Ahmet Koray Enis

21 April 2015

<u>BCNF:</u>

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

<u>3NF</u>

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

<u>3NF</u>

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

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

10	CS101	Davenport
20	CS101	Davenport
10	CS224	Sawyer
20	CS224	Sawyer

Redundancy

StuNo,Crs $\rightarrow$ Prof

StuNo,Crs = stuNo,Crs,Prof

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

LHS contains a key

RHS is a part of a key

Prof $\rightarrow$ Crs

Example:

R(ssn, name , address, hobby)

10, Ali, Bilkent, hiking

10, Ali, Bilkent, karate

20, Ali, Bilkent, karate

Ssn, hobby is the only key. Ssn ⇒ name violates 3NF because name is not a part of a superkey, ssn is not a superkey.

<u>3NF Decomposition</u>

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 ⇒ R1 and R2 is lossless if and only if it is correct.

T={ABH ⇒ CK, A ⇒ D, C ⇒ E, BCH ⇒ F, F ⇒ AD, E ⇒ F,BH ⇒ E}

U={ BH ⇒ C, BH ⇒ K, A ⇒ D, C ⇒ E, F ⇒ A, E ⇒ F}

U is a minimal cover of T

Step2: Partition U into sets U1,U2,…Un such that the LHS of all elements of U iare the same

U1={BH ⇒ C, BH ⇒ K}

U2={A ⇒ D}

U3={C ⇒ E}

U4={F ⇒ A}

U5={E ⇒ F}

Step3: For each Ui form a schema

Ri=(Ri,Ui) → set of dependencies
↓
Set of attributes

Where Ri contains all attributes mentioned in Ui. Each FD uf U will be in some Ri. Hence, decomposition is dependency preserving.

R1(BHCK; BH $\rightarrow$ C, BH $\rightarrow$ K)

R2(AD; A $\rightarrow$ D)

R3(CE; C $\rightarrow$ E)

R4(FA; F $\rightarrow$ A)

R5(EF; E $\rightarrow$ F)

Step4: If Ri is a superkeyof R, add schema Ro (Ro; { }) where Ro is a key of R.

Ro(BGH,{ })

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

R1 U R2 U …. Rn

- 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. Ro might be needed even if all attributes are accounted in R1 U R2 U …. Rn

Example 1: (A, B, C, D; { A $\rightarrow$ B, C $\rightarrow$ D})

AC = ABCD is the key

Step3: R1: (AB; A $\rightarrow$ B)

R2: (CD; C $\rightarrow$ D)

Step4: Ro(AC, { })

Add this for lossless decomposition for losslessness.