Normalization in DBMS:

Normalization in Database Management Systems (DBMS) is the process of organizing the attributes (columns) and relations (tables) in a database to reduce redundancy and dependency. The goal is to ensure that the data is stored in such a way that it minimizes data anomalies, such as update, insert, and delete anomalies. This is achieved by dividing large tables into smaller, more manageable ones while ensuring that relationships between the tables are well-defined.

Here are the common **normal forms** used in normalization, along with examples for each:

1. First Normal Form (1NF)

A table is in 1NF if:

- It has only atomic (indivisible) values.
- Each column contains only one value per record.
- Each record is unique (no duplicate rows).

Example: Consider the following table:

Student_ID	Name	Subjects
1	John	Math, Science
2	Mary	History, English

To convert this into 1NF, we split the multiple values in the "Subjects" column into individual rows:

Student_ID	Name	Subject
1	John	Math
1	John	Science
2	Mary	History
2	Mary	English

Now, each column contains atomic values.

---

2. Second Normal Form (2NF)

A table is in 2NF if:

- It is in 1NF.
- It has no partial dependency, meaning that all non-key attributes are fully dependent on the primary key.

Example: Consider the following table:

Student_ID	Course_ID	Student_Name	Course_Name
1	C101	John	Math
1	C102	John	Science
2	C101	Mary	Math

Here, the primary key is a composite key: (Student_ID, Course_ID).

- "Student_Name" depends only on "Student_ID" (partial dependency).
- "Course_Name" depends only on "Course_ID" (partial dependency).

To convert this into 2NF, we remove the partial dependencies by creating separate tables:
Student Table:

Student_ID	Student_Name
1	John
2	Mary

Course Table:

Course_ID	Course_Name
C101	Math
C102	Science

Enrollment Table:

Student_ID	Course_ID
1	C101

1	C102
2	C101

Now, all non-key attributes are fully dependent on the primary key.

---

3. Third Normal Form (3NF)

A table is in 3NF if:

- It is in 2NF.
- It has no transitive dependency, meaning non-key attributes do not depend on other non-key attributes.

Example: Consider the following table:

Student_ID	Name	Department	Dept_Head
1	John	CS	Dr. Smith
2	Mary	Math	Dr. Brown

Here, "Dept_Head" depends on "Department", not on the primary key "Student_ID" (transitive dependency).

To convert it into 3NF, we separate the "Department" and "Dept_Head" into a new table: **Student Table**:

Student_ID	Name	Department
1	John	CS
2	Mary	Math

Department Table:

Department	Dept_Head
CS	Dr. Smith
Math	Dr. Brown

Now, there is no transitive dependency.

4. Boyce-Codd Normal Form (BCNF)

A table is in BCNF if:

- It is in 3NF.
- For every non-trivial functional dependency, the left side of the dependency is a superkey.

Example: Consider the following table:

Course_ID	Instructor	Room
C101	Dr. Smith	R101
C102	Dr. Brown	R102

Here, "Room" depends on "Course_ID" (a candidate key), but "Instructor" depends on "Course_ID", and "Course_ID" is not a superkey.

To convert this into BCNF, we split the table into two: **Course Table**:

Course_ID	Instructor
C101	Dr. Smith
C102	Dr. Brown

Room Table:

Course_ID	Room
C101	R101
C102	R102

Now, both "Instructor" and "Room" are functionally dependent on the superkey (Course_ID).

---

5. Fourth Normal Form (4NF)

A table is in 4NF if:

- It is in BCNF.
- It has no multi-valued dependencies, where one attribute determines multiple independent values.

Example: Consider the following table:

Student_ID	Hobby	Language
1	Cricket	English
1	Football	Spanish

Here, "Student_ID" determines two independent values: "Hobby" and "Language". These are multivalued dependencies.

To convert this into 4NF, we split it into two tables: **Hobby Table**:

Student_ID	Hobby
1	Cricket
1	Football

Language Table:

Student_ID	Language
1	English
1	Spanish

Now, there are no multi-valued dependencies.

---

6. Fifth Normal Form (5NF)

A table is in 5NF if:

- It is in 4NF.
- It has no join dependency, meaning that data can be reconstructed by joining smaller tables without any loss of information.

This is the highest level of normalization and is often rare in practice. A table is typically decomposed into smaller tables that can be joined back together to recover the original data.

Summary of Normal Forms:

- **1NF**: Eliminate duplicate columns and ensure atomicity.
- **2NF**: Eliminate partial dependencies (i.e., every non-prime attribute must depend on the entire primary key).
- **3NF**: Eliminate transitive dependencies (i.e., non-prime attributes must not depend on other non-prime attributes).
- **BCNF**: Ensure that every determinant is a superkey.
- **4NF**: Eliminate multi-valued dependencies.
- **5NF**: Eliminate join dependencies and ensure that data is reconstructible from smaller tables.

This progression of normal forms helps to ensure that the database structure is efficient and that redundancy is minimized, improving data integrity.