Functional Dependencies

1) Definitions
    a. **<u>Functional Dependency</u>** (FD), X→A (X determines A), given any set of tuples with the same value(s) for X then the corresponding A values must be the same
    b. **<u>Superkey</u>**, X→{*all other attributes*}, no requirement to be minimal
    c. **<u>Candidate Key</u>**, a minimal **Superkey**
    d. **<u>Primary Key</u>**, one (arbitrarily) selected **Candidate Key**
2) Rules of Inference
    a. Armstrong's Axioms
        i. **<u>Reflexivity:</u>** XA→A [note: X can be the empty set]
        ii. **<u>Augmentation:</u>** Given X→A then XB→AB
        iii. **<u>Transitivity:</u>** Given X→A and A→B then X→B
    b. Corollaries
        i. **<u>Union:</u>** X→A and X→B then X→AB
        ii. **<u>Decomposition:</u>** X→AB then X→A and X→B
3) **<u>Closure of FD</u>** sets (F^+)
    a. The set of FD, the closure is the set of ALL FDs that can be implied using rules of inference
    b. Usually the closure only asks for non-trivial dependencies
    c. A **Trivial Dependency** is XA→A [note: X can be the empty set]
    d. Algorithm:
        i. Start with set of existing FDs
        ii. Apply rules of inference to determine new dependencies
        iii. Iterate until set does not enlarge
4) **<u>Attribute Closure</u>** (X^+)
    a. The complete set of attributes that can be inferred by X, X→Y
    b. Algorithm:
        i. Start with trivial X→X, so Y={X}
        ii. Loop through all FDs A→B [note: does not need to be F+]
        iii. If A is a subset of Y then add B to Y
        iv. Once a FD is used, it does not need to be considered again
        v. Iterate until set Y does not enlarge
5) **<u>Projection of F on X</u>** (F_X)
    a. Set of FDs, A→B, from F^+, such that all attributes in A and B are in X
6) **<u>Minimal Cover</u>**
    a. Not necessarily unique
    b. Algorithm:
        i. For each FD in F, X→A
        ii. Split into separate FDs such that A is a single attribute (Using corollary ii) and add to G
        iii. Minimize the left side of each FD in G
        iv. Remove all FDs (one-by-one) in G if without it, G^+ still equals F^+

Normal Forms

7) Normal Forms
   a. 1st NF: all attributes are atomic (no sets)
   b. 2nd NF: historical interest, you do not need to know
   c. 3rd NF: eliminates most redundancies
   d. BCNF: eliminates all redundancies
   e. 4NF, 5NF: stricter guarantees than BCNF
8) Decomposition
   a. Given relation R replace with 2 or more relations such that every attribute appears in a least one of the new relations
   b. **Lossless decomposition:** recombination using relational join produces EXACTLY same as pre-decomposition
   c. Decomposition of R into X and Y is lossless iff F^+ contains
      i. $X \cap Y \rightarrow X$ OR
      ii. $X \cap Y \rightarrow Y$
   d. **Dependency Preserving**: If R is decomposed into X and Y then $(F_x \cup F_y)^+ = F^+$
9) BCNF
   a. Algorithm to check:
      i. For each FD in F^+, $X \rightarrow A$
      ii. A is a subset of X (trivial dependency) OR
      iii. X is a superkey for R
      iv. If any FD does not meet either (ii) or (iii) then relation is NOT IN BCNF
   b. As an optimization, (a)(i) could be changed to for each FD in the minimal cover
   c. Algorithm to decompose into BCNF
      i. For each FD in the minimal cover that violates BCNF, $X \rightarrow A$
      ii. Decompose R into R-A and XA
      iii. Guaranteed to be lossless but may not be dependency preserving
10) 3NF
   a. Algorithm to check:
      i. For each FD in F^+, $X \rightarrow A$
      ii. A is a subset of X (trivial dependency) OR
      iii. X is a superkey for R OR
      iv. A is part of a candidate key
      v. If any FD does not meet either (ii), (iii), or (iv) then relation is NOT IN 3NF
   b. Algorithm to decompose into 3NF
      i. Decompose into BCNF
      ii. For each FD, $X \rightarrow A$, in the minimal cover that is NOT preserved
      iii. Add relation XA
   c. Algorithm Two to decompose into 3NF
      i. For each FD in the minimal cover, $X \rightarrow A$
      ii. Add relation XA
      iii. Recombine relations with same key
      iv. If no relation contains a superkey, create one