First Normal Form

Normalisation

Normalisation is the process used to ensure that the logical design process has produced high-quality structures for the tables within a database. By quality, we mean structures that will perform optimally, and do not contain any duplicate information.

If a logical structure is normalised, then each attribute in the structure will have the same importance. This is useful during the physicalisation process, when the database is actually created using SQL, because indexes can be assigned to each attribute to make searching and retrieval on that attribute faster.

Normalisation is a useful verification tool, which indicates amendments to be made to the logical design process, but does not substitute for the conceptual and logical design. All design stages should be undertaken to ensure that the design covers all aspects of the required system.

The Normalisation Process

The process of normalising a logical model can be summarised as follows:

1. Convert all table structures to First Normal Form
2. Identify Functional Dependencies
3. Convert all table structures from First Normal Form to Second Normal Form

4. Convert all table structures from Second Normal form to Third Normal Form and (very rarely) into Boyce-Codd Normal Form

Decomposition

This is the process of splitting up tables into smaller tables. It should:

- Be lossless – no information should be lost through the normalisation process. The original information must be able to be reconstructed from the normalised data without adding any new data. Querying the decomposed tables should achieve the same result as querying the original tables.
- Preserve dependencies – the relations between the different attributes and tables should not be lost. The new tables must have the same capacity to represent the integrity constraints as the original tables. The user should not be able to add data to the new tables that was not able to be added into the old tables.

First Normal Form

A table (or relation) is said to be in First Normal Form only if it satisfies the following condition:

- **The underlying domains contain simple atomic values**

This means that for every domain (or field) in each row of the table, there is only one value.

An example of a Non-Normalised table for a product ordering system is shown below.

Customer Name	**City**	**Phone**	Product ID 1	Product ID 2	Order Date	Quantity 1	Quantity 2	Sales-person	%Discount1	%Discount2
James Smith	London	07721 121121	23	24	12/12/2003	50	100	John Brown	10	15
Martin Jones	Manchester	01612249933		23	2/11/2002		50	Bob Jones		10
Alex Haley	London	020845522988		23	15/1/2003		150	John Brown		20

The primary key here consists of a combination of Customer Name, City and Phone. Multiple values exist for Product ID, Quantity and Discount. Therefore you cannot determine which quantity relates to which product. Consider also the situation where James Smith purchases another product. This would mean adding another column to the table.

In all normalisation operations, it is important to consider the operations we wish to perform on the table:

1. **Update** – how do we change the quantity associated with James Smith's purchase of Product ID 23 – we cannot identify whether this is 50 or 100, and this would involve changing only part of the value in the field.
2. **Insert** - How do we add another purchase for James Smith – this would mean modifying the Product ID and Quantity fields, but also making sure that existing values were not lost.
3. **Delete** – what happens James Smith changes his mind about the purchase of Product 23. Once again, this would involve identifying the quality associated with this product and only removing part values from the field.

Therefore the proposed structure does not allow us to perform standard operations on the table.

A normalised version of the above table is shown below.

Customer Name	**City**	**Phone**	Product ID	Date	Quantity	Salesperson	%Discount
James Smith	London	07721 121121	23	12/12/2003	50	John Brown	10
James Smith	London	07721 121121	24	15/12/2003	100	John Brown	15
Martin Jones	Manchester	01612249933	23	2/11/2002	50	Bob Jones	10
Alex Haley	London	020845522988	23	15/1/2003	150	John Brown	20

The process of normalisation to First Normal Form has ensured that there is only one value for each field in the table – i.e. that each field has atomic values. First Normal Form is often represented as 1NF and every relation in a database is in first normal form, as you cannot enter multiple values to into the fields for a relational database.

Ensuring that tables are in First Normal Form is the first step of the normalisation process. At this stage, the focus is on atomic values, and the fact that there are duplicate entries in the table should not be of concern, as these will be eliminated by the remaining steps in the normalisation process.