

Campus: Polo Ingleses

Curso: Desenvolvimento Full Stack

Disciplina: Nível 2 - Vamos Manter as Informações?

Turma: 9001

Semestre: 23.3

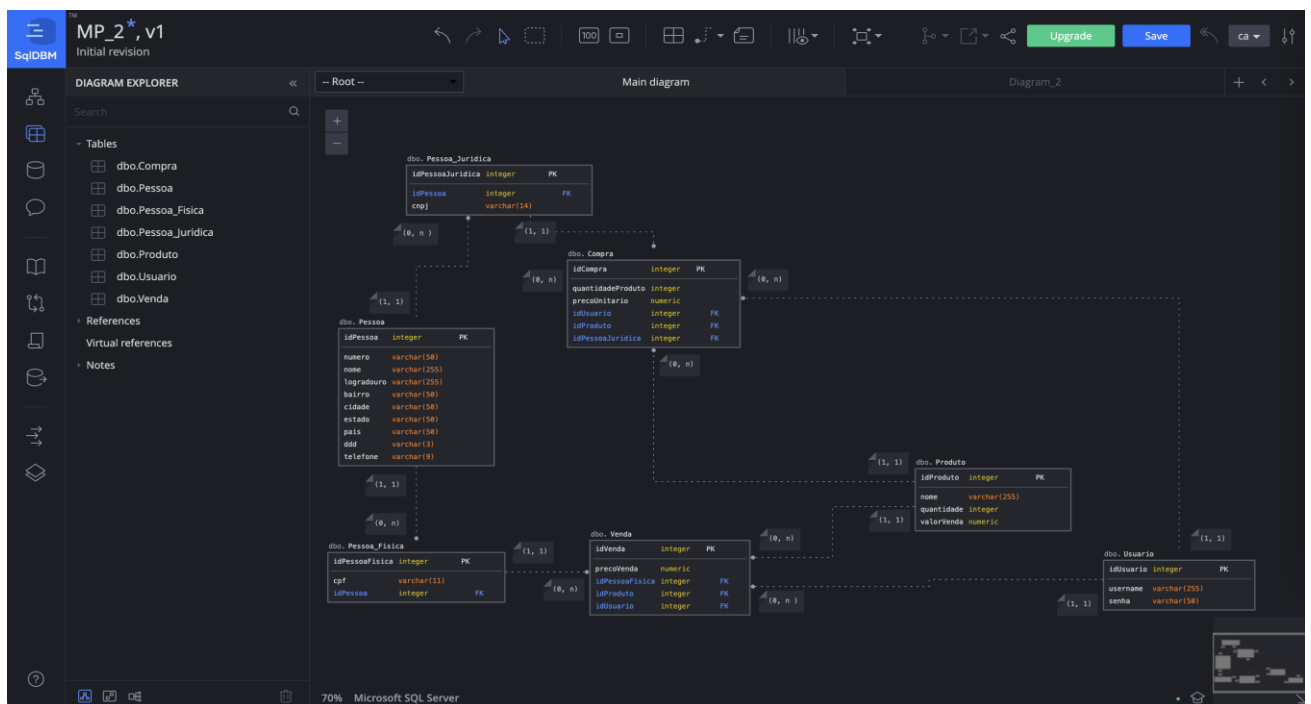
Aluna: Maria Carolina Knudsen Boabaid

1º Procedimento | Criando o Banco de Dados

O objetivo da prática é que o estudante evidencie as competências fundamentais na modelagem de bases de dados em um sistema, juntamente com a proficiência no emprego da sintaxe SQL para conceber as estruturas essenciais. Esse conhecimento prático abrange desde a definição apropriada de entidades e seus atributos até a formulação precisa de relacionamentos entre as tabelas.

Desta forma, a finalidade principal desta prática é capacitar o estudante a desenvolver uma base de dados coesa e eficiente, refletindo a compreensão sólida das práticas fundamentais de design e implementação de bancos de dados.

Modelagem física:



Comandos SQL utilizados para a criação do banco de dados:

```
CREATE TABLE "Usuario" (  
  "idUsuario" INTEGER NOT NULL UNIQUE,  
  "username" TEXT NOT NULL UNIQUE,  
  "senha" TEXT NOT NULL,  
  PRIMARY KEY ("idUsuario"));
```

```
CREATE TABLE "Produto" (  
  "idProduto" INTEGER NOT NULL UNIQUE,  
  "nome" TEXT NOT NULL,  
  "quantidade" INTEGER NOT NULL,  
  "valorVenda" NUMERIC NOT NULL,  
  PRIMARY KEY ("idProduto" AUTOINCREMENT));
```

```
CREATE TABLE "Pessoa" (  
  "idPessoa" INTEGER NOT NULL UNIQUE,  
  "nome" TEXT NOT NULL,  
  "numero" TEXT NOT NULL,  
  "logradouro" TEXT NOT NULL,  
  "bairro" TEXT NOT NULL,  
  "cidade" TEXT NOT NULL,  
  "estado" TEXT NOT NULL,  
  "pais" TEXT NOT NULL,  
  "ddd" TEXT NOT NULL,  
  "telefone" TEXT NOT NULL,  
  PRIMARY KEY ("idPessoa" AUTOINCREMENT));
```

```
CREATE TABLE "Venda" (  
  "idVenda" INTEGER NOT NULL UNIQUE,  
  "precoVenda" NUMERIC NOT NULL,  
  "idPessoaFisica" INTEGER NOT NULL,  
  "idUsuario" INTEGER NOT NULL,  
  "idProduto" INTEGER NOT NULL,  
  PRIMARY KEY ("idVenda" AUTOINCREMENT),  
  FOREIGN KEY ("idPessoaFisica") REFERENCES "Pessoa_Fisica"("idPessoaFisica"),  
  FOREIGN KEY ("idProduto") REFERENCES "Produto"("idProduto"),
```

```
FOREIGN KEY ("idUserario") REFERENCES "Usuario"("idUserario"));
```

```
CREATE TABLE "Compra" (  
  "idCompra" INTEGER NOT NULL UNIQUE,  
  "quantidadeProduto" INTEGER NOT NULL,  
  "precoUnitario" NUMERIC NOT NULL,  
  "idUserario" INTEGER NOT NULL,  
  "idProduto" INTEGER NOT NULL,  
  "idPessoaJuridica" INTEGER NOT NULL,  
  FOREIGN KEY ("idPessoaJuridica") REFERENCES "Pessoa_Juridica"("idPessoaJuridica"),  
  FOREIGN KEY ("idUserario") REFERENCES "Usuario"("idUserario"),  
  FOREIGN KEY ("idProduto") REFERENCES "Produto"("idProduto"),  
  PRIMARY KEY ("idCompra" AUTOINCREMENT));
```

```
CREATE TABLE "Pessoa_Fisica" (  
  "idPessoaFisica" INTEGER NOT NULL UNIQUE,  
  "idPessoa" INTEGER NOT NULL,  
  "cpf" TEXT NOT NULL UNIQUE,  
  FOREIGN KEY ("idPessoa") REFERENCES "Pessoa"("idPessoa"),  
  PRIMARY KEY ("idPessoaFisica" AUTOINCREMENT));
```

```
CREATE TABLE "Pessoa_Juridica" (  
  "idPessoaJuridica" INTEGER NOT NULL UNIQUE,  
  "cnpj" TEXT NOT NULL UNIQUE,  
  "idPessoa" INTEGER NOT NULL UNIQUE,  
  FOREIGN KEY ("idPessoa") REFERENCES "Pessoa"("idPessoa"),  
  PRIMARY KEY ("idPessoaJuridica" AUTOINCREMENT));
```

Como são implementadas as diferentes cardinalidades, basicamente 1X1, 1XN ou NxN, em um banco de dados relacional?

1 x 1: Ocorre quando uma incidência da tabela A pode ter somente uma referência a uma incidência da tabela B, e vice-versa. O relacionamento será desenhado pela chave primária da tabela A ou B (pode ser qualquer uma das tabelas dependendo da lógica do banco de dados), será recebida como chave estrangeira na outra tabela.

Geralmente, esse tipo de relacionamento é utilizado quando existe uma forte dependência entre as duas tabelas, mas as informações relacionadas a essas entidades são mantidas em

tabelas separadas por razões de organização ou para atender a requisitos específicos de normalização.

1 x N: Este relacionamento acontece quando uma incidência da tabela A pode ter referência a várias incidências da tabela B, contudo uma incidência da tabela B pode ter apenas uma referência a uma incidência da tabela A. Um exemplo prático é um cliente que pode ter várias compras, mas uma compra pode ter apenas um cliente.

Para desenhar esse relacionamento a tabela que apenas pode ter uma incidência, neste contexto a tabela B, receberá a chave primária da outra tabela (tabela A) como chave estrangeira.

N x N: Esta relação se configura quando muitas incidências da tabela A fazem referência a muitas incidências da tabela B. Um exemplo prático é um livro que pode ter muitos autores e autores que podem ter muitos livros.

Para desenhar esse tipo de relacionamento é preciso criar uma tabela associativa, a tabela C, que terá sua própria chave primária e receberá como chaves estrangeira as chaves primárias da tabela A e B.

Que tipo de relacionamento deve ser utilizado para representar o uso de herança em bancos de dados relacionais?

É possível fazer a 'Herança de Tabela por Classe' da classe presente na hierarquia de herança é mapeada para uma tabela distinta. Cada tabela armazena exclusivamente os atributos que são específicos da respectiva classe, além dos atributos herdados. Essa estratégia envolve a criação de uma tabela para a classe base e uma tabela separada para cada subclasse. As tabelas das subclasses referenciam a tabela da classe base, estabelecendo assim a relação hierárquica entre elas.