RDBMS - Normalization

By Dhandapani Yedappalli Krishnamurthi Sep 30, 2025

📌 What is Normalization in SQL?

👉 **Normalization** is the process of **organizing data in a database** to:

- Reduce **redundancy** (duplicate data)
- Improve **data integrity**
- Make queries more **efficient**

It breaks a big, messy table into smaller, related tables using **primary keys** and **foreign keys**.

📊 Example Before Normalization (Unnormalized Table)

Suppose we have this table:

StudentID	StudentName	Course1	Course2	Course3
1	Anil	Math	Physics	NULL
2	Priya	Math	English	History
3	Ramesh	Biology	NULL	NULL

🔴 Problems:

- Repeated course names across columns
- Hard to add a new course (need new column)
- Wasted space with NULLs

✅ Step-by-Step Normalization Forms

1NF (First Normal Form)

- Remove repeating groups
- Each column should hold only **atomic (single) values**

👉 Convert courses into rows instead of multiple columns:

StudentID	StudentName	Course
1	Anil	Math
1	Anil	Physics
2	Priya	Math
2	Priya	English
2	Priya	History
3	Ramesh	Biology

2NF (Second Normal Form)

- Table should be in **1NF**
- Remove **partial dependency** (non-key column depends only on part of composite key)

👉 Split into 2 tables:
Students

StudentID	StudentName
1	Anil
2	Priya
3	Ramesh

Enrollments

StudentID	Course
1	Math
1	Physics
2	Math
2	English
2	History
3	Biology

3NF (Third Normal Form)

- Table should be in **2NF**
- Remove **transitive dependency** (non-key column depends on another non-key column)

👉 If we had course details (like Course → Teacher), we separate further:
Courses

CourseID	CourseName	Teacher
101	Math	Mr. Raj
102	Physics	Ms. Meena
103	English	Mr. Arun
104	History	Ms. Sita
105	Biology	Dr. Kumar

Enrollments

StudentID	CourseID
1	101
1	102
2	101
2	103
2	104
3	105

🎯 Why Normalize?

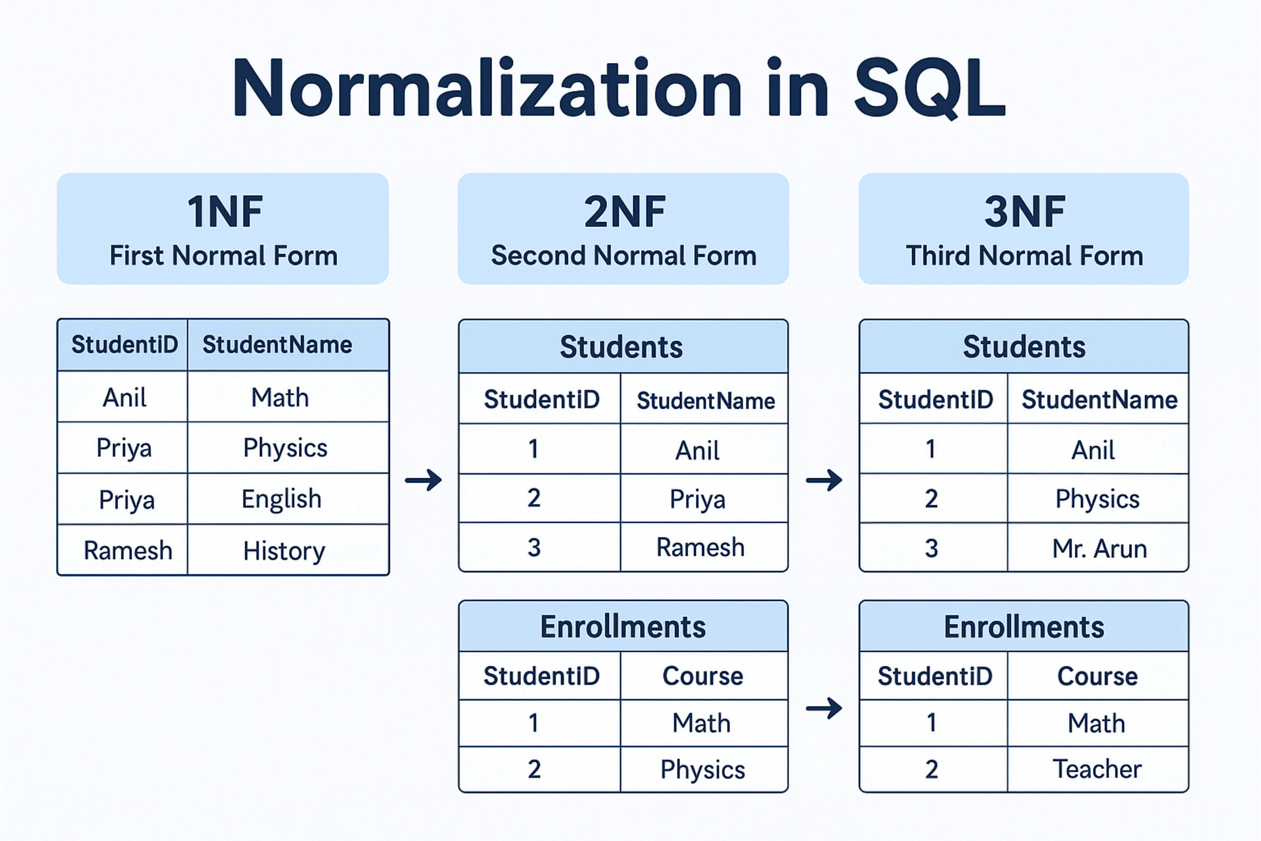
✅ Reduces data duplication
✅ Makes database flexible (easy to add new data)
✅ Maintains data integrity (consistent updates)
✅ Saves storage

⚖️ When Not to Normalize?

- In **Data Warehousing / BI systems**, sometimes we use **Denormalization** (like star schema) for faster reporting.

📌 Quick Recap of Normal Forms

1. **1NF** → No repeating groups (atomic values only)
2. **2NF** → No partial dependency (eliminate redundancy with composite keys)
3. **3NF** → No transitive dependency (non-key columns depend only on key)
   (Higher forms exist, but for beginners, up to 3NF is enough)



Normalization in SQL

1NF
First Normal Form

StudentID	StudentName
Anil	Math
Priya	Physics
Priya	English
Ramesh	History

→

2NF
Second Normal Form

Students

StudentID	StudentName
1	Anil
2	Priya
3	Ramesh

Enrollments

StudentID	Course
1	Math
2	Physics

→

3NF
Third Normal Form

Students

StudentID	StudentName
1	Anil
2	Physics
3	Mr. Arun

Enrollments

StudentID	Course
1	Math
2	Teacher

https://popsql.com/blog/normalization-in-sql

https://www.youtube.com/watch?v=GFQaEYEc8_8