

--Tabela Usuario

```
CREATE TABLE usuario (  
    idUsuario SERIAL PRIMARY KEY, -- Chave primária com incremento automático  
    nomeUsuario VARCHAR(100) NOT NULL, -- Nome do usuário, não pode ser nulo  
    senha VARCHAR(255) NOT NULL, -- Senha do usuário, não pode ser nulo  
    CPF CHAR(11) NOT NULL UNIQUE -- CPF com 11 caracteres, único  
);
```

--Tabela Mesas

```
CREATE TABLE mesas (  
    numeroMesa SERIAL PRIMARY KEY, -- Chave primária com incremento automático  
    usuarioMesa INT, -- Chave estrangeira que referencia o idUsuario da tabela usuario  
    CONSTRAINT fk_usuario FOREIGN KEY (usuarioMesa) REFERENCES usuario(idUsuario)  
);
```

--Tabela Bebidas

```
CREATE TABLE bebidas (  
    idBebida SERIAL PRIMARY KEY, -- Chave primária com incremento automático  
    nomeBebida VARCHAR(100) NOT NULL, -- Nome da bebida, não pode ser nulo  
    qntBebida INT NOT NULL, -- Quantidade disponível da bebida  
    valorBebida FLOAT NOT NULL, -- Valor da bebida com duas casas decimais  
    disponivelBebida BOOLEAN NOT NULL -- Indica se a bebida está disponível (TRUE ou FALSE)  
);
```

--Tabela Lanches

```
CREATE TABLE lanches (  
    idLanche SERIAL PRIMARY KEY, -- Chave primária com incremento automático  
    nomeLanche VARCHAR(100) NOT NULL, -- Nome do lanche, não pode ser nulo  
    valorLanche FLOAT NOT NULL, -- Valor do lanche em ponto flutuante  
    disponivelLanche BOOLEAN NOT NULL -- Indica se o lanche está disponível (TRUE ou FALSE)  
);
```

--Tabela Hamburguers

```
CREATE TABLE hamburguers (  
    idIngrediente SERIAL PRIMARY KEY, -- Chave primária com incremento automático  
    tipoIngrediente VARCHAR(50) NOT NULL, -- Tipo do ingrediente (ex: carne, pão, molho)  
    nomeIngrediente VARCHAR(100) NOT NULL, -- Nome do ingrediente, não pode ser nulo  
    valorIngrediente FLOAT NOT NULL, -- Valor do ingrediente em ponto flutuante  
    disponivelIngrediente BOOLEAN NOT NULL -- Indica se o ingrediente está disponível (TRUE ou FALSE)  
);
```

--TabelaPedidos

```
CREATE TABLE pedidos (  
    idPedido SERIAL PRIMARY KEY, -- Chave primária com incremento automático  
    lanchePedido INT, -- Chave estrangeira que referencia idLanche da tabela lanches  
    bebidaPedido INT, -- Chave estrangeira que referencia idBebida da tabela bebidas  
    hamburguerPedido INT, -- Chave estrangeira que referencia idIngrediente da tabela  
hamburguers  
    valorPedido FLOAT NOT NULL, -- Valor total do pedido  
    mesaPedido INT, -- Chave estrangeira que referencia numeroMesa da tabela mesas  
    statusPedido VARCHAR(50) NOT NULL, -- Status do pedido (ex: "em andamento",  
"concluído")  
    usuarioPedido INT, -- Chave estrangeira que referencia idUsuario da tabela usuario  
    CONSTRAINT fk_lanche FOREIGN KEY (lanchePedido) REFERENCES lanches(idLanche),  
    CONSTRAINT fk_bebida FOREIGN KEY (bebidaPedido) REFERENCES bebidas(idBebida),  
    CONSTRAINT fk_hamburguer FOREIGN KEY (hamburguerPedido) REFERENCES  
hamburguers(idIngrediente),  
    CONSTRAINT fk_mesa FOREIGN KEY (mesaPedido) REFERENCES mesas(numeroMesa),  
    CONSTRAINT fk_usuario FOREIGN KEY (usuarioPedido) REFERENCES usuario(idUsuario)  
);
```

--Tabela Conta

```
CREATE TABLE contas (  
    idConta SERIAL PRIMARY KEY, -- Chave primária com incremento automático  
    valorConta FLOAT NOT NULL, -- Valor total da conta  
    mesaConta INT, -- Chave estrangeira que referencia numeroMesa da tabela mesas  
    pedidoConta INT, -- Chave estrangeira que referencia idPedido da tabela pedidos  
    CONSTRAINT fk_mesaConta FOREIGN KEY (mesaConta) REFERENCES  
mesas(numeroMesa),  
    CONSTRAINT fk_pedidoConta FOREIGN KEY (pedidoConta) REFERENCES  
pedidos(idPedido)  
);
```