NORMALIZATION

A large database defined as a single relation may result in data duplication. This repetition of data may result in:

* Making relations very large.
* It isn't easy to maintain and update data as it would involve searching many records in relation.
* Wastage and poor utilization of disk space and resources.
* The likelihood of errors and inconsistencies increases.

So, to handle these problems, we should analyze and decompose the relations with redundant data into smaller, simpler, and well-structured relations that satisfy desirable properties. Normalization is a process of decomposing relations into relations with fewer attributes.

**What is Normalization?**

* Normalization is the process of organizing the data in the database.
* Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.
* Normalization divides the larger table into smaller and links them using relationships.
* The normal form is used to reduce redundancy from the database table.

**Why do we need Normalization?**

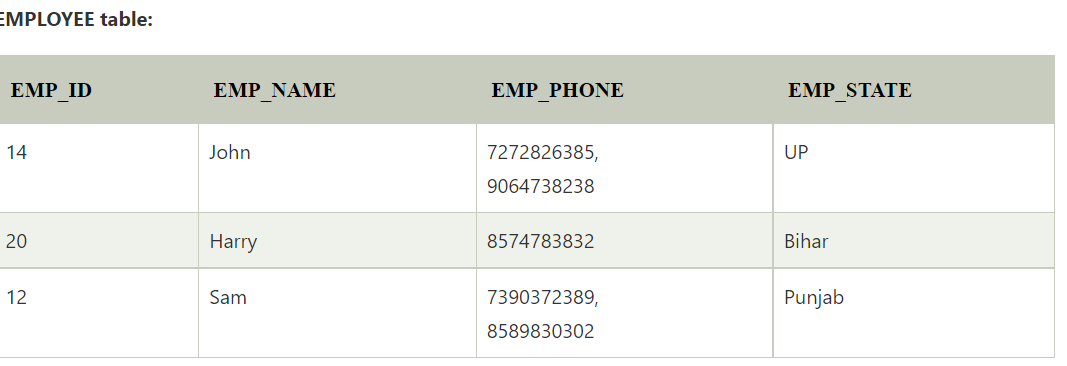
The main reason for normalizing the relations is removing these anomalies. Failure to eliminate anomalies leads to data redundancy and can cause data integrity and other problems as the database grows.

**Data modification anomalies can be categorized into three types:**

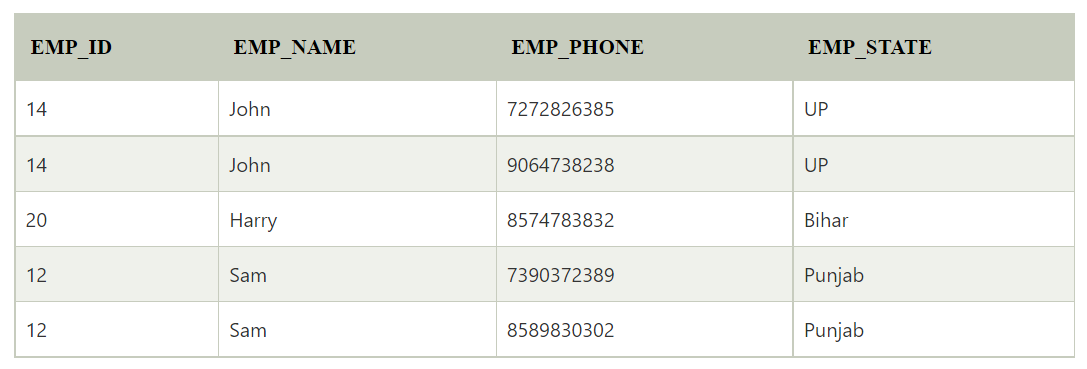
* **Insertion Anomaly:** Insertion Anomaly refers to when one cannot insert a new tuple into a relationship due to lack of data.
* **Deletion Anomaly:** The delete anomaly refers to the situation where the deletion of data results in the unintended loss of some other important data.
* **Updatation Anomaly:** The update anomaly is when an update of a single data value requires multiple rows of data to be updated.

**Normalization Process**

1. 1st NORMAL FORM: By definition, an entity that does not have any repeating columns or data groups can be termed as the First Normal Form. In the First Normal Form, every column is unique.



Above employee table is having duplicate values for EMP\_PHONE column so, the decomposition of the EMPLOYEE table into 1NF has been shown below:

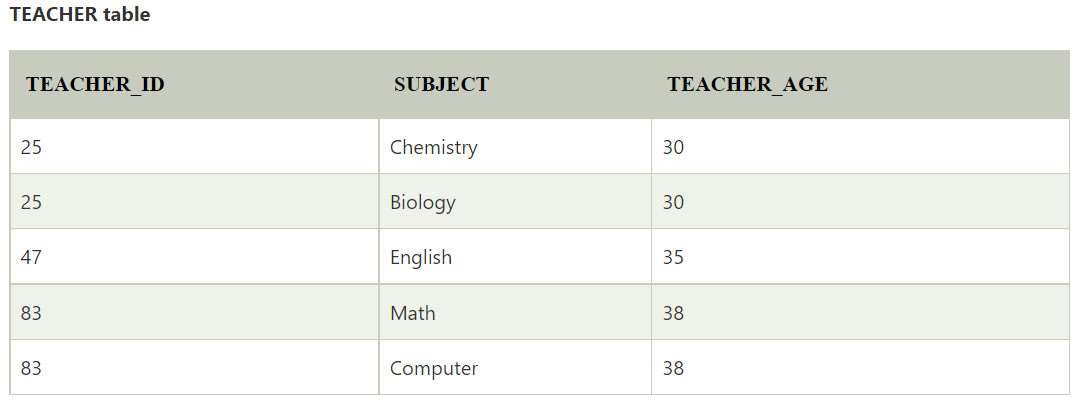


Rules for 1st NF

* 1. Each column should be atomic values.
  2. In each column the value stored should be of the same data type.
  3. Each column should have a unique name.
  4. Order in which the data is stored does not matter.

1. 2nd NORMAL FORM: By definition, an entity that is 1NF and one of its attributes is defined as the primary key and the remaining attributes are dependent on the primary key.
   1. It should be in 1st NF.
   2. And should not have any partial dependencies.

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.



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:

