

## 2NF and 3NF

### 2NF (Normal Form):

It should be 1NF and meet all the rules, and every non-key attribute is fully dependent on primary key. The second normal form eliminates partial dependencies on primary keys.

Example:

Suppose a college wants to store the data of a students and their project details with their student ids. They create a table that looks like this as shown below.

<StudentProjectDetails>

StudentID	ProjectID	StudentName	ProjectName
S1	P2	Raju	Deep learning
S2	P3	Ravi	Face detection
S3	P4	Sai	IoT Devices
S4	P5	shaik	Cloud Deployment

The prime key attributes are StudentID and ProjectID.

The non-prime attributes are StudentName and ProjectName.

The non-prime attributes should be functionally dependent on a part of a candidate key, to be partial dependent.

The StudentName can be determined by StudentID, which makes the relation partial dependent.

The ProjectName can be determined by ProjectID, which makes the relation partial dependent.

The <StudentProjectDetails> relation violates the 2NF in Normalization and is considered as bad database design.

To remove Partial Dependency and violation on 2NF, we can break it into two tables as shown below.

<StudentInfo>

StudentID	ProjectID	StudentName
S1	P2	Ravi
S2	P3	Raju
S3	P4	Sai
S4	P5	shak

<ProjectInfo>

ProjectID	ProjectName
P2	Deep learning
P3	Face detection
P4	IoT Devices
P5	Cloud Deployment

Now the relation is in 2<sup>nd</sup> Normal form of Database Normalization.

3NF (Normal Form):

A database is in third normal form if it satisfies the second normal form (2NF) and should not have transitive functional dependency.

By transitive functional dependency, we mean we have the following relationships in the table: A is functionally dependent on B. ( $A \twoheadrightarrow B$ )

B is functionally dependent on C. ( $B \twoheadrightarrow C$ )

In this case, C is transitively dependent on A via B. ( $A \twoheadrightarrow C$ )

Example:

Table\_Book

BookID	GenreID	GenreType	Price
1	1	General Knowledge	25.99
2	2	Stories	50.00
3	1	General Knowledge	25.99
4	3	Travel	12.88
5	2	Stories	50.00

In the table, [BookID] determines [GenreID], and [GenreID] determines [GenreType].

Therefore, [BookID] determines [GenreType] via [GenreID] and we have transitive functional dependency, and this structure does not satisfy third normal form (3NF).

To bring this table to third normal form (3NF), we split the table into two as follows:

Table\_Book

BookID	GenreID	Price
1	1	25.99
2	2	50.00
3	1	25.99
4	3	12.88
5	2	50.00

Table\_Genre

GenreID	GenreType
1	General Knowledge
2	Stories
3	Travel

Now all the non-key attributes are fully functional dependent only on the primary key. In [Table\_Book], both [GenreID] and [Price] are only dependent on [BookID]. In the [Table\_Genre], [GenreType] is only dependent on [GenreID].