CSC 370 - Midterm 02: Design Theory for Relational Databases - Study Sheet

Relations and Functional Dependencies (FD's)

Relational Data Model Attributes: Columns (set of attributes). Schema: Relation and attributes (i.e., $R(A_1, ..., A_n)$). Tuples: Unique row of values. Domains: Restriction on Data Types. Keys: $K \subseteq A$, $A \in R$ (e.g., R($\underline{A_1}$, ..., A_n)).
FD A (antecedent) functionally determines B (consequent).

Closure and Keys

Logical Rules Logical Inferences: $A \to B$.
Combining Rule: $A \to B$, $A \to C$; $A \to BC$. Splitting Rule: $A \to BC$; $A \to B$, $A \to C$. Transitivity: $A \to B$, $B \to C$; $A \to C$. Triviality: $A \to B$ if $B \subseteq A$. Semi-Triviality: $A \to B$ and $C = B \cap A$, C can be drop from B.

Closure of An Attribute Set Maximum set of attributes determined by X (i.e., X^+).

(Super)Key of A Relation Let C be the set of attributes of relation R and $A \subseteq C$. A is a superkey iff $\{A\}^+ = C$. A is a key iff A is a superkey and there is no set $A' \in C$ for A' is a superkey. R is a func. from minimal domain A onto range C.

Algorithm Closure of A Set of Attributes
Input: $\{A_1, ..., A_n\}$ and FD's S. **Output**: $\{A_1, ..., A_n\}^+$.
1. Split FD's of S into singletons. **2.** Initialize X to be $\{A_1, ..., A_n\}$. **3.** Repeatedly find FD $B_1, ..., B_m \to C$; B is in the set of attributes X but C is not and add C. **4.** Return when there are no more FD's that contribute to X.

Minimal Bases and Projecting FD's

Closure of A Set of FD's The closure of all possible subsets of the attributes (2^n).

Basis of A Set of FD's Two sets of FD's are equal iff they have the same closing set. Thus, F' is a basis for F if the closure in every subset of attributes is identical in F and F'.

Minimal Basis (MB) Given a set of FD's F, we say that they form a MB iff: **1.** Every FD has a singleton RHS. **2.** If any FD is removed or If we remove any attribute from the LHS of any FD, we no longer have a basis.

Projecting FD's Given a set of FD's and a relation S with attributes C, we can produce a set of FD's for S by **1.** Calculating the closure of every non-empty subset c of C to create a FD: $c \to c^+$. **2.** Simplify the set of $2^{|n|} - 1$ FD's into a MB.

Algorithm Projecting A Set of FD's
Input: Relations R with a set of FD's S and R_1 computed by $R_1 = \pi_L(R)$. **Output**: The set of FD's T that hold in R_1.
1. T is initially empty. **2.** For each set of attributes X; $X \subseteq L$, compute X^+. This computation is performed with respect to the set of FD's S, and may involve attributes that are in the schema of R but not in R_1. Add to T all non-trivial FD's $X \to A$; $A \in X^+ \wedge L$. **3.** Now, T is a basis for the FD's that hold in R_1, but may not be a MB.

Boyce-Codd Normal Form (BCNF)

Decomposing Relations A relation R with attributes $\{A_1, ..., A_n\}$. A decomposition of R splits it into two relations: S with attributes $\{B_1, ..., B_m\}$ and T with attributes $\{C_1, ..., C_k\}$; $A = B \cup C$, $S = \pi_{(B)}(R)$, and $T = \pi_{(C)}(R)$. Note: We can join each tuple of T to a unique tuple of S on the common A's; $A \in T \wedge S$ AND a key for either T or S.

BCNF A relation R with attribute set C is in BCNF iff for every non-trivial FD on R, $\{A_1, ..., A_n\} \to \{B_1, ..., B_m\}$, $\{A_1, ..., A_n\}$ is a superkey of R (i.e., $A^+ = C$).

Algorithm BCNFDecomp(R_0, F_0) (Nondeterministic)
Input: Relation R_0 with attribute set C_0 and FD's F_0.
Output: A decomposition of R_0 with all relations in BCNF.
1. If R_0 is in BCNF, Return R_0. **2.** Select a BCNF violation, $X \to Y$. **3.** Compute X^+. **4.** Let $R_1 := X^+$. **5.** Let $R_2 := (C \setminus X^+) \cup X$. **6.** Project F_0 to get FD for R_1 and R_2 denoted F_1 and F_2. Return BCNFDecomp(R_1, F_1) $\cup$ BCNFDecomp(R_2, F_2).

Third Normal Form (3NF)

3NF A relation R over attributes C with FD's F ($X \to Y$) is in 3NF iff every FD is in BCNF OR every attribute in Y is prime (an attribute that is a member of some key).

Algorithm Synthesis of 3NF Relations (Nondeterministic)
Input: A relation R and a set F of FD's that hold for R.
Output: A decomposition of R with all relations in 3NF. It is a lossless join and preserves all FD's.
1. Find a MB for F, say G. **2.** For each FD $X \to A$ in G, use XA as the schema of one of the relations in the decomposition.
3. If none of the relation schemas from step 2 is a superkey for R add another relation whose schema is a key for R.

Relational Algebra and Constraints

Set-Theoretic Operators: R and S must have identical schemas. Union: Retains all tuples. Intersection: Retains any tuples that are in R and S. Difference: Retains any tuples that only appears in the LHS.

Rename: Obtain $S(B, C, D)$ with $\rho_{S(B,C,D)}(R(A, B, C))$. Project: Apply $\pi_{(B,C)}(R(A, B, C))$ to drop column A and any new duplicate tuples from R. Select: Apply $\sigma_{A=a1}(R(A, B, C))$ to drop all tuples where $A \neq a1$. Cross Product ($\times$): Combine every tuple pairwise. Natural Join ($\bowtie$): Combine only tuples that agree on common columns. Theta Join ($\bowtie_{Condition}$): Combine only tuples for which the specified condition is true (take the product, select the correct tuples).

Constraint A limitation on data. $R = \emptyset$: The value of R must be empty (or $R \subseteq \emptyset$). $R \subseteq S$: Every tuple in the result of R must be in the result of S (or $R - S = \emptyset$).

Referential Integrity Constraint Asserts that a value appearing in one context also appears in another, related context. $\pi_A(R) \subseteq \pi_B(S)$ and $\pi_A(R) - \pi_B(S) = \emptyset$.

Key Constraints Express algebraically the constraint that a certain attribute or set of attributes is a key for a relation.

$$\rho_A(R) \bowtie_{A.a = B.a \wedge (A.b \neq B.b \vee A.c \neq B.c)} \rho_B(R) = \emptyset$$