

Script do Banco

1-Acessando o Postgres:

1.1-docker exec -it projetobd psql -U postgres -d postgres

2-Criação do Banco:

2.1-Criação do banco de dados:

```
CREATE DATABASE EsmalteriaDB;
```

2.2-Acessando o banco:

```
\c esmalteriadb
```

2.3-Usuários do sistema:

```
CREATE TABLE Usuario (  
    id_usuario SERIAL PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    telefone VARCHAR(20),  
    cep VARCHAR(10),  
    senha VARCHAR(100) NOT NULL,  
    tipo_usuario VARCHAR(20) CHECK (tipo_usuario IN ('cliente', 'funcionario', 'administrador'))  
NOT NULL  
);
```

2.4-Clientes:

```
CREATE TABLE Cliente (  
    id_usuario INT PRIMARY KEY,  
    pontos_fidelidade INT DEFAULT 0,  
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)  
);
```

2.5-Funcionários:

```
CREATE TABLE Funcionario (  
    id_usuario INT PRIMARY KEY,  
    cpf VARCHAR(14) UNIQUE NOT NULL,  
    data_nascimento DATE,  
    area_atuacao VARCHAR(100),  
    status VARCHAR(20) DEFAULT 'ativo' CHECK (status IN ('ativo', 'congelado', 'excluido')),  
    avaliacao_media DECIMAL(2,1) DEFAULT 0.0,  
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)  
);
```

2.6-Administradores:

```
CREATE TABLE Administrador (  
    id_usuario INT PRIMARY KEY,  
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)  
);
```

2.7-Tratamentos:

```
CREATE TABLE Tratamento (  
    id_tratamento SERIAL PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    descricao TEXT,  
    preco DECIMAL(10,2) NOT NULL  
);
```

2.8-Produtos:

```
CREATE TABLE Produto (  
    id_produto SERIAL PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    marca VARCHAR(100),  
    descricao TEXT,  
    preco DECIMAL(10,2) NOT NULL,  
    quantidade_estoque INT DEFAULT 0,  
    data_inclusao DATE,  
    nota_fiscal VARCHAR(50)  
);
```

2.9-Agendamentos:

```
CREATE TABLE Agendamento (  
    id_agendamento SERIAL PRIMARY KEY,  
    data DATE NOT NULL,  
    hora TIME NOT NULL,  
    status VARCHAR(20) DEFAULT 'marcado' CHECK (status IN ('marcado', 'cancelado',  
'realizado')),  
    id_cliente INT NOT NULL,  
    id_funcionario INT NOT NULL,  
    id_tratamento INT NOT NULL,  
    FOREIGN KEY (id_cliente) REFERENCES Cliente(id_usuario),  
    FOREIGN KEY (id_funcionario) REFERENCES Funcionario(id_usuario),  
    FOREIGN KEY (id_tratamento) REFERENCES Tratamento(id_tratamento)  
);
```

2.10-Pagamentos:

```
CREATE TABLE Pagamento (  

```

```

id_pagamento SERIAL PRIMARY KEY,
tipo_pagamento VARCHAR(20) CHECK (tipo_pagamento IN ('cartao', 'pix', 'dinheiro')) NOT
NULL,
valor DECIMAL(10,2) NOT NULL,
data DATE NOT NULL,
id_agendamento INT UNIQUE NOT NULL,
id_cliente INT NOT NULL,
id_funcionario INT NOT NULL,
nome_cartao VARCHAR(100),
bandeira_cartao VARCHAR(20),
numero_cartao VARCHAR(20),
cod_seguranca VARCHAR(4),
vencimento_cartao DATE,
senha_compra VARCHAR(20),
qr_code_pix TEXT,
chave_pix VARCHAR(100),
FOREIGN KEY (id_agendamento) REFERENCES Agendamento(id_agendamento),
FOREIGN KEY (id_cliente) REFERENCES Cliente(id_usuario),
FOREIGN KEY (id_funcionario) REFERENCES Funcionario(id_usuario)
);

```

2.11-Avaliações:

```

CREATE TABLE Avaliacao (
id_avaliacao SERIAL PRIMARY KEY,
nota INT CHECK (nota BETWEEN 1 AND 5),
comentario TEXT,
data DATE NOT NULL,
id_cliente INT NOT NULL,
id_funcionario INT NOT NULL,
FOREIGN KEY (id_cliente) REFERENCES Cliente(id_usuario),
FOREIGN KEY (id_funcionario) REFERENCES Funcionario(id_usuario)
);

```

2.12-Reservas de Produtos:

```

CREATE TABLE ReservaProduto (
id_reserva SERIAL PRIMARY KEY,
data_reserva DATE NOT NULL,
status VARCHAR(20) DEFAULT 'reservado' CHECK (status IN ('reservado', 'retirado')),
id_cliente INT NOT NULL,
id_produto INT NOT NULL,
FOREIGN KEY (id_cliente) REFERENCES Cliente(id_usuario),
FOREIGN KEY (id_produto) REFERENCES Produto(id_produto)
);

```

2.13-Histórico:

```
CREATE TABLE Historico (  
    id_historico SERIAL PRIMARY KEY,  
    data DATE NOT NULL,  
    valor_pago DECIMAL(10,2),  
    percent_funcionario DECIMAL(5,2),  
    percent_salao DECIMAL(5,2),  
    id_agendamento INT UNIQUE NOT NULL,  
    FOREIGN KEY (id_agendamento) REFERENCES Agendamento(id_agendamento)  
);
```

2.14-Sugestões:

```
CREATE TABLE Sugestao (  
    id_sugestao SERIAL PRIMARY KEY,  
    texto TEXT NOT NULL,  
    data DATE NOT NULL,  
    anonima BOOLEAN DEFAULT TRUE,  
    tipo_usuario VARCHAR(20) CHECK (tipo_usuario IN ('cliente', 'funcionario', 'administrador')),  
    id_usuario INT,  
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)  
);
```

3-População do Banco:

3.1-Populando tabela Usuario:

```
INSERT INTO Usuario (nome, email, telefone, cep, senha, tipo_usuario) VALUES  
( 'Maria Silva', 'maria@gmail.com', '11999999999', '01001-000', 'senha123', 'cliente'),      -- id 1  
( 'João Costa', 'joao@gmail.com', '21999999999', '22041-001', 'senha123', 'cliente'),      -- id  
2  
( 'Ana Manicure', 'ana@salon.com', '11988888888', '03023-000', 'senha456', 'funcionario'),    --  
id 3  
( 'Carlos Cabeleireiro', 'carlos@salon.com', '11977777777', '04044-000', 'senha456',  
'funcionario'), -- id 4  
( 'Admin Geral', 'admin@salon.com', '11911111111', '05050-000', 'admin123', 'administrador');  
-- id 5
```

3.2-Populando Cliente:

```
INSERT INTO Cliente (id_usuario, pontos_fidelidade) VALUES  
(1, 120),  
(2, 60);
```

3.3-Populando Funcionario:

```
INSERT INTO Funcionario (id_usuario, cpf, data_nascimento, area_atuacao, status,  
avaliacao_media) VALUES
```

```
(3, '123.456.789-00', '1990-05-20', 'Manicure', 'ativo', 4.6),  
(4, '987.654.321-00', '1985-10-10', 'Corte de cabelo', 'ativo', 4.8);
```

3.4-Populando Administrador:

```
INSERT INTO Administrador (id_usuario) VALUES  
(5);
```

3.5-Populando Tratamentos:

```
INSERT INTO Tratamento (nome, descricao, preco) VALUES  
( 'Corte de Cabelo', 'Corte masculino ou feminino', 50.00),      -- id 1  
( 'Manicure', 'Cuidado com as unhas e esmaltação', 35.00);      -- id 2
```

3.6-Populando Produtos:

```
INSERT INTO Produto (nome, marca, descricao, preco, quantidade_estoque, data_inclusao,  
nota_fiscal) VALUES  
( 'Esmalte Vermelho', 'Colorama', 'Esmalte vermelho vivo', 8.99, 15, CURRENT_DATE,  
'NF12345'),  
( 'Shampoo Hidratante', 'Pantene', 'Hidrata e dá brilho', 24.90, 10, CURRENT_DATE,  
'NF12346');
```

3.7-Populando Agendamentos:

```
INSERT INTO Agendamento (data, hora, status, id_cliente, id_funcionario, id_tratamento)  
VALUES  
( '2025-07-21', '16:00:00', 'realizado', 1, 3, 2), -- Maria com Ana  
( '2025-07-22', '14:00:00', 'realizado', 2, 4, 1); -- João com Carlos
```

3.8-Populando Pagamentos:

```
INSERT INTO Pagamento (tipo_pagamento, valor, data, id_agendamento, id_cliente,  
id_funcionario) VALUES  
( 'cartao', 35.00, '2025-07-21', 1, 1, 3),  
( 'pix', 50.00, '2025-07-22', 2, 2, 4);
```

3.9-Populando Avaliações:

```
INSERT INTO Avaliacao (nota, comentario, data, id_cliente, id_funcionario) VALUES  
(5, 'Excelente atendimento da Ana!', '2025-07-21', 1, 3),  
(4, 'Corte muito bom, mas atrasou.', '2025-07-22', 2, 4);
```

3.10-Populando Reserva de Produto:

```
INSERT INTO ReservaProduto (id_cliente, id_produto, data_reserva, status) VALUES  
(1, 1, '2025-07-20', 'pendente');
```

3.11-Populando Histórico:

```
INSERT INTO Historico (id_agendamento, valor_pago, percent_funcionario, percent_salao)  
VALUES
```

```
(1, 35.00, 70, 30),  
(2, 50.00, 60, 40);
```

3.12-Populando Sugestões:

```
INSERT INTO Sugestao (descricao, data_cadastro, anonimo, id_usuario) VALUES  
( 'Adicionar mais horários no sábado', CURRENT_DATE, true, 1),  
( 'Melhorar o layout do app', CURRENT_DATE, false, 2);
```

4-Observações:

O postgres foi acessado a partir de uma imagem no docker.