

# Script de Criação

```
create database if not exists dbfooddelivery;
use dbfooddelivery;

create table if not exists tb_usuario (
    usuario VARCHAR(20) not null,
    senha CHAR(8) not null,
    data_criacao DATETIME not null,
    data_ultima_atualizacao DATETIME,
    ativo BIT not null,
    primary key (usuario)
);

create table if not exists tb_voucher (
    codigo_promo VARCHAR(15) not null,
    valor DECIMAL(7,2),
    porcentagem DECIMAL(3,2),
    tipo_voucher BIT not null,
    data_inicio DATETIME not null,
    data_fim DATETIME not null,
    descricao VARCHAR(144) not null,
    quantidade INTEGER not null,
    primary key (codigo_promo)
);

create table if not exists tb_endereco (
    cep CHAR(8) not null,
    tipo_logradouro VARCHAR(12),
    logradouro VARCHAR(40),
    bairro VARCHAR(40),
    cidade VARCHAR(30) not null,
    uf CHAR(2) not null,
    primary key (cep)
);

create table if not exists tb_categoria_estabelecimento (
    categoria_estabelecimento_id INTEGER not null auto_increment,
    nome VARCHAR(40) not null unique,
    primary key (categoria_estabelecimento_id)
);

create table if not exists tb_dias_semana (
    dia_id INTEGER not null auto_increment,
    nome VARCHAR(15) not null unique,
    primary key (dia_id)
);

create table if not exists tb_categoria_cardapio (
    categoria_cardapio_id INTEGER not null auto_increment,
    nome VARCHAR(20) not null unique,
    primary key (categoria_cardapio_id)
);

create table if not exists tb_categoria_item (
    categoria_item_id INTEGER not null auto_increment,
    nome VARCHAR(20) not null unique,
    primary key (categoria_item_id)
);
```

```

create table if not exists tb_cliente (
    cliente_id INTEGER not null auto_increment,
    usuario VARCHAR(20) not null unique,
    cpf CHAR(11) not null unique,
    rg VARCHAR(11) not null,
    pre_nome VARCHAR(15) not null,
    sobrenome VARCHAR(80) not null,
    sexo CHAR(1),
    data_nascimento DATE not null,
    primary key (cliente_id),
    foreign key (usuario) references tb_usuario (usuario)
);

create table if not exists tb_endereco_cliente (
    endereco_cliente_id INTEGER not null auto_increment,
    cliente_id INTEGER not null,
    cep CHAR(8) not null,
    numero VARCHAR(8),
    ponto_referencia VARCHAR(40),
    complemento VARCHAR(40),
    primary key (endereco_cliente_id),
    foreign key (cliente_id) references tb_cliente (cliente_id),
    foreign key (cep) references tb_endereco (cep)
);

create table if not exists tb_estabelecimento (
    estabelecimento_id INTEGER not null auto_increment,
    categoria_estabelecimento_id INTEGER not null,
    cnpj CHAR(14) not null unique,
    razao_social VARCHAR(60) not null unique,
    nome_fantasia VARCHAR(60) not null unique,
    descricao VARCHAR(144),
    email VARCHAR(30) not null,
    nota DECIMAL(3,2) not null,
    primary key (estabelecimento_id),
    foreign key (categoria_estabelecimento_id) references tb_categoria_estabelecimento (categoria_estabelecimento_id)
);

create table if not exists tb_contato_estabelecimento (
    estabelecimento_id INTEGER not null,
    contato CHAR(11) not null,
    foreign key (estabelecimento_id) references tb_estabelecimento (estabelecimento_id)
);

create table if not exists tb_estabelecimento_endereco (
    estabelecimento_id INTEGER,
    cep CHAR(8),
    numero VARCHAR(8),
    ponto_referencia VARCHAR(40),
    complemento VARCHAR(40),
    primary key (estabelecimento_id , cep),
    foreign key (estabelecimento_id) references tb_estabelecimento (estabelecimento_id),
    foreign key (cep) references tb_endereco (cep)
);

create table if not exists tb_horario_funcionamento (
    estabelecimento_id INTEGER not null,
    dia_id INTEGER not null,
    horario_inicio TIME not null,
    horario_fim TIME not null,
    almoco BIT not null,
    foreign key (estabelecimento_id) references tb_estabelecimento (estabelecimento_id) ,
    foreign key (dia_id) references tb_dias_semana (dia_id)
);

```

```

create table if not exists tb_status(
    status_id integer not null auto_increment,
    nome varchar(30) not null unique,
    status_anterior integer,
    status_proximo integer,
    primary key(status_id),
    foreign key(status_anterior) references tb_status(status_id),
    foreign key(status_proximo) references tb_status(status_id)
);

create table if not exists tb_pedido (
    pedido_id INTEGER not null auto_increment,
    endereco_cliente_id INTEGER not null,
    estabelecimento_id INTEGER not null,
    cliente_id INTEGER not null,
    status_id INTEGER not null,
    data_hora_solicitacao DATETIME not null,
    data_hora_entrega DATETIME,
    total DECIMAL(7,2) not null,
    desconto DECIMAL(7,2) not null,
    primary key (pedido_id),
    foreign key (endereco_cliente_id) references tb_endereco_cliente (endereco_cliente_id),
    foreign key (estabelecimento_id) references tb_estabelecimento (estabelecimento_id),
    foreign key (cliente_id) references tb_cliente (cliente_id),
    foreign key (status_id) references tb_status (status_id)
);

create table if not exists tb_pedido_historico (
    pedido_id INTEGER not null,
    status INTEGER not null,
    data_hora DATETIME not null,
    foreign key (pedido_id) references tb_pedido (pedido_id)
);

create table if not exists tb_cartao (
    cartao_id INTEGER not null,
    cliente_id INTEGER not null,
    numero CHAR(16) not null,
    nome VARCHAR(20) not null,
    validade CHAR(5) not null,
    cvv CHAR(3) not null,
    primary key (cartao_id),
    foreign key (cliente_id) references tb_cliente (cliente_id)
);

create table if not exists tb_pagamento (
    pedido_id INTEGER not null,
    cartao_id INTEGER,
    codigo_promo VARCHAR(15),
    valor DECIMAL(7,2) not null,
    data_hora DATETIME not null,
    foreign key (pedido_id) references tb_pedido (pedido_id),
    foreign key (cartao_id) references tb_cartao (cartao_id),
    foreign key (codigo_promo) references tb_voucher (codigo_promo)
);

create table if not exists tb_cardapio (
    cardapio_id INTEGER not null auto_increment,
    categoria_cardapio_id INTEGER not null,
    estabelecimento_id INTEGER not null,
    nome VARCHAR(20),
    primary key (cardapio_id),
    foreign key (categoria_cardapio_id) references tb_categoria_cardapio (categoria_cardapio_id),
    foreign key (estabelecimento_id) references tb_estabelecimento (estabelecimento_id)
);

```

```

);

create table if not exists tb_item_cardapio (
    item_id INTEGER not null auto_increment,
    cardapio_id INTEGER not null,
    categoria_item_id INTEGER not null,
    nome VARCHAR(20) not null,
    preco_unitario DECIMAL(6,2) not null,
    descricao VARCHAR(30) not null,
    observacoes VARCHAR(144),
    primary key (item_id),
    foreign key (cardapio_id) references tb_cardapio (cardapio_id),
    foreign key (categoria_item_id) references tb_categoria_item (categoria_item_id)
);

create table if not exists tb_item_pedido (
    pedido_id INTEGER not null auto_increment,
    item_cardapio_id INTEGER not null,
    quantidade INTEGER not null,
    observacao VARCHAR(144),
    primary key (pedido_id, item_cardapio_id),
    foreign key (pedido_id) references tb_pedido (pedido_id),
    foreign key (item_cardapio_id) references tb_item_cardapio (item_id)
);

```