A \rightarrow B$, $B \rightarrow C$, $C \rightarrow D$, and $D \rightarrow A$
  - Compute the closure for each subset of the attributes in R
    - $\{A\}^+ = \{ABCD\}$, $\{B\}^+ = \{ABCD\}$, $\{C\}^+ = \{ABCD\}$, $\{D\}^+ = \{ABCD\}$
    - The other closures are all {ABCD}
  - There would be no closure satisfying the "more but not all" property
  - So there is no violation of BCNF

Exercise

- R(A, B, C, D, E) with FDs AB $\rightarrow$ C, C $\rightarrow$ E, E $\rightarrow$ A, and E $\rightarrow$ D
- Is R in BCNF?

Exercise

Not In BCNF

- R(A, B, C, D, E) with FDs AB $\rightarrow$ C, C $\rightarrow$ E, E $\rightarrow$ A, and E $\rightarrow$ D
  - Compute the closure for each subset of the attributes in R
    - $\{A\}^+ = \{A\}$, $\{B\}^+ = \{B\}$, $\{C\}^+ = \{ACDE\}$, $\{D\}^+ = \{D\}$, $\{E\}^+ = \{ADE\}$
  - $\{E\}^+ = \{ADE\}$ satisfies the "more but not all" property
  - So $\{E\}^+ = \{ADE\}$ indicates a violation of BCNF

Exercise

- R(A, B, C, D) with FDs AB → D, BD → C, CD → A, and AC → B
- Is R in BCNF?

Exercise

In BCNF

- R(A, B, C, D) with FDs $AB \rightarrow D$, $BD \rightarrow C$, $CD \rightarrow A$, and $AC \rightarrow B$
  - Compute the closure for each subset of the attributes in R
    - $\{A\}^+ = \{A\}$, $\{B\}^+ = \{B\}$, $\{C\}^+ = \{C\}$, $\{D\}^+ = \{D\}$
    - $\{AB\}^+ = \{BD\}^+ = \{CD\}^+ = \{AC\}^+ = \{ABCD\}$,
    - $\{AD\}^+ = \{AD\}$, $\{BC\}^+ = \{BC\}$
    - $\{ABC\}^+ = \{ABD\}^+ = \{BCD\}^+ = \{ACD\}^+ = \{ABCD\}$
    - $\{ABCD\}^+ = \{ABCD\}$
  - There is no closure satisfying the "more but not all" property
  - So there is no violation of BCNF

Roadmap

- Now we know how to check whether a table is in BCNF
- But if a table is not in BCNF, how can we improve it?
- We can decompose it into smaller tables
  - This is also called a normalization

BCNF Decomposition

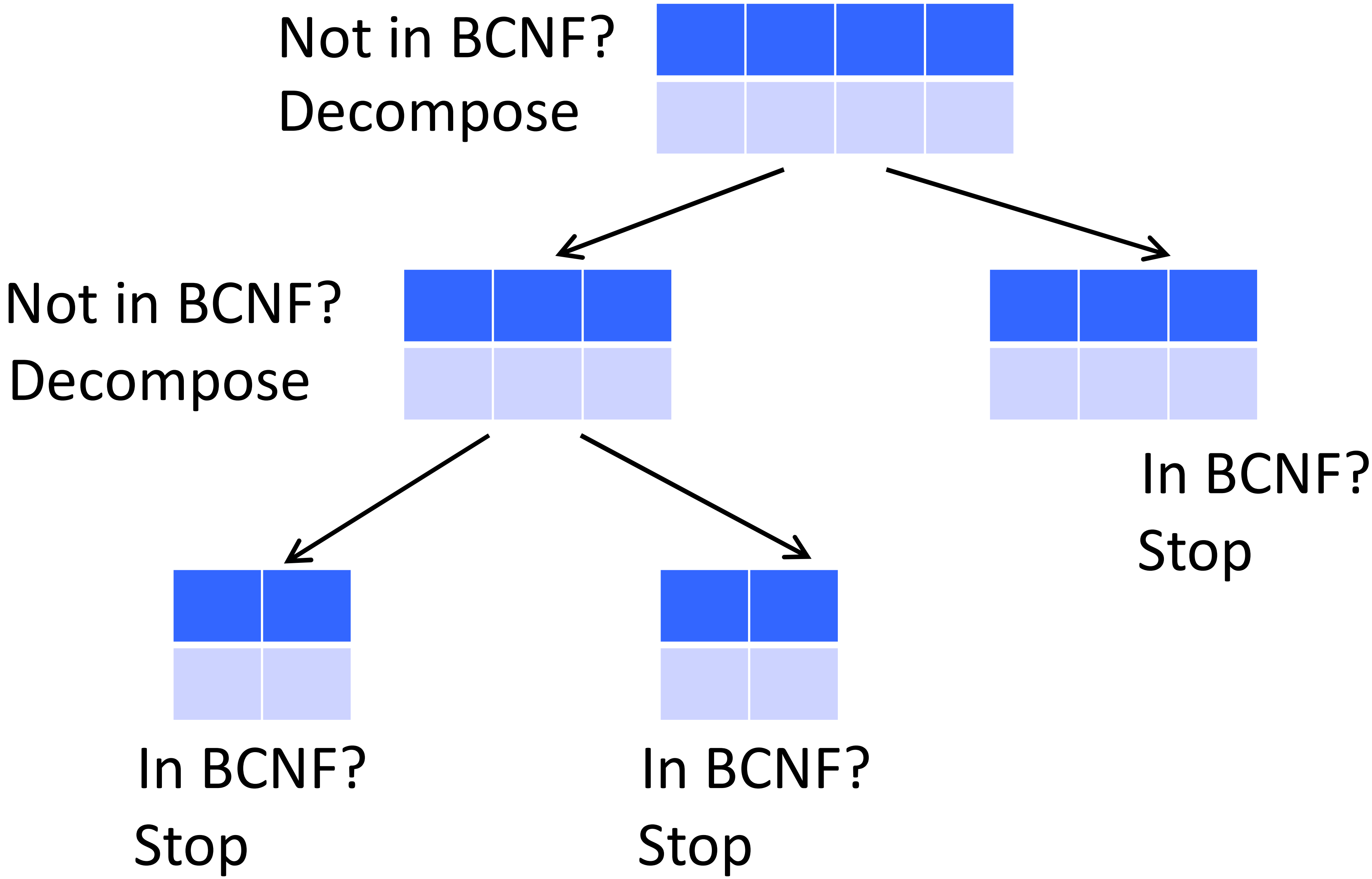
Name	NRIC	PhoneNumber	HomeAddress
Alice	1234	67899876	Jurong East
Alice	1234	83848384	Jurong East
Bob	5678	98765432	Pasir Ris

- Decomposing non-BCNF tables into smaller ones in BCNF

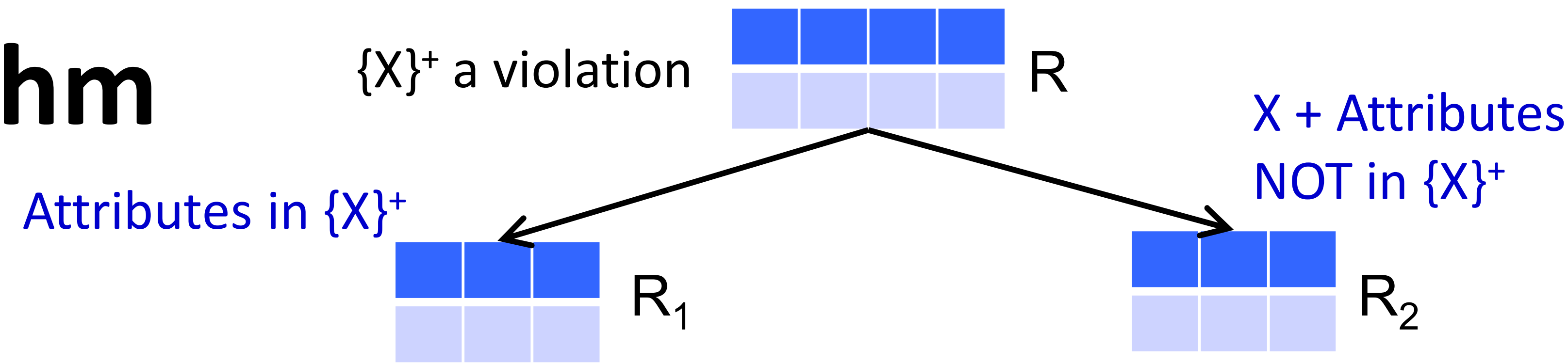
Name	NRIC	HomeAddress
Alice	1234	Jurong East
Bob	5678	Pasir Ris

NRIC	PhoneNumber
1234	67899876
1234	83848384
5678	98765432

Decompose, until all are in BCNF



Algorithm



- Input: a table R

1. Find a subset X of the attributes in R, such that its closure $\{X\}^+$ (i) contains more attributes than X does, but (ii) does not contain all attributes in R
2. Decompose R into two tables R_1 and R_2, such that
   - R_1 contains all attributes in $\{X\}^+$
   - R_2 contains all attributes in X as well as the attributes not in $\{X\}^+$
3. If R_1 is not in BCNF, further decompose R_1;
   If R_2 is not in BCNF, further decompose R_2

BCNF Decomposition: Example

Name	NRIC	PhoneNumber	HomeAddress
Alice	1234	67899876	Jurong East
Alice	1234	83848384	Jurong East
Bob	5678	98765432	Pasir Ris

- FD: NRIC → Name, HomeAddress

  1. Find a subset X of the attributes in R, such that its closure X^+ (i) contains more attributes than X, but (ii) does not contain all attributes in R
- $\{NRIC\}^+ = \{Name, NRIC, HomeAddress\}$

  2. Decompose R into two tables R_1 and R_2, such that
  - R_1 contains all attributes in X^+
  - R_2 contains all attributes in X as well as the attributes not in X^+
- R_1(Name, NRIC, HomeAddress), R_2(NRIC, PhoneNumber)

  3. Check if R_1 and R_2 are in BCNF, and so on. (Spoiler: they are in BCNF)

BCNF Decomposition: Example

Name	<u>NRIC</u>	<u>PhoneNumber</u>	HomeAddress
Alice	1234	67899876	Jurong East
Alice	1234	83848384	Jurong East
Bob	5678	98765432	Pasir Ris

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

<u>NRIC</u>	<u>PhoneNumber</u>
1234	67899876
1234	83848384
5678	98765432

BCNF Decomposition: Example

- R(A, B, C, D) with FDs A→B, B→C

1. Find a subset X of the attributes in R, such that its closure $\{X\}^+$ (i) contains more attributes than X, but (ii) does not contain all attributes in R

- $\{A\}^+ = \{A, B, C\}$

2. Decompose R into two tables R_1 and R_2, such that
    - R_1 contains all attributes in $\{X\}^+$
    - R_2 contains all attributes in X as well as the attributes not in $\{X\}^+$

- $R_1(A, B, C)$, $R_2(A, D)$

3. Check if R_1 and R_2 are in BCNF

- R_1: No, R_2: Yes

4. Further decompose R_1

BCNF Decomposition: Example

- $R(A, B, C, D)$ with FDs $A \rightarrow B$, $B \rightarrow C$
- $R_1(A, B, C)$, $R_2(A, D)$
- Further decompose R_1

1. Find a subset X of the attributes in R_1, such that its closure $\{X\}^+$ (i) contains more attributes than X, but (ii) does not contain all attributes in R

- $\{A\}^+ = \{A, B, C\}$, $\{B\}^+ = \{B, C\}$

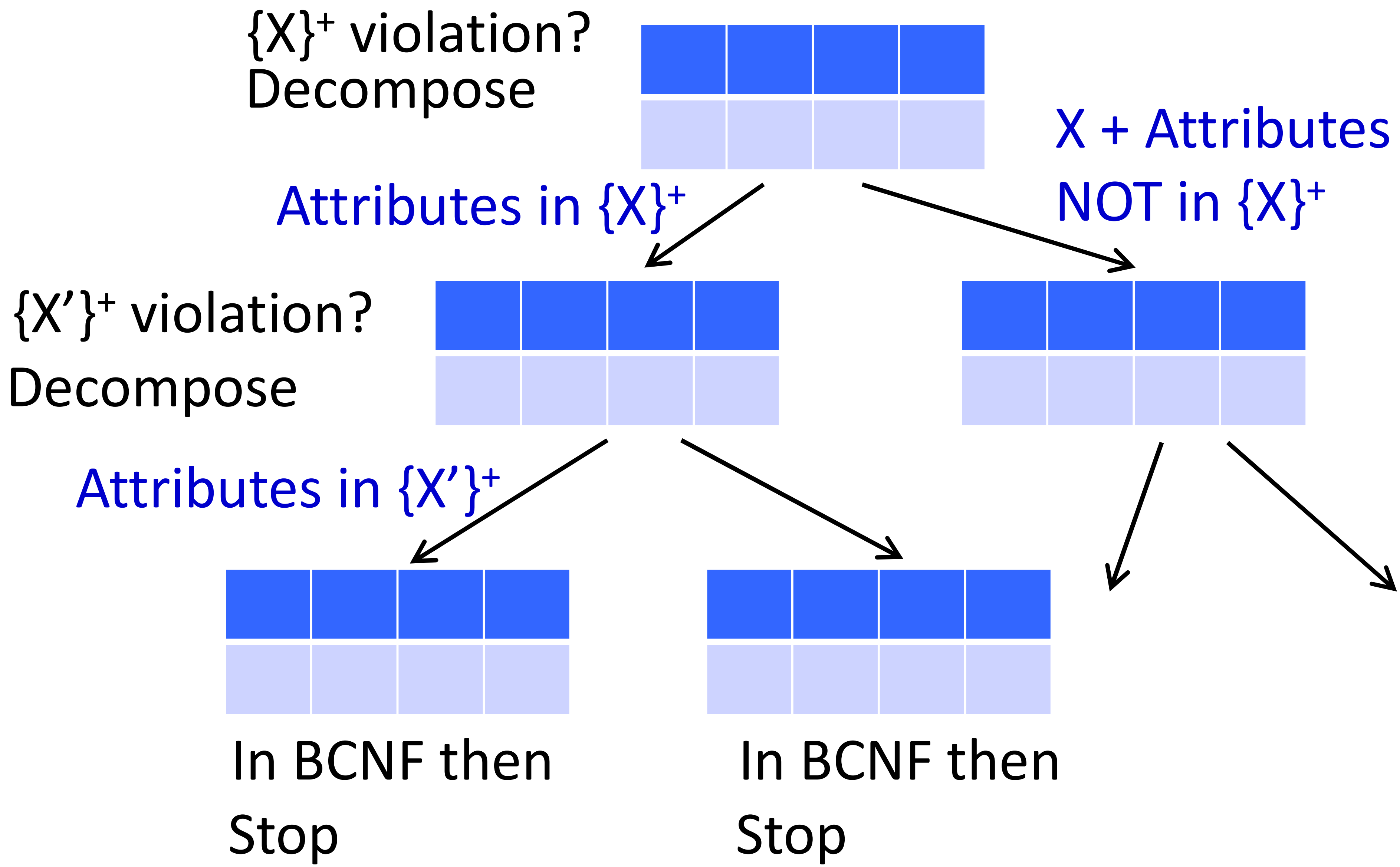
2. Decompose R_1 into two tables R_3 and R_4, such that
   - R_3 contains all attributes in $\{X\}^+$
   - R_4 contains all attributes in X as well as the attributes not in $\{X\}^+$

- $R_3(B, C)$, $R_4(A, B)$

3. Check if R_1 and R_2 are in BCNF

- R_3: Yes, R_4: Yes
- Final results: $R_3(B, C)$, $R_4(A, B)$, $R_2(A, D)$

Decompose, until all are in BCNF



Notes

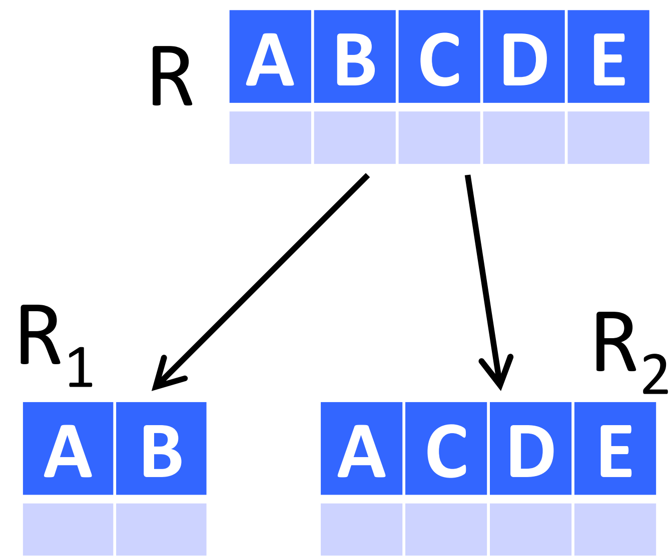
- The BCNF decomposition of a table may not be unique
- If a table has only two attributes, then it must be in BCNF
  - Therefore, you do not need to check tables with only two attributes

BCNF Decomposition: One More Issue

- Recall that, whenever we decompose a table R into two smaller tables R_1 and R_2, we need to check whether R_1 and R_2 satisfies BCNF
- This requires us to check the closures on R_1 and R_2
- We will explain how this can be done using an example

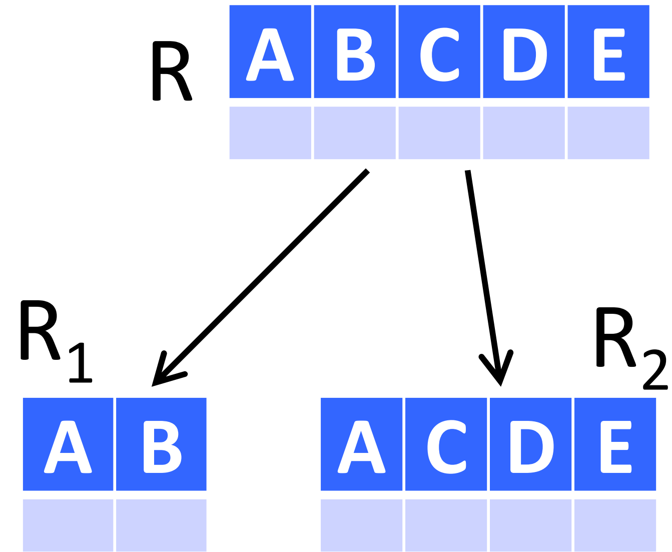
BCNF Decomposition: One More Issue

- Given: A→B, BC→D
- Step 1: Check if there is closure that indicates a violation of BCNF
  - $\{A\}^+ = \{A, B\}$
- Step 2: Decompose the table into two
  - First one: include all attributes in the closure, i.e, {A, B}
  - Second one: include A and all attributes NOT in the closure, i.e., {A, C, D, E}
- Now we need to check whether R_1 and R_2 are in BCNF
- R_1 is in BCNF; but what about R_2?



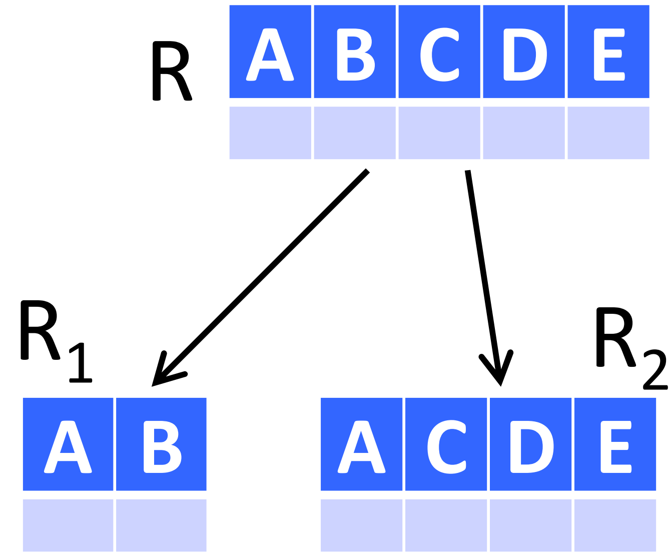
BCNF Decomposition: One More Issue

- Given: $A \rightarrow B$, $BC \rightarrow D$
- To check whether R_2 is in BCNF, we need to derive the closures for R_2
- But we don't know what FDs are there on R_2
- Solution:
  - Derive the closures on R
  - Then, project them onto R_2



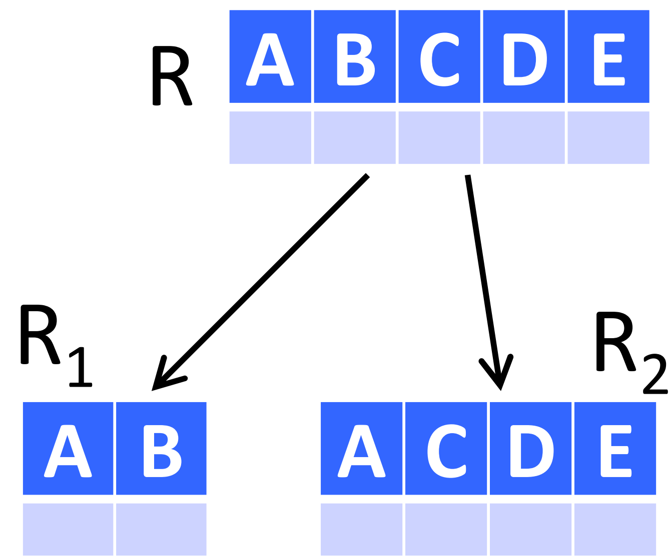
BCNF Decomposition: One More Issue

- Given: $A \rightarrow B$, $BC \rightarrow D$
- Deriving closures for R_2
  - Step 1: enumerate the attribute subsets in R_2
    - {A}
    - {C}
    - {D}
    - {E}
    - {AC}
    - {AD}
    - {AE}
    - {CD}
    - {CE}
    - {DE}
    - {ACD}
    - {ACE}
    - {ADE}
    - {CDE}



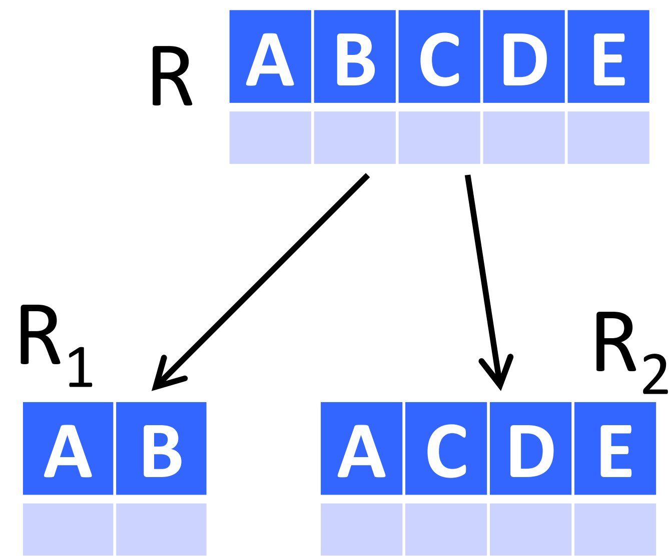
BCNF Decomposition: One More Issue

- Given: A→B, BC→D
- Deriving closures for R_2
  - Step 2: derive the closures of these attribute subsets on R
    - {A}
    - {C}
    - {D}
    - {E}
    - {AC}
    - {AD}
    - {AE}
    - {CD}
    - {CE}
    - {DE}
    - {ACD}
    - {ACE}
    - {ADE}
    - {CDE}



BCNF Decomposition: One More Issue

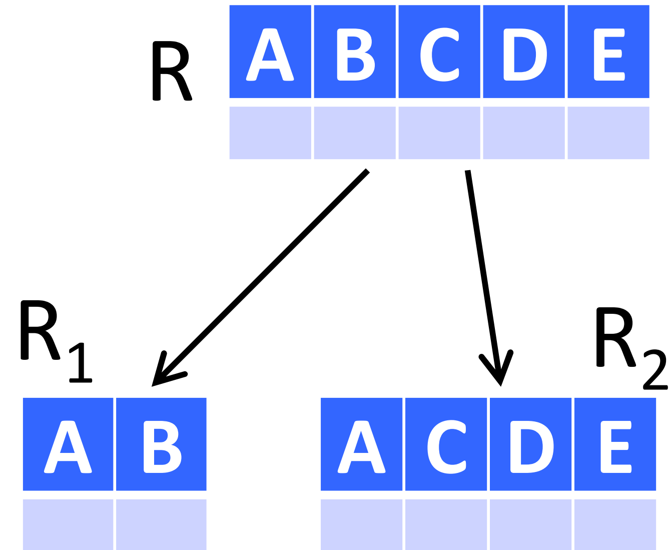
- Given: $A \rightarrow B$, $BC \rightarrow D$
- Deriving closures for R_2
  - Step 2: derive the closures of these attribute subsets on R



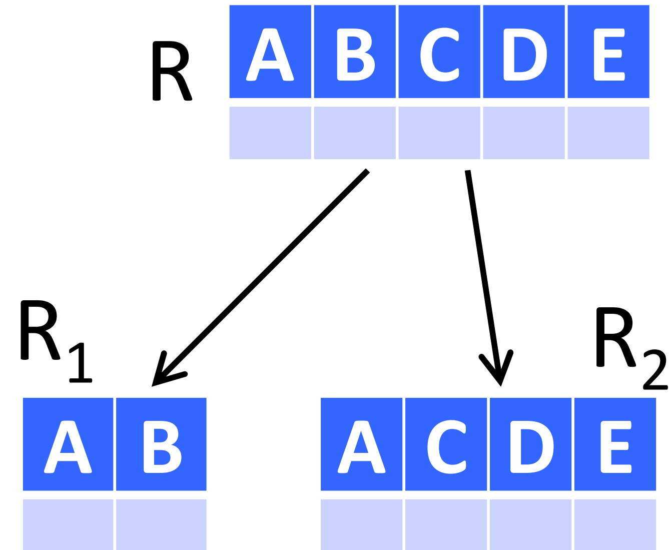
$\{A\}^+=\{AB\}$	$\{C\}^+=\{C\}$
$\{D\}^+=\{D\}$	$\{E\}^+=\{E\}$
$\{AC\}^+=\{ABCD\}$	$\{AD\}^+=\{ABD\}$
$\{AE\}^+=\{ABE\}$	$\{CD\}^+=\{CD\}$
$\{CE\}^+=\{CE\}$	$\{DE\}^+=\{DE\}$
$\{ACD\}^+=\{ABCD\}$	$\{ACE\}^+=\{ABCDE\}$
$\{ADE\}^+=\{ABCDE\}$	$\{CDE\}^+=\{CDE\}$

BCNF Decomposition: One More Issue

- Given: $A \rightarrow B$, $BC \rightarrow D$
- Deriving closures for R_2
  - Step 3: Project these closures onto R_2, by removing irrelevant attributes
    - $\{A\}^+=\{AB\}$ $\quad$ $\{C\}^+=\{C\}$
    - $\{D\}^+=\{D\}$ $\quad$ $\{E\}^+=\{E\}$
    - $\{AC\}^+=\{ABCD\}$ $\quad$ $\{AD\}^+=\{ABD\}$
    - $\{AE\}^+=\{ABE\}$ $\quad$ $\{CD\}^+=\{CD\}$
    - $\{CE\}^+=\{CE\}$ $\quad$ $\{DE\}^+=\{DE\}$
    - $\{ACD\}^+=\{ABCD\}$ $\quad$ $\{ACE\}^+=\{ABCDE\}$
    - $\{ADE\}^+=\{ABCDE\}$ $\quad$ $\{CDE\}^+=\{CDE\}$

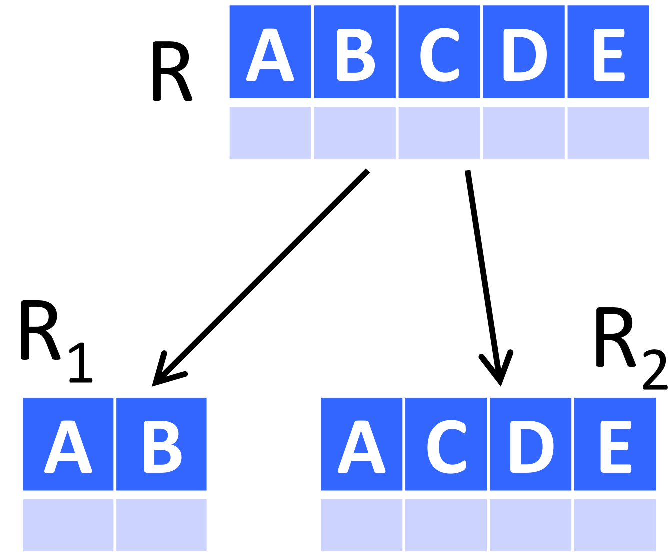


BCNF Decomposition: One More Issue



- Given: A→B, BC→D
- Deriving closures for R_2
  - Step 3: Project these closures onto R_2, by removing irrelevant attributes
    - $\{A\}^+=\{AB\}$
    - $\{C\}^+=\{C\}$
    - $\{D\}^+=\{D\}$
    - $\{E\}^+=\{E\}$
    - $\{AC\}^+=\{ABCD\}$
    - $\{AD\}^+=\{ABD\}$
    - $\{AE\}^+=\{ABE\}$
    - $\{CD\}^+=\{CD\}$
    - $\{CE\}^+=\{CE\}$
    - $\{DE\}^+=\{DE\}$
    - $\{ACD\}^+=\{ABCD\}$
    - $\{ACE\}^+=\{ABCDE\}$
    - $\{ADE\}^+=\{ABCDE\}$
    - $\{CDE\}^+=\{CDE\}$

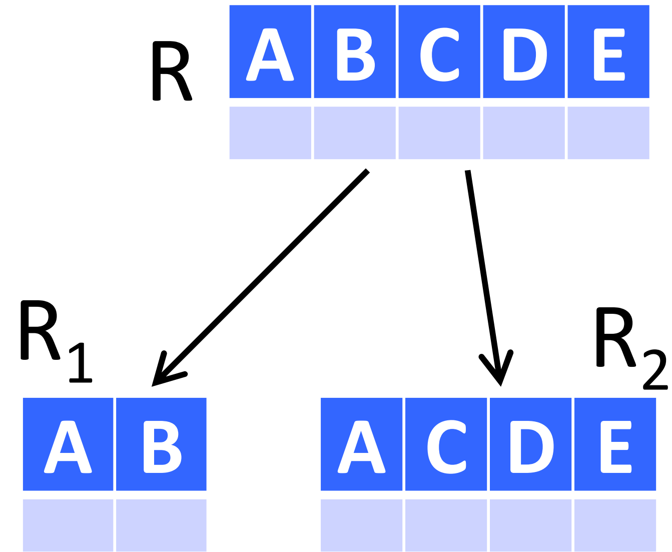
BCNF Decomposition: One More Issue



- Given: A→B, BC→D
- Deriving closures for R_2
  - Step 3: Project these closures onto R_2, by removing irrelevant attributes
    - $\{A\}^+=\{A\}$
    - $\{C\}^+=\{C\}$
    - $\{D\}^+=\{D\}$
    - $\{E\}^+=\{E\}$
    - $\{AC\}^+=\{ACD\}$
    - $\{AD\}^+=\{AD\}$
    - $\{AE\}^+=\{AE\}$
    - $\{CD\}^+=\{CD\}$
    - $\{CE\}^+=\{CE\}$
    - $\{DE\}^+=\{DE\}$
    - $\{ACD\}^+=\{ACD\}$
    - $\{ACE\}^+=\{ACDE\}$
    - $\{ADE\}^+=\{ACDE\}$
    - $\{CDE\}^+=\{CDE\}$

BCNF Decomposition: One More Issue

- Given: A→B, BC→D
- Deriving closures for R_2
  - Step 3: Project these closures onto R_2, by removing irrelevant attributes
    - $\{A\}^+=\{A\}$
    - $\{D\}^+=\{D\}$
    - $\boxed{\{AC\}^+=\{ACD\}}$
    - $\{AE\}^+=\{AE\}$
    - $\{CE\}^+=\{CE\}$
    - $\{ACD\}^+=\{ACD\}$
    - $\{ADE\}^+=\{ACDE\}$
    - $\{C\}^+=\{C\}$
    - $\{E\}^+=\{E\}$
    - $\{AD\}^+=\{AD\}$
    - $\{CD\}^+=\{CD\}$
    - $\{DE\}^+=\{DE\}$
    - $\{ACE\}^+=\{ACDE\}$
    - $\{CDE\}^+=\{CDE\}$



- This closure violates BCNF
- So R2 is not in BCNF

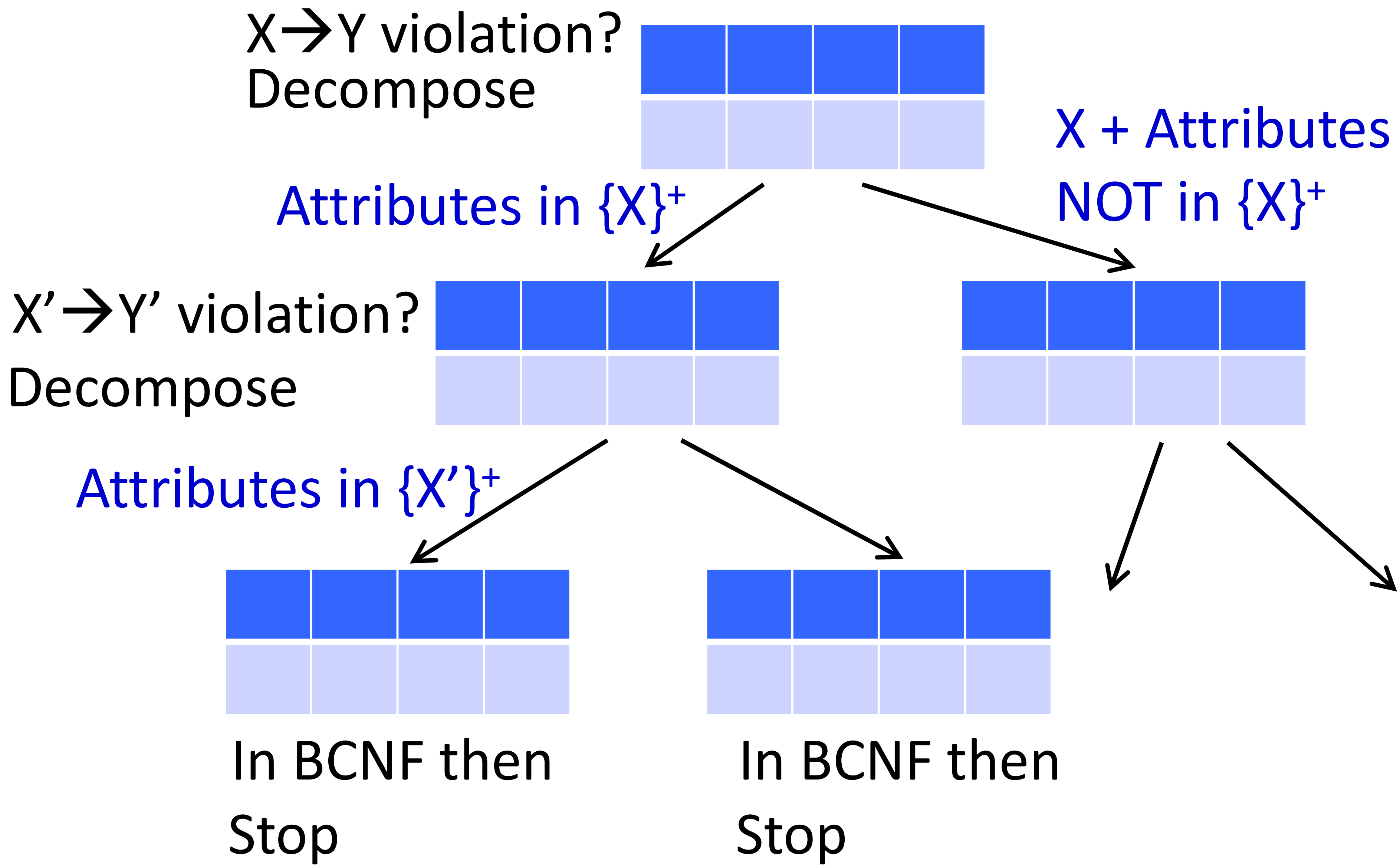
Projection of Closures/FDs

- In general, if we are to derive the closures on a table R_i that is decomposed from a table R, we can
  - First, enumerate the attribute subsets of R_i
  - For each subset, derive its closure on R
  - Project each closure onto R_i by removing those attributes that do not appear in R_i
- These projected closures can then be used to
  - Decide whether R_i is in BCNF
  - Further decompose R_i (if R_i violates BCNF)

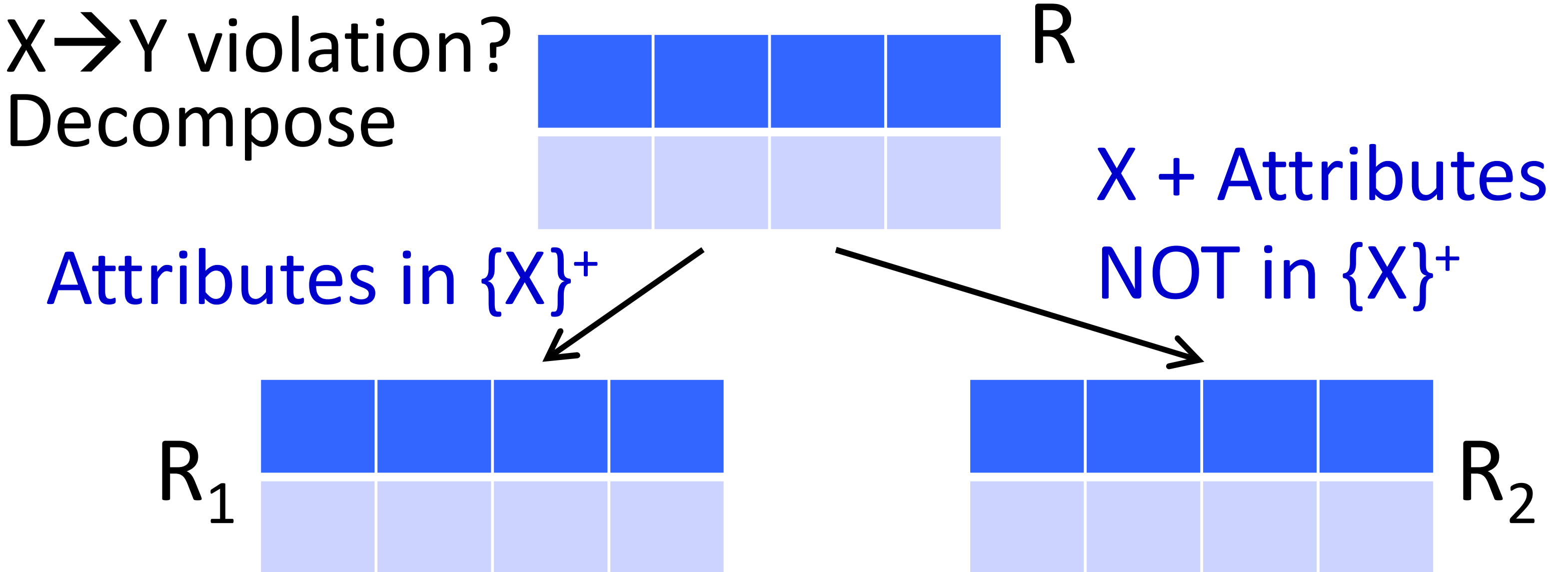
Question

- Why does the BCNF decomposition algorithm work?
- Why can it eliminate violations of BCNF?

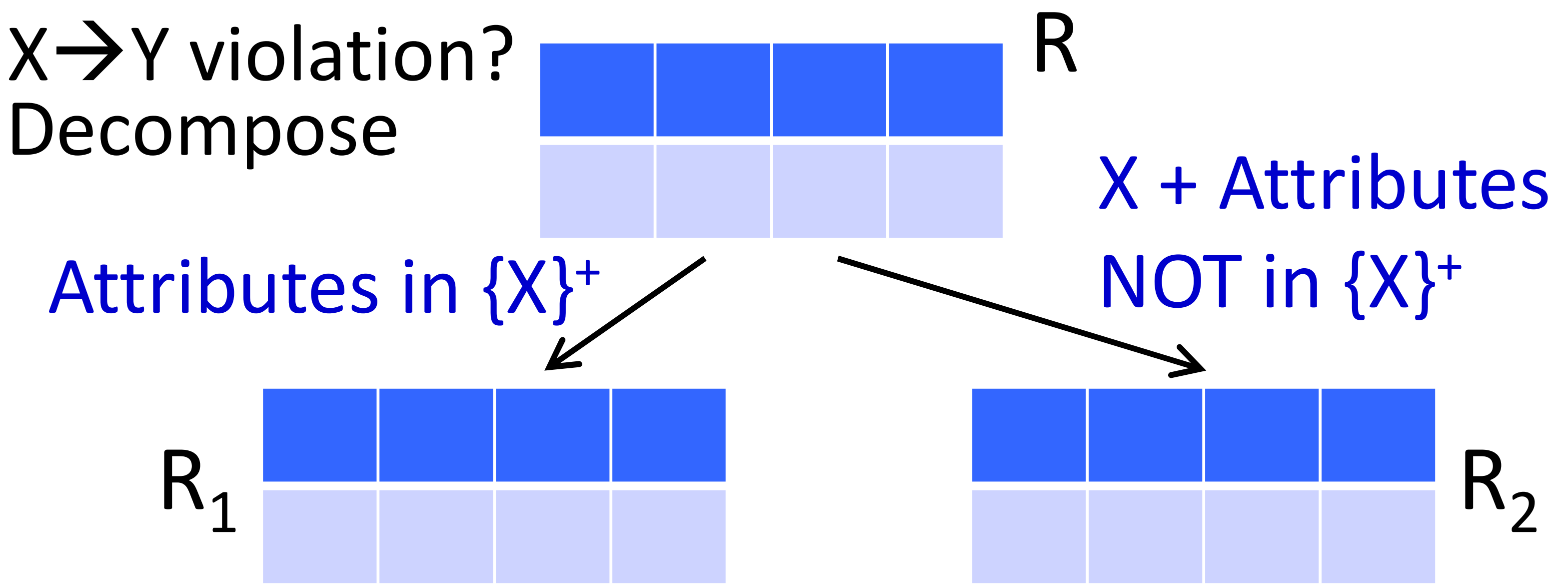
BCNF Decomposition Algorithm



BCNF Decomposition Algorithm

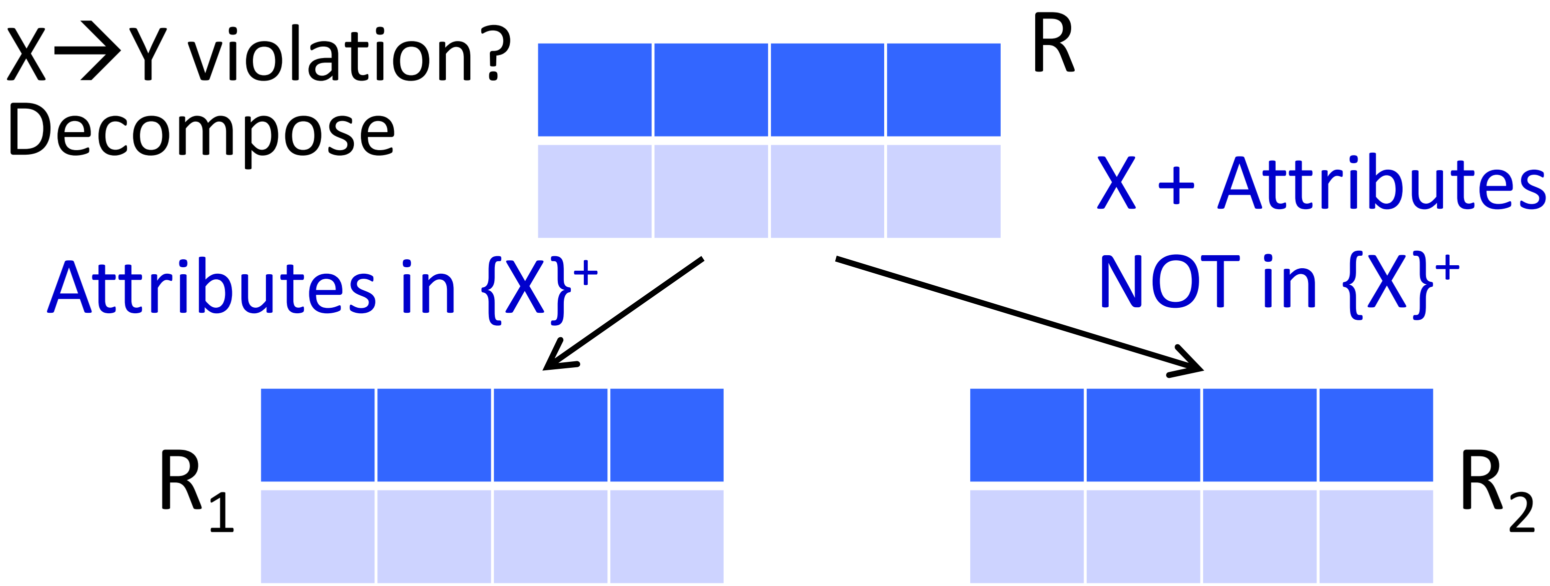


BCNF Decomposition Algorithm



- X→Y is no longer a BCNF violation on R_1
- X→Y is no longer an FD on R_2
- So this decomposition step gets rid of one BCNF violation

BCNF Decomposition Algorithm



- In general, each decomposition step removes at least one BCNF violation
- Recursive decomposition ==> all violations will be removed in the end

Exercise

- R(A, B, C, D) with FDs $A \rightarrow B$, $A \rightarrow C$

1. Find a subset X of the attributes in R, such that its closure X^+ (i) contains more attributes than X, but (ii) does not contain all attributes in R
2. Decompose R into two tables R_1 and R_2, such that
   - R_1 contains all attributes in X^+
   - R_2 contains all attributes in X as well as the attributes not in X^+
3. If R_1 is not in BCNF, further decompose R_1;
   If R_2 is not in BCNF, further decompose R_2

Exercise

- $R(A, B, C, D)$ with FDs $A \rightarrow B$, $A \rightarrow C$

1. Find a subset X of the attributes in R, such that its closure X^+ (i) contains more attributes than X, but (ii) does not contain all attributes in R

- $\{A\}^+ = \{A, B, C\}$

2. Decompose R into two tables R_1 and R_2, such that
   - R_1 contains all attributes in X^+
   - R_2 contains all attributes in X as well as the attributes not in X^+

- $R_1(A, B, C)$, $R_2(A, D)$

3. Check if R_1 and R_2 are in BCNF

- Yes. Final results: $R_1(A, B, C)$, $R_2(A, D)$

Exercise

- R(A, B, C, D) with FDs BC→D, D→A, A→B

1. Find a subset X of the attributes in R, such that its closure X^+ (i) contains more attributes than X, but (ii) does not contain all attributes in R
2. Decompose R into two tables R_1 and R_2, such that
   - R_1 contains all attributes in X^+
   - R_2 contains all attributes in X as well as the attributes not in X^+
3. If R_1 is not in BCNF, further decompose R_1;
   If R_2 is not in BCNF, further decompose R_2

Exercise

- R(A, B, C, D) with FDs BC→D, D→A, A→B

1. Find a subset X of the attributes in R, such that its closure X^+ (i) contains more attributes than X, but (ii) does not contain all attributes in R

- $\{A\}^+ = \{A, B\}$

2. Decompose R into two tables R_1 and R_2, such that
   - R_1 contains all attributes in X^+
   - R_2 contains all attributes in X as well as the attributes not in X^+

- $R_1(A, B)$, $R_2(A, C, D)$

3. Check if R_1 and R_2 are in BCNF

- R_1: Yes. R_2: No
- Further decompose R2

Exercise

- $R(A, B, C, D)$ with FDs $BC \rightarrow D$, $D \rightarrow A$, $A \rightarrow B$
- $R_1(A, B)$, $R_2(A, C, D)$
- Further decompose R_2

1. Find a subset X of the attributes in R_2, such that its closure X^+ (i) contains more attributes than X, but (ii) does not contain all attributes in R_2

- $\{A\}^+ = \{A\}$, $\{C\}^+ = \{C\}$, $\{D\}^+ = \{A, D\}$,

2. Decompose R_1 into two tables R_3 and R_4, such that
   - R_3 contains all attributes in X^+
   - R_4 contains all attributes in X as well as the attributes not in X^+

- $R_3(A, D)$, $R_4(C, D)$

3. Check if R_3 and R_4 are in BCNF

- Yes. Final results: $R_1(A, B)$, $R_3(A, D)$, $R_4(C, D)$

Properties of BCNF

- Good properties
  - No update or deletion or insertion anomalies
  - Small redundancy
  - The original table can always be reconstructed from the decomposed tables

Table Reconstruction

Name	NRIC	PhoneNumber	HomeAddress
Alice	1234	67899876	Jurong East
Alice	1234	83848384	Jurong East
Bob	5678	98765432	Pasir Ris

- SELECT * FROM R1, R2
  WHERE R1.NRIC = R2.NRIC

This is called a "Lossless Join"

Name	NRIC	HomeAddress
Alice	1234	Jurong East
Bob	5678	Pasir Ris

R1

NRIC	PhoneNumber
1234	67899876
1234	83848384
5678	98765432

R2

Lossless Join Decomposition

- Say we decompose a table R into two tables R_1 and R_2
- The decomposition guarantees lossless join, whenever the common attributes in R_1 and R_2 constitute a superkey of R_1 or R_2
- Example
  - R(A, B, C) decomposed into R_1(A, B) and R_2 (B, C), with B being a superkey of R_2
  - R(A, B, C, D) decomposed into R_1(A, B, C) and R_2 (B, C, D), with BC being a superkey of R_1

Why BCNF guarantees lossless join?

- Decompose R into two tables R_1 and R_2, such that
  - R_1 contains all attributes in $\{X\}^+$
  - R_2 contains all attributes in X as well as the attributes not in $\{X\}^+$
- X is the set of common attributes between R_1 and R_2
- X is a superkey of R_1

Properties of BCNF

- Good properties
  - No update or deletion or insertion anomalies
  - Small redundancy
  - The original table can always be reconstructed from the decomposed tables
- Bad properties
  - Dependencies may not be preserved
  - We will talk about it in the next lecture