Ao utilizar o parâmetro cascade de um relacionamento, vemos que existem muitas variações de operações. Veremos na prática como cada uma delas pode ser utilizada.

Imagine que temos duas entidades: Post e Comment. Um Post pode ter muitos Comments. Se excluirmos um Post, o que acontecerá com os Comments relacionados a ele? Os tipos de cascata têm o objetivo de responder a este tipo de questões.

Os diferentes tipos de cascata são:

1. PERSIST : se você persistir a entidade Post, os Comments relacionados também serão persistidos.
2. MERGE : se você mesclar os detalhes de um Post, os Comments relacionados também serão mesclados.
3. REMOVE : se você remover um Post, os Comments relacionados também serão removidos.
4. REFRESH : se você atualizar o Post, também atualizará os Comments relacionados.
5. DETACH : se um Post foi desanexado, todos os Comments relacionados serão desanexados também.
6. ALL : se você executar qualquer uma das operações acima em um Post, essa operação será propagada para todos os Comments relacionados.

**Exemplo de código**

Para esclarecer estas definições, veja como isso se parece no código:

@Entity

**public** **class** **Post** {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

**private** Long id;

@OneToMany(mappedBy = "post", cascade = CascadeType.ALL)

**private** List<Comment> comments = **new** **ArrayList**<>();

//getters and setters

}

@Entity

**public** **class** **Comment** {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

**private** Long id;

@ManyToOne

@JoinColumn(name = "post\_id", nullable = false)

**private** Post post;

//getters and setters

}

COPIAR CÓDIGO

Nesse exemplo, usamos CascadeType.ALL, então todas as operações serão aplicadas aos Comments quando as operações correspondentes forem realizadas no Post. É importante que tenhamos conhecimento dessas variações para entender qual delas se aplica ao programa que estamos desenvolvendo.