Normalization is the process of simplifying the design of a database so that it achieves the optimum structure

Normalization theory gives us the concept of normal forms to assist in achieving the optimum structure. The normal forms are a linear progression of rules that you apply to your database, with each higher normal form achieving a better, more efficient design. The normal forms are:

- First Normal Form
- Second Normal Form
- Third Normal Form
- Boyce Codd Normal Form
- Fourth Normal Form
- Fifth Normal Form

We will discuss normalization through Third Normal Form.

The Normal Forms are based on relations rather than tables. A relation is a special type of table that has the following attributes:

1. They describe one entity.
2. They have no duplicate rows; hence there is always a primary key.
3. The columns are unordered.
4. The rows are unordered.

First Normal Form

First Normal Form (1NF) says that all column values must be atomic.

The word atom comes from the Latin atomis, meaning indivisible (or literally "not to cut"). 1NF dictates that, for every row-by-column position in a given table, there exists only one value, not an array or list of values. The benefits from this rule should be fairly obvious. If lists of values are stored in a single column, there is no simple way to manipulate those values. Retrieval of data becomes much more laborious and difficult to generalize.

Table: tblOrder1

OrderId	CustomerId	Items
1	4	5 hammer, 3 screwdriver, 6 monkey wrench
2	23	1 hammer
3	15	2 deluxe garden hose, 2 economy nozzle
4	2	15 10' 2x4 untreated pine board
5	23	1 screwdriver
6	2	5 key

Record: 1 of 6

Table: tblOrder2

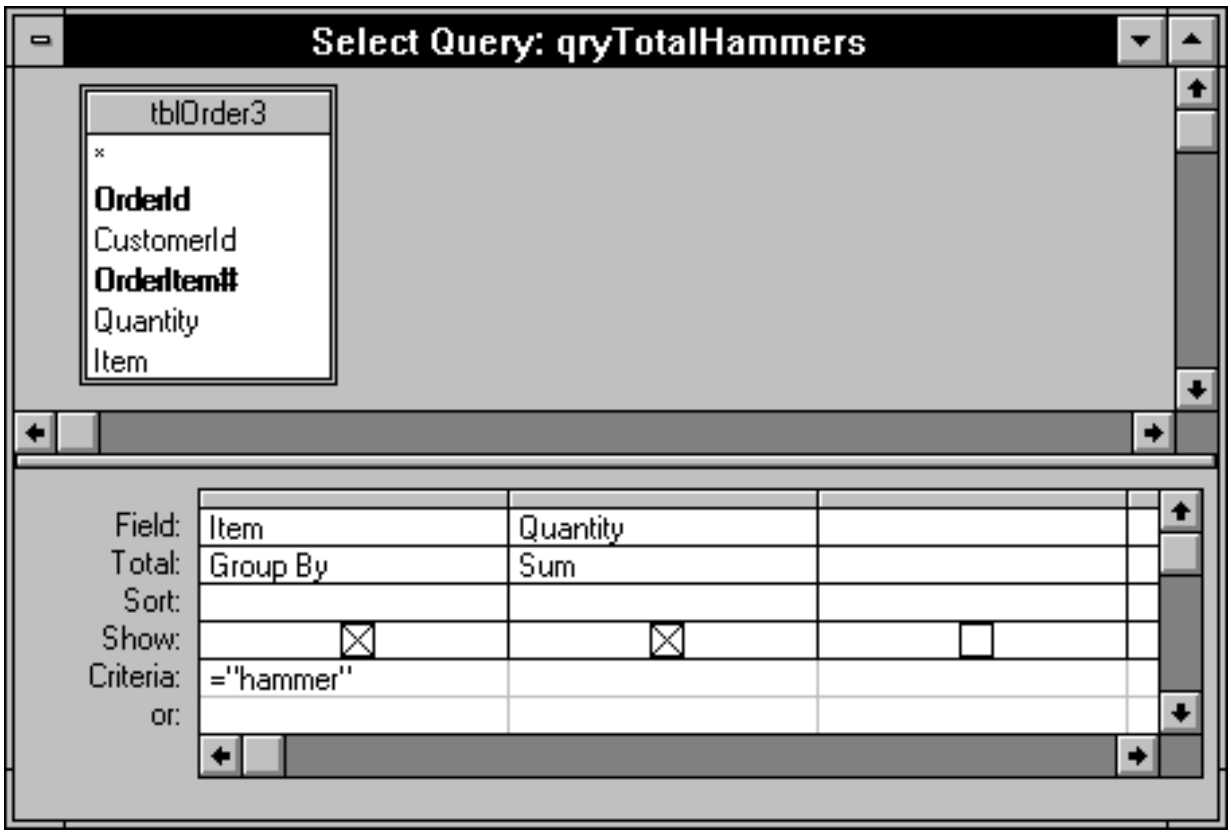
OrderId	CustomerId	Quant1	Item1	Quant2	Item2	Quant3	Item3
1	4	5	hammer	3	screwdriver	6	monkey wrench
2	23	1	hammer				
3	15	2	deluxe garden hose	2	economy nozzle		
4	2	15	10' 2x4 untreated pine board				
5	23	1	phillips screwdriver				
6	2	5	key				

Record: 1 of 6

Table: tblOrder3

OrderId	CustomerId	OrderItem#	Quantity	Item
1	4	1	5	hammer
1	4	2	3	screwdriver
1	4	3	6	monkey wrench
2	23	1	1	hammer
3	15	1	2	deluxe garden hose
3	15	2	2	ecomomy nozzle
4	2	1	15	10' 2x4 untreated pine board
5	23	1	1	screwdriver
6	2	1	5	key

Record: 1 of 9



Second Normal Form

A table is said to be in Second Normal Form (2NF), if it is in 1NF and every non-key column is fully dependent on the (entire) primary key. Put another way, tables should only store data relating to one "thing" (or entity) and that entity should be described by its primary key.

To determine if tblOrder4 meets 2NF, you must first note its primary key. The primary key is a composite of OrderId and OrderItem#. Thus, in order to be 2NF, each non-key column (i.e., every column other than OrderId and OrderItem#) must be fully dependent on the primary key. In other words, does the value of OrderId and OrderItem# for a given record imply the value of every other column in the table? The answer is no. Given the OrderId, you know the customer and date of the order, without having to know the OrderItem#. Thus, these two columns are not dependent on the entire primary key which is composed of both OrderId and OrderItem#. For this reason tblOrder4 is not 2NF.

You can achieve Second Normal Form by breaking tblOrder4 into two tables. The process of breaking a non-normalized table into its normalized parts is called **decomposition**. Since tblOrder4 has a composite primary key, the decomposition process is straightforward. Simply put everything that applies to each order in one table and everything that applies to each order item in a second table

Table: tblOrder4

OrderId	CustomerId	OrderDate	OrderItem#	Quantity	ProductId	ProductDescription
1	4	5/1/94	1	5	32	hammer
1	4	5/1/94	2	3	2	screwdriver
2	23	5/9/94	1	1	32	hammer
3	15	7/4/94	1	2	113	deluxe garden hose
3	15	7/4/94	2	2	121	ecomomy nozzle
4	2	8/1/94	1	15	1024	10' 2x4 untreated pine boards
5	23	8/2/94	1	1	2	screwdriver
6	2	8/2/94	1	5	52	key

Record: 1 of 8

Table: tblOrder

OrderId	CustomerId	OrderDate
1	1	5/1/94
2	3	5/9/94
3	1	7/4/94
4	2	8/1/94
5	1	8/2/94
6	2	8/2/94

Record: 1 of 6

Table: tblOrderDetail

OrderId	OrderItem#	Quantity	ProductId	ProductDescription
1	1	5	32	hammer
1	2	3	2	screwdriver
2	1	1	32	hammer
3	1	2	113	deluxe garden hose
3	2	2	121	ecomomy nozzle
4	1	15	1024	10' 2x4 untreated pine boards
5	1	1	2	screwdriver
6	1	5	52	key

Record: 1 of 8

Third Normal Form

A table is said to be in Third Normal Form (3NF), if it is in 2NF and if all non-key columns are mutually independent.

Table: tblOrderDetail1

OrderId	OrderItem#	Quantity	ProductId
1	1	5	32
1	2	3	2
2	1	1	32
3	1	2	113
3	2	2	121
4	1	15	1024
5	1	1	2
6	1	5	52

Record: 1 of 8

Table: tblProduct

ProductId	ProductDescription
2	screwdriver
32	hammer
52	key
113	deluxe garden hose
121	ecomomy nozzle
1024	10' 2x4 untreated pine boards

Record: 1 of 6

Note: An Anomaly is simply an error or inconsistency in the database. A poorly designed database runs the risk of introducing numerous anomalies. There are three types of anomalies:

1. **Insertion**—an anomaly that occurs during the insertion of a record. For example, the insertion of a new row causes a calculated total field stored in another table to report the wrong total.
2. **Deletion**—an anomaly that occurs during the deletion of a record. For example, the deletion of a row in the database deletes more information than you wished to delete.
3. **Update**—an anomaly that occurs during the updating of a record. For example, updating a description column for a single part in an inventory database requires you to make a change to thousands of rows.