# Các loại quan hệ

Nhat Tung Le

# One-To-One (Một một)

- Mối quan hệ One-To-One xác định một sự tương ứng duy nhất giữa hai bản ghi trong hai bảng khác nhau.
- Trong mối quan hệ One-To-One, một bản ghi trong bảng A chỉ liên kết với một bản ghi duy nhất trong bảng B và ngược lại.
- Để ám chỉ mối quan hệ One-To-One, bạn có thể sử dụng các annotation như @OneToOne hoặc @JoinColumn.

```java
@Entity
public class Teacher {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    @OneToOne(mappedBy = "teacher", cascade = CascadeType.
    private TeacherDetail teacherDetail;

    // Constructors, getters, setters, etc.
}
```

72

50 x 5 = 250 kb

x 500

OneToOne

```java
@Entity
public class TeacherDetail {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String address;

    @OneToOne
    @JoinColumn(name = "teacher_id")
    private Teacher teacher;

    // Constructors, getters, setters, etc.
}
```

48

# One-To-Many (Một - Nhiều)

- Mối quan hệ One-To-Many xác định một sự liên kết một chiều từ một bản ghi trong bảng A đến nhiều bản ghi trong bảng B.
- Trong mối quan hệ One-To-Many, một bản ghi trong bảng A có thể liên kết với nhiều bản ghi trong bảng B, nhưng mỗi bản ghi trong bảng B chỉ liên kết với một bản ghi duy nhất trong bảng A.
- Để ám chỉ mối quan hệ One-To-Many, bạn có thể sử dụng các annotation như @OneToMany, @ManyToOne hoặc @JoinColumn.

```java
@Entity
public class Teacher {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    @OneToMany(mappedBy = "teacher")
    private List<Course> courses;

    // Other properties, constructors, and getters/setters
}
```

```java
@Entity
public class Course {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    @ManyToOne
    @JoinColumn(name = "teacher_id")
    private Teacher teacher;

    // Other properties, constructors, and getters/setters
}
```

# Many-To-One (Nhiều - Một)

- Mối quan hệ Many-To-One là đảo ngược của mối quan hệ One-To-Many.
- Mối quan hệ Many-To-One xác định một sự liên kết một chiều từ nhiều bản ghi trong bảng A đến một bản ghi trong bảng B.
- Trong mối quan hệ Many-To-One, một bản ghi trong bảng B có thể liên kết với nhiều bản ghi trong bảng A, nhưng mỗi bản ghi trong bảng A chỉ liên kết với một bản ghi duy nhất trong bảng B.
- Để ám chỉ mối quan hệ Many-To-One, bạn có thể sử dụng các annotation như @ManyToOne, @OneToMany hoặc @JoinColumn.

```java
@Entity
public class Teacher {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    @OneToMany(mappedBy = "teacher")
    private List<Course> courses;

    // Other properties, constructors, and getters/setters
}
```

```java
@Entity
public class Course {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    @ManyToOne
    @JoinColumn(name = "teacher_id")
    private Teacher teacher;

    // Other properties, constructors, and getters/setters
}
```

# Many-To-Many (Nhiều – Nhiều)

- Mối quan hệ Many-To-Many là một loại mối quan hệ giữa hai thực thể trong hệ thống quản lý cơ sở dữ liệu mà mỗi thực thể có thể liên kết với nhiều thực thể khác trong thực tế, và ngược lại. Điều này có nghĩa là mỗi thực thể trong mối quan hệ Many-To-Many có thể tồn tại độc lập và được liên kết với một hoặc nhiều thực thể khác trong mối quan hệ.

- Mối quan hệ Many-To-Many là một trong những mối quan hệ phổ biến và quan trọng trong quản lý cơ sở dữ liệu, đặc biệt khi có sự tương tác nhiều-nhiều giữa các thực thể.

```java
@Entity
public class Student {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    @ManyToMany(cascade = CascadeType.ALL)
    @JoinTable(name = "student_course",
            joinColumns = @JoinColumn(name = "student_id"),
            inverseJoinColumns = @JoinColumn(name = "course_id"))
    private List<Course> courses;

    // Constructors, getters, setters, etc.
}
```

```java
@Entity
public class Course {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    @ManyToMany(mappedBy = "courses")
    private List<Student> students;

    // Constructors, getters, setters, etc.
}
```

Many - To - One

One - To - Many