NORMALIZATION

The topic of normalizing tables draws much attention in database design. Normalization helps you avoid redundancies and inconsistencies in your data. The main idea in normalization is to reduce tables to a set of columns where all the non-key columns depend on the entire primary key of the table. If this is not the case, the data can become inconsistent during updating.

NORMALIZATION RULES

Here are brief descriptions of the normal forms presented later:

First - At each row and column position in the table there exists one value, never a set of values.

Second - Each column that is not in the key provides a fact that depends on the entire key.

Third - Each non-key column provides a fact that is independent of other non-key columns and depends only on the key.

Fourth - No row contains two or more independent multi-valued facts about an entity.

First Normal Form (1NF)

A table satisfies the requirement of the first normal form if for each row-and-column position in the table there exists one value, never a set of values. A table that is in first normal form does not necessarily meet the test for higher normal forms.

For example, the following table violates first normal form because the WAREHOUSE column contains several values for each occurrence of PART.

TEST: Are there any repeating groups?

Violating 1NF

PART (Primary Key)	WAREHOUSE
P0010	Warehouse A, Warehouse B, Warehouse C
P0020	Warehouse B, Warehouse D

Conforming to 1NF

PART (Primary Key)	WAREHOUSE (Primary Key)	QUANTITY
P0010	Warehouse A	400
P0010	Warehouse B	543
P0010	Warehouse C	329
P0020	Warehouse B	200
P0020	Warehouse D	278

Second Normal Form (2NF)

A table is in the second normal form if each column that is not in the key provides a fact that depends on the entire key.

This means that all data that is not part of the primary key must depend on all of the columns in the key. This reduces repetition among database tables.

Second normal form is violated when a non-key column is a fact about a subset of a composite key, as in the following example. An inventory table records quantities of specific parts stored at particular warehouses; its columns are shown in the following example.

TEST: Does any attribute depend on part of the key not the primary key?

Violating 2NF

PART (Primary Key)	WAREHOUSE (Primary Key)	QUANTITY	WAREHOUSE_ADDR
P0010	Warehouse A	400	1608 New Field Road
P0010	Warehouse B	543	4141 Greenway Drive
P0010	Warehouse C	329	171 Pine Lane
P0020	Warehouse B	200	4141 Greenway Drive
P0020	Warehouse D	278	800 Massey Street

Here, the key consists of the PART and the WAREHOUSE columns together. Because the column WAREHOUSE_ADDRESS depends only on the value of WAREHOUSE, the table violates the rule for second normal form.

The problems with this design are:

- The warehouse address is repeated in every record for a part stored in that warehouse.
- If the address of the warehouse changes, every row referring to a part stored in that warehouse must be updated.
- Because of the redundancy, the data might become inconsistent, with different records showing different addresses for the same warehouse.

- If at some time there are no parts stored in the warehouse, there might be no row in which to record the warehouse address.

To satisfy second normal form, the Table, would have to be split into the following two tables:

Part-Stock table conforming to Second Normal Form (2NF)

PART (Primary Key)	WAREHOUSE (Primary Key)	QUANTITY
P0010	Warehouse A	400
P0010	Warehouse B	543
P0010	Warehouse C	329
P0020	Warehouse B	200
P0020	Warehouse D	278

Warehouse table conforming to Second Normal Form (2NF)

WAREHOUSE (Primary Key)	WAREHOUSE_ADDR
Warehouse A	1608 New Field Road
Warehouse B	4141 Greenway Drive
Warehouse C	171 Pine Lane
Warehouse D	404 Greenway Blvd
Warehouse E	800 Massey Street

However, there is a performance consideration in having the two tables in second normal form. Application programs that produce reports on the location of parts must join both tables to retrieve the relevant information.

Third Normal Form (3NF)

A table is in third normal form if each non-key column provides a fact that is independent of other non-key columns and depends only on the key.

Third normal form is violated when a non-key column is a fact about another non-key column. For example, the first table in the following example contains the columns EMPNO and WORKDEPT. Suppose a column DEPTNAME is added. The new column depends on WORKDEPT, whereas the primary key is the column EMPNO; thus the table now violates third normal form.

Changing DEPTNAME for a single employee, John Parker, does not change the department name for other employees in that department. The inconsistency that results is shown in the updated version of the table in the following example.

TEST: Does any attribute depend on another attribute that is not the primary key?

Unnormalized Employee-Department Table Before Update

EMPNO Primary Key	FIRSTNAME	LASTNAME	DEPT	DEPTNAME
000290	John	Parker	E11	Operations
000320	Ramlal	Mehta	E21	Software Support
000310	Maude	Setright	E11	Operations

The following example shows the content of the table following an update to the DEPTNAME column for John Parker. Note that there are now two different department names used for department number (DEPT) E11:

Unnormalized Employee-Department Table After Update of John Parker's Department name (DEPTNAME). Information in table has become inconsistent.

EMPNO (Primary Key)	FIRSTNAME	LASTNAME	DEPT	DEPTNAME
000290	John	Parker	E11	Installation Mgmt
000320	Ramlal	Mehta	E21	Software Support
000310	Maude	Setright	E11	Operations

The table can be normalized by providing a new table, with columns for DEPT and DEPTNAME. In that case, an update like changing a department name is much easier—the update only has to be made to the new table. An SQL query that shows the department name along with the employee name is more complex to write because it requires joining the two tables. This query will probably also take longer to execute than the query of a single table.

Normalized Employee Table

EMPNO (Primary Key)	FIRSTNAME	LASTNAME	DEPT
000290	John	Parker	E11
000320	Ramlal	Mehta	E21
000310	Maude	Setright	E11

Normalized Department Table

DEPT (Primary Key)	DEPTNAME
E11	Operations
E21	Software Support

Fourth Normal Form (4NF)

A table is in fourth normal form if no row contains two or more independent multi-valued facts about an entity.

Consider these entities: Employees, Skills, and Languages. An employee can have several skills and know several languages. There are two relationships, one between employees and skills, and one between employees and languages. A table is not in fourth normal form if it represents both relationships, as in the following example:

Violating 4NF

EMPNO (Primary Key)	SKILL (Primary Key)	LANGUAGE (Primary Key)
000130	Data Modelling	English
000130	Database Design	English
000130	Application Design	English
000130	Data Modelling	Spanish
000130	Database Design	Spanish
000130	Application Design	Spanish

Instead, the relationships should be represented in two tables, as in the following examples.

Employee-Skill Table in Fourth Normal Form (4NF)

EMPNO (Primary Key)	SKILL (Primary Key)
000130	Data Modelling
000130	Database Design
000130	Application Design

Employee-Language Table in Fourth Normal Form (4NF)

EMPNO (Primary Key)	LANGUAGE (Primary Key)
000130	English
000130	Spanish

If, however, the facts are interdependent—that is, the employee applies certain languages only to certain skills—then the table should not be split.

Any data can be put into fourth normal form. A good rule when designing a database is to arrange all data in tables in fourth normal form, and then decide whether the result gives you an acceptable level of performance. If it does not, you are at liberty to denormalize your design.

"The key, the whole key, and nothing but the key, so help me Codd."