

# Relacionamentos - JPA

---

## Relacionamento na prática

---

Na prática, o Mapeamento cria a Relação de equivalência abaixo:

Banco de dados		Linguagem Orientada a Objetos
Tabela	→	Classe
Coluna	→	Atributo
Registro	→	Objeto

## Estratégias para geração da Chave Primária

---

Estratégia	Descrição
<b>GenerationType.AUTO</b>	Valor padrão, deixa com o provedor de persistência a escolha da estratégia mais adequada de acordo com o Banco de dados.
<b>GenerationType.IDENTITY</b>	Informamos ao provedor de persistência que os valores a serem atribuídos ao identificador único serão gerados pela coluna de auto incremento do banco de dados. Assim, um valor para o identificador é gerado para cada registro inserido no banco. Alguns bancos de dados podem não suportar essa opção.

Estratégia	Descrição
<b>GenerationType.SEQUENCE</b>	Informamos ao provedor de persistência que os valores serão gerados a partir de uma sequence. Caso não seja especificado um nome para a sequence, será utilizada uma sequence padrão, a qual será global, para todas as entidades. Caso uma sequence seja especificada, o provedor passará a adotar essa sequence para criação das chaves primárias. Alguns bancos de dados podem não suportar essa opção, como o MySQL por exemplo.
<b>GenerationType.TABLE</b>	Com a opção TABLE é necessário criar uma tabela para gerenciar as chaves primárias. Por causa da sobrecarga de consultas necessárias para manter a tabela atualizada, essa opção é pouco recomendada.

## Tipos de Relacionamentos

Cardinalidade	Anotação	Descrição
<b>1:1</b>	@OneToOne	<p>No relacionamento OneToOne, um Objeto pode pertencer a apenas um outro Objeto, ou seja, é uma ligação um para um.</p> <p>No Banco de dados significa que cada linha de uma Entidade se refere a <b>apenas</b> uma linha de outra Entidade.</p>

Cardinalidade	Anotação	Descrição
<b>1:N</b>	@OneToMany	<p>No relacionamento OneToMany, um Objeto é referenciado com muitos Objetos.</p> <p>Em Bancos de dados relacionais, cada linha de uma entidade é referenciada a muitos registros filhos em outra entidade.</p> <p>O importante é que os registros filhos não podem ter vários pais. Em uma relação OneToMany entre a Tabela A e a Tabela B, cada linha da Tabela A está ligada a 0, 1 ou muitas linhas da Tabela B.</p> <p>Numa Relação Unidirecional, a relação será uma relação OneToMany porque o lado Um da relação manda no Relacionamento.</p>
<b>N:1</b>	@ManyToOne	<p>No relacionamento ManyToOne, um Objeto é referenciado com outro Objeto que contém valores únicos.</p> <p>Em Bancos de dados Relacionais, esses relacionamentos são aplicáveis usando o par Chave Estrangeira (Lado N da Relação) e a Chave Primária (Lado 1 da Relação), entre as tabelas. O Lado N também deve ter uma Chave Primária.</p> <p>Numa Relação Bidirecional, essa anotação será utilizada para referenciar o Lado Um da Relação, através de um Objeto, que funcionará como a Chave Estrangeira.</p>

Cardinalidade	Anotação	Descrição
N:M	@ManyToMany	<p>No relacionamento ManyToMany, muitos Objetos são referenciados com muitos outros Objetos.</p> <p>Em Bancos de dados Relacionais, o relacionamento Muitos para Muitos uma ou mais linhas de uma Entidade são associadas a mais de uma linha em outra Entidade.</p> <p>Os relacionamentos ManyToMany exigem um pouco mais de trabalho do que os outros relacionamentos.</p> <p>Em um Banco de dados, um Relacionamento ManyToMany envolve uma terceira Tabela intermediária, que faz referência a ambas as outras tabelas armazenando as 2 Chaves estrangeiras.</p>

## Tipos de Cascade

Tipo	Descrição
<b>PERSIST</b>	Ele propaga a operação de persistir um objeto <b>Pai</b> para um objeto <b>Filho</b> , assim quando salvar a Entidade Cliente, também será salvo todas as Entidades Telefone associadas.
<b>MERGE</b>	Ele propaga a operação de atualização de um objeto <b>Pai</b> para um objeto <b>Filho</b> , assim quando atualizadas as informações da Entidade Cliente, também será atualizado no banco de dados todas as informações das Entidades Telefone associadas.
<b>REMOVE</b>	Ele propaga a operação de remoção de um objeto <b>Pai</b> para um objeto <b>Filho</b> , assim quando remover a Entidade Cliente, também será removida todas as entidades Telefone associadas.

Tipo	Descrição
<b>REFRESH</b>	Ele propaga a operação de recarregar de um objeto <b>Pai</b> para um objeto <b>Filho</b> , assim, quando houver atualização no banco de dados na Entidade Cliente, todas as entidades Telefone associadas serão recarregadas do banco de dados.
<b>ALL</b>	Corresponde a todas as operações acima (MERGE, PERSIST, REFRESH e REMOVE).
<b>DETACH</b>	A operação de desanexação remove a entidade do contexto persistente. Quando usamos CascadeType.DETACH, a entidade filha também é removida do contexto persistente

## 1.2. Hibernate

O **Hibernate** é um framework para o mapeamento objeto-relacional escrito na linguagem Java, cujo objetivo é diminuir a complexidade entre os programas Java, baseado no modelo Orientado a Objetos, que precisam trabalhar com um banco de dados do modelo Relacional (presente na maioria dos SGBD's). Em especial, no desenvolvimento de consultas e atualizações dos dados.

