

## Database Normalization Report

Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. This is achieved by dividing large tables into smaller ones and defining relationships between them.

### 1NF - First Normal Form

A table is in 1NF if:

- It has only atomic (indivisible) values.
- Each column contains only one value per record.

Example Before (UNF):

StudentID | Name | Courses

S01 | Reem | DB, AI

Example After (1NF):

StudentID | Name | Course

S01 | Reem | DB

S01 | Reem | AI

### 2NF - Second Normal Form

A table is in 2NF if:

- It is in 1NF
- All non-key attributes are fully functionally dependent on the entire primary key.

This typically applies to tables with composite keys.

Example Before (1NF):

StudentID, CourseID | Name, CourseName, Department

Name and Department depend only on StudentID.

After (2NF):

- Student(StudentID, Name, Department)
- Enrollment(StudentID, CourseID)
- Course(CourseID, CourseName)

### 3NF - Third Normal Form

A table is in 3NF if:

- It is in 2NF
- It has no transitive dependencies (non-key attributes depending on other non-keys).

Example Before (2NF):

EmployeeID | Name | Department | Manager

Manager depends on Department, not directly on EmployeeID.

After (3NF):

- Employee(EmployeeID, Name, Department)
- Department(Department, Manager)

## **De-normalization**

De-normalization is the process of deliberately introducing redundancy into a database design to improve read performance.

- Why apply it?

To speed up SELECT queries and reduce JOIN complexity.

- When to apply?

In reporting systems, OLAP databases, or when performance is more critical than update speed.

Example: Instead of joining Customer and Orders every time, you might store CustomerName in the Orders table.