Course Name: **A Level (1st Sem)** Subject : **Introduction to DBMS**

Topic: **Database Normalization – Second Normal Form (Part 4)** Date: **18-May-2020**

Database Normalization – Second Normal Form (2NF)

Second Normal Form (2NF)

A relation (table) is said to in 2NF:

- **If it is in 1NF.**
- **Relation must not contain any partial dependency.**

"If non prime attribute(s) is/are dependent on part of the candidate key, then it is called partial dependency."

Or

"No non prime attribute is dependent on the proper subset of prime attributes of table."

Prime Attributes:

All those attributes in the table which are part of the candidate key are known prime attributes.

Non-prime Attributes:

All those attributes which are not parts of candidate key attributes (i.e. except the attributes of candidate key) are known as non-prime attributes.

No partial dependency is allowed in 2NF

If x → y holds in any relation, then there should not be any proper subset z of x for which z → a also holds; where x is prime attribute(s) and y, a is non prime attribute(s).

Example:

Suppose R(A B C D) is relational schema and set of functional dependency :

F: AB → C

B → D

Find out the relation R is in 2NF or not? If not decompose it in 2NF.

Solution:

AB^+ is candidate key in above table because AB^+ = ABCD

(The closure of AB contains all the attributes of R)

AB – Prime attribute (because AB are the part of the candidate key)

CD – Non prime attribute (because CD are not the part of the candidate key)

Now, the functional dependency AB → C follows the rule of 2NF,

But **functional dependency B → D violates the rule of 2NF, because attribute B which is prime attribute (part of the candidate key) is determining the non prime attribute D.** It is partial dependency and this type of partial dependency is not allowed in 2NF.

Therefore, to convert the relation R(A B C D) in 2NF, It is divided into two relations R1, R2 as following:

R1 (A B C)

R2(B D)

In R1 (A B C), AB is candidate key and, since AB → C holds.

In R2 (B D), B is candidate key since B → D holds.

Now R1 and R2 are following the rules of 2NF.

The B → D violates the rule of 2NF in R, so BD is kept in separate table.

Exercise:

1. Suppose a relational schema R (A B C D E), and

FDs: AB → C

D → E

Check out the relation R is in 2NF or not? If not decompose it in 2NF.