Normalization and Functional Dependencies

- 1NF
- Redundancy and Anomalies
- Functional Dependencies
- Attribute Closure
- Keys and Super keys
- 2NF
- 3NF
- BCNF
- Minimal Cover Algorithm
- 3NF Synthesis Algorithm
- Decomposition of Tables

Attribute Closure

- Find all attributes dependent on a particular set of attributes.
- The closure of a set of attributes, X, is designated by X^+

Attribute Closure Algorithm Under FD Set F

- *closure* := X; // since $X \subseteq X^+$
  repeat
  old := *closure*;
  if there is an FD $Z \rightarrow V$ in ***F*** such that
  $Z \subseteq$ *closure*
  then *closure* := *closure* $\cup$ V
  until *old* = *closure*

- If $T \subseteq$ *closure* then $X \rightarrow T$ is implied by ***F***

Problem

- Let R = {A, B, C, D, E, F}
- Let the FD set be
  - ABF → C
  - CF → B
  - CD → A
  - BD → AE
  - C → F
  - B → F
- Find the closure of ABC

ABC+ : ABC

: F

Keys and Super Keys

- A set of attributes, X, in a super key for a table T if $X \subseteq T$ and X -> T
- Another way of saying this is that $T \subseteq X^+$
- A set of attributes, X, in a key for a table T if it has the super key property and no proper subset of X has the super key property

Problem

- Let R = {A, B, C, D, E, F}
- Let the FD set be
  - ABF → C
  - CF → B
  - CD → A
  - BD → AE
  - C → F
  - B → F
- Is ABF a super key for R?
- Is ABD a super key for R?
- What attribute must be part of any key for R?

ABF+: ABF
: C

ABD+: ABD
: E
: F
: C

Is ABD a key?
BD+: BD
: AE
: F
: C

Create 3NF Tables

- Identify all attributes, R, and FDs, F
  - A table containing all attributes in R is called the universal table
  - The designers must work with the customers to identify R and F
  - The FDs in F represent "real world" constraints of the data that can be entered into the database
- Create a minimal cover FD set, G, from F
- Apply the 3NF synthesis algorithm using the FD set G and the set of attributes R

Minimal Cover Set

- A minimal cover set, G, of an FD set F is an FD set such that
  - G is equivalent to F
  - No FD can be removed from G to create a “smaller” FD set equivalent to F
  - No FD in G can have an attribute removed from the FD to create a “smaller” FD set equivalent to F
- Minimal cover sets are not unique

Minimal Cover Algorithm for FD Set F

- Step 1: Make all RHS single attributes
  - Use decomposition of RHS on all FDs
- Step 2: Remove redundant attributes from LHS

```
G = F
 repeat
     old = G
     for each XB -> A ∈ G
             if X -> A is implied by G (i.e. A ∈ X+)
             then G = G – {XB -> A } ∪ {X -> A}
 until old == G  (i.e. keep going until G does not change)
```

Minimal Cover Algorithm for FD Set F

- Step 3: Remove redundant FDs from G (G was produced in step 2)

```
H = G
repeat
        old = H
        For each X -> A ∈ H
            if H is equivalent to H – {X -> A} (i.e. A ∈ X+
                where X+ is found using FD set  H – {X -> A})
            then H = H – {X -> A}
until old == H (i.e. keep going until H does not change)
```

Minimal Cover Algorithm for FD Set F

- Step 4: Combine FDs that have the same LHS
  - Use the Union rule
  - Sometimes this step is considered part of the 3NF synthesis algorithm

Problem

- Let R = {A, B, C, D, E, F}
- Let the FD set be
  - ABF → C
  - CF → B
  - CD → A
  - BD → AE
  - C → F
  - B → F
- Find a minimal cover FD set.

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

Problem

- Let R = {A, B, C, D, E, F}
- Let the FD set be
  - ABF $\rightarrow$ C
  - CF $\rightarrow$ B
  - CD $\rightarrow$ A
  - BD $\rightarrow$ AE
  - C $\rightarrow$ F
  - B $\rightarrow$ F
- Create a set of 3NF tables from R and the FD set.

Decomposition of Tables

- Lossless Decomposition
  - A decomposition of T into T1 and T2 is a lossless if and only if T1 ∩ T2 -> T1 or T1 ∩ T2 -> T2
  - A decomposition of T into T1 and T2 is a lossless if for every valid T (valid relative to the FDs) T = T1 Natural Join T2

Decomposition of Tables

- Dependency Preserving Decomposition
  - Let T_1 and T_2 be a decomposition of T with FD set F
  - Let F_1 and F_2 be the FDs from F^+ that lie in T_1 and T_2 respectively
  - The decomposition is dependency preserving if and only if $F^+ = F_1 \cup F_2$

Decomposition of Tables

- The 3NF synthesis algorithm is equivalent to a series of lossless, dependency preserving decompositions into a set of 3NF tables
- A lossless decomposition of the universal table into a set of BCNF tables is possible but the decomposition might not be dependency preserving

Decomposition of Tables

- To remove a 3NF or BCNF violator through decomposition do the following
  - Let T contain attributes X, attributes Y and attribute A
  - Let X -> A be violator that lies in T
  - Decompose T into T1 and T2 where T1 contains attributes X and attribute A and T2 contains attributes X and attributes Y
  - The decomposition is lossless because $X = T1 \cap T2$ and X is a super key for T1

Problem

- 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

Problem

- Find keys for the universal table
- Create a minimal cover FD set
- Create a set of 3NF tables
- If any of the tables are not in BCNF decompose them into BCNF tables

Problem

- Universal table {A,B,C,D,E,G,H,K,L,M}
- FDs
  - ABE -> CK
  - AB -> D
  - C -> BE
  - EG -> DHK
  - D -> L
  - DL -> EK
  - KL -> DM

Problem

- Find keys for the universal table
- Create a minimal cover FD set
- Create a set of 3NF tables
- If any of the tables are not in BCNF decompose them into BCNF tables