## Normalization in DBMS

Normalization is a process of organizing data in a database to reduce redundancy and dependency. It helps in maintaining data consistency, accuracy, and integrity. Normalization is an important concept in database design as it helps in optimizing database performance and reducing storage requirements. Normalization is a technique used to organize data in a database by dividing large tables into smaller tables and defining relationships between them. It helps in eliminating redundant data and improving data consistency.

For example, in a hotel booking database, instead of storing all customer information in one table, we can divide it into multiple tables such as customer details, room details, and booking details. This way, we can avoid duplicate entries and maintain data accuracy.

**First Normal Form (1NF):**

First Normal Form (1NF) is the first level of normalization in which each table has a primary key and all columns contain atomic values. It eliminates duplicate data and ensures data consistency.

For example, in a hotel booking database, we can create separate tables for customer details, room details, and booking details. Each table would have a primary key, such as customer ID, room ID, and booking ID respectively. This way, we can avoid duplicate entries and maintain data accuracy.

## Second Normal Form (2NF):

Second Normal Form (2NF) is the second level of normalization in which all non-key columns are dependent on the primary key. It eliminates partial dependencies and ensures data consistency.

For example, in a hotel booking database, we can create separate tables for customer details, room details, and booking details. Each table would have a primary key, such as customer ID, room ID, and booking ID respectively. The non-key columns, such as customer name and room type, would be dependent on the primary key. This way, we can avoid partial dependencies and maintain data accuracy.

**Third Normal Form (3NF):**

Third Normal Form (3NF) is the third level of normalization in which all non-key columns are independent of each other. It eliminates transitive dependencies and ensures data consistency.

For example, in a hotel booking database, we can create separate tables for customer details, room details, and booking details. Each table would have a primary key, such as customer ID, room ID, and booking ID respectively. The non-key columns, such as customer name and room type, would be independent of each other. We can create another table for room types and link it to the room details table using a foreign key. This way, we can avoid transitive dependencies and maintain data accuracy.