Normalization and Functional Dependencies

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

- Attribute values are atomic
  - This part is assumed for any relational database
  - Object-relational extensions to the relational model might violate 1NF depending on your definition of atomic

- Sometimes 1NF includes the requirement that a table has a primary key
2NF

• A table T is in 2NF
  – If there are no non-trivial dependencies,  $X \rightarrow A$, that lie in T, where X is a proper subset of a key and A is not a prime attribute

• No non-prime attribute is functionally dependent on a proper subset of a key

• Sometimes this is phrased as no partial key dependencies exists in the table
Redundancy and Anomalies

- Consider combining all Library tables into one table
  - What attribute(s) could be the primary key for the table?
- Redundancy
  - The name of an author will appear in many places (once for each loan of a copy written by the author)
- Update Anomaly
  - If the customers name changes it must be changed in many places
- Delete Anomaly
  - If all loans for a copies of books written by an author are deleted all information about the author is lost. Why?
- Insert Anomaly
  - A new author cannot be added to the database until at least one loan for a copy of a book written by the author is added.
- Normalization (3NF and BCNF) reduces redundancy and eliminates the anomalies described above.
Functional Dependencies

• In the following let letters late in the alphabet represent sets of attributes and letters early in the alphabet represent individual attributes.

• Functional Dependencies (X -> A or X -> Y) are constraints on the data that can be entered into the database.

• If the FD, X -> A, holds for a database then if t1 and t2 are tuples that contain the attributes X and attribute A (and possibly other attributes) and if the tuples have the same values for attributes X they must have the same value for attribute A.
Functional Dependencies (FDs)

- FDs can entail or imply other FDs (Armstrong’s Axioms)
  - Reflexivity: if Y is a subset of X then X → Y
  - Augmentation: if X → Y then XZ → YZ
  - Transitivity: if X → Y and Y → Z then X → Z
  - Union: if X → A and X → B then X → AB
  - Decomposition: if X → AB then X → A and X → B

- The closure of a set of FDs, F, is designated by F⁺
- Two FD sets, F and G, are equivalent iff F⁺ = G⁺
- Equivalency of two FD sets can be shown by showing that the FDs in F are implied by the FDs in G and the FDs in G are implied by the FDs in F
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

- \( \text{closure} := X; \) \hspace{1cm} // \text{since} \; X \subseteq X^+

repeat
  \( \text{old} := \text{closure}; \)
  if there is an FD \( Z \rightarrow V \) in \( F \) such that \( Z \subseteq \text{closure} \)
  then \( \text{closure} := \text{closure} \cup V \)
until \( \text{old} = \text{closure} \)

- If \( T \subseteq \text{closure} \) then \( X \rightarrow T \) is implied by \( F \)
Problem

• Let $R = \{A, B, C, D, E, F\}$
• Let the FD set be
  – $ABF \rightarrow C$
  – $CF \rightarrow B$
  – $CD \rightarrow A$
  – $BD \rightarrow AE$
  – $C \rightarrow F$
  – $B \rightarrow F$
• Find the closure of $ABC$
Keys and Super Keys

• A set of attributes, $X$, in a super key for a table $T$ if $X \subseteq T$ and $X \rightarrow T$

• 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
  – $ABF \rightarrow C$
  – $CF \rightarrow B$
  – $CD \rightarrow A$
  – $BD \rightarrow AE$
  – $C \rightarrow F$
  – $B \rightarrow F$
• Is $ABF$ a super key for $R$?
• Is $ABD$ a super key for $R$?
• What attribute must be part of any key for $R$?
3NF

• A table $T$ is in 3NF
  – if for all non-trivial dependencies, $X \rightarrow A$, that lie in $T$, $X$ is a super key or $A$ is a prime attribute

• An FD is a 3NF violator for table $T$
  – if it is a non-trivial dependency, $X \rightarrow A$, that lies in the $T$ where $X$ is not a super key and $A$ is not a prime attribute.

• A prime attribute is an attribute that is part of some key
BCNF

• A table $T$ is in BCNF
  – if for all non-trivial dependencies, $X \rightarrow A$, 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, $X \rightarrow A$, that lies in $T$ where $X$ is not a super key.
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 \( G = G - \{XB \rightarrow A\} \cup \{X \rightarrow A\} \)
  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

\text{old} = H

For each $X \rightarrow A \in H$

if $H$ is equivalent to $H \setminus \{X \rightarrow A\}$ (i.e., $A \in X^+$ where $X^+$ is found using FD set $H \setminus \{X \rightarrow A\}$)

then $H = H \setminus \{X \rightarrow A\}$

until \text{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
  – $ABF \rightarrow C$
  – $CF \rightarrow B$
  – $CD \rightarrow A$
  – $BD \rightarrow AE$
  – $C \rightarrow F$
  – $B \rightarrow F$
• 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
  
  \[ \begin{align*} 
  &\text{ABF} \to C \\
  &\text{CF} \to B \\
  &\text{CD} \to A \\
  &\text{BD} \to AE \\
  &\text{C} \to F \\
  &\text{B} \to F 
  \end{align*} \]

• 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 $T_1 \cap T_2 \rightarrow T_1$ or $T_1 \cap T_2 \rightarrow T_2$
  – A decomposition of T into T1 and T2 is a lossless if for every valid T (valid relative to the FDs) $T = T_1 \text{ Natural Join } T_2$
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
  - $A \rightarrow E$
  - $BE \rightarrow D$
  - $AD \rightarrow BE$
  - $BDH \rightarrow E$
  - $AC \rightarrow E$
  - $F \rightarrow A$
  - $E \rightarrow B$
  - $D \rightarrow H$
  - $BG \rightarrow F$
  - $CD \rightarrow A$
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 $\rightarrow$ CK
  – AB $\rightarrow$ D
  – C $\rightarrow$ BE
  – EG $\rightarrow$ DHK
  – D $\rightarrow$ L
  – DL $\rightarrow$ EK
  – KL $\rightarrow$ 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