Library Management System Case Study

Reader.java

```
package com.example.library.entity;
import jakarta.persistence.*;
import lombok.*;
import java.util.List;
@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Reader {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;
    private String email;

    @OneToMany(mappedBy = "reader")
    private List<Book> books;
}
```

Author.java

```
package com.example.library.entity;
import jakarta.persistence.*;
import lombok.*;
```

```
import java.util.List;
@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Author {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    @OneToMany(mappedBy = "author")
    private List<Book> books;
}
```

Category.java

```
package com.example.library.entity;
import jakarta.persistence.*;
import lombok.*;
import java.util.List;
@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Category {
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    @OneToMany(mappedBy = "category")
    private List<Book> books;
}
```

Book.java

```
package com.example.library.entity;
import jakarta.persistence.*;
import lombok.*;
import java.time.LocalDate;
@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Book {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String title;
    private LocalDate publishDate;

    @ManyToOne
```

```
    @JoinColumn(name = "reader_id")
    private Reader reader;

    @ManyToOne
    @JoinColumn(name = "category_id")
    private Category category;

    @ManyToOne
    @JoinColumn(name = "author_id")
    private Author author;
}
```

Repositories (com.example.library.repository)

```
public interface ReaderRepository extends JpaRepository<Reader, Long> {}
public interface BookRepository extends JpaRepository<Book, Long> {}
public interface AuthorRepository extends JpaRepository<Author, Long> {}
public interface CategoryRepository extends JpaRepository<Category, Long> {}
```

Controller (LibraryController.java)

```
package com.example.library.controller;
import com.example.library.entity.*;
import com.example.library.repository.*;
import lombok.RequiredArgsConstructor;
import org.springframework.web.bind.annotation.*;

import java.util.List;
@RestController
@RequestMapping("/api")
@RequiredArgsConstructor
```

```
public class LibraryController {

    private final ReaderRepository readerRepo;
    private final BookRepository bookRepo;
    private final AuthorRepository authorRepo;
    private final CategoryRepository categoryRepo;

    // Add Category
    @PostMapping("/categories")
    public Category addCategory(@RequestBody Category category) {
        return categoryRepo.save(category);
    }

    @GetMapping("/categories")
    public List<Category> getCategories() {
        return categoryRepo.findAll();
    }

    // Add Author
    @PostMapping("/authors")
    public Author addAuthor(@RequestBody Author author) {
        return authorRepo.save(author);
    }

    @GetMapping("/authors")
    public List<Author> getAuthors() {
        return authorRepo.findAll();
```

```
    }

    // Add Reader
    @PostMapping("/readers")
    public Reader addReader(@RequestBody Reader reader) {
        return readerRepo.save(reader);
    }

    @GetMapping("/readers")
    public List<Reader> getReaders() {
        return readerRepo.findAll();
    }

    // Add Book
    @PostMapping("/books")
    public Book addBook(@RequestBody Book book) {
        return bookRepo.save(book);
    }

    @GetMapping("/books")
    public List<Book> getBooks() {
        return bookRepo.findAll();
    }
}
```

---

application.properties

```
spring.datasource.url=jdbc:mysql://localhost:3306/library_db
```

```
spring.datasource.username=root
spring.datasource.password=yourpassword
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

Main Class

```
package com.example.library;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
public class LibraryManagementApplication {
    public static void main(String[] args) {
        SpringApplication.run(LibraryManagementApplication.class, args);
    }
}
```

MySQL Command

library_db.sql

```
-- Create the database
CREATE DATABASE IF NOT EXISTS library_db;
USE library_db;

-- Create Reader table
CREATE TABLE reader (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL
```

```
);

-- Create Author table
CREATE TABLE author (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL
);

-- Create Category table
CREATE TABLE category (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL
);

-- Create Book table with foreign keys
CREATE TABLE book (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(200) NOT NULL,
    publish_date DATE,
    reader_id BIGINT,
    category_id BIGINT,
    author_id BIGINT,

    CONSTRAINT fk_book_reader FOREIGN KEY (reader_id) REFERENCES reader(id)
ON DELETE SET NULL,
    CONSTRAINT fk_book_category FOREIGN KEY (category_id) REFERENCES
category(id),
    CONSTRAINT fk_book_author FOREIGN KEY (author_id) REFERENCES author(id)
```

);

-- Sample Data Insertion

-- Insert into Category

INSERT INTO category (name) VALUES ('Fiction'), ('Technology'), ('History');

-- Insert into Author

INSERT INTO author (name) VALUES ('George Orwell'), ('Isaac Asimov'), ('Yuval Noah Harari');

-- Insert into Reader

INSERT INTO reader (name, email) VALUES

('Alice', 'alice@gmail.com'),

('Bob', 'bob@example.**com');**

-- Insert into Book

INSERT INTO book (title, publish_date, reader_id, category_id, author_id) VALUES

('1984', '1949-06-08', 1, 1, 1),

('Foundation', '1951-01-01', NULL, 2, 2),

('Sapiens', '2011-06-04', 2, 3, 3);