

# Normalization in DBMS - Notes

Normalization in DBMS is a process used to minimize data redundancy and eliminate undesirable characteristics like insertion, update, and deletion anomalies. It divides larger tables into smaller, more manageable pieces and establishes relationships between them.

## Objectives of Normalization:

- To remove redundant data.
- To ensure data dependencies make sense.
- To protect data integrity.
- To make database design more efficient and flexible.

## Normal Forms in DBMS:

Normalization is carried out through a series of 'normal forms' — each form represents a specific level of database organization. The most commonly used forms are:

### 1NF (First Normal Form):

A table is in 1NF if:

- Each cell contains only a single (atomic) value.
- Each record is unique.

**Example:** Splitting multi-valued attributes into separate rows.

### 2NF (Second Normal Form):

A table is in 2NF if:

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

**Example:** Removing partial dependency (when a non-key attribute depends on only part of a composite key).

### 3NF (Third Normal Form):

A table is in 3NF if:

- It is in 2NF.
- It has no transitive dependencies (non-key attributes should not depend on other non-key attributes).

**Example:** Removing columns that are not directly dependent on the primary key.

### BCNF (Boyce-Codd Normal Form):

A stronger version of 3NF. A table is in BCNF if:

- It is in 3NF.
- For every functional dependency ( $X \rightarrow Y$ ), X should be a super key.

**Example:** When multiple candidate keys exist and anomalies still appear in 3NF.

## Advantages of Normalization:

- Reduces data redundancy.
- Improves data integrity.
- Easier maintenance and modification.

- Saves storage space.
- Enhances query performance in some cases.

**Disadvantages of Normalization:**

- Complex queries due to multiple table joins.
- May affect performance in read-heavy systems.
- Requires careful design and understanding of data relationships.

**Conclusion:**

Normalization is a vital process in database design that ensures efficiency, consistency, and integrity. However, in real-world systems, a balance is often maintained between normalization and performance by selectively denormalizing data where necessary.