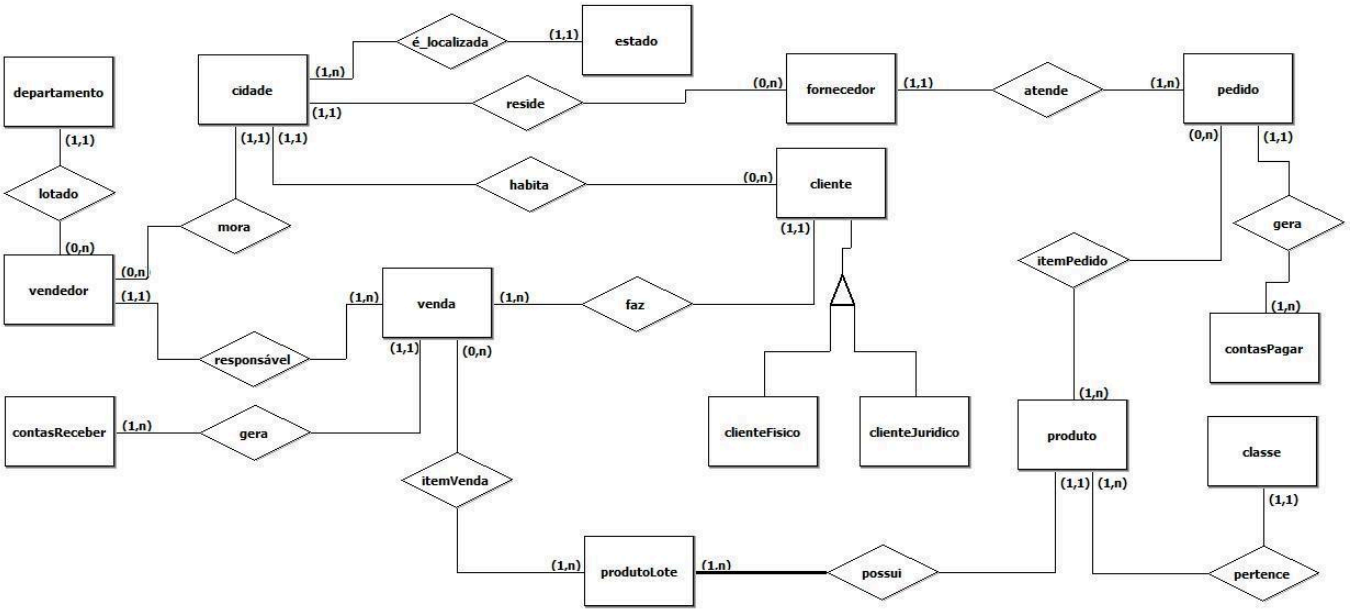

SISTEMA DE CONTROLE DE VENDAS

Crie outro banco de dados, depois faça a criação das seguintes tabelas.

Observação: As tabelas abaixo são as mesmas criadas anteriormente, porém com mais informações, desta vez elas ficarão completas, a imagem acima é uma representação visual do banco. Utilizem o mesmo script criado anteriormente, as informações novas estarão em destaque.

Tabelas departamento e estado: sem mudanças.

Tabela cidade:

- O campo codCidade é de numeração automática, sendo chave primária da tabela;
- O campo nome é texto, de tamanho variável máximo 50, sendo uma chave candidata, de preenchimento obrigatório;
- O campo siglaEstado é de texto, de tamanho fixo 2 e de preenchimento obrigatório, sendo uma chave estrangeira da tabela estado.
  As restrições de integridade referencial devem ser especificadas na sequência da definição do campos siglaEstado, assim temos que um estado somente poderá ser apagado quando não existirem cidades a ele relacionadas, e quando um estado for atualizado, essa atualização deverá ser propagada as cidades.
  A resolução desta tabela está descrita abaixo, notem que pode ser um pouco diferente do que foi visto anteriormente. Existem diferentes maneiras de declarar uma FK.

```
Unset
CREATE TABLE cidade (
        codCidade serial PRIMARY KEY,
```

```
        nome varchar(50) UNIQUE NOT NULL,
        siglaEstado char(2) NOT NULL REFERENCES estado(siglaEstado) ON
DELETE no action ON UPDATE cascade);
```

CREATE TABLE cidade (
codCidade serial PRIMARY KEY,
nome varchar(50) UNIQUE NOT NULL,
siglaEstado char(2) NOT NULL REFERENCES estado(siglaEstado) ON DELETE no action ON UPDATE cascade
);

Tabela vendedor:

- O campo codVendedor é de numeração automática, sendo chave primária da tabela;
- O campo nome é de texto, de tamanho variável máximo 60, de preenchimento obrigatório;
- O campo dataNascimento é um campo tipo data;
- O campo endereco é de texto, de tamanho variável máximo 60;
- O campo cep é tamanho fixo máximo 8;
- O campo telefone é caracter, de tamanho variável máximo 20;
- O campo codCidade é inteiro;
- A dataContratacao é um campo data, que terá como valor padrão a data atual de inserção do registro, no caso de uma data não ter sido especificada;
- O campo codDepartamento é inteiro;

As restrições de integridade referencial devem ser especificadas no final da definição da tabela: quando um departamento for apagado, na tabela vendedor este código de departamento deverá receber o valor null, e quando for atualizado, essa atualização deverá ser propagada aos vendedores;

Com relação às restrições de cidade, temos que quando uma cidade for apagada deverá receber o valor padrão 1, e se a mesma for atualizada, essa atualização deverá ser propagada aos vendedores.

Unset

```
CREATE TABLE vendedor(
       codvendedor serial PRIMARY KEY,
       nome varchar(60) NOT NULL,
       dataNascimento date,
       endereco varchar(60),
       cep char(8),
       telefone varchar(20),
```

```
        codCidade int default 1,
        dataContratacao date default (current_date),
        codDepartamento int,
        FOREIGN     KEY     (codDepartamento)     REFERENCES     departamento
(codDepartamento) ON DELETE set null
        ON UPDATE cascade,
        FOREIGN  KEY  (codCidade)  REFERENCES  cidade  (codCidade)  ON  DELETE
cascade ON UPDATE cascade);
```

CREATE TABLE vendedor(
codvendedor serial PRIMARY KEY,
nome varchar(60) NOT NULL,
dataNascimento date,
endereco varchar(60),
cep char(8),
telefone varchar(20),
codCidade int default 1,
dataContratacao date default (current_date),
codDepartamento int,
FOREIGN KEY (codDepartamento) REFERENCES departamento (codDepartamento) ON DELETE set null
ON UPDATE cascade,
FOREIGN KEY (codCidade) REFERENCES cidade (codCidade) ON DELETE cascade ON UPDATE cascade
);

Tabela cliente:

- O campo codCliente é um campo de numeração automática, sendo chave primária da tabela;
- O campo endereco é de texto, de tamanho variável máximo 60;
- O campo codCidade é inteiro e de preenchimento obrigatório, sendo uma chave estrangeira da tabela cidade;
- O campo telefone é de texto, de tamanho variável máximo 20;
- O campo tipo é caracter, de tamanho fixo máximo 1;
- A dataCadastro é um campo data, que terá como valor padrão a data atual de cadastro, no caso de uma data não ter sido especificada;
- O campo cep é tamanho fixo máximo 8;

As restrições de integridade referencial devem ser especificadas com nome: uma cidade nunca poderá ser apagada se existirem clientes que nela residem, e se a mesma for atualizada, essa atualização deverá ser propagada ao cliente. Exemplo:

CONSTRAINT fk_cli_cid FOREIGN KEY (codCidade) REFERENCES cidade (codCidade) ON DELETE no action ON UPDATE cascade);

Essa tabela utiliza uma maneira diferente de declarar chave estrangeira (neste caso é adicionado o nome 'fk_cli_cid' para identificação), então o script também foi disponibilizado. Para as próximas tabelas vocês podem escolher qual método utilizar, porém recomendo o primeiro método por ser mais prático.

```
Unset
CREATE TABLE cliente (
	codCliente serial PRIMARY KEY,
	endereco varchar(60),
	codCidade int NOT NULL,
	telefone varchar(20),
	tipo char(1),
	dataCadastro date default (current_date),
	cep char(8),
	CONSTRAINT fk_cli_cid FOREIGN KEY (codCidade) REFERENCES cidade
(codCidade) ON DELETE no action ON UPDATE cascade);
```

CREATE TABLE cliente (
codCliente serial PRIMARY KEY,
endereco varchar(60),
codCidade int NOT NULL,
telefone varchar(20),
tipo char(1),
dataCadastro date default (current_date),
cep char(8),
CONSTRAINT fk_cli_cid FOREIGN KEY (codCidade) REFERENCES cidade (codCidade) ON DELETE no action ON UPDATE cascade);

Tabela clienteFisico:

- O campo codCliente é um inteiro, sendo chave primária desta tabela, e chave estrangeira da tabela cliente;
- O campo nome é de texto, de tamanho variável máximo 50, de preenchimento obrigatório;
- O campo dataNascimento é um campo data;
- O campo cpf é caracter, de tamanho variável máximo 11, de preenchimento obrigatório, sendo uma chave candidata;
- O campo rg é caracter, de tamanho variável máximo 8;

Como restrições de integridade referencial temos que um cliente nunca poderá ser apagado se existirem um ou mais registros na tabela cliente_fisica a ele associados, e se o mesma for atualizado, essa atualização deverá ser propagada ao cliente_fisica. (Devem ser especificadas ao lado do campo)

```
CREATE TABLE clienteFisico (
   codCliente int PRIMARY KEY AUTO_INCREMENT,
   nome VARCHAR(50) NOT NULL,
   dataNascimento DATE,
   cpf VARCHAR(11) NOT NULL UNIQUE,
   rg VARCHAR(8),
    FOREIGN KEY (codCliente) REFERENCES client (codCliente) ON DELETE NO ACTION ON UPDATE CASCADE
   );
```

Tabela clienteJuridico:

- O campo codCliente é um inteiro, sendo chave primária desta tabela, e chave estrangeira da tabela cliente;
- O campo nomeFantasia é de texto, de tamanho variável máximo 80, sendo uma chave candidata;
- O campo razaoSocial é de texto, de tamanho variável máximo 80, sendo uma chave candidata, de preenchimento obrigatório;
- O campo ie (inscrição estadual) é de texto, de tamanho variável máximo 20, de preenchimento obrigatório, sendo uma chave candidata;
- O campo cgc é de texto, de tamanho variável máximo 20, sendo uma chave candidata, de preenchimento obrigatório;

Como restrições de integridade referencial temos que um cliente nunca poderá ser apagado se existirem um ou mais registros na tabela cliente_juridica a ele associados, e se o mesma for atualizado, essa atualização deverá ser propagada ao cliente_juridica. (Devem ser especificadas ao lado do campo)

```
CREATE TABLE clienteJuridico (
   codCliente INT AUTO_INCREMENT PRIMARY KEY,
   nomeFantasia VARCHAR(80) UNIQUE,
   razaoSocial VARCHAR(80) UNIQUE NOT NULL,
   ie VARCHAR(20) UNIQUE NOT NULL,
   cgc VARCHAR(20) UNIQUE NOT NULL,
      FOREIGN KEY (codCliente) REFERENCES cliente (codCliente) ON DELETE NO ACTION ON UPDATE CASCADE
   );
```

Tabela classe: sem mudanças.

Tabela produto:

- O campo codProduto é um campo de numeração automática, sendo chave primária da tabela;
- O campo descrição é de texto, de tamanho variável máximo 40, sendo de preenchimento obrigatório;
- O campo unidadeMedida é de texto, de tamanho fixo máximo 2;
- O campo embalagem é de texto, de tamanho variável máximo 15, que terá como valor padrão padrão ‘Fardo’, no caso de uma embalagem não ter sido especificada;
- O campo codClasse é um inteiro, sendo chave estrangeira da tabela classe;
- O campo precoVenda é um numérico com tamanho 14, com 2 casas decimais;
- O campo estoqueMinimo é um numérico com tamanho 14, com 2 casas decimais;

  Como restrições de integridade referencial temos que quando uma classe for apagada ou alterada, essa classe deve ficar como nula nesta tabela.

```
CREATE TABLE produto (
   codProduto INT AUTO_INCREMENT PRIMARY KEY,
   descricao VARCHAR(40) NOT NULL,
   unidadeMedida CHAR(2),
   embalagem VARCHAR(15) DEFAULT "Fardo",
   codClasse INT,
   precoVenda DECIMAL (14, 2),
   estoqueMinimo DECIMAL (14,2),
    FOREIGN KEY (codClasse) REFERENCES classe (codClasse) ON DELETE SET NULL
ON UPDATE SET NULL
   );
```

Tabela produtoLote:

- O campo codProduto é um campo inteiro, sendo chave primária da tabela e chave estrangeira da tabela produto;
- O campo numeroLote é um campo inteiro, sendo chave primária da tabela;
- O campo quantidadeAdquirida é um numérico com tamanho 14, com 2 casas decimais;
- O campo quantidadeVendida é um numérico com tamanho 14, com 2 casas decimais;
- O campo precoCusto é um numérico com tamanho 14, com 2 casas decimais;
- O campo dataValidade é um campo de data;

Como restrições de integridade referencial temos que quando um produto for apagado todos os seus lotes também o serão, e quando o mesmo for atualizado, essa atualização deverá ser propagada ao lote.

```
CREATE TABLE produtoLote (
   codProduto INT AUTO_INCREMENT,
   numeroLote INT,
   quantidadeAdquirida DECIMAL (16, 2),
   quantidadeVendida DECIMAL (16, 2),
   precoCusto DECIMAL (16, 2),
   dataValidade DATE,
   PRIMARY KEY(codProduto, numeroLote),
      FOREIGN KEY (codProduto) REFERENCES produto (codProduto) ON DELETE
CASCADE ON UPDATE CASCADE
   );
```

Tabela venda:

- O campo codVenda é um campo inteiro, sendo chave primária da tabela;
- O campo codCliente é um campo inteiro, sendo chave estrangeira da tabela cliente;
- O campo codVendedor é um campo inteiro, sendo chave estrangeira da tabela vendedor;
- O campo dataVenda é um campo de data, que tem como valor padrão a data atual, sendo que outras datas podem ser especificadas;
- O campo enderecoEntrega é caracter, de tamanho variável máximo 60;
- O campo status é caracter, de tamanho variável máximo 30;

Como restrições de integridade referencial temos que quando um vendedor for apagado da base de dados, na venda será assumido o valor padrão 100 – que representa o registro 'Não cadastrado', e quando o mesmo for atualizado, essa atualização deverá ser propagada à venda. Quando um cliente for apagado todas as vendas relativas a ele também o devem ser, sendo que atualizações não são permitidas em clientes que possuem vendas relacionadas a ele.

```
CREATE TABLE venda (
   codVenda INT PRIMARY KEY,
   codCliente INT,
   codVendedor INT DEFAULT 100,
   dataVenda DATE DEFAULT (CURRENT_DATE),
   enderecoEntrega VARCHAR(60),
   estatus VARCHAR(30),
```

```
    FOREIGN KEY (codCLiente) REFERENCES cliente (codCliente) ON DELETE CASCADE,
    FOREIGN KEY (codVendedor) REFERENCES vendedor (codVendedor) ON DELETE SET NULL ON UPDATE CASCADE
  );
```

Tabela itemVenda:

- O campo codVenda é um campo inteiro, sendo chave primária da tabela e estrangeira da tabela venda;
- O campo codProduto é um campo inteiro, sendo chave primária da tabela, e chave estrangeira da tabela lote (chave estrangeira de lote é composta por codProduto e numeroLote);
- O campo numeroLote é um campo inteiro, sendo chave primária da tabela, e chave estrangeira da tabela lote;
- O campo quantidade é um numérico com tamanho 14, com 2 casas decimais, sendo seu preenchimento obrigatório e com valor positivo;

  Como restrições de integridade referencial temos que quando uma venda for apagada todos os seus itens também o serão, e quando a mesma for atualizada, essa atualização deverá ser propagada a venda. Um lote de produtos somente poderá ser apagado se não existirem itens de vendas relacionados.

```
CREATE TABLE itemVenda (
  codVenda INT AUTO_INCREMENT,
  codProduto INT,
  numeroLote INT,
  quantidade DECIMAL(16, 2) NOT NULL CHECK (quantidade > 0),
  PRIMARY KEY (codProduto, codVenda, numeroLote),
   FOREIGN KEY (codVenda) REFERENCES venda (codVenda) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (codProduto) REFERENCES produto (codProduto),
    FOREIGN KEY (codProduto, numeroLote) REFERENCES produtoLote (codProduto, numeroLote) ON DELETE NO ACTION
);
```

Tabela fornecedor:

- O campo codFornecedor é um inteiro, sendo chave primária desta tabela;
- O campo nomeFantasia é caracter, de tamanho variável máximo 80, sendo uma chave candidata;
- O campo razaoSocial é caracter, de tamanho variável máximo 80, sendo uma chave candidata, de preenchimento obrigatório;

- O campo ie (inscrição estadual) é caracter, de tamanho variável máximo 20, de preenchimento obrigatório, sendo uma chave candidata;
- O campo cgc é caracter, de tamanho variável máximo 20, sendo uma chave candidata, de preenchimento obrigatório;
- O campo endereco é caracter, de tamanho variável máximo 60;
- O campo telefone é caracter, de tamanho variável máximo 20;
- O campo codCidade é inteiro, sendo uma chave estrangeira da tabela cidade;

Como restrições de integridade referencial temos que uma cidade nunca poderá ser apagada se existirem fornecedores que nela residem, e se a mesma for atualizada, essa atualização deverá ser propagada ao fornecedor.

```
CREATE TABLE fornecedor (
   codFornecedor INT AUTO_INCREMENT PRIMARY KEY,
   nomeFantasia VARCHAR(80) UNIQUE,
   razaoSocial VARCHAR(80) UNIQUE NOT NULL,
   ie VARCHAR(20) UNIQUE NOT NULL,
   cgc VARCHAR(20) UNIQUE NOT NULL,
   endereco VARCHAR(60),
   telefone VARCHAR(20),
   codCidade INT,
      FOREIGN KEY (codCidade) REFERENCES cidade (codCidade) ON DELETE NO ACTION ON UPDATE CASCADE
   );
```

Tabela pedido:

- O campo codPedido é serial, sendo chave primária desta tabela;
- O campo dataRealizacao é um campo de data, que tem como valor padrão a data atual, sendo que outras datas podem ser especificadas;
- O campo dataEntrega é um campo de data;
- O campo codFornecedor é um campo inteiro;
- O campo valor é um campo numérico de tamanho 20 com duas casas decimais;

Como restrições de integridade referencial temos que um fornecedor quando for apagado essa deleção será propagada à tabela pedido, e se o mesmo for atualizado, deverá ficar como nulo no fornecedor.

```
CREATE TABLE pedido (
   codPedido INT AUTO_INCREMENT PRIMARY KEY,
   dataRealização DATE DEFAULT (CURRENT_DATE),
   dataEntrega DATE,
   codFornecedor INT,
```

```
    valor DECIMAL (22,2),
        FOREIGN KEY (codFornecedor) REFERENCES fornecedor (codFornecedor) ON
DELETE CASCADE ON UPDATE SET NULL
    );
```

Tabela itemPedido:

- O campo codPedido é inteiro, sendo chave primária e estrangeira desta tabela;
- O campo codProduto é inteiro, sendo chave primária e estrangeira desta tabela;
- O campo quantidade é um numérico com tamanho 14, com 2 casas decimais, sendo seu preenchimento obrigatório e com valor positivo;

  Como restrições de integridade referencial temos que quando um pedido for apagado todos os seus itens também o serão, e quando o mesmo for atualizado, essa atualização deverá ser propagada ao pedido.

```
CREATE TABLE itemPedido (
    codPedido INT,
    codProduto INT,
    quantidade DECIMAL (16, 2) NOT NULL CHECK (quantidade > 0),
    PRIMARY KEY (codPedido, codProduto),
     FOREIGN KEY (codPedido) REFERENCES pedido (codPedido) ON DELETE CASCADE
ON UPDATE CASCADE,
    FOREIGN KEY (codProduto) REFERENCES produto (codProduto)
    );
```

Tabela contasPagar:

- O campo codTitulo é inteiro, sendo chave primária da tabela;
- O campo dataVencimento é um campo de data, de preenchimento obrigatório;
- O campo parcela é um campo inteiro;
- O campo codPedido é inteiro sendo chave estrangeira da tabela pedido;
- O campo valor é um campo numérico de tamanho 20 com duas casas decimais;
- O campo dataPagamento é um campo de data;
- O campo localPagamento é um campo de texto, de tamanho variável máximo 80;
- O campo juros é um campo numérico de tamanho 12 com duas casas decimais;
- O campo correcaoMonetaria é um campo numérico de tamanho 12 com duas casas decimais

  Como restrições de integridade referencial temos que quando um pedido for apagado todas as suas contas a pagar também o serão, e quando o mesmo for atualizado, essa atualização deverá ser propagada à tabela.

```
CREATE TABLE contasPagar (
```

```
    codTitulo INT PRIMARY KEY,
    dataVencimento DATE NOT NULL,
    parcela INT,
    codPedido INT,
    valor DECIMAL(22,2),
    dataPagamento DATE,
    localPagamento VARCHAR(80),
    juros DECIMAL (14,2),
    correcaoMonetaria DECIMAL (14, 2),
    FOREIGN KEY(codPedido) REFERENCES pedido (codPedido) ON DELETE CASCADE
);
```

Tabela contasReceber:

- O campo codTitulo é inteiro, sendo chave primária da tabela;
- O campo dataVencimento é um campo de data, de preenchimento obrigatório;
- O campo codVenda é inteiro sendo chave estrangeira da tabela venda, de preenchimento obrigatório;
- O campo parcela é um campo inteiro;
- O campo valor é um campo numérico de tamanho 20 com duas casas decimais;
- O campo dataPagamento é um campo de data;
- O campo localPagamento é um de texto caracter, de tamanho variável máximo 80;
- O campo juros é um campo numérico de tamanho 12 com duas casas decimais;
- O campo correcaoMonetaria é um campo numérico de tamanho 12 com duas casas decimais

Como restrições de integridade referencial temos que quando uma venda for apagada todas as suas contas a receber também o serão, e quando a mesma for atualizado, essa atualização deverá ser propagada à tabela.

```
CREATE TABLE contasReceber (
    codTitulo INT PRIMARY KEY,
    dataVencimento DATE NOT NULL,
    codVenda  INT NOT NULL,
    parcela INT,
    valor DECIMAL (22, 2),
    dataPagamento DATE,
    localPagamento VARCHAR(80),
    juros DECIMAL (14, 2),
    correcaoMonetaria DECIMAL (14, 2),
   FOREIGN KEY (codVenda) REFERENCES venda (codVenda) ON DELETE
CASCADE  ON UPDATE CASCADE
    );
```
