



- 

```
create table Cliente (  
    idCli int primary key not null auto_increment,  
    nomeCli varchar(45) not null,  
    emailCli varchar(45) not null,  
    telCli varchar(45) not null  
);
```

Criando a tabela 'Cliente' onde temos 'idCli' como chave primária, recebendo parâmetros 'not null' (não nulo) e 'auto_increment' para receber um id automaticamente. 'nomeCli', 'emailCli' e 'telCli' que recebem as informações do mesmo, guardando no tipo de dado varchar.

- 

```
create table EnderecoCli (  
    idEnderecoCli int primary key not null auto_increment,  
    idCli int,  
    ruaCli varchar(45) not null,  
    bairroCli varchar(45) not null,  
    cepCli varchar (8) not null,  
    foreign key (idCli) references Cliente(idCli)  
);
```


Criando a tabela 'EnderecoCli' para registrar a rua, o bairro e o cep de cada cliente da lanchonete. 'idEnderecoCli' chave primária que não pode ser nula (not null) e recebe um id automático com 'auto_increment'. Chave estrangeira 'idCli' referencia cada cliente na tabela.

- ```
create table Funcionario (
 idFuncionario int primary key not null auto_increment,
 nomeFunc varchar(45) not null,
 cargoFunc varchar(45) not null,
 dtContrato date not null,
 salarioFunc decimal(6,2) not null,
 dtNasc date,
 sexo enum('M', 'F')
);
```

Criando a tabela 'Funcionario' para registrar seu nome, cargo, data de contrato, salario, data de nascimento e sexo. Chave primária 'idFuncionario' 'not null' (não nulo) e 'auto\_increment'. Data de contrato recebe o tipo 'date' para armazenar datas. 'SalarioFunc' recebe o tipo 'decimal' que pode ter 6 casas ao total e somente duas após a vírgula. 'Data de nascimento' recebe 'date' para datas. 'Sexo' recebe 'enum' para receber somente os valores 'M' para masculino e 'F' para feminino.


- ```
create table EnderecoFuncionario (  
    idEnderecoFuncionario int primary key not null auto_increment,  
    idFuncionario int,  
    ruaFunc varchar(45) not null,  
    bairroFunc varchar(45) not null,  
    numeroFunc int,  
    foreign key (idFuncionario) references Funcionario(idFuncionario)  
);
```

Criando a tabela 'EnderecoFuncionario' para registrar a rua, o bairro e o numero. 'idEnderecoFuncionario' chave primária auto incrementável que não pode ser nulo. 'idFuncionario' chave estrangeira que referencia a tabela Funcionario.

- 


```
create table Pedido (  
    nmrPedido int primary key not null auto_increment,  
    idCli int,  
    idFuncionario int,  
    dataPed date not null,  
    horaPed time not null,  
    estadoPed varchar(45),  
    foreign key (idCli) references Cliente(idCli),  
    foreign key (idFuncionario) references Funcionario(idFuncionario)  
);
```

Criando a tabela 'Pedido' para registrar o pedido de cada cliente. nmrPedido chave primária não nula e auto incrementável. Guarda a data do pedido, hora do pedido e estado do pedido(Pendente, pronto, em andamento). Chaves estrangeiras 'idCli' referencia a tabela 'Cliente' e 'idFuncionario' referencia a tabela Funcionario.

- 


```
create table Pedido_ItemMenu (  
    idPedidoItem int primary key not null auto_increment,  
    nmrPedido int,  
    codItem int,  
    qntdItem int not null,  
    tempoPreparo time,  
    sttsEntrega varchar(20),  
    foreign key (nmrPedido) references Pedido(nmrPedido),  
    foreign key (codItem) references ItemMenu(codItem)  
);
```

Criando a tabela-associativa 'Pedido_ItemMenu' para guardar o tempo de preparo do pedido, o status da entrega e a quantidade de itens. Chaves estrangeiras 'nmrPedido' para referenciar a tabela 'Pedido' e 'codItem' para referenciar cada item que tem na tabela 'ItemMenu'.

- 


```
create table ItemMenu (  
    codItem int primary key not null auto_increment,  
    nomeItem varchar(45) not null,  
    descItem varchar(45) not null,  
    precoItem decimal(4,2) not null,  
    categoriaItem varchar(45)  
);
```

Criando a Tabela 'ItemMenu' que mostra cada item que possui no cardápio. Cada item possui um código(PK), o nome, a descrição, o preço de cada e a categoria(Hambúrguer, sobremesa, salada).

- 


```
create table Fornecedor (  
    idFornecedor int primary key not null auto_increment,  
    nomeEmpresa varchar(45) not null,  
    representante varchar(45) not null,  
    telFornecedor varchar(45) not null,  
    emailFornecedor varchar(45) not null,  
    listaIngredientes varchar(45) not null  
);
```

Criando a tabela 'Fornecedor' para cadastrar cada empresa que abastece a lanchonete. 'idFornecedor' (PK), registra o nome da empresa, o representante, o telefone, o email e a lista de ingredientes que cada empresa deve fornecer.

- 


```
create table EnderecoFornecedor (  
    idEnderecoFornecedor int primary key not null auto_increment,  
    idFornecedor int,  
    cepForne varchar(8) not null,  
    ruaForne varchar(45) not null,  
    numeroForne varchar(45) not null,  
    bairroForne varchar(45) not null,  
    cidadeForne varchar(45) not null,  
    estadoForne varchar(45) not null,  
    foreign key (idFornecedor) references Fornecedor(idFornecedor)  
);
```

Criando a tabela 'EnderecoFornecedor' para cadastrar onde cada empresa está localizada. 'idEnderecoFornecedor' (PK), guarda os registros do cep do fornecedor, rua, numero, bairro, cidade e estado. Chave estrangeira 'idFornecedor' para referenciar a tabela 'Fornecedor'.

- 


```
create table Ingredientes (  
    idIngrediente int primary key not null auto_increment,  
    idFornecedor int,  
    nomeIng varchar(45) not null,  
    estoqueDisponivel tinyint not null,  
    foreign key (idFornecedor) references Fornecedor(idFornecedor)  
);
```

Criando a tabela 'Ingredientes' para registrar o nome e o estoque disponível de cada. 'idIngrediente' (PK) e chave estrangeira 'idFornecedor' para saber o que será preciso reabastecer.

- 

```
create table ItemMenu_Ingredientes (  
    idItemMenu_Ingredientes int primary key not null auto_increment,  
    codItem int,  
    idIngrediente int,  
    qntdIngredientes int,  
    foreign key (codItem) references ItemMenu(codItem),  
    foreign key (idIngrediente) references Ingredientes(idIngrediente)  
);
```

Criando a tabela-associativa 'ItemMenu_Ingredientes' para guardar a quantidade de ingredientes em cada item do menu. 'idItemMenu_Ingredientes(PK) e chaves estrangeiras 'codItem' para referenciar os itens do menu e 'idIngrediente' para saber qual ingrediente utilizar no item do menu.

- 

```
create table Pedido_Ingredientes (  
    idPedido_Ingredientes int primary key not null auto_increment,  
    nmrPedido int,  
    idIngrediente int,  
    qntdPedido int,  
    foreign key (nmrPedido) references Pedido(nmrPedido),  
    foreign key (idIngrediente) references Ingredientes(idIngrediente)  
);
```

Criando a tabela-associativa 'Pedido_Ingredientes' para verificar quantos ingredientes será utilizado em cada pedido. 'idPedido_Ingredientes' (PK) e chaves estrangeiras 'nmrPedido' para referenciar a tabela Pedido e 'idIngrediente' para referenciar a tabela 'Ingredientes'.