Normalization

Database normalization is the process of organizing the fields and tables of a relational database to minimize redundancy and dependency. Normalization usually involves dividing large tables into smaller (and less redundant) tables and defining relationships between them. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database using the defined relationships.

Normalization of Database

Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a two step process that puts data into tabular form by removing duplicated data from the relation tables.

Normalization is used for mainly two purposes,

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

Problem without Normalization:
Without Normalization, it becomes difficult to handle and update the database, without facing data loss. Insertion, Updation and Deletion Anomalies are very frequent if Database is not normalized.

Normalization Rule:
Normalization rule are divided into following normal form.

1. First Normal Form
2. Second Normal Form
3. Third Normal Form
4. BCNF

First Normal Form (1NF)

A row of data cannot contain repeating group of data i.e. each column must have a unique value. Each row of data must have a unique identifier.

Second Normal Form (2NF)

A table to be normalized to Second Normal Form should meet all the needs of First Normal Form and there must not be any partial dependency of any column on primary key.

It means that for a table that has concatenated primary key, each column in the table that is not part of the primary key must depend upon the entire concatenated key for its existence. If any column depends only on one part of the concatenated key, then the table fails Second normal form.

Third Normal Form (3NF)

Third Normal form applies that every non-prime attribute of table must be dependent on primary key. The transitive functional dependency should be removed from the table. The table must be in Second Normal form.

Boyce and Codd Normal Form (BCNF)

Boyce and Codd Normal Form is a higher version of the Third Normal form. This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF.

What is Normalization? Why should we use it?

Normalization is a database design technique which organizes tables in a manner that reduces redundancy and dependency of data**.**

It divides larger tables to smaller tables and link them using relationships**.**

The inventor of the relational model **Edgar Codd** proposed the theory of normalization with the introduction of FirstNormal Form and he continued to extend theory with Second and Third Normal Form. Later he joined with **Raymond F. Boyce** to develop the theory of Boyce-Codd Normal Form**.**

1NF Rules

- Each table cell should contain single value.
- Each record needs to be unique.

2NF Rules

- Rule 1- Be in 1NF
- Rule 2- Single Column Primary Key

It is clear that we can't move forward to make our simple database in 2nd Normalization form unless we partition the table above.

3NF Rules

- Rule 1- Be in 2NF
- Rule 2- Has no transitive functional dependencies

To move our 2NF table into 3NF we again need to need divide our table.

Boyce-Codd Normal Form (BCNF)

Even when a database is in 3rd Normal Form, still there would be anomalies resulted if it has more than one **Candidate** Key.

Sometimes is BCNF is also referred as **3.5 Normal Form.**

4th Normal Form

If no database table instance contains two or more, independent and multi-valued data describing the relevant entity, then it is in 4th Normal Form.

5th Normal Form

A table is in 5th Normal Form only if it is in 4NF and it cannot be decomposed in to any number of smaller tables without loss of data.

A table is in fifth normal form (5NF) or Project-Join Normal Form (PJNF) if it is in 4NF and it cannot have a lossless decomposition into any number of smaller tables.

6th Normal Form

6th Normal Form is not standardized yet however it is being discussed by database experts for some time. Hopefully we would have clear standardized definition for 6th Normal Form in near future.

The first normal form (1NF)

The first normal form states that a database table is a representation of an entity in the system you are building. Examples of entities are order, customer, booking, hotel, product, etc. Each row in the database table represents one instance of an entity. For example in a customer table each row represents one customer.

Primary key

Rule: each table has a primary key, consisting of the smallest possible number of fields.

Atomicity

Rule: fields are not duplicated in a row and each field contains only one value.

Row order should not matter

Rule: The order of rows in a table should not matter.

The second normal form (2NF)

In order for a database to be normalized according to the second normal form it must first be normalized according to the rules of the first normal form. The second normal deals with data redundancy.

Data redundancy

Rule: Non-primary key fields must be dependent on the primary key.

The third normal form (3NF)

The third normal form deals with **transitive dependencies**. A transitive dependency between database fields exists when the value of a non-key field is determined by the value of another non-key field. For a database to be in the third normal form it must first be in the second normal form.

Transitive dependencies

Rule: there can be no transitive dependencies between fields in a table.

Description of Normalization

Normalization is the process of organizing data in a database. This includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.

First Normal Form

- Eliminate repeating groups in individual tables.
- Create a separate table for each set of related data.
- Identify each set of related data with a primary key.

Do not use multiple fields in a single table to store similar data.

Second Normal Form

- Create separate tables for sets of values that apply to multiple records.
- Relate these tables with a foreign key.

Records should not depend on anything other than a table's primary key (a compound key, if necessary).

Third Normal Form

- Eliminate fields that do not depend on the key.

Values in a record that are not part of that record's key do not belong in the table. In general, any time the contents of a group of fields may apply to more than a single record in the table, consider placing those fields in a separate table.

Other Normalization Forms

Fourth normal form, also called Boyce Codd Normal Form (BCNF), and fifth normal form do exist, but are rarely considered in practical design. Disregarding these rules may result in less than perfect database design, but should not affect functionality.

Normalization

This is a step by step process of removing different kinds of redundancy and anomaly at each step. At each step a specific rule is followed to remove specific kind of impurity in order to give the database a slim and clean look.

First Normal Form (1NF)

A relation is said to be in 1NF if it contains no non-atomic values and each row can provide a unique combination of values. The above table in UNF can be processed to create the following table in 1NF.

Second Normal Form (2NF)

A relation is said to be in 2NF f if it is already in 1NF and each and every attribute fully depends on the primary key of the relation. Speaking inversely, if a table has some attributes which is not dependant on the primary key of that table, and then it is not in 2NF.

Third Normal Form (3NF)

A relation is said to be in 3NF, if it is already in 2NF and there exists no **transitive dependency** in that relation. Speaking inversely, if a table contains transitive dependency, then it is not in 3NF, and the table must be split to bring it into 3NF.

Boyce-Code Normal Form (BCNF)

A relationship is said to be in BCNF if it is already in 3NF and the left hand side of every dependency is a candidate key. A relation which is in 3NF is almost always in BCNF. These could be same situation when a 3NF relation may not be in BCNF the following conditions are found true.

1. The candidate keys are composite.
2. There are more than one candidate keys in the relation.
3. There are some common attributes in the relation.

Fourth Normal Form (4NF)

When attributes in a relation have multi-valued dependency, further Normalization to 4NF and 5NF are required.

Let us first find out what multi-valued dependency is:

A **multi-valued dependency** is a typical kind of dependency in which each and every attribute within a relation depends upon the other, yet none of them is a unique primary key.

Fifth Normal Form (5NF)

These relations still have a problem. While defining the 4NF we mentioned that all the attributes depend upon each other.