DATABASE NORMALIZATION FORMS AND RULES

1NF – First Normal Form

- Each cell has atomic (single) value.
- Each record is unique.
- Columns contain same data type.

Goal: Remove repeating groups.

2NF – Second Normal Form

- Must be in 1NF.
- No partial dependency (non-key depends on full key).

Goal: Eliminate dependencies on part of composite key.

3NF – Third Normal Form

- Must be in 2NF.
- No transitive dependency (non-key depends only on key).

Goal: Remove indirect dependencies.

BCNF – Boyce-Codd Normal Form

- Must be in 3NF.
- For each dependency $X \rightarrow Y$, X must be a superkey.

Goal: Handle overlapping candidate keys.

4NF – Fourth Normal Form

- Must be in BCNF.
- No multi-valued dependencies.

Goal: Eliminate independent multi-valued facts.

5NF – Fifth Normal Form

- Must be in 4NF.
- Remove join dependencies.

Goal: Decompose complex many-to-many relations.

6NF – Sixth Normal Form

- Must be in 5NF.
- Remove temporal dependencies.

Goal: Support temporal versioning.

SQL EXAMPLES FROM 1NF TO 5NF

```
-- Unnormalized Table
CREATE TABLE Enrollment_UNF (
StudentID INT,
StudentName VARCHAR(50),
Courses TEXT,
Instructors TEXT
);

-- 1NF
CREATE TABLE Enrollment_1NF (
StudentID INT,
StudentName VARCHAR(50),
Course VARCHAR(50),
Instructor VARCHAR(50)
);

-- 2NF
CREATE TABLE Students (
StudentID INT PRIMARY KEY,
StudentName VARCHAR(50)
);
CREATE TABLE Courses (
CourseID INT PRIMARY KEY,
CourseName VARCHAR(50)
);
CREATE TABLE Enrollment_2NF (
StudentID INT,
CourseID INT,
PRIMARY KEY (StudentID, CourseID)
);

-- 3NF
CREATE TABLE Instructors (
InstructorID INT PRIMARY KEY,
InstructorName VARCHAR(50)
```

```
);
CREATE TABLE CourseInstructor (
CourseID INT PRIMARY KEY,
InstructorID INT
);

-- BCNF
CREATE TABLE CourseInstructor_BCNF (
InstructorID INT PRIMARY KEY,
CourseID INT
);

-- 4NF
CREATE TABLE StudentHobbies (
StudentID INT,
Hobby VARCHAR(50),
PRIMARY KEY (StudentID, Hobby)
);
CREATE TABLE StudentLanguages (
StudentID INT,
Language VARCHAR(50),
PRIMARY KEY (StudentID, Language)
);

-- 5NF
CREATE TABLE StudentCourseInstructor (
StudentID INT,
CourseID INT,
InstructorID INT,
PRIMARY KEY (StudentID, CourseID, InstructorID)
);
```