Documentation On Data Normalization

Database normalization is the process of structuring a relational database to reduce data redundancy and improve data integrity. It organizes columns and tables to ensure that dependencies are properly enforced by database integrity constraints. Normalization aims to eliminate anomalies that can occur during data insertion, deletion, and updates.

**1. First Normal Form (1NF) .**

A table is in **1NF** if it meets the following criteria:

* Each column contains only atomic (indivisible) values. This means a single cell can't hold multiple values.
* There are no repeating groups of columns. This means you shouldn't have columns like phone1, phone2, phone3 in a single row.

**Example:** Consider a table of students and the courses they are enrolled in.

| StudentID | StudentName | CourseName |
| --- | --- | --- |
| 101 | Alice | Math, Physics |
| 102 | Bob | Chemistry |

This table is **not in 1NF** because the CourseName column for StudentID 101 contains multiple values.

**To convert to 1NF:** Break down the multi-valued column into separate rows.

| StudentID | StudentName | CourseName |
| --- | --- | --- |
| 101 | Alice | Math |
| 101 | Alice | Physics |
| 102 | Bob | Chemistry |

**2. Second Normal Form (2NF) .**

A table is in **2NF** if it meets the following criteria:

* It is already in **1NF**.
* All non-key attributes are fully functionally dependent on the primary key. This means no non-key attribute can depend on only a part of a composite primary key.

A **composite primary key** is a primary key made of two or more attributes.

**Example:** Consider a table with student enrollment details where the composite primary key is (StudentID, CourseName).

| StudentID | CourseName | StudentName | InstructorName |
| --- | --- | --- | --- |
| 101 | Math | Alice | Mr. Smith |
| 101 | Physics | Alice | Ms. Jones |
| 102 | Chemistry | Bob | Mr. Smith |

This table is **not in 2NF** because StudentName depends only on StudentID (part of the primary key) and not the full composite key. Similarly, InstructorName depends only on CourseName.

**To convert to 2NF:** Create separate tables for the partial dependencies.

**Table 1: Student Details** (Primary Key: StudentID)

| StudentID | StudentName |
| --- | --- |
| 101 | Alice |
| 102 | Bob |

**Table 2: Course Details** (Primary Key: CourseName)

| CourseName | InstructorName |
| --- | --- |
| Math | Mr. Smith |
| Physics | Ms. Jones |
| Chemistry | Mr. Smith |

**Table 3: Enrollment Details** (Primary Key: (StudentID, CourseName)

| StudentID | CourseName |
| --- | --- |
| 101 | Math |
| 101 | Physics |
| 102 | Chemistry |

**3. Third Normal Form (3NF) .**

A table is in **3NF** if it meets the following criteria:

* It is already in **2NF**.
* There are no transitive dependencies. A **transitive dependency** occurs when a non-key attribute is dependent on another non-key attribute.

**Example:** Consider a table with student details.

| StudentID | StudentName | City | ZipCode |
| --- | --- | --- | --- |
| 101 | Alice | New York | 10001 |
| 102 | Bob | Los Angeles | 90210 |

The primary key is StudentID. The ZipCode is dependent on the City, and City is dependent on StudentID. This is a transitive dependency: StudentID -> City -> ZipCode.

This table is **not in 3NF**.

**To convert to 3NF:** Create a separate table for the transitive dependency.

**Table 1: Student Details** (Primary Key: StudentID)

| StudentID | StudentName | City |
| --- | --- | --- |
| 101 | Alice | New York |
| 102 | Bob | Los Angeles |

**Table 2: City\_ZipCode** (Primary Key: City)

| City | ZipCode |
| --- | --- |
| New York | 10001 |
| Los Angeles | 90210 |

**4.Boyce-Codd Normal Form (BCNF).**

**BCNF** is a stricter version of **3NF**. A table is in BCNF if and only if for every non-trivial functional dependency X→Y, X is a superkey. A **superkey** is any set of attributes that uniquely identifies a row in a table. It's essentially a superset of a primary key. BCNF addresses anomalies that 3NF misses, especially when a table has multiple candidate keys that overlap.

**Example:** Consider a table (StudentID, Subject, Professor) where:

* StudentID and Subject uniquely identify a professor: (StudentID, Subject) -> Professor.
* Professor also uniquely identifies a subject: Professor -> Subject (each professor teaches only one subject).

| StudentID | Subject | Professor |
| --- | --- | --- |
| 101 | Physics | Prof. White |
| 102 | Chemistry | Prof. Black |
| 103 | Physics | Prof. White |

Export to Sheets

This table is in **3NF** because there are no transitive dependencies. However, it's **not in BCNF** because the functional dependency Professor -> Subject has Professor as the determinant, but Professor is not a superkey.

To convert to BCNF, we decompose the table:

**Table 1: Student\_Professor**

| StudentID | Professor |
| --- | --- |
| 101 | Prof. White |
| 102 | Prof. Black |
| 103 | Prof. White |

**Table 2: Professor\_Subject**

| Professor | Subject |
| --- | --- |
| Prof. White | Physics |
| Prof. Black | Chemistry |

**5.Fourth Normal Form (4NF).**

A table is in **4NF** if it's in BCNF and has no multi-valued dependencies. A **multi-valued dependency** exists when the presence of one or more rows implies the presence of one or more other rows in the same table. It usually involves a single determinant (attribute) that determines multiple sets of values.

**Example:** Consider a table with a student and their multiple activities and hobbies.

| StudentID | Activity | Hobby |
| --- | --- | --- |
| 101 | Playing Football | Reading |
| 101 | Playing Football | Hiking |
| 101 | Swimming | Reading |
| 101 | Swimming | Hiking |

This table is **not in 4NF** because there are multi-valued dependencies: StudentID ->> Activity and StudentID ->> Hobby. The activities and hobbies are independent of each other.

To convert to 4NF, we decompose the table into separate tables for each multi-valued dependency:

**Table 1: Student\_Activities**

| StudentID | Activity |
| --- | --- |
| 101 | Playing Football |
| 101 | Swimming |

**Table 2: Student\_Hobbies**

| StudentID | Hobby |
| --- | --- |
| 101 | Reading |
| 101 | Hiking |

**5.Fifth Normal Form (5NF)**

A table is in **5NF** if it's in 4NF and has no join dependencies. A **join dependency** exists when a table can be losslessly decomposed into multiple tables, and when those tables are joined back together, they produce the original table without any extra, spurious rows. 5NF deals with complex real-world constraints where a table can be broken down into three or more tables without losing information. It is also known as Project-Join Normal Form (PJ/NF). 5NF is rarely implemented in practice because the decomposition is often complex and the benefits are minimal.