

# Database Normalization

## 1. First Normal Form (1NF)

**Definition:** A table is in **First Normal Form (1NF)** if:

- All the data values are atomic, meaning each column contains only indivisible values.
- Each record (row) in the table is unique, identified by a primary key.
- Each column contains only one type of data.

**Characteristics:**

- **Atomic Values:** No repeating groups or arrays.
- **Uniqueness:** No duplicate rows; each row must have a unique identifier, typically a primary key.

**Example:** Consider a table that stores information about students and their courses:

StudentID	Name	Courses
1	Alice	Math, Science
2	Bob	History, English

This table is not in 1NF because the "Courses" column contains multiple values (Math, Science). To convert this into 1NF, we should split these values into separate rows:

StudentID	Name	Course
1	Alice	Math
1	Alice	Science
2	Bob	History
2	Bob	English

Now, each column contains atomic values, and the table is in 1NF.

## 2. Second Normal Form (2NF)

**Definition:** A table is in **Second Normal Form (2NF)** if:

- It is already in 1NF.
- All non-key attributes are fully dependent on the entire primary key.

**Characteristics:**

- **No Partial Dependency:** Non-key attributes must depend on the entire composite primary key (if a composite key exists), not just a part of it.

**Example:** Consider the following table:

StudentID	Course	Instructor
1	Math	Dr. Smith
1	Science	Dr. Brown
2	History	Dr. White
2	English	Dr. Green

If the primary key is a composite key (**StudentID**, **Course**), this table is in 1NF, but not in 2NF. The **Instructor** column depends only on the **Course**, not on the **StudentID**. This is a partial dependency.

To convert it into 2NF, split the table into two tables:

**Students-Courses Table:**

StudentID	Course
1	Math
1	Science
2	History
2	English

**Courses-Instructors Table:**

Course	Instructor
Math	Dr. Smith
Science	Dr. Brown

History     Dr. White

English     Dr. Green

Now, both tables are in 2NF because all non-key attributes are fully dependent on the primary key.

### 3. Third Normal Form (3NF)

**Definition:** A table is in **Third Normal Form (3NF)** if:

- It is already in 2NF.
- All the attributes are functionally dependent only on the primary key, meaning there are no transitive dependencies.

**Characteristics:**

- **No Transitive Dependency:** Non-key attributes should not depend on other non-key attributes.

**Example:** Consider the following table:

StudentID	Course	Instructor	Department
1	Math	Dr. Smith	Mathematics
1	Science	Dr. Brown	Science
2	History	Dr. White	History
2	English	Dr. Green	Humanities

This table is in 2NF, but not in 3NF. The **Department** column depends on the **Instructor**, which is a non-key attribute, creating a transitive dependency.

To convert it into 3NF, split the table as follows:

**Students-Courses Table:**

StudentID	Course	Instructor
1	Math	Dr. Smith
1	Science	Dr. Brown
2	History	Dr. White

**Instructors-Departments Table:**

<b>Instructor</b>	<b>Department</b>
Dr. Smith	Mathematics
Dr. Brown	Science
Dr. White	History
Dr. Green	Humanities

Now, the tables are in 3NF because all attributes depend only on the primary key, and there are no transitive dependencies.

**Summary**

- **1NF**: Ensures that each column contains atomic values and each row is unique.
- **2NF**: Ensures that all non-key attributes are fully dependent on the entire primary key.
- **3NF**: Ensures that all attributes are only dependent on the primary key, eliminating transitive dependencies.