**Second Normal Form (2NF) in Relational Databases**

The Second Normal Form (2NF) is a level of database normalization that builds on the principles of the First Normal Form (1NF). A table is in 2NF if it satisfies the following conditions:

1. **It is in First Normal Form (1NF):**
   * The table has a primary key.
   * All columns contain atomic (indivisible) values.
   * Each column contains only one value per row, and the data is stored in a tabular format with no repeating groups.
2. **No Partial Dependency:**
   * Every non-prime attribute (an attribute that is not part of a candidate key) must depend on the whole primary key, not just part of it.
   * This applies to tables with composite primary keys (keys that consist of two or more columns). Partial dependency means that a non-prime attribute depends on only one part of the composite key, which violates 2NF.

**Key Concepts:**

* **Full Functional Dependency:** A non-prime attribute is fully functionally dependent on the entire primary key, not just part of it.
* **Eliminating Partial Dependencies:** Tables with partial dependencies are split into smaller tables so that each non-prime attribute is fully dependent on the primary key.

**Second Normal Form in ER Diagrams**

Entity-Relationship (ER) Diagrams are conceptual tools to design relational databases. To achieve 2NF in an ER diagram:

* Ensure that all non-prime attributes are fully dependent on the primary key of the entity.
* Redesign relationships and entities to eliminate partial dependencies by creating new entities or relationships as needed.

**Steps to Achieve 2NF in ER Diagrams:**

1. Identify composite keys in entities or relationships.
2. Check if any non-prime attribute depends only on part of the composite key.
3. Break down entities or relationships into smaller ones to ensure all attributes fully depend on the primary key.

**Example:**

Consider a table storing student-course details:

* **Table Name:** StudentCourse
* **Attributes:** StudentID, CourseID, StudentName, CourseName, Grade
* **Composite Primary Key:** (StudentID, CourseID)

**Problem:**

* StudentName depends only on StudentID.
* CourseName depends only on CourseID.
* Grade depends on the full key (StudentID, CourseID).

**Solution to Achieve 2NF:**

1. Split the table into three smaller tables:
   * Student: (StudentID, StudentName)
   * Course: (CourseID, CourseName)
   * StudentCourse: (StudentID, CourseID, Grade)

**ER Diagram Changes:**

* Create entities for Student, Course, and StudentCourse.
* Link StudentCourse with Student and Course using relationships.

This ensures that all non-prime attributes are fully dependent on the respective primary keys, satisfying 2NF.