Database Normalization to Third Normal Form (3NF)

Introduction

Third Normal Form (3NF) is a database normalization level used to ensure that the data is stored efficiently and to reduce redundancy. To achieve 3NF, a database must already be in Second Normal Form (2NF), and it must meet the following criteria:

1. It is in 2NF.
2. It has no transitive dependencies, meaning non-prime attributes (those not part of any candidate key) depend only on the candidate keys and not on other non-prime attributes.

Steps to Normalize a Database to 3NF

Step 1: Ensure the database is in First Normal Form (1NF)

1NF requires that:

- Each table has a primary key.
- All values in the table are atomic (indivisible).
- Each column contains only one value per row.
- Entries in a column are of the same data type.

Step 2: Ensure the database is in Second Normal Form (2NF)

2NF requires that:

- The database is in 1NF.
- All non-key attributes are fully functionally dependent on the primary key. This means that there are no partial dependencies; non-key attributes cannot depend on just part of the primary key in composite key tables.

Step 3: Ensure the database is in Third Normal Form (3NF)

Database Normalization to Third Normal Form (3NF)

3NF requires that:

- The database is in 2NF.

- All non-key attributes are non-transitively dependent on the primary key, meaning that non-key attributes depend directly on the primary key and not on other non-key attributes.

Example

Let's consider an example to demonstrate the normalization process.

Example Database (Unnormalized)

A table named `Student` with the following columns:

StudentID	StudentName	CourseID	CourseName	Instructor
1	Alice	101	Math	Dr. Smith
2	Bob	102	Physics	Dr. Jones
3	Charlie	101	Math	Dr. Smith
4	Dave	103	Chemistry	Dr. Brown

Step 1: Convert to 1NF

Ensure that all values are atomic and there is a primary key.

StudentID	StudentName	CourseID	CourseName	Instructor
1	Alice	101	Math	Dr. Smith
2	Bob	102	Physics	Dr. Jones
3	Charlie	101	Math	Dr. Smith

4	Dave	103	Chemistry	Dr. Brown

Step 2: Convert to 2NF

Remove partial dependencies by creating separate tables.

`Student` table:

StudentID	StudentName
1	Alice
2	Bob
3	Charlie
4	Dave

`Course` table:

CourseID	CourseName	Instructor
101	Math	Dr. Smith
102	Physics	Dr. Jones
103	Chemistry	Dr. Brown

Step 3: Convert to 3NF

Ensure there are no transitive dependencies.

The tables created in 2NF are already in 3NF because all non-key attributes are dependent on the primary key and there are no transitive dependencies.

Database Normalization to Third Normal Form (3NF)

So, the final normalized database in 3NF consists of:

`Student` table:

StudentID	StudentName
1	Alice
2	Bob
3	Charlie
4	Dave

`Course` table:

CourseID	CourseName	Instructor
101	Math	Dr. Smith
102	Physics	Dr. Jones
103	Chemistry	Dr. Brown