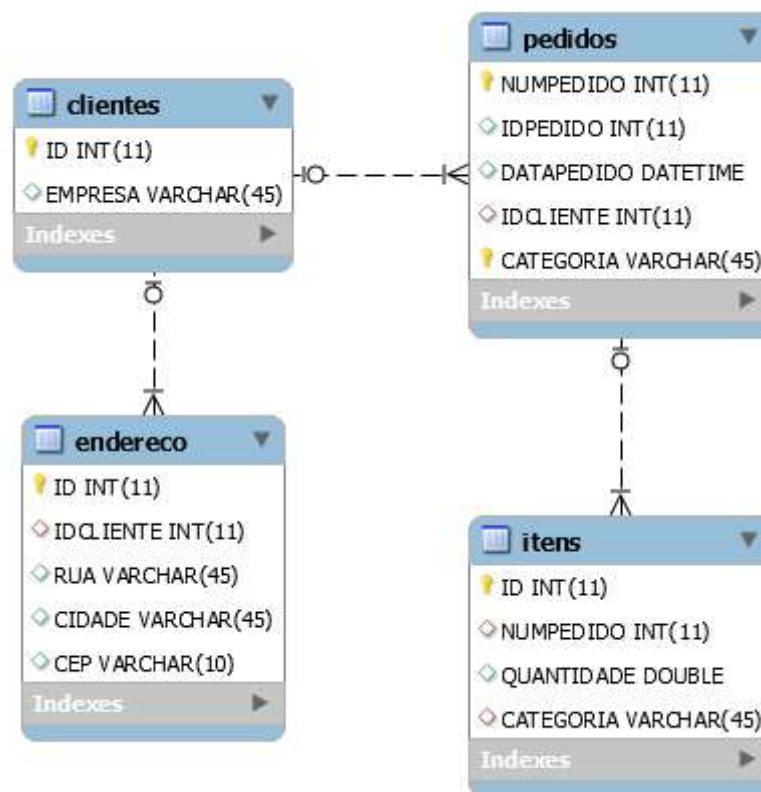


Roteiro para desenvolvimento da aula sobre JPA – Chave Composta (Exercício)

Neste roteiro desenvolveremos uma aplicação baseada em JPA, onde o aluno desenvolverá todos os passos. A obtenção da API JPA é a mesma utilizada no roteiro anterior, onde apresentamos o projeto JPA.

Vamos seguir este roteiro para o desenvolvimento da aplicação:

1. Criar um banco de dados chamado **vendasjpa**, cujo modelo é dado a seguir:



Nas tabelas **endereco** e **itens**, o campo **ID** é auto-incremento. Observe que a tabela **pedidos** possui uma chave composta.

2. Criar um projeto **JPA Project** chamado **Exercicio_ChaveComposta**.
3. Criar as entidades definidas abaixo. É importante que o aluno crie as entidades manualmente, para seu aprendizado:

```

package br.com.fiap.entity;

import java.io.Serializable;
import java.util.LinkedHashSet;
import java.util.Set;

import javax.persistence.CascadeType;
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity
@Table(name="clientes")
public class Cliente implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    @Column(name="ID")
    private int id;

    @Column(name="EMPRESA", length=45)
    private String empresa;

    @OneToMany(cascade = CascadeType.ALL, fetch=FetchType.LAZY,
mappedBy="cliente")
    private Set<Pedido> pedidos = new LinkedHashSet<Pedido>();

    @OneToMany(cascade = CascadeType.ALL, fetch=FetchType.LAZY,
mappedBy="cliente")
    private Set<Endereco> enderecos = new LinkedHashSet<Endereco>();

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getEmpresa() {
        return empresa;
    }

    public void setEmpresa(String empresa) {
        this.empresa = empresa;
    }

    public Set<Pedido> getPedidos() {
        return pedidos;
    }

    public void setPedidos(Set<Pedido> pedidos) {
        this.pedidos = pedidos;
    }

    public Set<Endereco> getEnderecos() {
        return enderecos;
    }

    public void setEnderecos(Set<Endereco> enderecos) {
```

```

        this.enderecos = enderecos;
    }
}

```

```

package br.com.fiap.pk;

import java.io.Serializable;

import javax.persistence.Column;
import javax.persistence.Embeddable;

@Embeddable
public class PedidosPK implements Serializable {

    private static final long serialVersionUID = 1L;

    @Column(name="NUMPEDIDO")
    private int codigo;

    @Column(name="CATEGORIA")
    private String categoria;

    public int getCodigo() {
        return codigo;
    }

    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }

    public String getCategoria() {
        return categoria;
    }

    public void setCategoria(String categoria) {
        this.categoria = categoria;
    }

    @Override
    public boolean equals(Object o){
        if(o instanceof PedidosPK){
            PedidosPK pk = (PedidosPK)o;
            if(this.getCodigo() != pk.getCodigo()){
                return false;
            }
            if(!this.getCategoria().equals(pk.getCategoria())){
                return false;
            }
            return true;
        }
        return false;
    }
}

```

```

@Override
public int hashCode(){
    return this.getCategoria().hashCode() + this.getCodigo();
}
}
    
```

```

package br.com.fiap.entity;

import java.io.Serializable;
import java.util.Date;
import java.util.LinkedHashSet;
import java.util.Set;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.EmbeddedId;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

import br.com.fiap.pk.PedidosPK;

@Entity
@Table(name="pedidos")
public class Pedido implements Serializable {

    private static final long serialVersionUID = 1L;

    @EmbeddedId
    private PedidosPK pedidoPK;

    @Temporal(value=TemporalType.TIMESTAMP)
    @Column(name="DATAPEDIDO")
    private Date dataPedido;

    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="IDCLIENTE")
    private Cliente cliente;

    @OneToMany(cascade=CascadeType.ALL, fetch=FetchType.LAZY, mappedBy="pedido")
    private Set<Item> itens = new LinkedHashSet<Item>();

    public Date getDataPedido() {
        return dataPedido;
    }

    public void setDataPedido(Date dataPedido) {
        this.dataPedido = dataPedido;
    }
}
    
```

```

    public Cliente getCliente() {
        return cliente;
    }

    public void setCliente(Cliente cliente) {
        this.cliente = cliente;
    }

    public Set<Item> getItens() {
        return itens;
    }

    public void setItens(Set<Item> itens) {
        this.itens = itens;
    }

    public PedidosPK getPedidoPK() {
        return pedidoPK;
    }

    public void setPedidoPK(PedidosPK pedidoPK) {
        this.pedidoPK = pedidoPK;
    }
}

```

```

package br.com.fiap.entity;

import java.io.Serializable;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;

@Entity
@Table(name="endereco")
public class Endereco implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @Column(name="ID")
    private int id;

    @Column(name="RUA")
    private String rua;
}

```

```

@Column(name="CIDADE")
private String cidade;

@Column(name="CEP")
private String cep;

@ManyToOne(fetch=FetchType.LAZY)
@JoinColumn(name="IDCLIENTE")
private Cliente cliente;

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getRua() {
    return rua;
}

public void setRua(String rua) {
    this.rua = rua;
}

public String getCidade() {
    return cidade;
}

public void setCidade(String cidade) {
    this.cidade = cidade;
}

public String getCep() {
    return cep;
}

public void setCep(String cep) {
    this.cep = cep;
}

public Cliente getCliente() {
    return cliente;
}

public void setCliente(Cliente cliente) {
    this.cliente = cliente;
}
}

```

```

package br.com.fiap.entity;

import java.io.Serializable;

```

```
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.JoinColumns;
import javax.persistence.ManyToOne;
import javax.persistence.Table;

@Entity
@Table(name="itens")
public class Item implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @Column(name="ID")
    private int id;

    @Column(name="QUANTIDADE")
    private double quantidade;

    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumns({@JoinColumn(name="NUMPEDIDO",referencedColumnName="NUMPEDIDO"),
        @JoinColumn(name="CATEGORIA",referencedColumnName="CATEGORIA")})
    private Pedido pedido;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public double getQuantidade() {
        return quantidade;
    }

    public void setQuantidade(double quantidade) {
        this.quantidade = quantidade;
    }

    public Pedido getPedido() {
        return pedido;
    }

    public void setPedido(Pedido pedido) {
        this.pedido = pedido;
    }
}
```

4. Incluir as classes (entidades) no arquivo **persistence.xml** e mantê-lo como no modelo abaixo:

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1"
    xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
    <persistence-unit name="jpaVendas" transaction-type="RESOURCE_LOCAL">
        <class>br.com.fiap.entity.Cliente</class>
        <class>br.com.fiap.entity.Endereco</class>
        <class>br.com.fiap.entity.Pedido</class>
        <class>br.com.fiap.entity.Item</class>
        <class>br.com.fiap.pk.PedidosPK</class>
        <properties>
            <property name="hibernate.hbm2ddl.auto" value="update" />
            <property name="hibernate.format_sql" value="true" />
            <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLDialect" />
            <property name="javax.persistence.jdbc.url"
                value="jdbc:mysql://localhost:3306/vendasjpa" />
            <property name="javax.persistence.jdbc.user" value="root" />
            <property name="javax.persistence.jdbc.password" value="fiap" />
            <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver" />
        </properties>
    </persistence-unit>
</persistence>
```

5. Escrever a classe **VendasHelper**, contendo métodos auxiliares para realizar a persistência:

```
package br.com.fiap.helper;

import javax.persistence.EntityManager;

import br.com.fiap.entity.Cliente;

public class VendasHelper {

    private EntityManager em;

    public VendasHelper(EntityManager em){
        this.em = em;
    }

    public void salvar(Clientes cliente) throws Exception{
        try {
            em.getTransaction().begin();
            em.persist(cliente);
            em.getTransaction().commit();
        } catch (Exception e) {
            em.getTransaction().rollback();
            throw e;
        }
    }
}
```


6. Para testar a aplicação, criar um cliente com todas as suas dependências, como no exemplo a seguir:

```
package br.com.fiap.programa;

import java.util.Date;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;
import javax.swing.JOptionPane;

import br.com.fiap.entity.Cliente;
import br.com.fiap.entity.Endereco;
import br.com.fiap.entity.Item;
import br.com.fiap.entity.Pedido;
import br.com.fiap.helper.VendasHelper;
import br.com.fiap.pk.PedidosPK;

public class TestaAplicacao {
    public static void main(String[] args) {
        EntityManagerFactory emf =
Persistence.createEntityManagerFactory("jpaVendas");
        EntityManager em = emf.createEntityManager();

        VendasHelper dao = new VendasHelper(em);

        try {
            //Definindo o cliente
            Cliente cliente = new Cliente();
            cliente.setId(10);
            cliente.setEmpresa("Fiap");

            //Definindo o Endereco
            Endereco endereco = new Endereco();
            endereco.setRua("Lins de Vasconcelos");
            endereco.setCidade("São Paulo");
            endereco.setCep("01538-001");
            endereco.setCliente(cliente);

            //Defnindo o pedido
            PedidosPK pkpedido = new PedidosPK();
            pkpedido.setCodigo(100);
            pkpedido.setCategoria("Livros");

            Pedido pedido = new Pedido();
            pedido.setDataPedido(new Date());
            pedido.setPedidoPK(pkpedido);
            pedido.setCliente(cliente);

            //Definindo dois itens
            Item item1 = new Item();
            item1.setQuantidade(2);

            Item item2 = new Item();
            item2.setQuantidade(3);
```

```

        //Fazendo as associações
        pedido.getItems().add(item1);
        pedido.getItems().add(item2);

        cliente.getEnderecos().add(endereco);
        cliente.getPedidos().add(pedido);

        dao.salvar(cliente);
        JOptionPane.showMessageDialog(null, "Cliente incluído com
sucesso!");
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Bom trabalho a todos!

Prof. Rafael Matsuyama