Database Design: Second Normal Form (2NF)

Second Normal Form (2NF) is a crucial step in designing a relational database. To achieve 2NF, you must first ensure the database is in First Normal Form (1NF). A database is in 2NF if it is in 1NF and all non-key attributes are fully functionally dependent on the primary key.

Here are the steps to convert a database to 2NF:

Step 1: Ensure the database is in 1NF

1. Remove repeating groups: Each column should contain atomic values, and each row should be unique.
2. Create a primary key: Each table should have a primary key that uniquely identifies each record.

Step 2: Identify Partial Dependencies

In 2NF, we eliminate partial dependencies. A partial dependency occurs when a non-key attribute is dependent on part of a composite primary key rather than the whole primary key.

Step 3: Remove Partial Dependencies

1. Create new tables: For each partial dependency, create a new table.
2. Move the partially dependent attributes: Move the attributes involved in the partial dependency to the new table.
3. Create foreign keys: Establish a relationship between the new table and the original table using foreign keys.

Example

Let's consider an example with a table that contains student courses.

StudentCourses Table (1NF):

StudentID	CourseID	CourseName	Instructor	InstructorOffice
1	101	Math	Dr. Smith	Room 201
1	102	History	Dr. Brown	Room 202
2	101	Math	Dr. Smith	Room 201

Step 1: Identify the primary key and ensure 1NF

- Primary Key: (StudentID, CourseID)

- The table is already in 1NF as there are no repeating groups and each row is unique.

Step 2: Identify partial dependencies

- CourseName, Instructor, and InstructorOffice depend on CourseID, not on the combination of StudentID and CourseID.

Step 3: Remove partial dependencies

1. Create new tables to eliminate partial dependencies:

Students Table:

StudentID
1
2

Courses Table:

CourseID	CourseName	Instructor	InstructorOffice
101	Math	Dr. Smith	Room 201
102	History	Dr. Brown	Room 202

StudentCourses Table:

StudentID	CourseID
1	101
1	102
2	101

2. Establish relationships using foreign keys:

- In the StudentCourses table, StudentID references Students table and CourseID references Courses table.

Now, the database is in 2NF as all non-key attributes are fully functionally dependent on the whole primary key.

Final Design in 2NF

1. Students Table:

StudentID
1
2

2. Courses Table:

CourseID	CourseName	Instructor	InstructorOffice
101	Math	Dr. Smith	Room 201
102	History	Dr. Brown	Room 202

3. StudentCourses Table:

StudentID	CourseID
1	101
1	102
2	101

This design ensures that all non-key attributes are fully functionally dependent on the primary key, achieving Second Normal Form (2NF).