

Dúvidas em relacionamentos hibernate

Perguntada 5 anos, 3 meses atrás Modified 5 anos, 3 meses atrás Vista 148 vezes



0



alguém poderia me tirar algumas dúvidas com relação ao hibernate? Bom tenho 2 tabelas produtos e vendas com um relacionamento n:n que gerou uma tabela item_venda

eu fiz isso nos meus dois modelos para mapear:

Classe Produtos:



```
@Entity(name="Produtos")
@Table(name = "produtos")
public class Produtos {
    @Id
    @GeneratedValue
    private Long codProduto;
    @Column(name="preco_Venda")
    private Double preco_Venda;
    @Column(name="nome")
    private String nome;
}
```

a classe Venda:

```
@Entity(name="Venda")
@Table(name = "venda")
public class Venda {
    @Id
    @GeneratedValue
    private Long codVenda;
    @Column(name="data_venda")
    private Date dataVenda;
    @OneToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "idCaixa")
    private Caixa caixa;
    @ManyToOne(cascade = CascadeType.ALL)
    @JoinColumn(name="produtos_idVenda", referencedColumnName="idVenda", nullable=false)
    private List<Lista_Produtos> listaProdutos;
}
```

bom eu não preciso ter essa minha tabela item_venda como uma classe

uma das minhas dúvidas é, com esse mapeamento o que eu precisaria para inserir um valor nessa tabela item_venda.

Outra dúvida é em relação a minha classe Produtos:

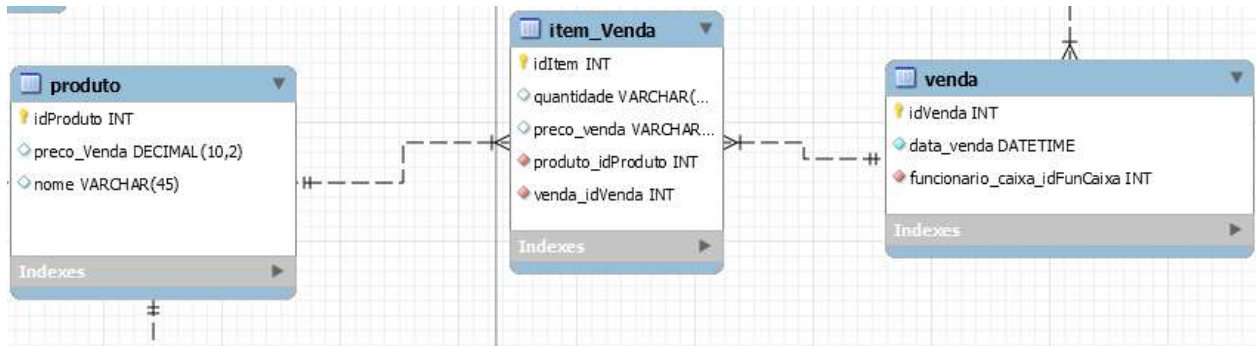
Eu precisaria por uma composição de Venda? ex:

```
private Venda venda;
```

```
//
```

ou eu precisaria criar + uma classe item_Venda

// para ilustrar meu relacionamento no banco de dados:



hibernate

Compartilhar Melhorar esta pergunta Seguir

perguntada 30/07/2019 às 23:51

g Joao Spirit
101 7

1 Resposta

Ordenado por: Maior pontuação (predefinição)



0



Neste caso você não tem como fugir de ter a tabela **ItemVenda** como entidade no seu projeto, justamente devido a estrutura dela. Se fosse apenas uma simples tabela de relação entre **Produto** e **Venda** daí sim seria possível pois o *hibernate* saberia como incluir os dados necessários (as *fks* das tabelas relacionadas).

De qualquer forma, na situação que você tem, realmente é um **N-N** entre **Produto-Venda**, no entanto é um **1-N** entre **Produto-ItemVenda** e **Venda-ItemVenda**. Então você deverá ter na sua entidade **Venda** um `@OneToMany` para `ItemVenda` (e também mapear o inverso no **ItemVenda** como `@ManyToOne`)

Exemplo de mapeamento:

```
@Entity
class Venda {
    /* outros fields omitidos... */

    @OneToMany(mappedBy="venda")
    private List<ItemVenda> items;
}

@Entity
class ItemVenda {
    /* omitindo outros campos.. */

    @ManyToOne
    @JoinColumn(name="id_venda")
    private Venda venda;

    @ManyToOne
```

```
@JoinColumn(name="id_produto")
private Produto produto;
}

@Entity
class Produto {
    /* omitindo outros campos.. */

    // Este mapeamento é opcional, coloca se fizer sentido pra o seu negócio..
    @OneToMany(mappedBy="produto")
    private List<ItemVenda> vendas;
}
```

Um exemplo de como você poderia montar esses dados pra persistir seria mais ou menos assim:

```
Caixa caixa = new Caixa(1);
Produto produto = new Produto(); /* omitindo inicialização do produto */

Venda venda = new Venda();
venda.setDataVenda(new Date());
venda.setCaixa(caixa);

venda.addItem(new ItemVenda(produto, 2 /* quantidade */));

/* salvar entidade com hibernate.. */
```

Compartilhar

editada 31/07/2019 às 13:14

respondida 31/07/2019 às 12:56

Melhore esta resposta Seguir



Israel Merljak

933 6 10

mt obrigado mano!! – Joao Spirit 31/07/2019 às 19:23
