**Normalization**

**Normalization** is a series of steps followed to obtain a database design that allows for efficient access and storage of data. These steps reduce data redundancy and the chances of data becoming inconsistent.

**Contents**

* [1 First Normal Form](http://www.orafaq.com/wiki/Normalization#First_Normal_Form)
* [2 Second Normal Form](http://www.orafaq.com/wiki/Normalization#Second_Normal_Form)
* [3 Third Normal Form](http://www.orafaq.com/wiki/Normalization#Third_Normal_Form)
* [4 Fourth Normal Form](http://www.orafaq.com/wiki/Normalization#Fourth_Normal_Form) or Boyce Codd Normalization
* [5 Fifth Normal Form](http://www.orafaq.com/wiki/Normalization#Fifth_Normal_Form)

**First Normal Form**

First Normal Form eliminates repeating groups by putting each into a separate table and connecting them with a one-to-many relationship.

Two rules follow this definition:

* Each table has a [primary key](http://www.orafaq.com/wiki/Primary_key) made of one or several fields and uniquely identifying each record
* Each field is atomic, it does not contain more than one value

For instance, assuming a table WAGON to follow each wagon in every station.

wagon\_id integer (ex. 101)

description string (ex. 'wagon\_type, empty\_weight, capacity, designer, design\_date')

state string (ex. 'under repair')

timestamp datetime (ex. '22/12/2008 17:37')

station string (ex. 'New York Grand Central')

The [primary key](http://www.orafaq.com/wiki/Primary_key) is (wagon\_id, timestamp).  
This table is not in 1NF because "description" is not atomic. To move it in 1NF we have to split "description" field in its components:

**wagon\_id** integer

wagon\_type string

empty\_weight number

capacity number

designer string

design\_date datetime

state string

timestamp datetime

station string

**Second Normal Form**

Second Normal Form eliminates [functional dependencies](http://www.orafaq.com/wiki/Functional_dependency) on a partial key by putting the fields in a separate table from those that are dependent on the whole key.

In our example, "wagon\_type", "empty\_weight", "capacity"... only depends on "wagon\_id" but not on "timestamp" field of the [primary key](http://www.orafaq.com/wiki/Primary_key), so this table is not in 2NF. In order to reach 2NF, we have to split the table in two in the way that each field of each table depends on all the fields of each [primary key](http://www.orafaq.com/wiki/Primary_key):

**wagon\_id** integer

wagon\_type string

empty\_weight number

capacity number

designer string

design\_date datetime

**wagon\_id**  integer

**timestamp** datetime

state string

station string

**Third Normal Form**

Third Normal Form eliminates [functional dependencies](http://www.orafaq.com/wiki/Functional_dependency) on non-key fields by putting them in a separate table. At this stage, all non-key fields are dependent on the key, the whole key and nothing but the key.

In our example, in the first table it is most likely that "empty\_weight", "capacity", "designer" and "design\_date" depend on "wagon\_type", so we have to split this table in two:

**wagon\_id** integer

**wagon\_type** string

**wagon\_type** string

empty\_weight number

capacity number

designer string

design\_date datetime

Now our example with its 3 tables is in 3NF.

**Fourth Normal Form**

Fourth Normal Form separates independent multi-valued facts stored in one table into separate tables.

In the last table of our example, it is clear that "empty\_weight" and "capacity" are interesting for every day wagon management whereas "designer" and "design\_date" for historical purpose. So we can think they will never be used at the same time and have to split the table in two to fit the 4NF.

***wagon\_type*** string

empty\_weight number

capacity number

***wagon\_type*** string

designer string

design\_date datetime

**Fifth Normal Form**

Fifth Normal Form breaks out data redundancy that is not covered by any of the previous normal forms.