W23-CSC 430 - 530

Lecture 19

Feb 13th, 2023

Recap: A good decomposition is characterized by
(i) the resultant union of smaller tables should recreate the same attributes as in the original table.
(ii) Reassembly should produce original content; also referred to as the "Lossless Join" property.

Recap: There are three Normal forms discussed so far namely: 1NF, 2NF, 3NF.

§ Boyce Codd (BCNF) {Relational design by decomposition}

Def(i) A relation R is in BCNF w.r.to a set of fd's if, for all fd's of the form $\alpha \to \beta$ where $\alpha, \beta \in R$, at least one of the following holds:

a) $\alpha \to \beta$ is a trivial fd; or
b) α is the superkey of the relation R.

Def(ii) A relation R is in BCNF if whenever a non-trivial fd $\alpha \to \beta$ holds in R, then α is a superkey of R.

Comparisons of Normal Forms:

Normal Forms	Test	
First (1NF)	Relation should have no multivalued attributes	
Second (2NF)	For relations where primary key contains multiple attributes, no fd should have a partial key on the determinant (α)	ensures $\alpha \to \beta$ totalprime non-prime
Third (3NF)	There should be no transitive dependany where the determinant (α) of the fd has a non prime attribute.	gets rid of $\alpha \to \beta$ np np
Boyce-Codd (BCNF)	There should be no non-trivial fd's where the determinant (α) has non-prime attributes.	gets rid of $\alpha \to \beta$ np P

Example: R (<u>A</u> <u>B</u> C)

fds : { $AB \to C$, $C \to B$ }

non-trivial fd. where α has non prime attributes.

R (A B C)

loops back its a problem.

The above relation is in 3NF but not in BCNF

Example: Based on the def 2. Let $A \rightarrow B$ be a non-trivial fd on relation $R(A\,B\,C\,D)$.

R:

A	B	C	D
a →	10	1	20
a →	10	4	50

for $A \rightarrow B$, this holds true

violation: However if R is in BCNF; 'A' would have to be a superkey and ∴ there would be no duplicates in the attribute 'A'.

Algorithm: BCNF decomposition

Input: R & FD's for R.

Output: Decomposition of R into BCNF relations with "lossless join".

step 1: compute keys for R.

step 2: Repeat until all relations are in BCNF:

a) Pick any R' with fd $A \rightarrow B$ that violates BCNF.

b) Decompose R' into $R_1'(A\,B)$ and $R_2'(A, \text{"rest"})$

c) compute closures for R_1' & R_2'

d) compute keys for R_1' & R_2'

Idea: R'

A	B	rest

$A \rightarrow B$

is not prime (BCNF violation)

R_1' | A | B | $\bowtie$ R_2' | A | Rest |

Example: Given $R(A\,B\,C\,D\,E\,F\,G\,H)$

$A \rightarrow BCG$

$G \rightarrow H$

$D \rightarrow EF$

decompose using BCNF.

Solution: Steps: Compute candidate keys of R.

Given fd: $A \rightarrow BCG$ compute closure

$A^+: ABCG$ (not key)

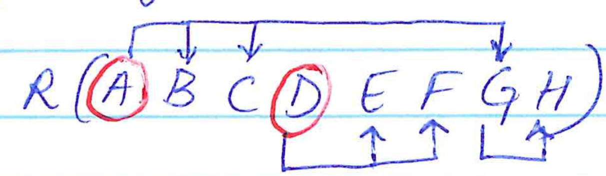
given fd: $G \rightarrow H$ compute closure

$G^+: GH$ (not key)

given fd: $D \rightarrow EF$ compute closure

$D^+: DEF$ (not key)

using edge diagram:



$AD^+: ADBCGEFH$ ⇐ is the candidate key.

Step 2: Check fd's to have key on LHS

$A \rightarrow BCG$ → partial dependency, not in BCNF

$G \rightarrow H$ → not in BCNF

$D \rightarrow EF$ → partial depend. not in BCNF

Repeat Step 3: pick an fd that violates BCNF and decompose R into $R_1(\alpha\beta)$ & $R_2(\alpha, \text{rest})$.

$R(A\ B\ C\ D\ E\ F\ G\ H)$

using $G \to H$

$\to R_1(GH)$ $\quad G^+: GH$ ✓

$\to R_2(GABCDEF)$

using $D \to EF$

$\to R_{21}(DEF)$ $\quad D^+: DEF$ ✓

$\to R_{22}(DGABC)$

using $A \to BCG$

$\to R_{221}(ABCG)$ $\quad A^+: ABCG$ ✓

$\to R_{222}(AD)$

$R(ABCDEFGH)$

$\to R_1(GH)$

$\to R_2(DEF)$

$\to R_3(ABCG)$

$\to R_4(AD)$

in BCNF.