


Jakarta EE Web Total

Herança nas Entidades




Tópicos Abordados



- `@MappedSuperclass`
- Herança de uma classe “normal”
- Herança entre entidades
 - Estratégia #1
 - Uma tabela por hierarquia de classes
 - Estratégia #2
 - Uma tabela por entidade
 - Estratégia #3
 - Uma tabela por subclasse

`@MappedSuperclass`



- Uma entidade pode herdar propriedades de uma superclasse
 - Ela deve ser anotada com `@MappedSuperclass`

```
@MappedSuperclass
public class Pessoa {

    @Id
    private Integer id;

    // getters & setters...
}
```

Não é gerada uma tabela Pessoa

@MappedSuperclass

```

@Entity(name = "PFISICA")
public class PessoaFisica extends Pessoa {
    private String cpf;
    // getters & setters...
}

@Entity(name = "PJURIDICA")
public class PessoaJuririca extends Pessoa {
    private String cnpj;
    // getters & setters...
}

```

PFISICA	
id	integer
cpf	varchar(255)

PJURIDICA	
id	integer
cnpj	varchar(255)

Herança de Uma Classe "Normal"

- Uma entidade pode herdar de uma classe "normal"
 - Que não é uma entidade
 - Que é não é uma @MappedSuperclass
- Neste caso, as propriedades que não pertencem à entidade não poderão ser persistidas

Herança de Uma Classe "Normal"

```

public class Pessoa {
    private String nome;
}

```

O atributo nome não é mapeado para o BD

```

@Entity(name = "PFISICA")
public class PessoaFisica extends Pessoa {
    @Id
    private Integer id;
    private String cpf;
}

@Entity(name = "PJURIDICA")
public class PessoaJuririca extends Pessoa {
    @Id
    private Integer id;
    private String cnpj;
}

```

PFISICA	
id	integer
cpf	varchar(255)

PJURIDICA	
id	integer
cnpj	varchar(255)

Herança Entre Entidades



- Uma entidade pode herdar de outra
- No BD, o mapeamento da herança pode seguir uma das seguintes estratégias
 - Uma tabela por hierarquia de classes
 - Uma tabela por entidade
 - Uma tabela por subclasse (com possibilidade de realizar um *join* entre tabelas)



Estratégia de Herança #1



- Uma tabela por hierarquia de classes

```
@Entity
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
public abstract class Pessoa {

    @Id
    @GeneratedValue
    private Integer id;

    private String nome;
}
```

@Inheritance
define a estratégia

Se @Inheritance não for especificada,
SINGLE_TABLE é usada por padrão



Estratégia de Herança #1



```
@Entity
public class PessoaFisica extends Pessoa {

    private String cpf;
}
```

```
@Entity
public class PessoaJuridica extends Pessoa {

    private String cnpj;
}
```

PESSOA	
DTYPE	varchar(31)
id	Integer
nome	varchar(255)
cnpj	varchar(255)
cpf	varchar(255)

Indica o tipo da
entidade (PessoaFisica
ou PessoaJuridica)



Configurando a Coluna *DTYPE*

```

@Entity
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
@DiscriminatorColumn(
    name = "TIPO",
    discriminatorType = DiscriminatorType.INTEGER,
    length = 5)
public abstract class Pessoa {
    //...
}

```

Nome da coluna

Tipo do discriminador (STRING, INTEGER, CHAR)

Tamanho do campo

```

@Entity
@DiscriminatorValue("1")
public class PessoaFisica extends Pessoa {
    //...
}

```

Define o valor a ser usado no discriminador

Estratégia de Herança #2

- Uma tabela por entidade

```

@Entity
@Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)
public abstract class Pessoa {
    //...
}

```

PFISICA		PJURIDICA	
id	integer	id	integer
nome	varchar(255)	nome	varchar(255)
cpf	varchar(255)	cnpj	varchar(255)

As colunas se repetem em cada tabela

Dependendo do *persistence provider* que você use, esta estratégia pode não estar disponível

Estratégia de Herança #3

- Uma tabela por subclasse

```

@Entity
@Inheritance(strategy = InheritanceType.JOINED)
public abstract class Pessoa {
    //...
}

```

As tabelas das subclasses referenciam a tabela da superclasse

PESSOA	
id	integer
nome	varchar(255)

PFISICA	
id	integer
cpf	varchar(255)

PJURIDICA	
id	integer
cnpj	varchar(255)