

EXPERIÊNCIA PRÁTICA 4 — MINIMUNDO – BeautyCare

1. SCRIPT DE CRIAÇÃO DO BANCO DE DADOS

```
CREATE DATABASE beutycare;  
USE beutycare;
```

-- Tabela Cliente

```
CREATE TABLE cliente (  
    id_cliente INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100),  
    telefone VARCHAR(20),  
    email VARCHAR(120),  
    data_nascimento DATE  
);
```

-- Tabela Profissional

```
CREATE TABLE profissional (  
    id_profissional INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100),  
    especialidade VARCHAR(100),  
    telefone VARCHAR(20)  
);
```

-- Tabela Serviço

```
CREATE TABLE servico (  
    id_servico INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100),  
    duracao_minutos INT,  
    preco DECIMAL(10,2)  
);
```

-- Tabela Agendamento

```
CREATE TABLE agendamento (  
    id_agendamento INT AUTO_INCREMENT PRIMARY KEY,  
    data DATE,  
    horario TIME,  
    id_cliente INT,  
    id_profissional INT,  
    id_servico INT,  
    status VARCHAR(30),  
    FOREIGN KEY (id_cliente) REFERENCES cliente(id_cliente),  
    FOREIGN KEY (id_profissional) REFERENCES profissional(id_profissional),  
    FOREIGN KEY (id_servico) REFERENCES servico(id_servico)  
);
```

```
-- Tabela Atendimento
CREATE TABLE atendimento (
    id_atendimento INT AUTO_INCREMENT PRIMARY KEY,
    id_agendamento INT UNIQUE,
    observacoes TEXT,
    status_final VARCHAR(30),
    FOREIGN KEY (id_agendamento) REFERENCES agendamento(id_agendamento)
);
```

```
-- Tabela Pagamento
CREATE TABLE pagamento (
    id_pagamento INT AUTO_INCREMENT PRIMARY KEY,
    id_atendimento INT UNIQUE,
    valor DECIMAL(10,2),
    forma_pagamento VARCHAR(30),
    data_pagamento DATE,
    FOREIGN KEY (id_atendimento) REFERENCES atendimento(id_atendimento)
);
```

2. SCRIPT DE INSERT (povoar tabelas)

-- Clientes

```
INSERT INTO cliente (nome, telefone, email, data_nascimento) VALUES
("Ana Maria", "11988887777", "ana@gmail.com", "1995-08-10"),
("Beatriz Souza", "11977776666", "bia@gmail.com", "1999-02-28"),
("Carla Monteiro", "11966665555", "carla@gmail.com", "1988-11-12");
```

-- Profissionais

```
INSERT INTO profissional (nome, especialidade, telefone) VALUES
("Mariana Lopes", "Limpeza de Pele", "11999998888"),
("Juliana Azevedo", "Massoterapia", "11955554444");
```

-- Serviços

```
INSERT INTO servico (nome, duracao_minutos, preco) VALUES
("Limpeza de Pele", 60, 120.00),
("Massagem Relaxante", 45, 150.00),
("Peeling Químico", 40, 200.00);
```

-- Agendamentos

```
INSERT INTO agendamento (data, horario, id_cliente, id_profissional, id_servico,
status) VALUES
("2025-11-20", "14:00", 1, 1, 1, "agendado"),
("2025-11-21", "16:00", 2, 2, 2, "agendado"),
("2025-11-22", "10:00", 3, 1, 3, "agendado");
```

-- Atendimentos

```
INSERT INTO atendimento (id_agendamento, observacoes, status_final) VALUES  
(1, "Pele sensível, recomendado hidratação", "concluído"),  
(2, "Paciente relaxada, nenhum imprevisto", "concluído"),  
(3, "Reação leve, dentro do esperado", "concluído");
```

-- Pagamentos

```
INSERT INTO pagamento (id_atendimento, valor, forma_pagamento,  
data_pagamento) VALUES  
(1, 120.00, "Cartão", "2025-11-20"),  
(2, 150.00, "Pix", "2025-11-21"),  
(3, 200.00, "Cartão", "2025-11-22");
```

3. SELECTS (5 consultas com WHERE, ORDER BY, LIMIT, JOIN...)

SELECT 1 — Listar todos os clientes

```
SELECT * FROM cliente ORDER BY nome;
```

SELECT 2 — Agendamentos com nome do cliente e serviço

```
SELECT a.id_agendamento, c.nome AS cliente, s.nome AS servico, a.data, a.horario  
FROM agendamento a  
JOIN cliente c ON a.id_cliente = c.id_cliente  
JOIN servico s ON a.id_servico = s.id_servico;
```

SELECT 3 — Serviços acima de R\$150

```
SELECT * FROM servico WHERE preco > 150 ORDER BY preco DESC;
```

SELECT 4 — Histórico de atendimentos de um profissional

```
SELECT p.nome AS profissional, c.nome AS cliente, s.nome AS servico, a.data  
FROM agendamento a  
JOIN profissional p ON p.id_profissional = a.id_profissional  
JOIN cliente c ON c.id_cliente = a.id_cliente  
JOIN servico s ON s.id_servico = a.id_servico  
WHERE p.id_profissional = 1;
```

SELECT 5 — Top 2 serviços mais caros

```
SELECT nome, preco FROM servico ORDER BY preco DESC LIMIT 2;
```

4. UPDATES (3 comandos com condições)

UPDATE 1 — Alterar status de um agendamento

```
UPDATE agendamento SET status = 'concluído' WHERE id_agendamento = 1;
```

UPDATE 2 — Ajustar preço de um serviço

```
UPDATE servico SET preco = preco + 20 WHERE id_servico = 3;
```

UPDATE 3 — Atualizar telefone de cliente

UPDATE cliente SET telefone = "11944443333" WHERE id_cliente = 2;

5. DELETES — COM CONDIÇÃO

DELETE 1 — Deletar pagamentos com valor menor que 100

DELETE FROM pagamento WHERE valor < 100;

DELETE 2 — Remover agendamentos com status “cancelado”

DELETE FROM agendamento WHERE status = 'cancelado';

DELETE 3 — Excluir clientes cujo email está vazio

DELETE FROM cliente WHERE email IS NULL OR email = '';