First Normal Form (1NF)

Last Updated : 09 Sep, 2025

First Normal Form (1NF) ensures that the structure of a database table is organized in a way that makes it easier to manage and query.

- A relation is in first normal form if every attribute in that relation is single-valued attribute or it does not contain any composite or multi-valued attribute.
- It is the first and essential step in to reduce redundancy, improve data integrity and reducing anomalies in relational database design.

A relation (table) is said to be in First Normal Form (1NF) if:

- All the attributes (columns) contain only atomic (indivisible) values.
- Each column contains values of a single type.
- Each record (row) is unique, meaning it can be identified by a primary key.
- There are no repeating groups or arrays in any row.

Rules for First Normal Form (1NF) in DBMS

To follow the First Normal Form (1NF) in a database, these simple rules must be followed:

Every Column Should Have Single Values

Each column in a table must contain only one value in a cell. No cell should hold multiple values. If a cell contains more than one value, the table does not follow 1NF.

- **Example:** A table with columns like [Writer 1], [Writer 2], and [Writer 3] for the same book ID is not in 1NF because it repeats the same type of information (writers). Instead, all writers should be listed in separate rows.

All Values in a Column Should Be of the Same Type

Each column must store the same type of data. You cannot mix different types of information in the same column.

- **Example:** If a column is meant for dates of birth (DOB), you cannot use it to store names. Each type of information should have its own column.

Every Column Must Have a Unique Name

Each column in the table must have a unique name. This avoids confusion when retrieving, updating, or adding data.

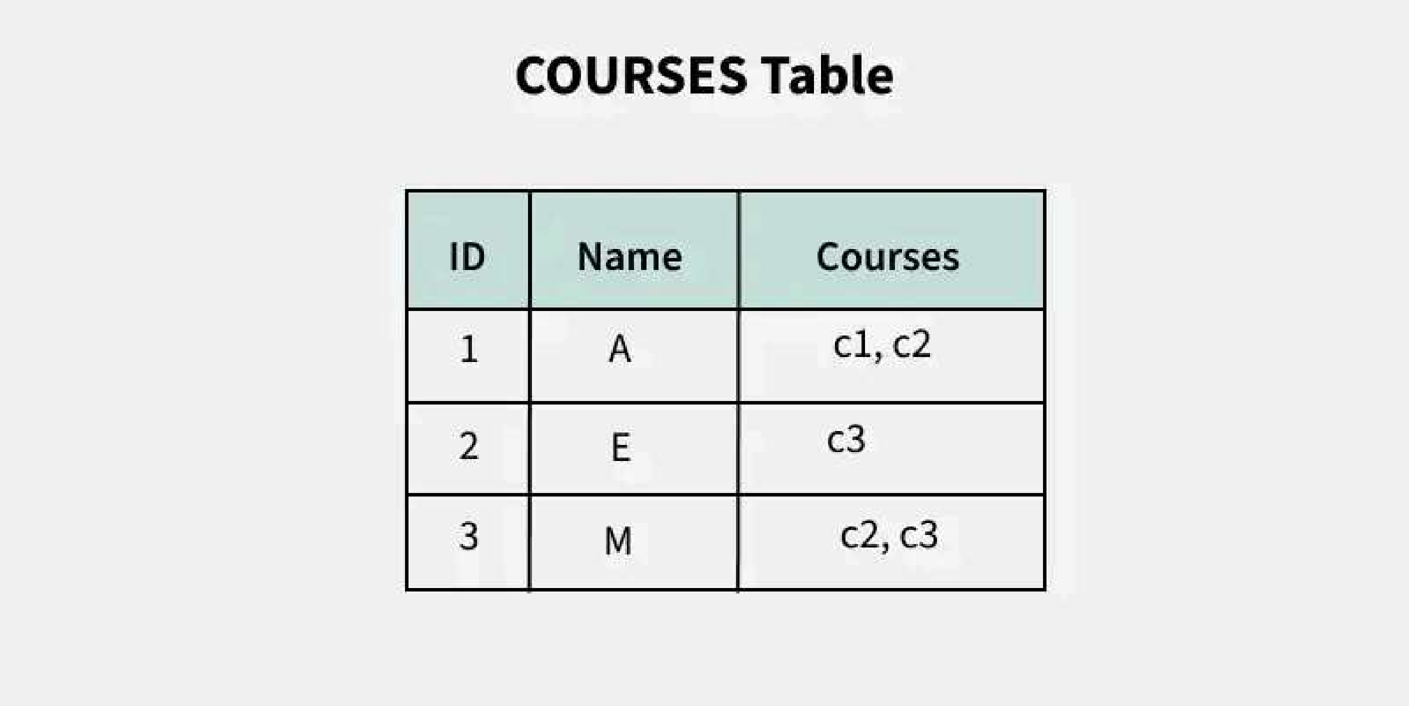
- **Example:** If two columns have the same name, the database system may not know which one to use.

The Order of Data Doesn’t Matter

In 1NF, the order in which data is stored in a table doesn’t affect how the table works. You can organize the rows in any way without breaking the rules.

Example:

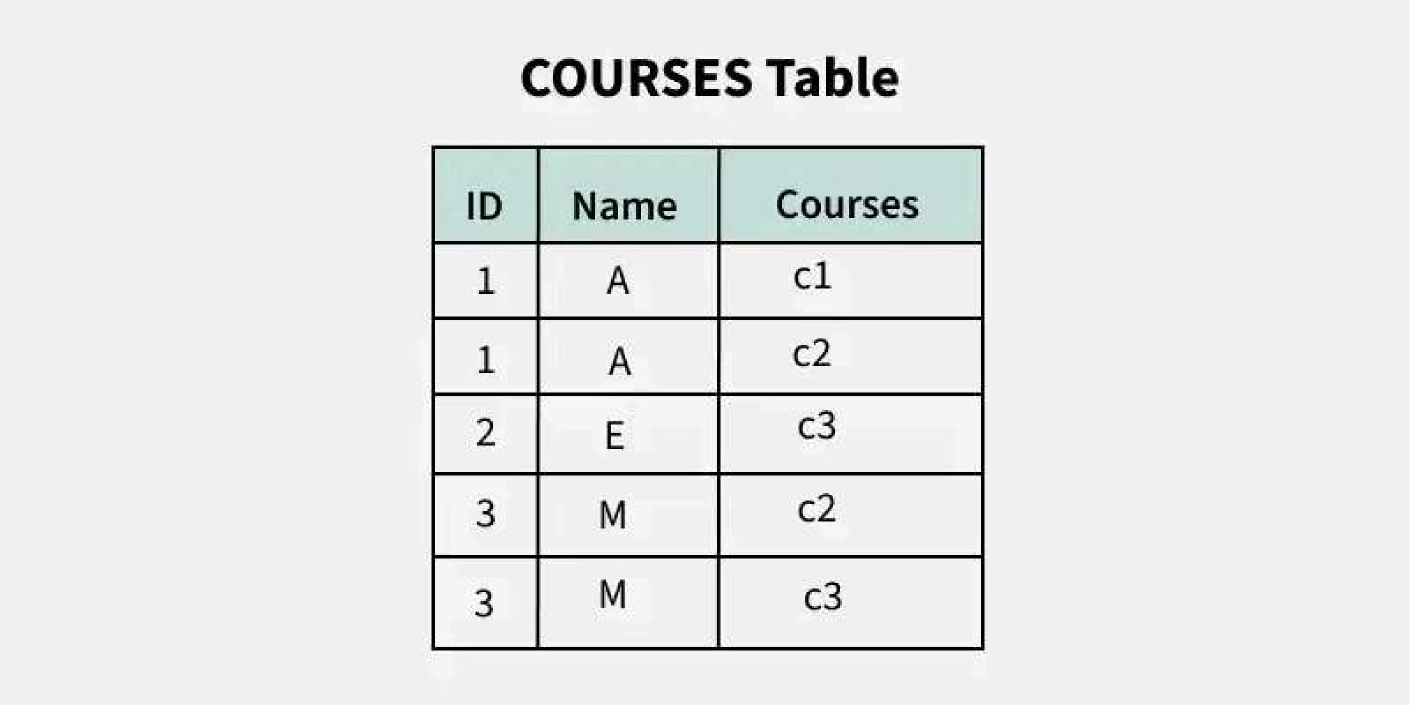
Consider the below COURSES Relation :



COURSES Table

ID	Name	Courses
1	A	c1, c2
2	E	c3
3	M	c2, c3

In the above table, Courses has a multi-valued attribute, so it is not in 1NF. To make the table in 1NF we have to remove the multivalued attributes from the table as given below:



COURSES Table

ID	Name	Courses
1	A	c1
1	A	c2
2	E	c3
3	M	c2
3	M	c3

1NF

Now the table is in 1NF as there is no multi-valued attribute present in the table.