

Normalization

SoS in Computer Science Application

PGDCA 203 : DBMS

Presented By
Shruti Dubey

Normalization in DBMS

- *Normalization* is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly. Let's discuss about anomalies first then we will discuss normal forms with examples.

Type of Normalization

Here are the most commonly used normal forms:

- First normal form(1NF)
- Second normal form(2NF)
- Third normal form(3NF)

First normal form (1NF)

First Normal Form is defined in the definition of relations (tables) itself. This rule defines that all the attributes in a relation must have atomic domains. The values in an atomic domain are indivisible units.

We re-arrange the relation (table) to convert it to First Normal Form.

Each attribute must contain only a single value from its pre-defined domain.

Course	Content
Programming	Java, c++
Web	HTML, PHP, ASP

Course	Content
Programming	Java
Programming	c++
Web	HTML
Web	PHP
Web	ASP

Second Normal Form (2NF)

Before we learn about the second normal form, we need to understand the following □

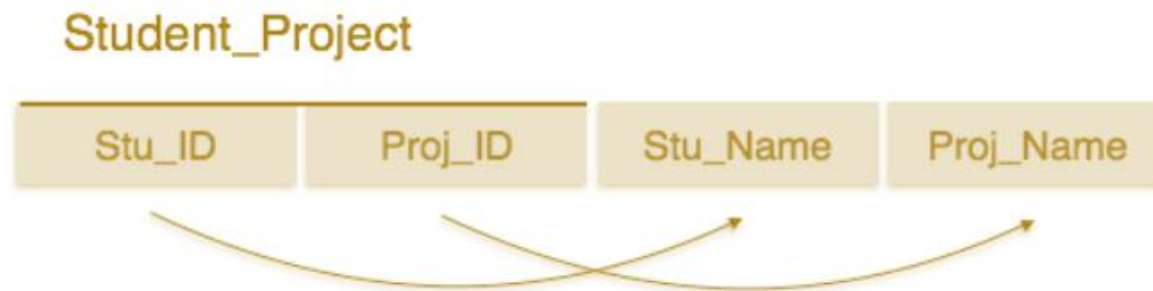
- *Prime attribute* □ An attribute, which is a part of the candidate key, is known as a prime attribute.
- *Non-prime attribute* □ An attribute, which is not a part of the prime key, is said to be a non-prime attribute.

2NF Continued..

- If we follow second normal form, then every non-prime attribute should be fully functionally dependent on prime key attribute. That is, if $X \twoheadrightarrow A$ holds, then there should not be any proper subset Y of X , for which $Y \twoheadrightarrow A$ also holds true.

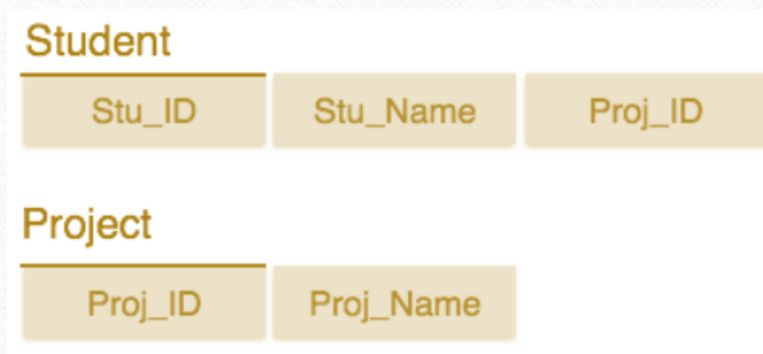
2NF Continued..

- We see here in Student_Project relation that the prime key attributes are Stu_ID and Proj_ID. According to the rule, non-key attributes, i.e. Stu_Name and Proj_Name must be dependent upon both and not on any of the prime key attribute individually.



2NF Continued..

we find that Stu_Name can be identified by Stu_ID and Proj_Name can be identified by Proj_ID independently. This is called *partial dependency*, which is not allowed in Second Normal Form.



We broke the relation in two as depicted in the above picture. So there exists no partial dependency.

Third Normal Form (3NF)

For a relation to be in Third Normal Form, it must be in Second Normal form and the following must satisfy.

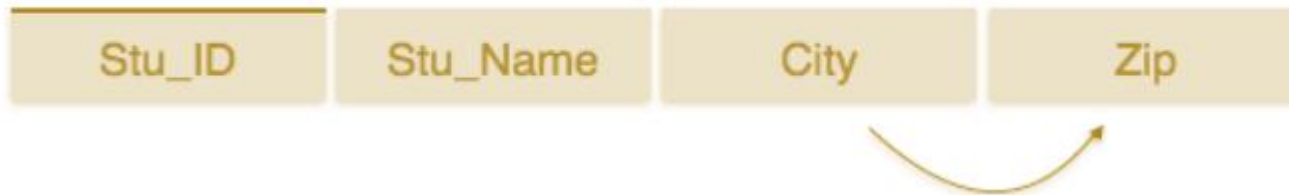
No non-prime attribute is transitively dependent on prime key attribute.

For any non-trivial functional dependency, $X \twoheadrightarrow A$, then either $\twoheadrightarrow X$ is a superkey or A is prime attribute.

3NF Continued..

We find that in the above Student_detail relation, Stu_ID is the key and only prime key attribute. We find that City can be identified by Stu_ID as well as Zip itself. Neither Zip is a superkey nor is City a prime attribute. Additionally, $\text{Stu_ID} \twoheadrightarrow \text{Zip} \twoheadrightarrow \text{City}$, so there exist *transitive dependency*.

Student_Detail



3NF Continued..

- To bring this relation into third normal form, we break the relation into two relations as follows:

Student_Detail

Stu_ID

Stu_Name

Zip

ZipCodes

Zip

City