**Rules of Functional Dependencies**

Below given are the Three most important rules for Functional Dependency:

* Reflexive rule –. If X is a set of attributes and Y is\_subset\_of X, then X holds a value of Y.
* Augmentation rule: When x -> y holds, and c is attribute set, then ac -> bc also holds. That is adding attributes which do not change the basic dependencies.
* Transitivity rule: This rule is very much similar to the transitive rule in algebra if x -> y holds and y -> z holds, then x -> z also holds. X -> y is called as functionally that determines y.

**Advantages of Functional Dependency**

* Functional Dependency avoids data redundancy. Therefore same data do not repeat at multiple locations in that database
* It helps you to maintain the quality of data in the database
* It helps you to defined meanings and constraints of databases
* It helps you to identify bad designs
* It helps you to find the facts regarding the database design

**Types of Functional Dependencies**

* **Multivalued dependency:**
* **Trivial functional dependency**:
* **Non-trivial functional dependency**:
* **Transitive dependency:**

Normalization is used for mainly two purposes,

* Eliminating redundant (useless) data.
* Ensuring data dependencies make sense i.e data is logically stored.

# First Normal Form (1NF)

* A relation will be 1NF if it contains an atomic value.
* It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attribute.
* First normal form disallows the multi-valued attribute, composite attribute, and their combinations.

**Example:** Relation EMPLOYEE is not in 1NF because of multi-valued attribute EMP\_PHONE.

**EMPLOYEE table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **EMP\_ID** | **EMP\_NAME** | **EMP\_PHONE** | **EMP\_STATE** |
| 14 | John | 7272826385, 9064738238 | UP |
| 20 | Harry | 8574783832 | Bihar |
| 12 | Sam | 7390372389, 8589830302 | Punjab |

The decomposition of the EMPLOYEE table into 1NF has been shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| **EMP\_ID** | **EMP\_NAME** | **EMP\_PHONE** | **EMP\_STATE** |
| 14 | John | 7272826385 | UP |
| 14 | John | 9064738238 | UP |
| 20 | Harry | 8574783832 | Bihar |
| 12 | Sam | 7390372389 | Punjab |
| 12 | Sam | 8589830302 | Punjab |

The First normal form simply says that each cell of a table should contain exactly one value.

| **Instructor's name** | **Course code** |
| --- | --- |
| Prof. George | (CS101, CS154) |
| Prof. Atkins | (CS152) |

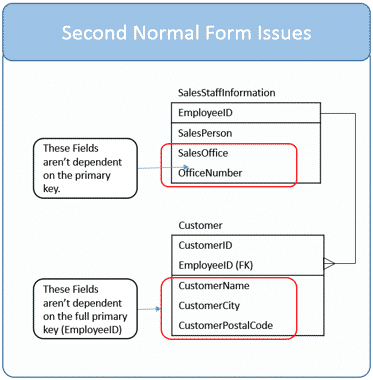
| **Instructor's name** | **Course code** |
| --- | --- |
| Prof. George | CS101 |
| Prof. George | CS154 |
| Prof. Atkins | CS152 |

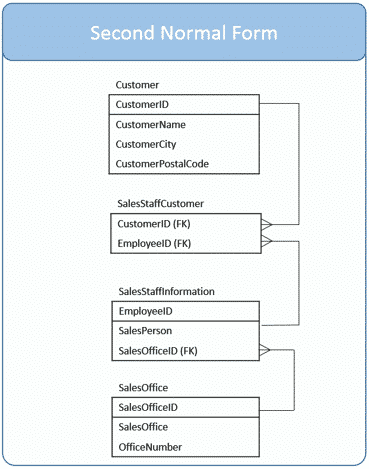
This way, if we want to edit some information related to CS101, we do not have to touch the data corresponding to CS154. Also, observe that each row stores unique information. There is no repetition. This is the First Normal Form.

|  |  |  |
| --- | --- | --- |
| **roll\_no** | **name** | **subject** |
| 101 | Akon | OS, CN |
| 103 | Ckon | Java |
| 102 | Bkon | C, C++ |

|  |  |  |
| --- | --- | --- |
| **roll\_no** | **name** | **subject** |
| 101 | Akon | OS |
| 101 | Akon | CN |
| 103 | Ckon | Java |
| 102 | Bkon | C |
| 102 | Bkon | C++ |

### 1NF Example





# Second Normal Form (2NF)

* In the 2NF, relational must be in 1NF.
* In the second normal form, all non-key attributes are fully functional dependent on the primary key
* In the second normal form, all the attributes will be functionally dependent on the primary key.

**Example:** Let's assume, a school can store the data of teachers and the subjects they teach. In a school, a teacher can teach more than one subject.

**TEACHER table**

|  |  |  |
| --- | --- | --- |
| **TEACHER\_ID** | **SUBJECT** | **TEACHER\_AGE** |
| 25 | Chemistry | 30 |
| 25 | Biology | 30 |
| 47 | English | 35 |
| 83 | Math | 38 |
| 83 | Computer | 38 |

In the given table, non-prime attribute TEACHER\_AGE is dependent on TEACHER\_ID which is a proper subset of a candidate key. That's why it violates the rule for 2NF.

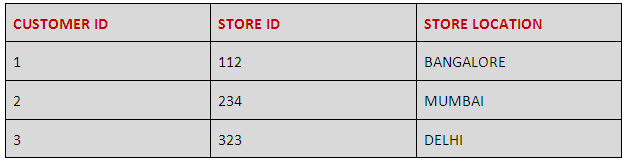
To convert the given table into 2NF, we decompose it into two tables:

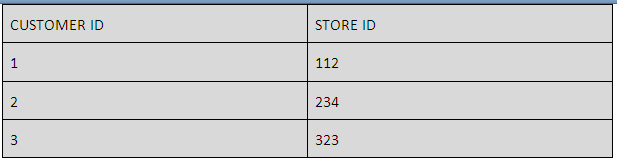
**TEACHER\_DETAIL table:**

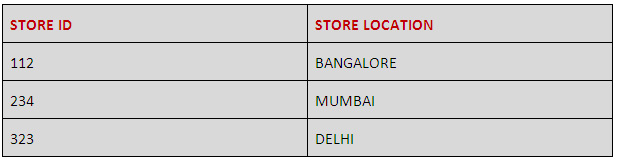
|  |  |
| --- | --- |
| **TEACHER\_ID** | **TEACHER\_AGE** |
| 25 | 30 |
| 47 | 35 |
| 83 | 38 |

**TEACHER\_SUBJECT table:**

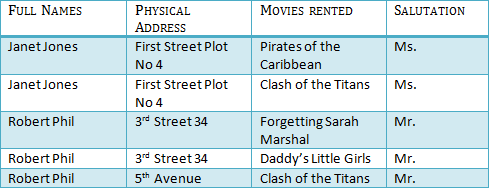
|  |  |
| --- | --- |
| **TEACHER\_ID** | **SUBJECT** |
| 25 | Chemistry |
| 25 | Biology |
| 47 | English |
| 83 | Math |
| 83 | Computer |







The above table is a composite one and has a composite primary key (CUSTOMER ID, STORE ID). The non-key attribute in this arrangement is STORE LOCATION. In the above case, the STORE LOCATION only depends on the STORE ID, which is the sole part of the primary key. Hence the table does not satisfy the second normal form.







### ****Fourth normal form****

A table is said to be in fourth normal form if there is no two or more, independent and multivalued data describing the relevant entity.

### ****Fifth normal form****

A table is in fifth Normal Form if:

* It is in fourth normal form.
* It cannot be subdivided into any smaller tables without losing some form of information.

### Candidate Key

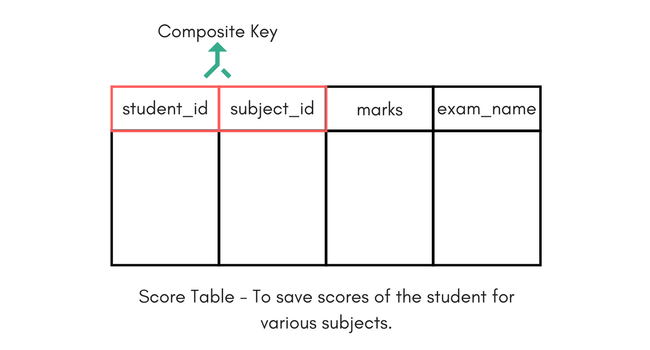
Candidate keys are defined as the minimal set of fields which can uniquely identify each record in a table. It is an attribute or a set of attributes that can act as a Primary Key for a table to uniquely identify each record in that table. There can be more than one candidate key.

For example, student\_id and phone both are candidate keys for table **Student**.

* A candiate key can never be NULL or empty. And its value should be unique.
* There can be more than one candidate keys for a table.
* A candidate key can be a combination of more than one columns(attributes).

### Composite Key

Key that consists of two or more attributes that uniquely identify any record in a table is called **Composite key**. But the attributes which together form the **Composite key** are not a key independently or individually.



In the above picture we have a **Score** table which stores the marks scored by a student in a particular subject.

In this table student\_id and subject\_id together will form the primary key, hence it is a composite key.

### 3NF Example

