

## Tema do Banco de Dados:

Para tornar as atividades mais interessantes e contextualizadas, vamos usar o tema de uma "**Escola de Música**".

### Estrutura básica:

- **Alunos:** Matrícula (PK), Nome, Telefone, Endereço, Data de Nascimento, Instrumento Principal.
- **Professores:** ID\_Professor (PK), Nome, Especialidade, Telefone.
- **Cursos:** ID\_Curso (PK), Nome do Curso, Descrição, ID\_Professor (FK).
- **Matrículas:** ID\_Matricula (PK), ID\_Aluno (FK), ID\_Curso (FK), Data da Matrícula.

### Atividades:

#### Nível Iniciante (10 atividades):

##### 1. Criar o banco de dados:

```
CREATE DATABASE EscolaDeMusica;
```

##### 2. Usar o banco de dados:

```
USE EscolaDeMusica;
```

##### 3. Criar a tabela Alunos com chave primária:

```
CREATE TABLE Alunos (  
    Matricula INT PRIMARY KEY,  
    Nome VARCHAR(255) NOT NULL,  
    Telefone VARCHAR(15),  
    Endereco VARCHAR(255),  
    DataNascimento DATE,  
    InstrumentoPrincipal VARCHAR(50)  
);
```

##### 4. Criar a tabela Professores com chave primária:

```
CREATE TABLE Professores (  
    ID_Professor INT PRIMARY KEY,  
    Nome VARCHAR(255) NOT NULL,  
    Especialidade VARCHAR(100),  
    Telefone VARCHAR(15)  
);
```

**5. Inserir 5 registros na tabela Alunos:**

-- Aluno 1

```
INSERT INTO Alunos (Matricula, Nome, Telefone, Endereco, DataNascimento, InstrumentoPrincipal)
VALUES (1, 'João Silva', '(11) 98765-4321', 'Rua A, 123', '2005-08-15', 'Violão');
```

-- Aluno 2

```
INSERT INTO Alunos (Matricula, Nome, Telefone, Endereco, DataNascimento, InstrumentoPrincipal)
VALUES (2, 'Maria Oliveira', '(11) 97654-3210', 'Rua B, 456', '2002-03-28', 'Piano');
```

-- Aluno 3

```
INSERT INTO Alunos (Matricula, Nome, Telefone, Endereco, DataNascimento, InstrumentoPrincipal)
VALUES (3, 'Pedro Santos', '(11) 96543-2109', 'Rua C, 789', '2004-11-02', 'Bateria');
```

-- Aluno 4

```
INSERT INTO Alunos (Matricula, Nome, Telefone, Endereco, DataNascimento, InstrumentoPrincipal)
VALUES (4, 'Ana Pereira', '(11) 95432-1098', 'Rua D, 1011', '2006-06-20', 'Violino');
```

-- Aluno 5

```
INSERT INTO Alunos (Matricula, Nome, Telefone, Endereco, DataNascimento, InstrumentoPrincipal)
VALUES (5, 'Lucas Souza', '(11) 94321-0987', 'Rua E, 1213', '2003-09-05', 'Violão');
```

**6. Inserir 3 registros na tabela Professores:**

-- Professor 1

```
INSERT INTO Professores (ID_Professor, Nome, Especialidade, Telefone)
VALUES (1, 'Carlos Martins', 'Violão', '(11) 91111-2222');
```

-- Professor 2

```
INSERT INTO Professores (ID_Professor, Nome, Especialidade, Telefone)
VALUES (2, 'Laura Almeida', 'Piano', '(11) 92222-3333');
```

-- Professor 3

```
INSERT INTO Professores (ID_Professor, Nome, Especialidade, Telefone)
VALUES (3, 'Ricardo Pereira', 'Bateria', '(11) 93333-4444');
```

**7. Listar todos os alunos:**

```
SELECT * FROM Alunos;
```

**8. Listar o nome e o telefone de todos os professores:**

```
SELECT Nome, Telefone FROM Professores;
```

**9. Listar os alunos que tocam violão como instrumento principal:**

```
SELECT * FROM Alunos WHERE InstrumentoPrincipal = 'Violão';
```

**10. Listar os professores com especialidade em piano:**

```
SELECT * FROM Professores WHERE Especialidade = 'Piano';
```

**Nível Intermediário (10 atividades):**

**1. Criar a tabela Cursos com chave primária e chave estrangeira para Professores:**

```
CREATE TABLE Cursos (  
    ID_Curso INT PRIMARY KEY,  
    NomeCurso VARCHAR(255) NOT NULL,  
    Descricao VARCHAR(255),  
    ID_Professor INT,  
    FOREIGN KEY (ID_Professor) REFERENCES Professores(ID_Professor)  
);
```

**2. Inserir 4 registros na tabela Cursos, associando cada curso a um professor existente:**

*-- Curso 1*

```
INSERT INTO Cursos (ID_Curso, NomeCurso, Descricao, ID_Professor)  
VALUES (1, 'Violão Iniciante', 'Curso para iniciantes em violão', 1);
```

*-- Curso 2*

```
INSERT INTO Cursos (ID_Curso, NomeCurso, Descricao, ID_Professor)  
VALUES (2, 'Piano Clássico', 'Curso de piano clássico', 2);
```

*-- Curso 3*

```
INSERT INTO Cursos (ID_Curso, NomeCurso, Descricao, ID_Professor)  
VALUES (3, 'Bateria Rock', 'Curso de bateria com foco em rock', 3);
```

-- Curso 4

INSERT INTO Cursos (ID\_Curso, NomeCurso, Descricao, ID\_Professor)

VALUES (4, 'Violão Avançado', 'Curso avançado de violão', 1);

**3. Criar a tabela Matriculas com chave primária e chaves estrangeiras para Alunos e Cursos:**

```
CREATE TABLE Matriculas (  
    ID_Matricula INT PRIMARY KEY,  
    ID_Aluno INT,  
    ID_Curso INT,  
    DataMatricula DATE,  
    FOREIGN KEY (ID_Aluno) REFERENCES Alunos(Matricula),  
    FOREIGN KEY (ID_Curso) REFERENCES Cursos(ID_Curso)  
);
```

**4. Inserir 5 registros na tabela Matriculas, associando alunos a cursos:**

-- Matrícula 1

INSERT INTO Matriculas (ID\_Matricula, ID\_Aluno, ID\_Curso, DataMatricula)

VALUES (1, 1, 1, '2023-03-01');

-- Matrícula 2

INSERT INTO Matriculas (ID\_Matricula, ID\_Aluno, ID\_Curso, DataMatricula)

VALUES (2, 2, 2, '2023-02-15');

-- Matrícula 3

INSERT INTO Matriculas (ID\_Matricula, ID\_Aluno, ID\_Curso, DataMatricula)

VALUES (3, 3, 3, '2023-04-10');

-- Matrícula 4

INSERT INTO Matriculas (ID\_Matricula, ID\_Aluno, ID\_Curso, DataMatricula)

VALUES (4, 4, 1, '2023-03-05');

-- Matrícula 5

INSERT INTO Matriculas (ID\_Matricula, ID\_Aluno, ID\_Curso, DataMatricula)

VALUES (5, 5, 4, '2023-04-20');

**5. Listar o nome do aluno, o nome do curso e a data da matrícula, utilizando JOIN:**

```
SELECT a.Nome AS NomeAluno, c.NomeCurso, m.DataMatricula  
FROM Alunos a  
JOIN Matriculas m ON a.Matricula = m.ID_Aluno  
JOIN Cursos c ON m.ID_Curso = c.ID_Curso;
```

**6. Listar os cursos ministrados pelo professor 'Nome do Professor' (substitua pelo nome de um professor da sua tabela):**

```
SELECT c.NomeCurso  
FROM Cursos c  
JOIN Professores p ON c.ID_Professor = p.ID_Professor  
WHERE p.Nome = 'Nome do Professor';
```

**7. Atualizar o telefone de um aluno específico:**

```
UPDATE Alunos SET Telefone = 'Novo Telefone' WHERE Matricula = 'Número da Matrícula';
```

**8. Adicionar uma coluna Email na tabela Professores:**

```
ALTER TABLE Professores  
ADD Email VARCHAR(255);
```

**9. Atualizar os emails dos professores:**

```
UPDATE Professores SET Email = 'carlos.martins@email.com' WHERE ID_Professor = 1;  
UPDATE Professores SET Email = 'laura.almeida@email.com' WHERE ID_Professor = 2;  
UPDATE Professores SET Email = 'ricardo.pereira@email.com' WHERE ID_Professor = 3;
```

**10. Apagar um curso específico da tabela Cursos:**

```
DELETE FROM Cursos WHERE ID_Curso = 'Número do Curso';
```

### Nível Avançado (10 atividades):

1. **Criar um índice na coluna Nome da tabela Alunos para otimizar buscas por nome:**

```
CREATE NONCLUSTERED INDEX IX_Alunos_Nome ON Alunos(Nome);
```

2. **Criar uma stored procedure para inserir um novo aluno:**

```
CREATE PROCEDURE sp_InserirAluno
    @Matricula INT,
    @Nome VARCHAR(255),
    @Telefone VARCHAR(15),
    @Endereco VARCHAR(255),
    @DataNascimento DATE,
    @InstrumentoPrincipal VARCHAR(50)
AS
BEGIN
    INSERT INTO Alunos (Matricula, Nome, Telefone, Endereco, DataNascimento,
InstrumentoPrincipal)
    VALUES (@Matricula, @Nome, @Telefone, @Endereco, @DataNascimento,
@InstrumentoPrincipal);
END;
```

3. **Utilizar a stored procedure sp\_InserirAluno para inserir um novo aluno:**

```
EXEC sp_InserirAluno 10, 'Novo Aluno', '(11) 99999-9999', 'Rua Teste, 123', '2003-05-10', 'Piano';
```

4. **Criar uma trigger que, ao inserir uma nova matrícula, atualiza a coluna TotalAlunos na tabela Cursos:**  
(Assumindo que você adicionou a coluna TotalAlunos na tabela Cursos)

```
CREATE TRIGGER trg_AtualizarTotalAlunos
ON Matriculas
AFTER INSERT
AS
BEGIN
    UPDATE Cursos
    SET TotalAlunos = TotalAlunos + 1
    WHERE ID_Curso IN (SELECT ID_Curso FROM inserted);
END;
```

5. **Criar uma view que mostra o nome do aluno, o nome do professor, o nome do curso e a data da matrícula:**

```
CREATE VIEW vw_InformacoesMatriculas AS

SELECT a.Nome AS NomeAluno, p.Nome AS NomeProfessor, c.NomeCurso, m.DataMatricula

FROM Alunos a

JOIN Matriculas m ON a.Matricula = m.ID_Aluno

JOIN Cursos c ON m.ID_Curso = c.ID_Curso

JOIN Professores p ON c.ID_Professor = p.ID_Professor;
```

6. **Utilizar a view vw\_InformacoesMatriculas para listar todas as informações das matrículas:**

```
SELECT * FROM vw_InformacoesMatriculas;
```

7. **Criar uma função que calcula a idade de um aluno com base na data de nascimento:**

```
CREATE FUNCTION fn_CalcularIdade (@DataNascimento DATE)

RETURNS INT

AS

BEGIN

RETURN DATEDIFF(YEAR, @DataNascimento, GETDATE());

END;
```

8. **Listar os alunos maiores de 18 anos usando a função fn\_CalcularIdade:**

```
SELECT *

FROM Alunos

WHERE dbo.fn_CalcularIdade(DataNascimento) > 18;
```

9. **Criar uma stored procedure que recebe o nome de um curso e aumenta o preço de todas as matrículas nesse curso em 10%:**

```
CREATE PROCEDURE sp_AumentarPrecoMatricula

@NomeCurso VARCHAR(255)

AS

BEGIN

UPDATE Matriculas

SET Valor = Valor * 1.10 -- Aumenta o valor em 10%

WHERE ID_Curso = (SELECT ID_Curso FROM Cursos WHERE NomeCurso = @NomeCurso);

END;
```

*(Assumindo que você adicionou a coluna Valor na tabela Matriculas)*

10. Utilizar a stored procedure **sp\_AumentarPrecoMatricula** para aumentar o preço das matrículas no curso de 'Violão':

```
EXEC sp_AumentarPrecoMatricula 'Violão';
```