

## Atividade Prática: CRUD em SQL com as Tabelas Pai e Filho

Contexto: Você está trabalhando no banco de dados de uma árvore genealógica, onde deve gerenciar as tabelas `pai` e `filho`. A tabela `pai` contém informações sobre os pais, e a tabela `filho` armazena dados sobre os filhos.

### Parte 1: Criação e Inserção

1. Crie o Banco de Dados `familia`.

```
CREATE DATABASE IF NOT EXISTS familia;
```

2. Use o banco de dados `familia`.

```
USE familia;
```

3. Crie a Tabela `pai` com os seguintes campos:

- `id\_pai` (chave primária e autoincremento)
- `nome\_pai` (texto)
- `idade\_pai` (inteiro)
- `email\_pai` (texto)

```
CREATE TABLE IF NOT EXISTS pai (  
    id_pai INT AUTO_INCREMENT PRIMARY KEY,  
    nome_pai VARCHAR(100),  
    idade_pai INT,  
    email_pai VARCHAR(100)  
);
```

4. Crie a Tabela `filho` com os seguintes campos:

- `id\_filho` (chave primária e autoincremento)
- `nome\_filho` (texto)
- `idade\_filho` (inteiro)
- `id\_pai` (inteiro, chave estrangeira para `pai`)

```
CREATE TABLE IF NOT EXISTS filho (  
    id_filho INT AUTO_INCREMENT PRIMARY KEY,  
    nome_filho VARCHAR(100),  
    idade_filho INT,  
    id_pai INT,  
    FOREIGN KEY (id_pai) REFERENCES pai(id_pai)  
);
```

5. Insira registros nas tabelas `pai` e `filho`:

- Insira 5 registros na tabela `pai`.
- Insira 10 registros na tabela `filho`, vinculando cada `filho` a um `pai` existente.

Exemplo para a tabela `pai`:

```
INSERT INTO pai (nome_pai, idade_pai, email_pai) VALUES  
( 'Carlos', 50, 'carlos@example.com'),  
( 'João', 45, 'joao@example.com'),  
( 'José', 40, 'jose@example.com'),  
( 'Antonio', 55, 'antonio@example.com'),  
( 'Paulo', 60, 'paulo@example.com');
```

Exemplo para a tabela `filho`:

```
INSERT INTO filho (nome_filho, idade_filho, id_pai) VALUES
('Lucas', 25, 1),
('Pedro', 22, 1),
('Ana', 20, 2),
('Julia', 18, 2),
('Gabriel', 15, 3),
('Fernanda', 12, 3),
('Mateus', 10, 4),
('Laura', 8, 4),
('Marcos', 5, 5),
('Luiza', 3, 5);
```

## Parte 2: Consultas

6. Selecione todos os registros da tabela `pai`.

```
SELECT * FROM pai;
```

7. Selecione todos os registros da tabela `filho` que pertencem a um pai específico.

```
SELECT * FROM filho WHERE id_pai = 1;
```

8. Liste todos os filhos com idade superior a 10 anos.

```
SELECT nome_filho, idade_filho FROM filho WHERE idade_filho > 10;
```

9. Conte quantos filhos existem para cada pai.

```
SELECT id_pai, COUNT(*) AS total_filhos FROM filho GROUP BY id_pai;
```

10. Liste todos os pais que têm filhos com menos de 5 anos.

```
SELECT pai.nome_pai  
FROM pai  
JOIN filho ON pai.id_pai = filho.id_pai  
WHERE filho.idade_filho < 5;
```

### Parte 3: Atualização

11. Atualize a idade de um pai específico.

```
UPDATE pai SET idade_pai = 52 WHERE nome_pai = 'Carlos';
```

12. Atualize o nome de um filho.

```
UPDATE filho SET nome_filho = 'Lucas Silva' WHERE nome_filho = 'Lucas';
```

### Parte 4: Deleção

13. Remova um filho da tabela `filho`.

```
DELETE FROM filho WHERE nome_filho = 'Pedro';
```

14. Remova um pai e todos os seus filhos.

```
DELETE FROM filho WHERE id_pai = 1;  
DELETE FROM pai WHERE id_pai = 1;
```

## Parte 5: Questões Avançadas

15. Liste todos os pais e a média de idade de seus filhos.

```
SELECT pai.nome_pai, AVG(filho.idade_filho) AS media_idade_filhos  
FROM pai  
JOIN filho ON pai.id_pai = filho.id_pai  
GROUP BY pai.nome_pai;
```