




JULY 31, 2023

ORM CONCEPT WITH HIBERNATE

ASSIGNMENT I

LECTURER: UDARA SAN
BSC (HONS) COMPUTER SCIENCE
IJSE : Panadura



Part 1: CRUD Operations on a Single Entity

Objective:

In this part of the assignment, the objective is to assess the students' ability to use Hibernate for performing CRUD operations on a single entity. Students will focus on the `Book` entity and its associated attributes.

Tasks:

1. Set up a Java project and configure Hibernate with a suitable database (e.g., MySQL, H2) for this assignment.
2. Create the necessary Java classes representing the `Book` entity with appropriate annotations for mapping.
3. Implement the following CRUD operations for the `Book` entity:
 - Create a new book and save it to the database.
 - Retrieve a book by its ID from the database.
 - Update an existing book's attributes (e.g., title, ISBN) and save the changes to the database.
 - Delete a book from the database.
4. Test the CRUD operations to ensure they are working correctly by using a simple console-based application.

Part 2: Relationships between Entities (One-to-One, One-to-Many, Many-to-Many)

Objective:

In this part of the assignment, the objective is to assess the students' understanding of creating and managing various types of relationships between entities using Hibernate. The students will work with two entities: `Book` and `Author`.

Tasks:

1. Create the necessary Java classes representing the `Book` and `Author` entities with appropriate annotations for mapping.
2. Implement the following relationships between entities using Hibernate annotations:
 - One-to-One: Each book can have only one author. Update the `Book` entity to have a one-to-one relationship with the `Author` entity.
 - One-to-Many: Each author can have multiple books. Update the `Author` entity to have a one-to-many relationship with the `Book` entity.
 - Many-to-Many: Each book can have multiple authors, and each author can have multiple books. Implement a many-to-many relationship between the `Book` and `Author` entities.

3. Implement the following tasks related to relationships:

- Save the `Author` entity independently, and then associate it with the corresponding `Book` using the one-to-one relationship.
- Save the `Book` entities independently, and then associate each book with its respective `Author` using the one-to-many relationship.
- Implement the many-to-many relationship by introducing a join table (association table) to manage the relationship between `Book` and `Author` entities. Save `Book` and `Author` entities independently, and then associate them using the many-to-many relationship.

4. Test the relationships by creating authors and books, ensuring that the relationships are correctly maintained.

Submission:

Students should submit their Java project, including all the source code and configuration files, as a GitHub public URL. They should create a public GitHub repository named appropriately, such as "Hibernate-ORM-Assignment" or similar. The repository should contain separate folders for Part 1 and Part 2 of the assignment, each containing the relevant code.

In the repository, students should include:

- The Java source code for both parts of the assignment.
- Any required configuration files (e.g., `pom.xml` if using Maven).
- A `README.md` file providing a brief description of the assignment, instructions on how to run the application, and any important notes.

Students should ensure that the repository is publicly accessible and does not contain any sensitive information.

Note:

Encourage students to follow best practices for code organization, and version control using Git. Remind them to avoid committing any sensitive information, such as database credentials or personal data. Emphasize that the submitted repository should be complete, self-contained, and able to be cloned and run without additional dependencies or configurations.