**Exercise 001:**

Let's consider a simple example of a one-to-many relationship between two entities: **Department** and **Employee**. Each department can have multiple employees.

1. Define the **Department(id, name)** and **Employee(id, name, salary)** entities with a one-to-many relationship.
2. Create a sample Java program to insert a department and its employees into the database.
3. Retrieve and display the employees of a specific department.

**Exercise002:**

Let's consider a simple example of a one-to-many relationship between two entities: **Course** and **Student**. Each course can have multiple students.

1. Define the **Course** and **Student** entities with a one-to-many relationship.
2. Create a sample Java program to insert courses and students into the database.
3. Retrieve and display the students enrolled in a specific course.

**Exercise003:**

Let's create a more comprehensive example that includes all CRUD operations (Create, Read, Update, Delete) for a simple one-to-many relationship between **Author** and **Book** entities.

Authors can write multiple books, and each book is associated with one author.

**Exercise004:**

In this exercise, we'll create a simple library management system with two entities: **Library** and **Book**, where each library can have multiple books.

Define the Library and Book entities with a one-to-many relationship.

Implement CRUD operations for both libraries and books using Hibernate. The operations should include creating, reading, updating, and deleting records.

Create a Java program to demonstrate these CRUD operations.