IDB Lecture 20: Normal Forms

Redundancy Principle don't repeat constrained information in a table. Every FD should define a key.

Boyce Codd Normal Form (BCNF)

"Problems with bad designs are caused by FDs $X \to Y$ where X is not a key."

A relation with FDs F is in **BCNF** if $\forall X \to Y \in F$

1. $Y \subseteq X$ (the FD is trivial), OR
2. X is a key (the closure contains all all attributes in the relation).

- a database is BCNF is all relations are BCNF

Decompositions

Given a set of attributes U and a set of FDs F, a **decomposition** of (U, F) is a set

$$(U_1, F_1), ..., (U_n, F_n)$$

such that $U = \bigcup_{i=1}^{n} U_i$ and F is a set of FDs over U_i

- dont really understand

Criteria for good decompositions

Lossnessness No information is lost

Dependency Preservation no constraints are lost

- formal definitions missed

Projections of FDs

The **projection** of F on $V \subseteq U$ is a subset of the closure containing only the attributes of V.

BCNF Decomposition Algorithm

Algorithm *BCNF-Decomposition(U: {Attribute}, F : {{FD}}) → S : {({Attribute}, {FD})}:*

1. $S := \{(U, F)\}$
2. **while** $\exists (U_i, F_i) \in S$ **not** BCNF:
3. $\quad S[U_i, F_i] :=$ ***decompose(***U_i, F_i)
4. **if** $\exists \{U_i \mid (U_i, F_i) \in S, U_i \subseteq U_j, (U_j, F_j) \in S\}$
5. $\quad$ **remove** $S[U_i, F_i]$
6. **return** S

Algorithm *Decompose(*U_i *: {Attribute},* F_i *: {FD})) -> ({U}, {F}):*

1. find $(X \to Y) \in F$ **not** BCNF
2. V, $Z := C_F(X)$, $U - V$
3. **return** $(V, \pi_V(F))$ **and** $(XZ, \pi_{XZ}(F))$