Mapeando Relacionamentos Básicos com JPA

Prof. Roussian Gaioso

Tópicos

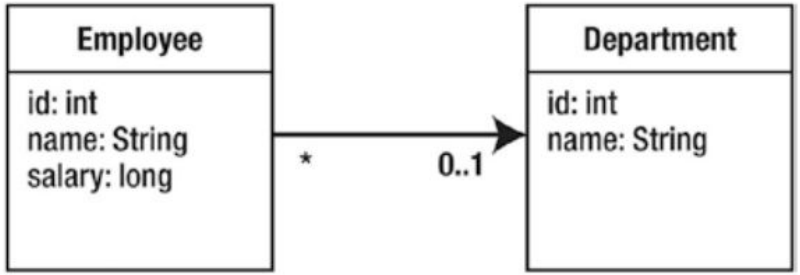
1. Mapeamento Unidirecional Many-to-One
2. Mapeamento Unidirecional One-to-One
3. Mapeamento Bidirecional One-to-One
4. Mapeamento Bidirecional One-to-Many
5. Mapeamento Bidirecional Many-to-Many
6. Relacionamentos Many-to-Many com Estado
7. Relacionamentos e Coleções de Elementos

Mapeamento Unidirecional Many-to-One

Mapeamento Unidirecional Many-to-One

Um mapeamento **Many-to-One** é definido anotando o atributo na entidade de origem (o atributo que se refere à entidade de destino) com a anotação **@ManyToOne.**

Mapeamento Unidirecional Many-to-One



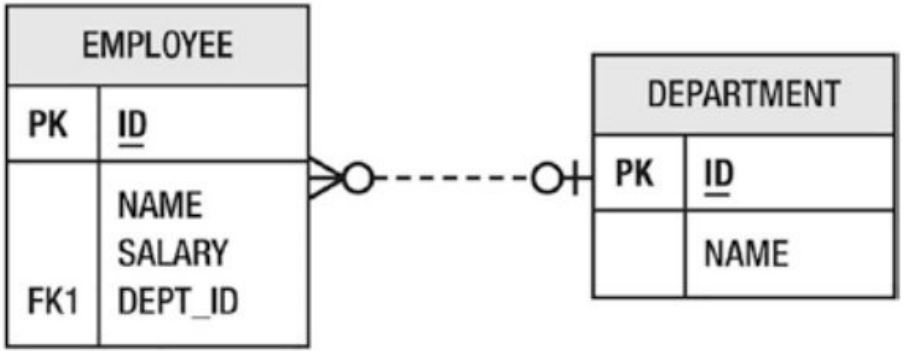

```java
@Entity
public class Employee {
    // ...
    @ManyToOne
    private Department department;
    // ...
}
```

Mapeamento Unidirecional Many-to-One

Notas

1. No banco de dados, um mapeamento de relacionamento significa que uma tabela tem uma referência a outra tabela.
2. No banco de dados, o termo de uma coluna que se refere a uma chave (geralmente a chave primária) em outra tabela é **uma coluna de chave estrangeira.**

Banco de Dados



Mapeamento Unidirecional Many-to-One

Notas

3. Em JPA, o termo de uma coluna que se refere a uma chave em outra tabela é chamado de **coluna de junção** e a anotação **@JoinColumn** é a anotação usada para configurar esses tipos de colunas.

```
@Entity
public class Employee {
    // ...
    @ManyToOne
    @JoinColumn(name="DEPT_ID")
    private Department department;
    // ...
}
```

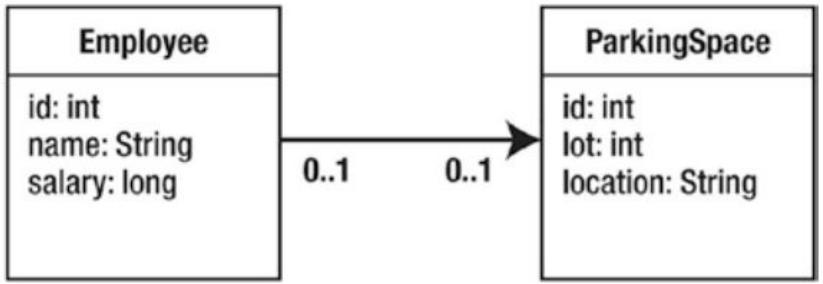
Mapeamento Unidirecional One-to-One

Mapeamento Unidirecional One-to-One

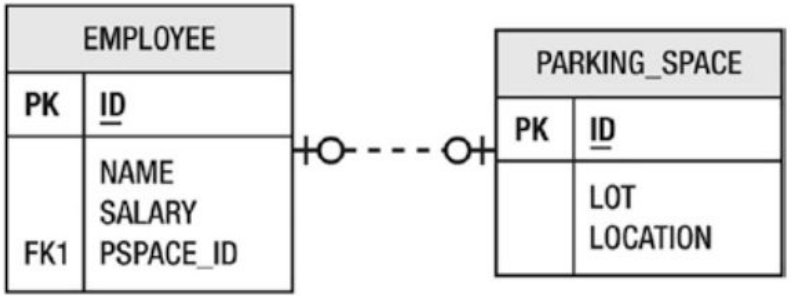
Um mapeamento **One-to-One** é definido anotando o atributo na entidade de origem (o atributo que se refere à entidade de destino) com a anotação **@OneToOne.**

Mapeamento Unidirecional One-to-One

Diagrama de Classe



Banco de Dados



Mapeamento Unidirecional One-to-One

```java
@Entity
public class Employee {
    // ...
    @OneToOne
    @JoinColumn(name="PSPACE_ID")
    private ParkingSpace parkingSpace;
    // ...
}
```

Mapeamento Bidirecional One-to-One

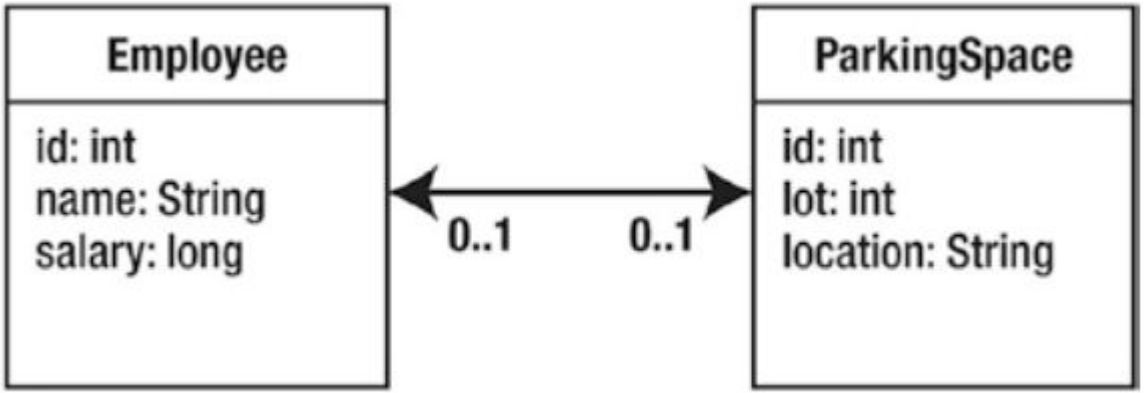
Mapeamento Bidirecional One-to-One

- A entidade de destino do One-to-One geralmente tem um relacionamento com a entidade de origem.
- Quando este é o caso, é chamado de relacionamento bidirecional um One-to-One.

Mapeamento Bidirecional One-to-One

Temos **dois mapeamentos One-to-One** separados, um em cada direção, mas a combinação dos dois é chamada de relacionamento **One-to-One bidirecional.**

Mapeamento Bidirecional One-to-One



Mapeamento Bidirecional One-to-One

O elemento **mappedBy** deve ser especificado na anotação @**OneToOne** na entidade que não define uma coluna de junção.

Mapeamento Bidirecional One-to-One

```java
@Entity
public class Employee {
    // ...
    @OneToOne
    @JoinColumn(name="PSPACE_ID")
    private ParkingSpace parkingSpace;
    // ...
}
```

```java
@Entity
public class ParkingSpace {
    @Id
    private long id;
    // ...
    @OneToOne(mappedBy="parkingSpace")
    private Employee employee;
    // ...
}
```

Mapeamento Bidirecional One-to-Many

Mapeamento Bidirecional One-to-Many

Quando um relacionamento é bidirecional, na verdade existem dois mapeamentos, um para cada direção.

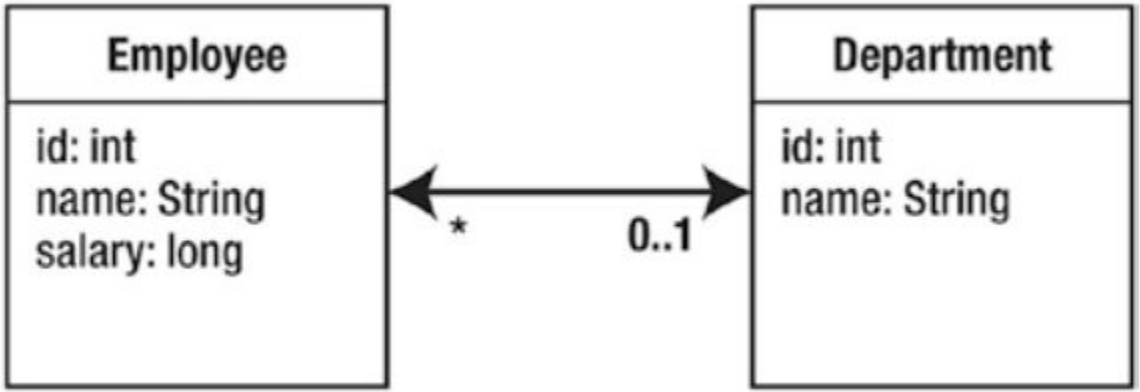
Um relacionamento bidirecional de One-to-Many sempre implica um mapeamento de Many-to-One de volta à origem,

Mapeamento Bidirecional One-to-Many

Quando um relacionamento é bidirecional, na verdade existem dois mapeamentos, um para cada direção.

Um relacionamento bidirecional de One-to-Many sempre implica um mapeamento de Many-to-One de volta à origem,

Mapeamento Bidirecional One-to-Many



Mapeamento Bidirecional One-to-Many

Há **dois pontos** importantes a serem lembrados ao definir relacionamentos bidirecionais de **One-to-Many** (ou **Many-to-One**):

1. No lado **Many-to-One**, a coluna de junção deve ser definida.
2. No lado **One-to-Many**, o elemento **mappedBy** deve ser usado.

Mapeamento Bidirecional One-to-Many

```java
@Entity
public class Employee {
    // ...
    @ManyToOne
    @JoinColumn(name="DEPT_ID")
    private Department department;
    // ...
}
```

```java
@Entity
public class Department {
    @Id private long id;
    // ...
    @OneToMany(mappedBy="department")
    private Collection<Employee> employees;
    // ...
}
```

Mapeamento Bidirecional Many-to-Many

Mapeamento Bidirecional Many-to-Many

Um **mapeamento Many-to-Many** é expresso nas entidades de origem e destino como uma anotação **@ManyToMany** nos atributos de coleção.

Mapeamento Bidirecional Many-to-Many

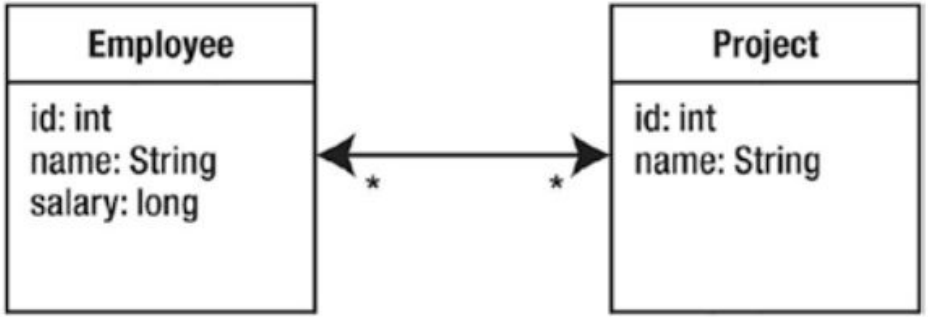
- Como a multiplicidade de ambos os lados de um relacionamento Many-to-Many é plural, devemos usar uma **terceira tabela** para associar os dois tipos de entidade.
- Essa tabela de associação é chamada de **tabela de junção** e cada relacionamento Many-to-Many deve ter uma.

Mapeamento Bidirecional Many-to-Many

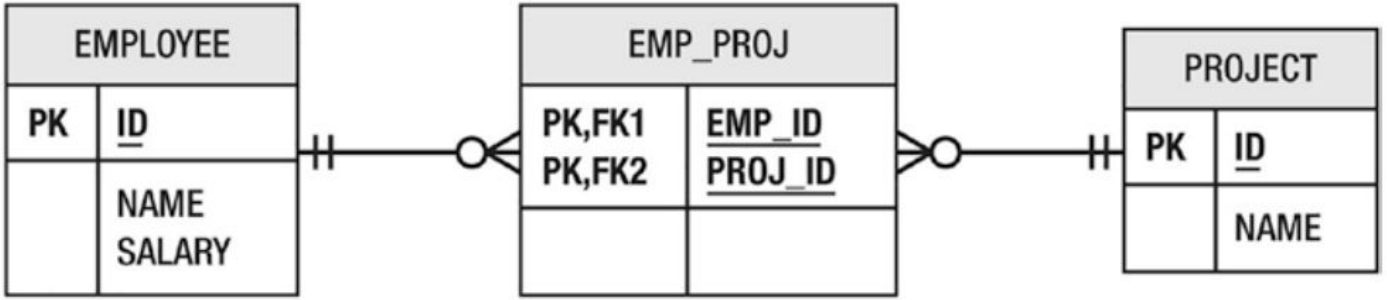
- A anotação **@JoinTable** é usada para configurar a tabela de junção para o relacionamento.
- A coluna de junção é descrita no elemento **joinColumns**, enquanto a coluna de junção ao lado inverso é especificada pelo elemento **inverseJoinColumns**.

Mapeamento Bidirecional Many-to-Many

Diagrama de classe



Banco de Dados



Mapeamento Bidirecional Many-to-Many

```
@Entity
public class Employee {
    // ...
    @ManyToMany
    @JoinTable(name="EMP_PROJ",
        joinColumns=@JoinColumn(name="EMP_ID"),
        inverseJoinColumns=@JoinColumn(name="PROJ_ID"))
    private Collection<Project> projects;
    // ...
}
```

Mapeamento Bidirecional Many-to-Many

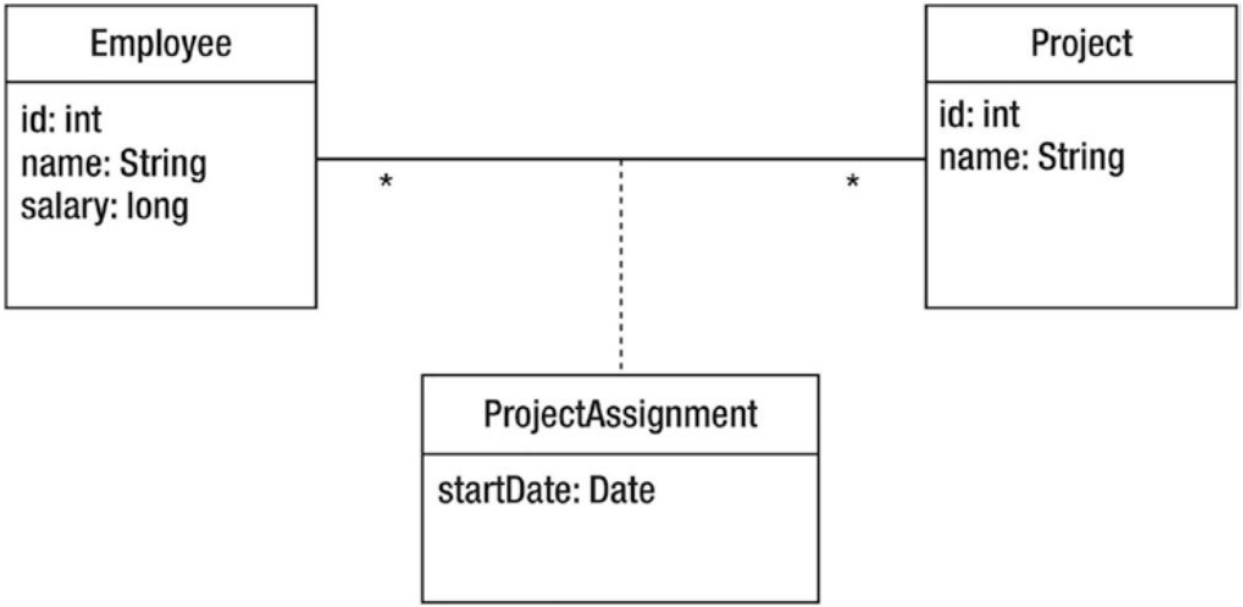
```java
@Entity
public class Project {
    @ManyToMany(mappedBy="projects")
    private Collection<Employee> employees;
    // ...
}
```

Relacionamentos Many-to-Many com Estado

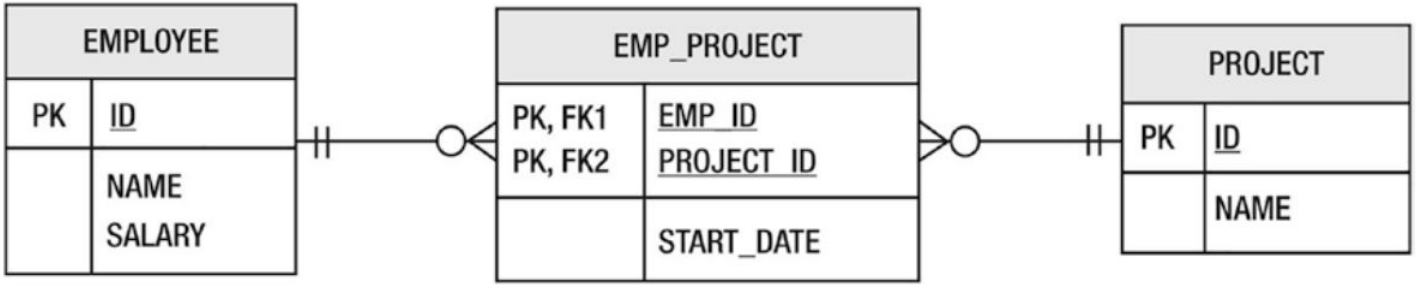
Relacionamentos Many-to-Many com Estado

Há momentos em que um relacionamento tem um estado associado a ele.

Relacionamentos Many-to-Many com Estado



Relacionamentos Many-to-Many com Estado



Relacionamentos Many-to-Many com Estado

```java
@Entity
public class Employee {
    // ...
    @OneToMany
    private Collection<ProjectAssignment> projectAssignments;
    // ...
}
```

```java
@Entity
public class Project {
    // ...
    @OneToMany
    private Collection<ProjectAssignment> projectAssignments;
    // ...
}
```

Relacionamentos Many-to-Many com Estado

```java
@Entity
public class ProjectAssignment {
    // ...
    @EmbeddedId
    private ProjectAssignmentPK id;
    // ...
}
```

```java
@Embeddable
public class ProjectAssignmentPK {
    @ManyToOne
    @JoinColumn(name="project_id")
    Project project;

    @ManyToOne
    @JoinColumn(name="employee_id")
    Employee employee;
}
```

Relacionamentos e Coleções de Elementos

Relacionamentos e Coleções de Elementos

Coleções de elementos contêm objetos que são dependentes da entidade de referência e podem ser recuperados apenas por meio da entidade que os contém.

Relacionamentos e Coleções de Elementos

Uma coleção de elementos é indicada pela anotação **@ElementCollection**.

- Podemos especificar uma tabela de coleção usando uma anotação **@CollectionTable**, que nos permite designar o nome da tabela, bem como a coluna de junção.

Dúvidas?

Obrigado!

Contato:
Roussian Gaioso
roussian.comp@gmail.com