Cascade Types in Hibernate

In configuring collections in hbm file, also in mapping one-to-many, many-to-many mappings, the collection element (say, **list**) in the hbm file contains an attribute **cascade.**

Example:

```
…

<class name="Student" table="student_tbl_100">
<id name="studentid">
<generator class="native"></generator>
</id>
<property name="name"></property>


<map name="courses" table="course_tbl_100" cascade="all">
<key column="id"></key>
<index column="course_id" type="string"></index>
<element column="course_name" type="string"></element>
</map>
</class>

…
```

The cascade type can also be mentioned in an annotation as shown below.

Different Cascade Types in Hibernate

Hibernate provides several types of cascade options that can be used to manage the relationships between entities. Here are the different cascade types in Hibernate:

1. **CascadeType.ALL**
2. **CascadeType.PERSIST**
3. **CascadeType.MERGE**
4. **CascadeType.REMOVE**

5. **CascadeType.REFRESH**
6. **CascadeType.DETACH**
7. **CascadeType.REPLICATE**
8. **CascadeType.SAVE_UPDATE**

These cascade types can be used individually or in combination to manage the relationships between entities based on the requirements of the application. It is important to use cascade types carefully, as they can lead to unintended consequences if not used properly.

1. CascadeType.ALL

CascadeType.ALL is a cascading type in Hibernate that specifies that all state transitions (create, update, delete, and refresh) should be cascaded from the parent entity to the child entities.

Example:

Java

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


    @OneToMany(mappedBy = "customer", cascade =
CascadeType.ALL)
    private Set<Order> orders = new HashSet<>();


    // getters and setters
```

```
}

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

    @ManyToOne
    @JoinColumn(name = "customer_id")
    private Customer customer;

    // getters and setters
}
```

Explanation: When a Customer entity is persisted, updated, or deleted, all associated Order entities will also be persisted, updated, or deleted.

2. CascadeType.PERSIST

CascadeType.PERSIST is a cascading type in Hibernate that specifies that the create (or persist) operation should be cascaded from the parent entity to the child entities.

3. CascadeType.MERGE

CascadeType.MERGE is a cascading type in Hibernate that specifies that the update (or merge) operation should be cascaded from the parent entity to the child entities.

4. CascadeType.REMOVE

CascadeType.REMOVE is a cascading type in Hibernate that specifies that the delete operation should be cascaded from the parent entity to the child entities.

5. CascadeType.REFRESH

CascadeType.REFRESH is a cascading type in Hibernate that specifies that the refresh operation should be cascaded from the parent entity to the child entities.

6. CascadeType.DETACH

CascadeType.DETACH is a cascading type in Hibernate that specifies that the detach operation should be cascaded from the parent entity to the child entities.

7. CascadeType.REPLICATE

CascadeType.REPLICATE is a cascading type in Hibernate that specifies that the replicate operation should be cascaded from the parent entity to the child entities.

8. CascadeType.SAVE_UPDATE

CascadeType.SAVE_UPDATE is a cascading type in Hibernate that specifies that the save or update operation should be cascaded from the parent entity to the child entities.