

Normalization is a data analysis technique to design a database system. It produces a relational schema with the following properties:

- Removing unnecessary data Redundancy, e.g. duplicate data
- Remove Anomalies, e.g. Removing Update/Delete/Insert Anomalies

StudentNum	CourseNum	Student Name	Address	Course
S21	9201	Jones	Edinburgh	Accounts
S21	9267	Jones	Edinburgh	physics
S24	9267	Smith	Glasgow	physics
S30	9201	Richards	Manchester	Accounts
S30	9322	Richards	Manchester	Maths

An **Update Anomaly** exists when one or more instances of duplicated data is updated, but not all. For example, consider Jones moving address - you need to update all instances of Jones's address.

A **Delete Anomaly** exists when certain attributes are lost because of the deletion of other attributes. For example, consider what happens if Student S30 is the last student to leave the course - All information about the course is lost.

An **Insert Anomaly** occurs when certain attributes cannot be inserted into the database without the presence of other attributes. For example, this is the converse of delete anomaly - we can't add a new course unless we have at least one student enrolled on the course.

Rules of Normalization:

1 NF:

Each table cell should contain a single value, i.e. **No repeating GROUP of data**

Each attribute contains smallest meaningful value, i.e., **Atomic value** (indivisible)

Contains **NO derived values**, e.g. Age vs DOB

Each record needs to be unique.

Has **Primary Key**

2 NF:

Be in 1NF - Every table with a single PK is automatically in 2NF

Single Column Primary Key

Every Non-Key attributes are dependent on **entire PK**

Tables with repeating VALUES in a field are not in 2NF, because all fields does not relate to the entire PK, i.e. NO repeating VALUES

Normally, 2NF only meaningful when there is a concatenated key

3 NF:

Be in 2NF

Has **NO transitive** functional dependencies

A transitive functional dependency is when changing a non-key column, might cause any of the other non-key columns to change, e.g. changing the non-key column Name may change Salutation.

Eliminate Transitive Dependency in 3NF: No non-key attributes are dependent on any other non-key attributes, e.g. dept_code -> department_name

Example 1:

Customer			
Customer ID	First Name	Last Name	Phone
123	Tom	Smith	555-123-4567, 555-123-5678
124	San	Jose	555-123-8888; 555-123-8889
125	John	Doe	555-123-9999

Violates 1NF: Phone numbers is **not atomic**, it can be sub-divided

Customer ID	First Name	Last Name	Phone 1	Phone 2
123	Tom	Smith	555-123-4567	555-123-5678
124	San	Jose	555-123-8888	555-123-8889
125	John	Doe	555-123-9999	

Technically, phone number is **atomic** and does not violate the requirement for values to be indivisible. However, the two phone number columns still form a **repeating group**.

What if there is more phone numbers? How many phone columns should there be?

Adding an extra telephone number may require the table to be reorganized by the addition of a new column rather than just having a new row (tuple) added. Also, there NULL value in Phone 2 for CustomerID 125.

1 NF:

Customer			
Customer ID	First Name	Last Name	Phone
123	Tom	Smith	555-123-4567
123	Tom	Smith	555-123-5678
124	San	Jose	555-123-8888
124	San	Jose	555-123-8889
125	John	Doe	555-123-9999

Re-arranging table ensures no row contains more than one phone number and now phone number into **atomic**, i.e. indivisible.

Note that the Customer ID is no longer unique. To uniquely identify a row, we need to use a combination of Customer ID and Phone Number. The value of the combination is unique although each column separately contains repeated values. Being able to uniquely identify a row (tuple) is a requirement of **1NF**.

2 NF:

Customer			Phone		
Customer ID	First Name	Last Name	Phone ID	Customer ID	Phone
123	Tom	Smith	1	123	555-123-4567
124	San	Jose	2	123	555-123-5678
125	John	Doe	3	124	555-123-8888
			4	124	555-123-8889
			5	125	555-123-9999

Every Non-Key attributes are dependent on **entire PK**

3 NF:

Has **NO transitive** functional dependencies. **No** non-key attributes are dependent on any other non-key attributes.

2 NF became 3 NF.

Note: Information collected from various sources, data and explanations may vary from the source to meet the class requirements.