

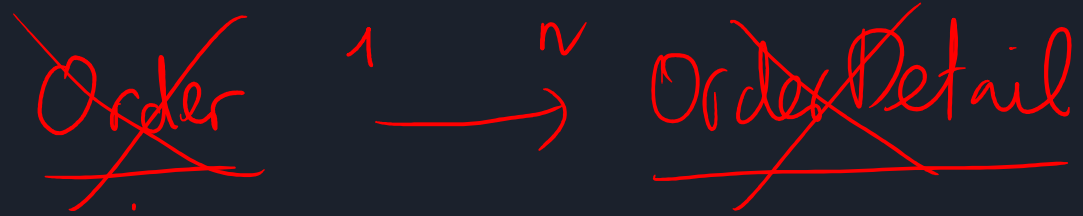
A decorative graphic on the left side of the slide featuring a blue parallelogram and a light green parallelogram, both tilted at an angle, set against a dark blue background with diagonal stripes.

Các loại Cascade

Nhat Tung Le

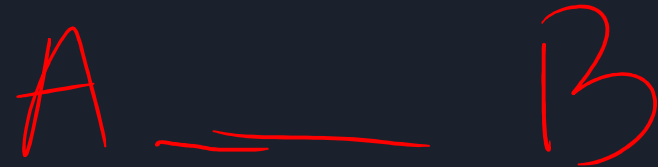


Cascade



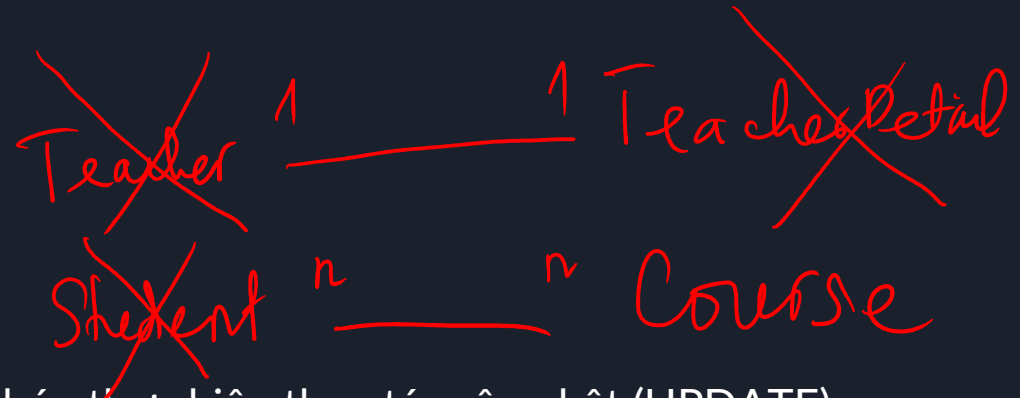
- Cascade (tiếng Việt là "Điều khiển tự động cập nhật") là một tính năng quan trọng giúp tự động lan truyền các thay đổi giữa các mối quan hệ của các Entity khi thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên một thực thể cha (parent entity).
- Nó giúp đơn giản hóa việc quản lý các mối quan hệ phức tạp trong cơ sở dữ liệu và tránh các lỗi thao tác không đồng nhất.

Các loại Cascade



- **CascadeType.ALL**: Đây là sự kết hợp của tất cả các loại Cascade khác. Nếu bạn chỉ định CascadeType.ALL trên một quan hệ, tất cả các thao tác CRUD (Create, Read, Update, Delete) trên thực thể cha sẽ được tự động lan truyền xuống các thực thể con liên quan.
- **CascadeType.PERSIST** (hay CascadeType.SAVE_UPDATE): Cascade cho phép thực hiện thao tác tạo mới (CREATE) trên thực thể cha và tự động lan truyền nó xuống các thực thể con liên quan, giúp lưu các thực thể con mới trong cùng một phiên Session.

Các loại Cascade



- **CascadeType.MERGE**: Cascade cho phép thực hiện thao tác cập nhật (UPDATE) trên thực thể cha và tự động lan truyền nó xuống các thực thể con liên quan, giúp cập nhật các thực thể con theo thay đổi của thực thể cha trong cùng một phiên Session.
- **CascadeType.REMOVE** (hoặc **CascadeType.DELETE**): Cascade cho phép thực hiện thao tác xóa (DELETE) trên thực thể cha và tự động lan truyền nó xuống các thực thể con liên quan, giúp xóa các thực thể con khi thực thể cha bị xóa.
- **CascadeType.REFRESH**: Cascade cho phép thực hiện thao tác làm mới (REFRESH) trên thực thể cha và tự động lan truyền nó xuống các thực thể con liên quan, giúp làm mới các thực thể con dựa trên dữ liệu từ thực thể cha.

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

    private String name;

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

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

```
@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.
}
```


• DE WIT

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

    private String name;

    @OneToMany(mappedBy = "course", cascade = {CascadeType.PERSIST, CascadeType.REMOVE})
    private List<Student> students;

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



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

    private String name;

    @ManyToOne
    @JoinColumn(name = "course_id")
    private Course course;

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