3 NF Example 2

3NF Synthesis Algorithm

- Input: Set of attributes R and FDs F
- Step 1: Create a minimal cover for F called G
- Step 2. For each FD in G create a table. Call the tables T_1,T_2, …
- Step 3: If none of the T_i contain a super key for the universal table create a new table containing the attributes of a key for the universal table

Create a Set of 3 NF Tables From the Following

- Let R = ABCDEFGH
- Let the FD set be
  - A→E
  - BE→D
  - AD→BE
  - BDH→E
  - AC→E
  - F→A
  - E→B
  - D→H
  - BG→F
  - CD→A

Minimal Cover Step 1
Decompose RHS

- A→E
- BE→D
- AD→BE
- BDH→E
- AC→E
- F→A
- E→B
- D→H
- BG→F
- CD→A

- A→E
- BE→D
- AD→B
- AD->E
- BDH→E
- AC→E
- F→A
- E→B
- D→H
- BG→F
- CD→A

Minimal Cover Step 2
Remove Redundant Attributes on LHS

- A→E
- BE→D
- AD→B
- AD->E
- BDH→E
- AC→E
- F→A
- E→B
- D→H
- BG→F
- CD→A

E+ = EBDH
A+ = AEBDH
BD+ = BDHE

- A→E
- E→D
- A→B
- ~~A->E~~
- BD→E
- ~~A→E~~
- F→A
- E→B
- D→H
- BG→F
- CD→A

Minimal Cover Step 3
Remove Redundant Dependencies

- A→E
- E→D
- A→B
- BD→E
- F→A
- E→B
- D→H
- BG→F
- CD→A

A+ = AEBDH (without A-> B)

- A→E
- E→D
- BD→E
- F→A
- E→B
- D→H
- BG→F
- CD→A

Minimal Cover Step 4
Combine Dependencies with common LHS

- A→E
- E→D
- BD→E
- F→A
- E→B
- D→H
- BG→F
- CD→A

- A→E
- E→BD
- BD→E
- F→A
- D→H
- BG→F
- CD→A

3NF Synthesis Step 2
Make a table for each Dependency

- A→E
- E→BD
- BD→E
- F→A
- D→H
- BG→F
- CD→A

R1 = {A, E}
R2 = {B, D, E}
R3 = {F, A}
R4 = {D, H}
R5 = {B, F, G}
R6 = {A, C, D}

3NF Synthesis Step 3

Does the attribute closure of the attributes of one of the tables include all attributes in the universal table?

In this case no

- A→E
- E→BD
- BD→E
- F→A
- D→H
- BG→F
- CD→A

R1 = {A, E}
R2 = {B, D, E}
R3 = {F, A}
R4 = {D, H}
R5 = {B, F, G}
R6 = {A, C, D}
R7 = {C, D, G}

CG+ = CG
CDG+ = CDGAEBHF

Are the Tables in BCNF
No

- A→E
- E→BD
- BD→E
- F→A
- D→H
- BG→F
- CD→A

R1 = {A, E}
R2 = {B, D, E}
R3 = {F, A}
R4 = {D, H}
R5 = {B, F, G}
R6 = {A, C, D}
R7 = {C, D, G}

- A table T is in BCNF
  - if for all non-trivial dependencies, X -> A, that lie in T, X is a super key
- An FD is a BCNF violator for table T
  - if it is a non-trivial dependency, X -> A, that lies in T where X is not a super key.

In R6 A -> D is a BCNF Violator
Decompose R6 into
R6a = {A, D}
R6b = {A, C}