



# SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)  
Coimbatore - 641035.



*Accredited by NBA - AICTE and Accredited by NAAC - UGC with 'A++' Grade  
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai*

## Department of Computer Applications

Course Code: 23CAT606

Course Name: Java Programming

Unit : V

Topic : CRUD Operation using DAO  
and Spring API

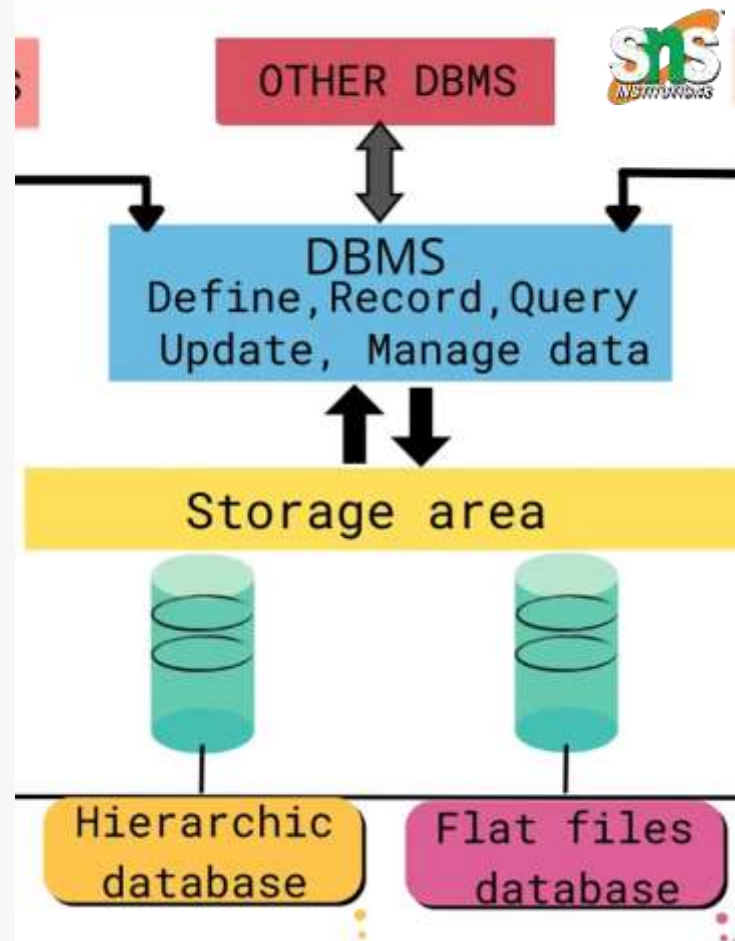




# Introduction to CRUD Operations

CRUD operations are the fundamental building blocks of database management. They enable you to Create, Read, Update, and Delete data, providing the essential functionality for any application that interacts with a database.

## DATABASE MANAGEMENT SYSTEM





# Understanding Data Access Object (DAO) Pattern

## Separation of Concerns

DAO pattern separates the data access logic from the business logic, making the application more modular and maintainable.

## Abstraction

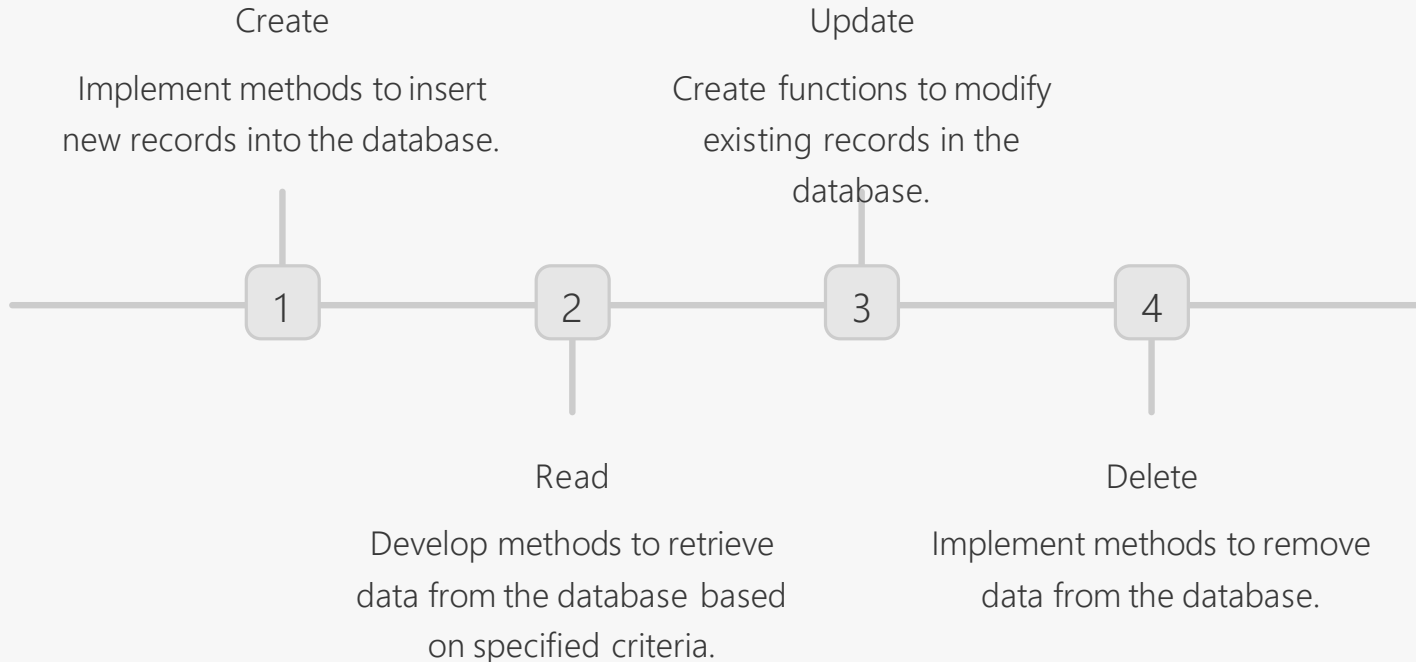
DAO provides an abstraction layer, allowing the application to work with a consistent API regardless of the underlying database implementation.

## Testability

DAO makes it easier to write unit tests for the data access layer, improving the overall quality of the application.



# Implementing CRUD Operations with DAO





# Integrating DAO with Spring API

1

## Dependency Injection

Use Spring's Dependency Injection to manage the lifecycle of DAO components.

3

## Exception Handling

Integrate Spring's exception handling mechanisms with the DAO layer.

2

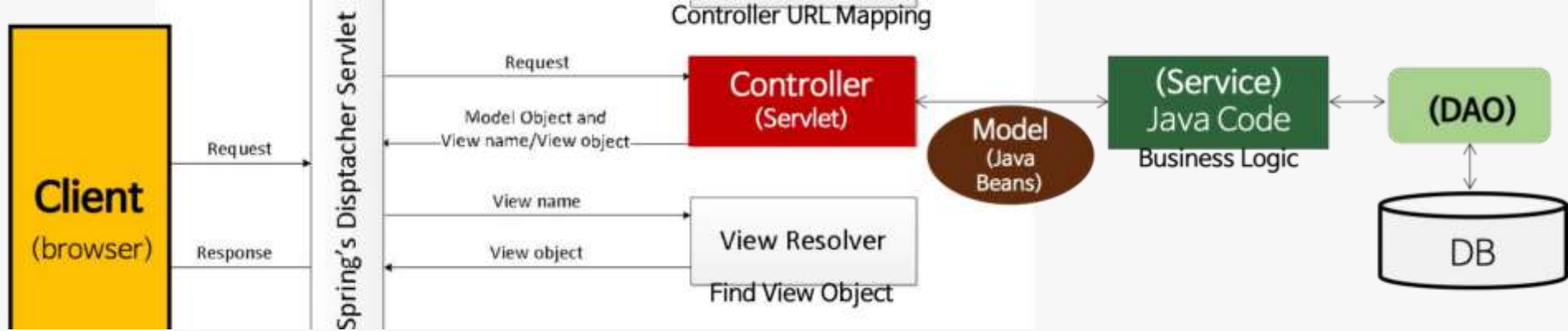
## Transaction Management

Leverage Spring's transaction management capabilities to ensure data integrity.

4

## Data Mapping

Utilize Spring's data mapping tools to seamlessly convert between database and application objects.



## Spring MVC and CRUD Operations

### Controller Layer

Implement CRUD actions as Spring MVC controller methods, handling incoming requests and invoking the appropriate DAO operations.

### View Layer

Create views (e.g., JSPs, Thymeleaf templates) to render the data retrieved by the DAO layer and present it to the user.

### Service Layer

Encapsulate the business logic and coordinate the interaction between the controller and DAO layers.



# Exception Handling in CRUD Operations

1

## Anticipate Exceptions

Identify potential exceptions that can occur during CRUD operations and create appropriate exception handling mechanisms.

2

## Centralized Handling

Implement a centralized exception handling strategy, using Spring's exception handling features.

3

## Meaningful Responses

Provide meaningful error messages and responses to users, ensuring a smooth user experience.



# Optimizing CRUD Performance with Spring



## Caching

Leverage Spring's caching abstraction to improve the performance of read operations.



## Batch Processing

Implement batch processing for create, update, and delete operations to reduce database roundtrips.



## Asynchronous Processing

Use Spring's async support to offload long-running CRUD operations to separate threads.





# Conclusion and Best Practices

|                          |   |
|--------------------------|---|
| Separation of Concerns   | Ensure a clear separation between the data access, business, and presentation layers.                     |
| Error Handling           | Implement a robust and user-friendly exception handling strategy.   |
| Performance Optimization | Apply techniques like caching, batch processing, and asynchronous processing to improve CRUD performance. |
| Testability              | Write comprehensive unit and integration tests to ensure the reliability of CRUD operations.              |