Normalization

By Dhandapani Yedappalli Krishnamurthi Sep 30, 2025

🔹 Normal Forms (Steps of Normalization)

1NF (First Normal Form)

- Each column should have **atomic values** (no multiple values in a single field).
- No repeating groups.

✅ Example (Unnormalized Table):

StudentID	StudentName	Subjects
1	Kapil	Math, Science
2	Sneha	English, History

❌ Problem: Multiple values in "Subjects".
👉 **1NF Conversion:**

StudentID	StudentName	Subject
1	Kapil	Math
1	Kapil	Science
2	Sneha	English
2	Sneha	History

2NF (Second Normal Form)

- Must be in **1NF**.
- No **partial dependency** (non-key attribute should not depend on part of a composite key).

✅ Example (1NF Table):

StudentID	Subject	Teacher
1	Math	Sangeetha
1	Science	Tharun

❌ Problem: "Teacher" depends only on "Subject", not on the full composite key (StudentID + Subject).
👉 **2NF Conversion:**

Students Table

StudentID	StudentName

1	Kapil
2	Sneha

Subjects Table

Subject	Teacher
Math	Sangeetha
Science	Tharun
English	Tharun
History	Sangeetha

StudentSubjects Table

StudentID	Subject
1	Math
1	Science
2	English
2	History

3NF (Third Normal Form)

- Must be in **2NF**.
- No **transitive dependency** (non-key column depending on another non-key column).

✅ Example (2NF Table):

Subject	Teacher	TeacherPhone
Math	Sangeetha	9991112222
Science	Tharun	8881112222

❌ Problem: "TeacherPhone" depends on "Teacher", not directly on "Subject".
👉 **3NF Conversion:**

Subjects Table

Subject	Teacher
Math	Sangeetha
Science	Tharun

Teachers Table

Teacher	TeacherPhone
Sangeetha	9991112222
Tharun	8881112222

A transitive dependency is a type of functional dependency in a database where a non-prime attribute (an attribute that is not part of any candidate key) depends on another non-prime attribute, instead of directly depending on the primary key.

In simpler words:
👉 A transitive dependency occurs when A → B and B → C, so A → C, where:

- **A is usually the primary key.**
- **B is a non-key attribute.**
- **C is another non-key attribute.**

Example:

Consider a table Students:

StudentID	StudentName	Department	HOD
101	Kapil	CSE	Dr. Rao
102	Sneha	ECE	Dr. Mehta

StudentID → Department (Student ID determines Department)

- **Department → HOD (Department determines HOD)**
- **So, by transitivity: StudentID → HOD**

Here, HOD is transitively dependent on StudentID via Department.

Why it matters:

- **Transitive dependencies cause data redundancy and update anomalies.**
- **To remove them, we normalize the table into Third Normal Form (3NF) by separating the transitive part into another table.**

For the example:
Students Table:

StudentID	StudentName	Department
101	Kapil	CSE
102	Sneha	ECE

Department Table:

Department	HOD
CSE	Dr. Rao
ECE	Dr. Mehta

👉 In short:
Transitive dependency = Non-key attribute depends on another non-key attribute (through the primary key).
Would you like me to also show you a diagram/visual illustration of transitive dependency to make it crystal clear?

BCNF (Boyce-Codd Normal Form)

- A stronger version of **3NF**.
- Every determinant must be a **candidate key**.

👉 Example:
If a course has multiple instructors, but one instructor can only teach one subject → BCNF helps resolve this by further splitting.

🔹 Benefits of Normalization

- Eliminates redundancy.
- Prevents anomalies.
- Ensures data integrity.
- Easier to maintain and scale.

✅ **Summary:**

- **1NF:** Atomic values, no repeating groups.
- **2NF:** No partial dependency.
- **3NF:** No transitive dependency.
- **BCNF:** Every determinant must be a candidate key.