

# AGENDA:

## Optimizing Database Management:

Check out this concept



# Normalization

# and

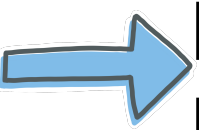
# Denormalization

DON'T  
FORGET

Follow me on

<https://www.linkedin.com/in/dhwanimehta05/>





# Data Normalization vs Data Denormalization



## Data Normalization

### What is it?

Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. It involves splitting large tables into smaller, related tables and linking them using relationships.

### Example:

Imagine you run a library, and you have a database to keep track of books and authors. Without normalization, your database might look like this:

BookID	BookTitle	AuthorName	AuthorBirthdate
1	To Kill a Mockingbird	Harper Lee	1926-04-28
2	Go Set a Watchman	Harper Lee	1926-04-28
3	1984	George Orwell	1903-06-25

In this unnormalized table, if you need to change the birthdate of Harper Lee, you have to update multiple rows. This is redundant and error-prone.

After normalization, you might have two tables:

### Books Table:

BookID	BookTitle	AuthorID
1	To Kill a Mockingbird	1
2	Go Set a Watchman	1
3	1984	2

### Authors Table:

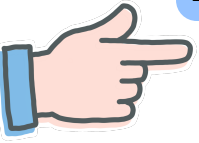
AuthorID	AuthorName	AuthorBirthdate
1	Harper Lee	1926-04-28
2	George Orwell	1903-06-25

### Advantages:



- Reduces data redundancy.
- Ensures data consistency and integrity.
- Easier to update and maintain data.

#### Disadvantages:



- More complex queries involving joins.
- Can lead to performance issues with very large datasets and complex joins.

## Data Denormalization

### What is it?

Denormalization is the process of combining normalized tables into larger tables to improve read performance. It involves introducing redundancy to speed up data retrieval.

### Example:

Using the same library example, a denormalized version of your database might look like the initial unnormalized table:

BookID	BookTitle	AuthorName	AuthorBirthdate
1	To Kill a Mockingbird	Harper Lee	1926-04-28
2	Go Set a Watchman	Harper Lee	1926-04-28
3	1984	George Orwell	1903-06-25

### Advantages:

- Faster read operations since all relevant data is in a single table.
- Simpler queries without the need for joins.

### Disadvantages:

- Increased data redundancy, leading to potential inconsistency.
- More storage space needed.
- More complex and error-prone updates.



### Scenario

Imagine you are managing an online store with thousands of products and categories.



- **Normalization:** You might have separate tables for products, categories, and suppliers. This helps maintain clean data and makes it easier to manage updates. If you need to change a supplier's contact information, you only update it in one place.
- **Denormalization:** If your main focus is on quickly retrieving product information for customers browsing the website, you might combine the product and category data into a single table to speed up read operations. This approach sacrifices some data integrity for performance.

## In Summary



- **Normalization** is about organizing data to minimize redundancy and ensure consistency. It's like organizing your kitchen, putting all spices in one drawer and utensils in another for easy maintenance and order.
- **Denormalization** is about making data retrieval faster by combining tables, even if it means having duplicate data. It's like having a frequently used set of spices on the countertop for quick access while cooking, even if you already have them in the drawer.

Both techniques have their place, and the choice depends on your specific needs: whether you prioritize data integrity and maintenance (normalization) or read performance (denormalization).

