# NORMALISATION

**First normal form (1NF)**

As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.

**Example**: Suppose a video-hosting site wants to store the names and contact details of its user. It creates a table that looks like this:

|  |  |  |  |
| --- | --- | --- | --- |
| user\_id | user\_name | Email\_id | user\_mobile |
| U\_01 | Monish De | monishde@gmail.com | 8912312390 |
| U\_02 | Bhawana | [bhawana@gmail.com](mailto:bhawana@gmail.com) | 8812121212  9900012222 |
| U\_03 | Dikshika | [dikshika3@gmail.com](mailto:dikshika3@gmail.com) | 7778881212 |
| U\_04 | Shreya | [Shreya13@gmail.com](mailto:Shreya13@gmail.com) | 9990000123  8123450987 |

Two customers (Bhawana & Shreya) are having two mobile numbers so the company stored them in the same field as you can see in the table above.

This table is **not in 1NF**as the rule says “each attribute of a table must have atomic (single) values”, the emp\_mobile values for employees Jon & Lester violates that rule.

To make the table complies with 1NF we should have the data like this:

|  |  |  |  |
| --- | --- | --- | --- |
| User\_id | user\_name | Email | user\_mobile |
| U\_01 | Monish De | monishde@gmail.com | 8912312390 |
| U\_02 | Bhawana | [bhawana@gmail.com](mailto:bhawana@gmail.com) | 8812121212 |
| U\_02 | Bhawana | [bhawana@gmail.com](mailto:bhawana@gmail.com) | 9900012222 |
| U\_03 | Dikshika | [dikshika3@gmail.com](mailto:dikshika3@gmail.com) | 7778881212 |
| U\_04 | Shreya | [shreya13@gmail.com](mailto:shreya13@gmail.com) | 9990000123 |
| U\_04 | Shreya | [shreya13@gmail.com](mailto:shreya13@gmail.com) | 8123450987 |

**Second normal form (2NF)**

A table is said to be in 2NF if both the following conditions hold:

* Table is in 1NF (First normal form)
* No non-prime attribute is dependent on the proper subset of any candidate key of table.

An attribute that is not part of any candidate key is known as non-prime attribute.

**Example**: Suppose a video site wants to store the data of user and their roles as uploader and viewer. They create a table that looks like this: Since a user can be video uplaoder or viewer or both, the table can have multiple rows for a user

|  |  |  |
| --- | --- | --- |
| User\_id | Role | user\_age |
| 01 | Uploader | 38 |
| 01 | Viewer | 38 |
| 02 | Viewer | 38 |
| 03 | Uploader | 40 |
| 03 | Viewer | 40 |

**Candidate Keys**: {user\_id,user\_role }  
**Non prime attribute**: user\_age

The table is in 1 NF because each attribute has atomic values. However, it is not in 2NF because non prime attribute user\_age is dependent on user\_id alone which is a proper subset of candidate key. This violates the rule for 2NF as the rule says “**no** non-prime attribute is dependent on the proper subset of any candidate key of the table”.

To make the table complies with 2NF we can break it in two tables like this:  
**user\_details table:**

|  |  |
| --- | --- |
| user\_id | user\_age |
| 111 | 38 |
| 222 | 38 |
| 333 | 40 |

**user\_role table:**

|  |  |
| --- | --- |
| user\_id | Role |
| 01 | uploader |
| 01 | viewer |
| 02 | viewer |
| 03 | Uploader |
| 03 | Viewer |

Now the tables comply with Second normal form (2NF).