

What is Normalization?

The goal of relational database design is to generate a set of relation schemes that allow us to store information without any redundant data. It also us to retrieve information easily & efficiently.

Normalization is the name given to the process of simplifying the relationship among data elements in a tuple. Normalization replaces a collection of data in a ~~tuple~~ record structure by another record design which is simple, more predictable, and therefore more manageable.

Need for Normalization -

Normalization reduces redundancy. Redundancy is un-necessary repetition of data. It can cause problems with storage, retrieval & updation of data.

Redundancy can lead to

1) Inconsistencies - errors are more likely to occur when information are repeated.

2) Update anomalies - insertion, modification & deletion of data may cause inconsistencies in a database. Inconsistency occurs when we perform updation or deletion of data in one relation while forgetting to make corresponding changes in other relations.

Such a relation in which the problem of redundancy & anomalies are removed by applying rules of normalization, is called normalized relation. A normalized relation has minimum redundancy with easy & fast accessing mechanism.

A fully normalized record consists of

- i) A primary key that identifies that entity.
- ii) A set of attributes that describe that entity.

The first step towards Normalization is to convert E-R model into tables or relations. Next step is to check the table for data redundancy & if required, change them to non-redundant form. This non-redundant model is then converted to a database definition, which achieves the objective of the database design phase.

First Normal Form - A table is in the 1NF, when it contains no repeating groups. The repeating columns or fields present in an unnormalized table are removed from the table and put into separate table or tables. These tables are dependant on the parent table from which it is derived. The key to these tables must also be a part of the parent table, so that the parent table and the derived tables can be related to each other.

When a table has no repeating groups, it is said to be in 1NF. This means, for each cell in a table, there can be only one value. This value should be atomic in the sense that it can not be decomposed into smaller pieces.

1NF says that
separate the repeating group.

Consider a relation consisting of details of sales made by each salesperson.

Employee no: 999

Employee name: XXXX

Store Branch: XXXXX

Department: XXXXV

Item Number	Item description	Sale Price
999	XXXXXXV	999.99

Un-normalised form (UNF) - Data in its elementary form is un-normalised. Therefore, we need to start with all the elementary data items from our data dictionary.

UNF Un-normalized file for salesperson will be

UNF

Employee no

Employee name

Store Branch

Department

Item number

Item description

Sale price

The first four attributes (i.e. employee no, Employee name, Store Branch, Department) are constant. The remaining three attributes (Item number, Item description, Sale price) contain data that changes and are repeated with

each salesperson. From the above structures, we find that repeated & non-repeated information are kept together, as a result some of the information are repeated unnecessarily and table is not in normalized form. Therefore to separate the repeating and non-repeating group we apply 1NF to the 1NF and we get following files

i) Salesperson with following data items

Employee number *
Employee name
Store branch
Department

ii) Sales with following attributes

Employee number +
Item number *
Item description
Sale price

First Normalization.

- i) The salesperson data file with employee number as primary key.
- ii) Sales file with employee number & item number as composite primary key. These two attributes are added to relate the records in the sales file to the salesperson file. The two attributes (Employee number, Item number) are used together for accessing data.

Second Normal Form (2NF) - This form says

Separate part key dependencies

A table is in 2NF, if all of its non-key fields are fully dependent on the whole key. This means that each field in a table must depend upon the entire key. Those that do not depend upon the combination key, are moved to another table on whose key they depend on. Structures which do not contain cascaded keys are automatically in the 2NF.

2NF makes sure that each non-key attribute