

NORMALIZATION

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

Reduce redundancy

Avoid update, insert, and delete anomalies

Maintain consistency

Improve efficiency

1. FIRST NORMAL FORM (1NF)

Definition

A relation is in **1NF** if:

All attributes contain **atomic (indivisible)** values.

No multivalued or repeating attributes.

Each record is unique.

Example (NOT in 1NF)

Employee Table

EmpID	EmpName	Skills	PhoneNumbers
101	Ravi	Java, Python	98765, 87654
102	Priya	SQL	76543

Problems:

Skills contains multiple values → *multivalued*

PhoneNumbers contains multiple values

Violates 1NF

Convert to 1NF

Split multi-valued attributes into separate rows.

Employee Table (1NF)

EmpID EmpName Skill PhoneNumber

101 Ravi Java 98765

101 Ravi Python 87654

102 Priya SQL 76543

Now all values are atomic → **1NF achieved**

2. SECOND NORMAL FORM (2NF)

Definition

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

It is in **1NF**

No partial dependency exists

A non-key attribute should NOT depend on part of a composite key.

☞ Applies only if a table has a **composite primary key**.

Example of Partial Dependency

OrderDetails(OrderID, ProductID → Composite Key)

OrderID ProductID ProductName Price Qty

Functional Dependencies:

{OrderID, ProductID} → Qty

ProductID → ProductName, Price (**partial dependency**)

Fixing 2NF Violation

Split into two tables:

1. Product Table

ProductID ProductName Price

2. OrderDetails

OrderID ProductID Qty

Now:

No partial dependency

Achieved 2NF

3. THIRD NORMAL FORM (3NF)

Definition

A relation is in 3NF if:

It is in 2NF

No **transitive dependency** exists:

$X \rightarrow Y$ and $Y \rightarrow Z$

means $X \rightarrow Z$ is *transitive*

Example of Transitive Dependency

Student Table

RollNo Name DeptID DeptName

Functional Dependencies:

$\text{RollNo} \rightarrow \text{DeptID}$

$\text{DeptID} \rightarrow \text{DeptName}$

So: **$\text{RollNo} \rightarrow \text{DeptName}$** (transitive)

Convert to 3NF

Split the table:

1. Student

| RollNo | Name | DeptID |

2. Department

| DeptID | DeptName |

Now:

No transitive dependency

Achieved 3NF

4. BOYCE–CODD NORMAL FORM (BCNF)

Definition

A relation is in **BCNF** if:

For every functional dependency $X \rightarrow Y$,
X must be a superkey.

More strict than 3NF.

BCNF Example

Lecture Hall Allocation Table:

Course Instructor Room

Assume:

Instructor $\rightarrow$ Room

Room $\rightarrow$ Instructor

(Course, Instructor) $\rightarrow$ Room

Problem:

Instructor is NOT a key

Room is NOT a key

Both determine other values $\rightarrow$ violates BCNF

Convert to BCNF

Split:

1. Instructor-Room

| Instructor | Room |

2. Course-Instructor

| Course | Instructor |

Now FDs:

Instructor $\rightarrow$ Room (Instructor is key)

Course $\rightarrow$ Instructor (Course is key)

BCNF achieved.

5. FOURTH NORMAL FORM (4NF)

Definition

A table is in **4NF** if:

It is in BCNF

It has **no non-trivial multivalued dependencies (MVDs)**

Example of MVD

A professor may teach **multiple subjects**

A professor may have **multiple offices**

Prof	Subject	Office
------	---------	--------

A	DBMS	Block-1
---	------	---------

A	DBMS	Block-2
---	------	---------

A	Networks	Block-1
---	----------	---------

A	Networks	Block-2
---	----------	---------

Here:

Prof $\twoheadrightarrow$ Subject

Prof $\twoheadrightarrow$ Office

Two independent multi-valued attributes $\rightarrow$ violates 4NF

Convert to 4NF

1. Prof-Subject

| Prof | Subject |

2. Prof-Office

| Prof | Office |

Now:

No multi-valued dependency in a single table

Achieved **4NF**

6. FIFTH NORMAL FORM (5NF)

Definition

A relation is in **5NF** if:

It is in 4NF

It cannot be decomposed further without losing information

Deals with **join dependencies**

Used when:

Data comes from **three or more independent entities**

Example

Supplier-Product-Region table:

Supplier Product Region

Assume:

A supplier can supply multiple products

A product can be supplied in multiple regions

Supplier works in multiple regions

These combine to create many rows.

To achieve 5NF, split into:

1. Supplier-Product

2. Product-Region

3. Supplier-Region

Now joins reconstruct the original table without anomalies.

CASE STUDY 1: Normalizing a RETAIL INVENTORY DATABASE**UNNORMALIZED RetailInventory Table**

ItemID	ItemName	Supplier	SupplierPhone	StoreLocations	PurchasePrices
101	Pen	ABC Co	98765	Chennai, Pune	10, 9.5

Problems

StoreLocations is multi-valued

PurchasePrices is multi-valued

Supplier details repeated → redundancy

Step 1: Convert to 1NF

Break multi-valued attributes:

ItemID	ItemName	Supplier	SupplierPhone	StoreLocation	PurchasePrice
101	Pen	ABC Co	98765	Chennai	10
101	Pen	ABC Co	98765	Pune	9.5

Step 2: Remove Partial Dependencies (2NF)

Assume Key = {ItemID, StoreLocation}

But Supplier depends **only on ItemID** → partial dependency.

Split:

Item

| ItemID | ItemName | Supplier |

ItemLocation

| ItemID | StoreLocation | PurchasePrice |

Supplier

| Supplier | SupplierPhone |

Step 3: Remove Transitive Dependencies (3NF)

Supplier → SupplierPhone is fine
No transitive dependencies remain.

Final BCNF / 4NF / 5NF

No multivalued dependencies.
Final database:

- ✓ **Item(ItemID, ItemName, SupplierID)**
 - ✓ **Supplier(SupplierID, SupplierName, SupplierPhone)**
 - ✓ **ItemLocation(ItemID, StoreLocation, PurchasePrice)**
-

CASE STUDY 2: Normalizing a LIBRARY DATABASE

UNNORMALIZED Library Table

BookID	Title	Authors	Member	BorrowDate
B101	DBMS	Navathe, Korth	Suresh	2023-01-01

Problems

Authors is multi-valued

Book borrowed by a member repeatedly → redundancy

1NF

Split authors:

BookID	Title	Author	Member	BorrowDate
--------	-------	--------	--------	------------

2NF & 3NF

Dependencies:

BookID → Title

BookID → Author (M:N relationship → separate table)

Member $\rightarrow$ BorrowDate depends on (BookID, Member)

Final 3NF Schema

✓ **Book(BookID, Title)**

✓ **Author(AuthorID, AuthorName)**

✓ **BookAuthor(BookID, AuthorID)**

✓ **Member(MemberID, MemberName)**

✓ **Borrow(BookID, MemberID, BorrowDate)**

FINAL SUMMARY TABLE OF NORMAL FORMS

Normal Form	Removes	Violation Example
1NF	Multivalued attributes, repeating groups	Skills = {Java, Python}
2NF	Partial dependency	ProductID $\rightarrow$ ProductName
3NF	Transitive dependency	DeptID $\rightarrow$ DeptName
BCNF	Determinant not a key	Instructor $\rightarrow$ Room
4NF	Multivalued dependency	Prof $\twoheadrightarrow$ Subject, Office
5NF	Join dependency	Supplier–Product–Region