Normalization in SQL is a process to organize data in a database to minimize redundancy and ensure data integrity. The process involves dividing a database into tables and defining relationships between them based on rules called **normal forms (NFs)**. Here's a simplified explanation of the **First Normal Form (1NF)**, **Second Normal Form (2NF)**, and **Third Normal Form (3NF):**

First Normal Form (1NF)

Definition:
A table is in **1NF** if:

1. All the values in each column are **atomic** (indivisible).
2. Each column contains values of a single type.
3. Each row is unique (there is a primary key).

Example (Non-1NF Table):

StudentID	Name	Courses
1	Alice	Math, Science
2	Bob	English

Here, the "Courses" column contains multiple values in a single cell.

1NF Table (Normalized):

StudentID	Name	Course
1	Alice	Math
1	Alice	Science
2	Bob	English

Second Normal Form (2NF)

Definition:
A table is in **2NF** if:

1. It is already in **1NF**.

2. All **non-key attributes** are fully dependent on the **entire primary key** (no partial dependency).

Partial Dependency Explanation:
If a table has a composite primary key (e.g., two columns as the primary key), then every non-key column must depend on **both** parts of the key, not just one.

Example (Non-2NF Table):

StudentID	Course	Instructor
1	Math	Dr. Smith
1	Science	Dr. Brown

Here, "Instructor" depends only on "Course," not on the entire primary key ("StudentID, Course").

2NF Table (Normalized):
Table 1: Enrollment

StudentID	Course
1	Math
1	Science

Table 2: CourseDetails

Course	Instructor
Math	Dr. Smith
Science	Dr. Brown

Third Normal Form (3NF)

Definition:
A table is in **3NF** if:

1. It is already in **2NF**.
2. There are no **transitive dependencies**, meaning no non-key attribute depends on another non-key attribute.

Example (Non-3NF Table):

StudentID	Course	Instructor	InstructorPhone
1	Math	Dr. Smith	123-456-7890
1	Science	Dr. Brown	987-654-3210

Here, "InstructorPhone" depends on "Instructor," not directly on the primary key.

3NF Table (Normalized):
Table 1: Enrollment

StudentID	Course
1	Math
1	Science

Table 2: CourseDetails

Course	Instructor
Math	Dr. Smith
Science	Dr. Brown

Table 3: InstructorDetails

Instructor	Phone
Dr. Smith	123-456-7890
Dr. Brown	987-654-3210

Key Takeaways:

- **1NF** eliminates duplicate columns and ensures each column has atomic values.
- **2NF** removes partial dependencies on composite keys.
- **3NF** removes transitive dependencies to avoid redundancy and maintain data integrity.

Here's a visual example using **SQL code** for creating and normalizing a table step-by-step:

Step 1: Non-Normalized Table

This is how a non-normalized table might look:

Table: Enrollment

StudentID	Name	Courses	Instructor	InstructorPhone
1	Alice	Math, Science	Dr. Smith	123-456-7890
2	Bob	English	Dr. Brown	987-654-3210

Step 2: 1NF (Atomic Columns)

To achieve **1NF**, split multi-valued columns into individual rows:

SQL Query to Normalize to 1NF:

```
CREATE TABLE Enrollment_1NF (
  StudentID INT,
  Name VARCHAR(50),
  Course VARCHAR(50),
  Instructor VARCHAR(50),
  InstructorPhone VARCHAR(15)
);

INSERT INTO Enrollment_1NF VALUES
(1, 'Alice', 'Math', 'Dr. Smith', '123-456-7890'),
(1, 'Alice', 'Science', 'Dr. Brown', '987-654-3210'),
(2, 'Bob', 'English', 'Dr. Brown', '987-654-3210');
```

1NF Table Output:

StudentID	Name	Course	Instructor	InstructorPhone
1	Alice	Math	Dr. Smith	123-456-7890
1	Alice	Science	Dr. Brown	987-654-3210
2	Bob	English	Dr. Brown	987-654-3210

Step 3: 2NF (Eliminate Partial Dependencies)

To achieve **2NF**, remove attributes that depend only on part of a composite primary key.

Normalization to 2NF:

Split the data into separate tables:

1. **Enrollment**: Tracks which students are enrolled in which courses.
2. **CourseDetails**: Stores course-specific information.

SQL Queries to Normalize to 2NF:

```
-- Enrollment Table
CREATE TABLE Enrollment_2NF (
  StudentID INT,
  Course VARCHAR(50)
);

INSERT INTO Enrollment_2NF VALUES
(1, 'Math'),
(1, 'Science'),
(2, 'English');

-- CourseDetails Table
CREATE TABLE CourseDetails (
```

```
    Course VARCHAR(50),
    Instructor VARCHAR(50),
    InstructorPhone VARCHAR(15)
);

INSERT INTO CourseDetails VALUES
('Math', 'Dr. Smith', '123-456-7890'),
('Science', 'Dr. Brown', '987-654-3210'),
('English', 'Dr. Brown', '987-654-3210');
```

2NF Tables Output:

Table 1: Enrollment_2NF

StudentID	Course
1	Math
1	Science
2	English

Table 2: CourseDetails

Course	Instructor	InstructorPhone
Math	Dr. Smith	123-456-7890
Science	Dr. Brown	987-654-3210
English	Dr. Brown	987-654-3210

Step 4: 3NF (Eliminate Transitive Dependencies)

To achieve **3NF**, remove transitive dependencies. InstructorPhone depends on Instructor, not Course.

Normalization to 3NF:

Split data further into:

1. **CourseDetails**: Links courses to instructors.
2. **InstructorDetails**: Stores instructor-specific information.

SQL Queries to Normalize to 3NF:

```
-- CourseDetails Table
CREATE TABLE CourseDetails_3NF (
  Course VARCHAR(50),
  Instructor VARCHAR(50)
);

INSERT INTO CourseDetails_3NF VALUES
('Math', 'Dr. Smith'),
('Science', 'Dr. Brown'),
('English', 'Dr. Brown');

-- InstructorDetails Table
CREATE TABLE InstructorDetails (
  Instructor VARCHAR(50),
  Phone VARCHAR(15)
);

INSERT INTO InstructorDetails VALUES
('Dr. Smith', '123-456-7890'),
('Dr. Brown', '987-654-3210');
```

3NF Tables Output:

Table 1: Enrollment_2NF

StudentID	Course
1	Math
1	Science
2	English

Table 2: CourseDetails_3NF

Course	Instructor
Math	Dr. Smith
Science	Dr. Brown
English	Dr. Brown

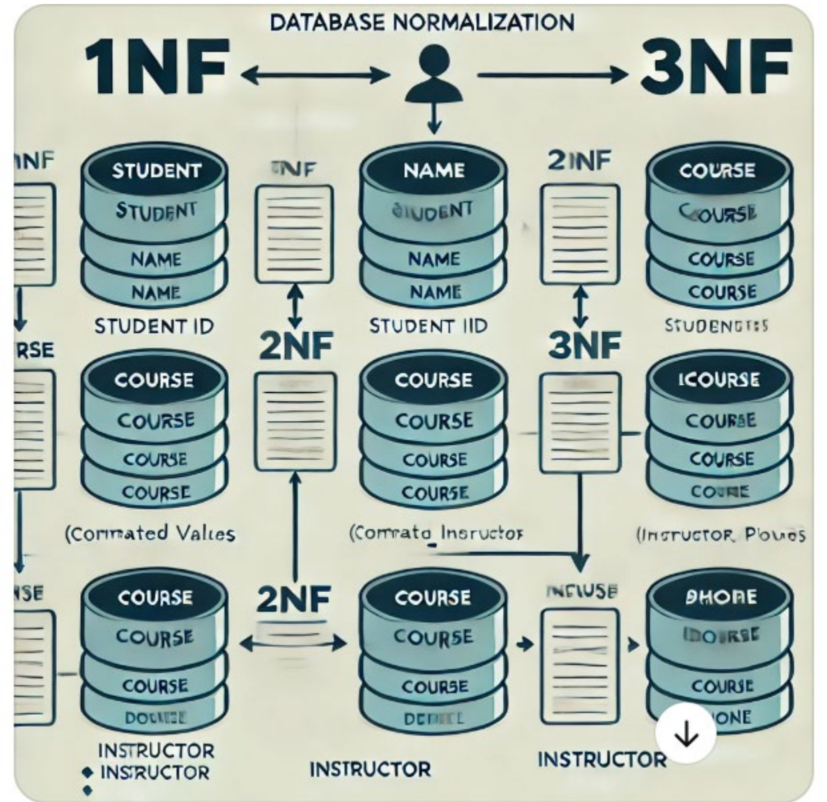
Table 3: InstructorDetails

Instructor	Phone
Dr. Smith	123-456-7890
Dr. Brown	987-654-3210

Earned Benefits of Normalization:

- **1NF:** Eliminated multi-valued columns.
- **2NF:** Removed partial dependencies by isolating course-related data.
- **3NF:** Removed transitive dependencies by isolating instructor-related data.

Here is a visual representation of the normalization process from 1NF to 3NF for the student enrollment database. The diagram demonstrates how the data is transformed and organized at each step for better clarity and data integrity.


DATABASE NORMALIZATION
1NF
3NF
STUDENT
NAME
STUDENT ID
2NF
COURSE
INSTRUCTOR