# Projeto Banco de Dados - PizzariaSenai

Este documento descreve o projeto completo de banco de dados para a PizzariaSenai, incluindo o modelo relacional, scripts SQL para criação das tabelas e um exemplo de CRUD e relatórios implementados em Python para integração com o SQL Server.

## Modelo Relacional

Cliente (ClienteID PK, Nome, Telefone, Logradouro, Numero, Bairro, Cidade, UF, CEP)  
Pizza (PizzaID PK, Sabor, Tamanho, Preco)  
Pedido (PedidoID PK, ClienteID FK, DataHora, ValorTotal)  
ItemPedido (ItemPedidoID PK, PedidoID FK, PizzaID FK, Quantidade, ValorUnitario)

## Script SQL Server

CREATE DATABASE PizzariaSenai;  
GO  
  
USE PizzariaSenai;  
GO  
  
CREATE TABLE Cliente (  
 ClienteID INT PRIMARY KEY IDENTITY(1,1),  
 Nome VARCHAR(100) NOT NULL,  
 Telefone VARCHAR(15) NOT NULL UNIQUE,  
 Logradouro VARCHAR(150) NOT NULL,  
 Numero VARCHAR(10) NOT NULL,  
 Bairro VARCHAR(50) NOT NULL,  
 Cidade VARCHAR(50) NOT NULL,  
 UF CHAR(2) NOT NULL,  
 CEP CHAR(9) NOT NULL  
);  
  
CREATE TABLE Pizza (  
 PizzaID INT PRIMARY KEY IDENTITY(1,1),  
 Sabor VARCHAR(50) NOT NULL,  
 Tamanho CHAR(1) NOT NULL CHECK (Tamanho IN ('P','M','G')),  
 Preco DECIMAL(10,2) NOT NULL  
);  
  
CREATE TABLE Pedido (  
 PedidoID INT PRIMARY KEY IDENTITY(1,1),  
 ClienteID INT NOT NULL,  
 DataHora DATETIME NOT NULL DEFAULT GETDATE(),  
 ValorTotal DECIMAL(10,2) NOT NULL,  
 FOREIGN KEY (ClienteID) REFERENCES Cliente(ClienteID)  
);  
  
CREATE TABLE ItemPedido (  
 ItemPedidoID INT PRIMARY KEY IDENTITY(1,1),  
 PedidoID INT NOT NULL,  
 PizzaID INT NOT NULL,  
 Quantidade INT NOT NULL CHECK (Quantidade > 0),  
 ValorUnitario DECIMAL(10,2) NOT NULL,  
 FOREIGN KEY (PedidoID) REFERENCES Pedido(PedidoID),  
 FOREIGN KEY (PizzaID) REFERENCES Pizza(PizzaID)  
);

## Script Python - CRUD + Relatórios

import pyodbc  
  
server = 'localhost'  
database = 'PizzariaSenai'  
username = 'sa'  
password = '123456'  
driver = '{ODBC Driver 17 for SQL Server}'  
  
conn = pyodbc.connect(  
 f'DRIVER={driver};SERVER={server};DATABASE={database};UID={username};PWD={password}'  
)  
cursor = conn.cursor()  
  
def inserir\_cliente(nome, telefone, logradouro, numero, bairro, cidade, uf, cep):  
 cursor.execute(  
 "INSERT INTO Cliente (Nome, Telefone, Logradouro, Numero, Bairro, Cidade, UF, CEP) VALUES (?, ?, ?, ?, ?, ?, ?, ?)",  
 (nome, telefone, logradouro, numero, bairro, cidade, uf, cep)  
 )  
 conn.commit()  
 print("Cliente inserido com sucesso.")  
  
def listar\_clientes():  
 cursor.execute("SELECT \* FROM Cliente")  
 for row in cursor.fetchall():  
 print(row)  
  
def atualizar\_cliente(cliente\_id, novo\_telefone):  
 cursor.execute("UPDATE Cliente SET Telefone = ? WHERE ClienteID = ?", (novo\_telefone, cliente\_id))  
 conn.commit()  
  
def deletar\_cliente(cliente\_id):  
 cursor.execute("DELETE FROM Cliente WHERE ClienteID = ?", (cliente\_id,))  
 conn.commit()  
  
def inserir\_pizza(sabor, tamanho, preco):  
 cursor.execute(  
 "INSERT INTO Pizza (Sabor, Tamanho, Preco) VALUES (?, ?, ?)",  
 (sabor, tamanho, preco)  
 )  
 conn.commit()  
  
def inserir\_pedido(cliente\_id, itens):  
 valor\_total = sum(qtd \* preco for (\_, qtd, preco) in itens)  
 cursor.execute(  
 "INSERT INTO Pedido (ClienteID, DataHora, ValorTotal) VALUES (?, GETDATE(), ?)",  
 (cliente\_id, valor\_total)  
 )  
 conn.commit()  
 pedido\_id = cursor.execute("SELECT SCOPE\_IDENTITY()").fetchone()[0]  
 for pizza\_id, qtd, preco in itens:  
 cursor.execute(  
 "INSERT INTO ItemPedido (PedidoID, PizzaID, Quantidade, ValorUnitario) VALUES (?, ?, ?, ?)",  
 (pedido\_id, pizza\_id, qtd, preco)  
 )  
 conn.commit()  
  
def listar\_pedidos\_cliente(cliente\_id):  
 cursor.execute(  
 "SELECT Pedido.PedidoID, Pedido.DataHora, Pedido.ValorTotal FROM Pedido WHERE ClienteID = ?",  
 (cliente\_id,)  
 )  
 for row in cursor.fetchall():  
 print(row)  
  
def faturamento\_por\_dia():  
 cursor.execute(  
 "SELECT CAST(DataHora AS DATE) AS Dia, SUM(ValorTotal) FROM Pedido GROUP BY CAST(DataHora AS DATE) ORDER BY Dia"  
 )  
 for row in cursor.fetchall():  
 print(row)  
  
def pizza\_mais\_vendida():  
 cursor.execute(  
 "SELECT TOP 1 Pizza.Sabor, SUM(ItemPedido.Quantidade) FROM ItemPedido JOIN Pizza ON ItemPedido.PizzaID = Pizza.PizzaID GROUP BY Pizza.Sabor ORDER BY SUM(ItemPedido.Quantidade) DESC"  
 )  
 print(cursor.fetchone())