



## Avaliação 5

**Feito em Equipe: Jaime Nicolas, Pedro Farley, Matheus Carvalho e Gerdson Costa.**

Objetivo: Construa um banco de dados usando a linha de comandos do SQLite3.  
Crie tabelas com seus atributos. Chave primária do tipo int com auto incremento, Atributos com campo texto não nulo. As tabelas deveram formar relacionamentos 1:1, 1:N e N:N.  
Inserir dados nas tabelas observando os relacionamentos.

### Tabela 1: Usuários (Relacionamento 1:1):

```
sqlite> CREATE TABLE Usuarios (  
...>     id INTEGER PRIMARY KEY AUTOINCREMENT,  
...>     nome TEXT NOT NULL  
...> );
```

### Tabela 2: Endereços (Relacionamento 1:1):

```
sqlite> CREATE TABLE Enderecos (  
...>     id INTEGER PRIMARY KEY AUTOINCREMENT,  
...>     endereco TEXT NOT NULL,  
...>     usuario_id INTEGER UNIQUE,  
...>     FOREIGN KEY (usuario_id) REFERENCES Usuarios(id)  
...> );
```

### Tabela 3: Departamentos (Relacionamento 1:N):

```
sqlite> CREATE TABLE Departamentos (  
...>     id INTEGER PRIMARY KEY AUTOINCREMENT,  
...>     nome TEXT NOT NULL  
...> );
```

### Tabela 4: Funcionários (Relacionamento 1:N):

```
sqlite> CREATE TABLE Funcionarios (  
...>     id INTEGER PRIMARY KEY AUTOINCREMENT,  
...>     nome TEXT NOT NULL,  
...>     departamento_id INTEGER,  
...>     FOREIGN KEY (departamento_id) REFERENCES Departamentos(id)  
...> );
```

### Tabela 5: Cursos (Relacionamento N:N):

```
sqlite> CREATE TABLE Cursos (  
...>     id INTEGER PRIMARY KEY AUTOINCREMENT,  
...>     nome TEXT NOT NULL  
...> );
```

### Tabela de junção para relacionamento N:N entre Funcionários e Cursos:

```
sqlite> CREATE TABLE FuncionariosCursos (  
...>     funcionario_id INTEGER,  
...>     curso_id INTEGER,  
...>     FOREIGN KEY (funcionario_id) REFERENCES Funcionarios(id),  
...>     FOREIGN KEY (curso_id) REFERENCES Cursos(id),  
...>     PRIMARY KEY (funcionario_id, curso_id)  
...> );
```

### Inserção de Dados:

#### Inserindo dados na tabela Usuarios:

```
INSERT INTO Enderecos (endereco, usuario_id) VALUES ('123 Main St', 1);  
INSERT INTO Enderecos (endereco, usuario_id) VALUES ('456 Elm St', 2);
```

#### Inserindo dados na tabela Enderecos (1:1 com Usuarios):

```
INSERT INTO Enderecos (endereco, usuario_id) VALUES ('123 Main St', 1);  
INSERT INTO Enderecos (endereco, usuario_id) VALUES ('456 Elm St', 2);
```

#### Inserindo dados na tabela Departamentos:

```
INSERT INTO Departamentos (nome) VALUES ('RH');  
INSERT INTO Departamentos (nome) VALUES ('TI');
```

#### Inserindo dados na tabela Funcionarios (1:N com Departamentos):

```
INSERT INTO Funcionarios (nome, departamento_id) VALUES ('Carol', 1);  
INSERT INTO Funcionarios (nome, departamento_id) VALUES ('David', 2);
```

#### Inserindo dados na tabela Cursos:

```
INSERT INTO Cursos (nome) VALUES ('Curso A');  
INSERT INTO Cursos (nome) VALUES ('Curso B');
```

#### Inserindo dados na tabela FuncionariosCursos (N:N entre Funcionarios e Cursos):

```
INSERT INTO FuncionariosCursos (funcionario_id, curso_id) VALUES (1, 1);  
INSERT INTO FuncionariosCursos (funcionario_id, curso_id) VALUES (1, 2);  
INSERT INTO FuncionariosCursos (funcionario_id, curso_id) VALUES (2, 2);
```

### Tabelas:

```
sqlite> .tables
```

Cursos

Enderecos

FuncionariosCursos

Departamentos

Funcionarios

Usuarios