

## TABELAS GERADAS A PARTIR DO DIAGRAMA ENTIDADE RELACIONAMENTO

### Observações:

- **Sublinhado** representa chave primária e **sobrelinha** representa chave estrangeira.
- As partes em **itálico** são apenas comentários para explicar alguns detalhes da tabela.

Funcionarios (ID, Nome, CPF, DataNascimento, AdministradorID)

*AdministradorID é a chave primária do outro funcionário (administrador) que o cadastrou.*

Caixas (FuncionarioID, EAdministrador)

*EAdministrador é um campo booleano, que diz se um caixa é administrador ou não (um administrador pode fazer tudo que um caixa faz).*

Entregadores (FuncionarioID, CNH, PlacaVeiculo)

Telefones (DDDeNumero, FuncionarioID)

*Telefone era um atributo multivalorado e virou uma tabela.*

Pedidos (ID, VolumeTotal, PesoTotal, ValorTotal, Status, Descricao, EntregadorID, DataHoraEntrega)

Produtos (ID, Nome, Preco, Foto, Peso, Volume, Descricao, CaixaID)

*Foto guardará o caminho da foto no sistema de arquivos.*

EnderecosEntrega (PedidoID, Latitude, Longitude, Descricao)

*Endereço é entidade fraca.*

CaixasGerenciamPedidos (CaixaID, PedidoID, DataHoraAbertura)

*Tabela gerada do relacionamento n para n.*

CaixasGerenciamProdutos (CaixaID, ProdutoID)

*Tabela gerada do relacionamento n para n.*

PedidosContemProdutos (PedidoID, ProdutoID, Quantidade)

*Tabela gerada do relacionamento n para n.*

## CÓDIGO SQL PARA ADICIONAR AS TABELAS ACIMA EM UM BANCO DE DADOS

```
CREATE TABLE IF NOT EXISTS Funcionarios (  
    ID INTEGER AUTO_INCREMENT,  
    Nome VARCHAR(255) NOT NULL,  
    CPF VARCHAR(11) NOT NULL UNIQUE,  
    DataNascimento DATETIME NOT NULL,  
    AdministradorID INTEGER NOT NULL,  
    PRIMARY KEY (ID),  
    FOREIGN KEY (AdministradorID ) REFERENCES Funcionarios(FuncionarioID)  
);
```

```
CREATE TABLE IF NOT EXISTS Caixas (  
    FuncionarioID INTEGER NOT NULL,  
    EAdministrador BOOL DEFAULT false,  
    PRIMARY KEY (FuncionarioID),  
    FOREIGN KEY (FuncionarioID ) REFERENCES Funcionarios (FuncionarioID),  
);
```

```
CREATE TABLE IF NOT EXISTS Entregadores (  
    FuncionarioID INTEGER NOT NULL,  
    CNH VARCHAR(20) NOT NULL,  
    Placa VARCHAR(7) NOT NULL,  
    PRIMARY KEY (FuncionarioID),  
    FOREIGN KEY (FuncionarioID ) REFERENCES Funcionarios (FuncionarioID),  
);
```

```
CREATE TABLE IF NOT EXISTS Telefones (  
    DDDeNumero VARCHAR(11) NOT NULL,  
    FuncionarioID INTEGER NOT NULL,  
    PRIMARY KEY (DDDeNumero),  
    FOREIGN KEY (FuncionarioID ) REFERENCES Funcionarios (FuncionarioID),  
);
```

```
CREATE TABLE IF NOT EXISTS Pedidos (  
    ID INTEGER AUTO_INCREMENT,  
    VolumeTotal FLOAT,  
    PesoTotal FLOAT,  
    ValorTotal FLOAT,  
    Status VARCHAR(20),  
    Descricao VARCHAR(800),  
    EntregadorID INTEGER NOT NULL,  
    DataHoraEntrega DATETIME,  
    PRIMARY KEY (ID),  
    FOREIGN KEY ( EntregadorID ) REFERENCES Entregadores (FuncionarioID),  
);
```

```
CREATE TABLE IF NOT EXISTS Produtos (  
    ID INTEGER AUTO_INCREMENT,  
    Nome VARCHAR(80) NOT NULL,  
    Preco FLOAT NOT NULL,  
    Foto VARCHAR(200),  
    Peso FLOAT,  
    Volume FLOAT,  
    Descricao VARCHAR(800),  
    CaixaID INTEGER NOT NULL,  
    PRIMARY KEY (ID),  
    FOREIGN KEY ( CaixaID ) REFERENCES Caixas (FuncionarioID),  
);
```

```
CREATE TABLE IF NOT EXISTS EnderecosEntrega (  
    PedidoID INTEGER NOT NULL,  
    Latitude FLOAT NOT NULL,  
    Longitude FLOAT NOT NULL,  
    Descricao VARCHAR(800),  
    PRIMARY KEY (PedidoID),  
    FOREIGN KEY ( PedidoID ) REFERENCES Pedidos (ID),  
);
```

```
CREATE TABLE IF NOT EXISTS CaixasGerenciamPedidos (  
    CaixaID INTEGER NOT NULL,  
    PedidoID INTEGER NOT NULL,  
    DataHoraAbertura DATETIME NOT NULL,  
    PRIMARY KEY (CaixaID),  
    PRIMARY KEY (PedidoID),  
    FOREIGN KEY ( PedidoID ) REFERENCES Pedidos (ID),  
    FOREIGN KEY ( CaixaID ) REFERENCES Caixas (FuncionarioID),  
);
```

```
CREATE TABLE IF NOT EXISTS CaixasGerenciamProdutos (  
    CaixaID INTEGER NOT NULL,  
    ProdutoID INTEGER NOT NULL,  
    PRIMARY KEY (CaixaID),  
    PRIMARY KEY (ProdutoID),  
    FOREIGN KEY ( ProdutoID ) REFERENCES Produtos (ID),  
    FOREIGN KEY ( CaixaID ) REFERENCES Caixas (FuncionarioID),  
);
```

```
CREATE TABLE IF NOT EXISTS PedidosContemProdutos (  
    ProdutoID INTEGER NOT NULL,  
    PedidoID INTEGER NOT NULL,  
    Quantidade INTEGER NOT NULL,  
    PRIMARY KEY (ProdutoID),  
    PRIMARY KEY (PedidoID),  
    FOREIGN KEY ( PedidoID ) REFERENCES Pedidos (ID),  
    FOREIGN KEY ( ProdutoID ) REFERENCES Produtos (ID),  
);
```