

Introduction to Functional Dependencies (FDs) Normal Form Schema Design

Cs143 Notes by Carlo Zaniolo

Update Anomalies

- Connected integrity constraints called Functional Dependencies
- Actually with certain FD structures
- Normal forms give simple rules that avoid these bad structures
- Decomposition: a process whereby a relation is replaced by its projections eventually generating NF relations
- The process is also driven by the FDs.

Functional Dependencies (FDs)

$X \rightarrow Y$ (where $X = A_1, \dots, A_n$, $Y = B_1, \dots, B_m$)

For any u_1, u_2 in R ,
if $u_1[X] = u_2[X]$, then $u_1[Y] = u_2[Y]$

No two tuples in R can have the same X values but different Y values. Then $X \rightarrow Y$ is an FD for R

- Special case: No two distinct tuples exist with the same X -values. Then X is a superkey
- X is a superkey for R iff for every attribute A of R , we have $X \rightarrow A$.
- A key is a minimal superkey.

BOYCE-CODD NORMAL FORM (BCNF)

Let a relation R be given and let G be the set of its FDs

Question: is this a good schema—no update anomalies?

According to Boyce and Codd: yes iff it is BCNF:

R is BCNF with respect to G :

iff for every non-trivial $X \rightarrow Y$, X is a key or a superkey

($X \rightarrow Y$ trivial means that Y is a subset of X)

Decomposition into BCNF

- Given a relation R, we find its FDs
- From these we find the keys,
- And then we ask the question: is R BCNF?
- If the answer is yes, we smile.
- If the answer is no: we decompose the original relation into a set of BCNF tables
- The decomposition algorithm is based on the notion of cover, which is based on the formal properties of FDs.

FDs' Properties

Reflexivity: If Y is a nonempty subset of X then $X \rightarrow Y$.

Augmentation: if $X \rightarrow Y$ and X is a subset of Z, then $Z \rightarrow Y$

Transitivity: $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$

This is a complete set of inference rules. Every other valid properties can be derived from these.

E.g. Pseudotransitivity, Additivity.

What is a valid property: one for which there is no counterexample is possible.

Transitive Closure

Transitive closure of a set of FDs G denoted G^+ : consists of the FDs in G plus those can be constructed with the previous rules

(Transitive) Closure of Attributes set X , denoted X^+ :

Start with $X^+ = X$ and repeat until no change in X^+ If there is $Y \rightarrow Z$ and Y is a subset of X^+ , add all attribute in Z to X^+

Cover of G : any subset of G that has the same transitive cover as G

Computation of minimal cover for G

Decomposition

- Take a relation $R(W)$ where $W = X \cup Y \cup Z$ and replace it with:
- $R_1 = \pi_{X \cup Y} R(W)$ and $R_2 = \pi_{X \cup Z} R(W)$

Theorem: If $X \rightarrow Y$ or $X \rightarrow Z$ then $R(W)$ can be reconstructed as the natural join of these two projections (lossless join property)

But while lossless join preserves the information (as needed for queries) it does not preserve the FD constraints (as needed for updates)

Example: $R(A, B, C)$ with $A \rightarrow B, B \rightarrow C, A \rightarrow C$.

Decomposition Algorithm into BCNF

Starting with a given set of FDs G .

Step1. Put G into canonical form G' . G' only contains FDs $X \rightarrow A$ where no $X' \rightarrow A$ holds for some proper subset X' of X .

Step2. For every $X \rightarrow A$ in G' compute X^+ .

If X is not a key then decompose the relation using $X \rightarrow X^+ - X$.

List the FDs in the projection and repeat this process until all relations are BCNF.

(Underscore the keys of the relations so produced)

Schema Directly from ER Diagrams

- One relation for each entity
- One relation for each relationship connecting the keys of the entities at the end of diamonds
- Combine the above if they have the same keys
- Special cases for
 - Weak entities and
 - Inheritance.

Example

$R(A, B, C, D, E, F)$

1. $AB \rightarrow C$
2. $B \rightarrow CD$
3. $BC \rightarrow D$
4. $D \rightarrow EF$
5. $E \rightarrow F$
6. $F \rightarrow E$

Decomposition Algorithm

- Lossless join property is achieved always,
- FD preservation is achieved in all cases but the rare case of Key Breaking dependencies
- E.g. $R(A, B, C)$ where $AB \rightarrow C$ and $C \rightarrow A$. Two keys: AB and BC . $C \rightarrow A$ violates BCNF and the decomposition yields the decomposition into $R(C, A)$ and $R(C, B)$ where the constraint $AB \rightarrow C$ is lost.
- So a decomposition into BCNF is not always feasible. To assure universal feasibility one needs to use Third Normal Form (3NF).

R is BCNF with respect to G :

iff for every non-trivial $X \rightarrow A$, either

- (i) X is a key or
- (ii) is an attribute of some key (which will be broken if we decompose since A will go into one projection and the remaining key attributes into the other)