

Ahmet Koray Enis

21 April 2015

BCNF:

1. Eliminates redundancies
2. Always possible to have a BCNF decomposition
3. It is possible to have more than one BCNF for a given relation
4. Lossless
5. Dependency preserving may not be possible

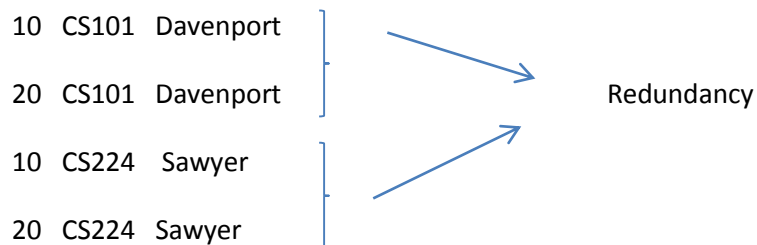
3NF

1. May be unable to eliminate redundancies
2. Always possible
3. Always lossless
4. Always dependency preserving

3NF

If relation R is in 3NF, if for every FD $X \twoheadrightarrow Y$

- $Y \subseteq X$
- X is a superkey of R
- Every $A \in Y$ is a part of some key of R
-
- Example: R(stuNO, Crs, Prof)



StuNo, Crs \twoheadrightarrow Prof

StuNo, Crs = stuNo, Crs, Prof

In 3NF, but not in BCNF because since Prof \twoheadrightarrow Crs is not a superkey of the relation.

LHS contains a key

RHS is a part of a key

Prof \twoheadrightarrow Crs

Example:

$R(\text{ssn}, \text{name}, \text{address}, \text{hobby})$

10, Ali, Bilkent, hiking

10, Ali, Bilkent, karate

20, Ali, Bilkent, karate

Ssn, hobby is the only key. $\text{Ssn} \rightarrow \text{name}$ violates 3NF because name is not a part of a superkey, ssn is not a superkey.

3NF Decomposition

Step 1: Compute minimal cover U of T . The 3NF decomposition is based on U but $U = T$ the same function dependencies will hold.

Note: A binary decomposition of $R \rightarrow R_1$ and R_2 is lossless if and only if it is correct.

$T = \{ABH \rightarrow CK, A \rightarrow D, C \rightarrow E, BCH \rightarrow F, F \rightarrow AD, E \rightarrow F, BH \rightarrow E\}$

$U = \{BH \rightarrow C, BH \rightarrow K, A \rightarrow D, C \rightarrow E, F \rightarrow A, E \rightarrow F\}$

U is a minimal cover of T

Step2: Partition U into sets U_1, U_2, \dots, U_n such that the LHS of all elements of U are the same

$U_1 = \{BH \rightarrow C, BH \rightarrow K\}$

$U_2 = \{A \rightarrow D\}$

$U_3 = \{C \rightarrow E\}$

$U_4 = \{F \rightarrow A\}$

$U_5 = \{E \rightarrow F\}$

Step3: For each U_i form a schema

$R_i = (R_i, U_i) \rightarrow \text{set of dependencies}$



Set of attributes

Where R_i contains all attributes mentioned in U_i . Each FD of U will be in some R_i . Hence, decomposition is dependency preserving.

$R_1(BHCK; BH \twoheadrightarrow C, BH \twoheadrightarrow K)$

$R_2(AD; A \twoheadrightarrow D)$

$R_3(CE; C \twoheadrightarrow E)$

$R_4(FA; F \twoheadrightarrow A)$

$R_5(EF; E \twoheadrightarrow F)$

Step4: If R_i is a superkey of R , add schema $R_o (R_o; \{ \})$ where R_o is a key of R .

$R_o(BGH, \{ \})$

- a. Note that R_o might be needed when not all attributes are necessarily contained in

$R_1 \cup R_2 \cup \dots \cup R_n$

- A missing attribute must be a part of all keys. (Since it is not in any FD of U , deriving/obtaining a key constraint of U involves the augmentation axiom)
- b. R_o might be needed even if all attributes are accounted in $R_1 \cup R_2 \cup \dots \cup R_n$

Example 1: $\{ A, B, C, D; \{ A \twoheadrightarrow B, C \twoheadrightarrow D \} \}$

$AC = ABCD$ is the key

Step3: $R_1: (AB; A \twoheadrightarrow B)$

$R_2: (CD; C \twoheadrightarrow D)$

Step4: $R_o(AC, \{ \})$

Add this for lossless decomposition for losslessness.