

## EXPERIÊNCIA PRÁTICA IV — SQL, DML E MANIPULAÇÃO DE DADOS

### Sistema de Gestão de Academia StrongFit

#### 1. Preparação do Ambiente e Estrutura do Banco

Este EP4 utiliza o modelo lógico do EP3 para criação e manipulação de dados no banco StrongFit.

Abaixo está o script de criação das tabelas (DDL), seguindo o modelo lógico aprovado no EP3.

##### 1.1 SCRIPT DE CRIAÇÃO DAS TABELAS (DDL)

```
CREATE TABLE PLANO (  
  id_plano INT PRIMARY KEY,  
  nome VARCHAR(50),  
  valor_mensal DECIMAL(10,2),  
  descricao VARCHAR(150)  
);
```

```
CREATE TABLE ALUNO (  
  id_aluno INT PRIMARY KEY,  
  nome VARCHAR(80),  
  data_nascimento DATE,  
  telefone VARCHAR(20),  
  email VARCHAR(80),  
  id_plano INT,  
  FOREIGN KEY (id_plano) REFERENCES PLANO(id_plano)  
);
```

```
CREATE TABLE INSTRUTOR (  
  id_instrutor INT PRIMARY KEY,  
  nome VARCHAR(80),  
  especialidade VARCHAR(80),  
  telefone VARCHAR(20)  
);
```

```
CREATE TABLE AVALIACAO_FISICA (  
  id_avaliacao INT PRIMARY KEY,  
  id_aluno INT,  
  id_instrutor INT,  
  peso DECIMAL(5,2),  
  altura DECIMAL(3,2),  
  imc DECIMAL(4,2),  
  data_avaliacao DATE,  
  FOREIGN KEY (id_aluno) REFERENCES ALUNO(id_aluno),  
  FOREIGN KEY (id_instrutor) REFERENCES INSTRUTOR(id_instrutor)  
);
```

```
CREATE TABLE TREINO (  
  id_treino INT PRIMARY KEY,  
  id_aluno INT,  
  id_instrutor INT,  
  nome_treino VARCHAR(60),  
  objetivo VARCHAR(120),  
  FOREIGN KEY (id_aluno) REFERENCES ALUNO(id_aluno),  
  FOREIGN KEY (id_instrutor) REFERENCES INSTRUTOR(id_instrutor)  
);
```

```
CREATE TABLE EXERCICIO (  
  id_exercicio INT PRIMARY KEY,  
  nome VARCHAR(60),  
  grupo_muscular VARCHAR(50)  
);
```

```
CREATE TABLE TREINO_EXERCICIO (  
  id_treino INT,  
  id_exercicio INT,  
  series INT,  
  repeticoes INT,  
  carga INT,  
  PRIMARY KEY (id_treino, id_exercicio),  
  FOREIGN KEY (id_treino) REFERENCES TREINO(id_treino),
```

```
FOREIGN KEY (id_exercicio) REFERENCES EXERCICIO(id_exercicio)
);
```

```
CREATE TABLE AULA_COLETIVA (
id_aula INT PRIMARY KEY,
nome VARCHAR(60),
dia_semana VARCHAR(15),
horario TIME,
id_instrutor INT,
FOREIGN KEY (id_instrutor) REFERENCES INSTRUTOR(id_instrutor)
);
```

```
CREATE TABLE PRESENCA_AULA (
id_presenca INT PRIMARY KEY,
id_aula INT,
id_aluno INT,
data_presenca DATE,
FOREIGN KEY (id_aula) REFERENCES AULA_COLETIVA(id_aula),
FOREIGN KEY (id_aluno) REFERENCES ALUNO(id_aluno)
);
```

## -----

## 2. INSERTS – POVOAMENTO DAS TABELAS

## -----

### 2.1 INSERTS PLANO

```
INSERT INTO PLANO VALUES
(1, 'Mensal', 99.90, 'Acesso total à academia'),
(2, 'Trimestral', 249.90, 'Plano de 3 meses com desconto'),
(3, 'Anual', 799.90, 'Plano anual completo');
```

### 2.2 INSERTS INSTRUTOR

```
INSERT INTO INSTRUTOR VALUES
(1, 'Carlos Silva', 'Musculação', '11988887777'),
(2, 'Mariana Souza', 'Funcional', '11977776666');
```

### 2.3 INSERTS ALUNO

INSERT INTO ALUNO VALUES

(1, 'Otávio Santos', '2002-11-04', '11995554444', 'otavio@gmail.com', 1),  
(2, 'Ana Pereira', '1999-02-20', '11991112222', 'ana@gmail.com', 2),  
(3, 'Lucas Mendes', '2001-08-15', '11990009999', 'lucas@gmail.com', 3);

## 2.4 INSERTS EXERCICIO

INSERT INTO EXERCICIO VALUES

(1, 'Supino Reto', 'Peito'),  
(2, 'Agachamento Livre', 'Pernas'),  
(3, 'Remada Curvada', 'Costas');

## 2.5 INSERTS TREINO

INSERT INTO TREINO VALUES

(1, 1, 1, 'Hipertrofia A', 'Foco em peito e tríceps'),  
(2, 2, 2, 'Funcional Básico', 'Circuito leve para condicionamento'),  
(3, 3, 1, 'Hipertrofia B', 'Foco em costas e bíceps');

## 2.6 INSERTS TREINO\_EXERCICIO

INSERT INTO TREINO\_EXERCICIO VALUES

(1, 1, 4, 12, 40),  
(1, 3, 4, 10, 35),  
(3, 2, 5, 8, 60);

## 2.7 INSERTS AULA\_COLETIVA

INSERT INTO AULA\_COLETIVA VALUES

(1, 'Yoga', 'Segunda', '08:00', 2),  
(2, 'Funcional Avançado', 'Quarta', '19:00', 2);

## 2.8 INSERTS PRESENCA\_AULA

INSERT INTO PRESENCA\_AULA VALUES

(1, 1, 1, '2025-02-01'),  
(2, 2, 2, '2025-02-02');

---

## 3. CONSULTAS SQL (SELECT)

---

### 3.1 Listar todos os alunos com seus respectivos planos

```
SELECT a.nome, p.nome AS plano
FROM ALUNO a
JOIN PLANO p ON a.id_plano = p.id_plano;
```

### 3.2 Treinos criados por cada instrutor

```
SELECT i.nome AS instrutor, t.nome_treino
FROM INSTRUTOR i
JOIN TREINO t ON i.id_instrutor = t.id_instrutor;
```

### 3.3 Exercícios usados no treino 1

```
SELECT e.nome
FROM EXERCICIO e
JOIN TREINO_EXERCICIO te ON te.id_exercicio = e.id_exercicio
WHERE te.id_treino = 1;
```

### 3.4 Aulas por dia da semana, ordenadas por horário

```
SELECT nome, dia_semana, horario
FROM AULA_COLETIVA
ORDER BY horario ASC;
```

### 3.5 Presenças registradas (JOIN aluno + aula)

```
SELECT a.nome AS aluno, ac.nome AS aula, p.data_presenca
FROM PRESENCA_AULA p
JOIN ALUNO a ON p.id_aluno = a.id_aluno
JOIN AULA_COLETIVA ac ON p.id_aula = ac.id_aula;
```

-----

## 4. UPDATE – MODIFICAÇÃO DE REGISTROS

-----

### 4.1 Alterar o telefone de um aluno

```
UPDATE ALUNO SET telefone = '11999998888' WHERE id_aluno = 1;
```

### 4.2 Atualizar carga em exercício

```
UPDATE TREINO_EXERCICIO SET carga = 50 WHERE id_treino = 1 AND id_exercicio = 1;
```

### 4.3 Alterar valor de plano

```
UPDATE PLANO SET valor_mensal = 89.90 WHERE id_plano = 1;
```

-----

## 5. DELETE – REMOÇÃO CONTROLADA

---

### 5.1 Remover presença indevida

DELETE FROM PRESENCA\_AULA WHERE id\_presenca = 1;

### 5.2 Remover exercício de um treino

DELETE FROM TREINO\_EXERCICIO WHERE id\_treino = 1 AND id\_exercicio = 3;

### 5.3 Remover avaliação física

DELETE FROM AVALIACAO\_FISICA WHERE id\_avaliacao = 1;

---

## 6. README.md – ORIENTAÇÕES DO PROJETO

---

Projeto StrongFit — EP4

Inclui:

- Scripts DDL
- Scripts DML (INSERT, UPDATE, DELETE, SELECT)
- Instruções para execução no MySQL Workbench
- Estrutura organizada em pastas