

Bases de Dados – Prof. Eduardo Corrêa

Aula 03 - Criação de Esquemas na Linguagem SQL

Esta aula prática tem dois objetivos: (i) apresentar a ferramenta SQLite Studio; (ii) demonstrar a forma básica para a criação de esquemas em SQL.

EXECUÇÃO DA PRÁTICA

PASSO 1: Executar o SQLite Studio

Vá até a pasta onde está armazenado o SQLite Studio e dê dois-cliques no programa “SQLiteStudio.exe” (a extensão “.exe” pode não estar aparecendo – observe então o desenho do ícone destacado na Figura 1). Será exibida uma tela similar à apresentada na Figura 2.






	sqlite.dll	20/07/2016 00:43	Extensão de aplica...	403 KB
	sqlite3.dll	20/07/2016 00:43	Extensão de aplica...	808 KB
	SQLiteStudio.exe	20/07/2016 00:43	Aplicativo	338 KB
	sqlitestudiocli.exe	20/07/2016 00:43	Aplicativo	314 KB
	tcl86.dll	20/07/2016 00:43	Extensão de aplica...	1.144 KB

Figura 1. Executável do SQLiteStudio

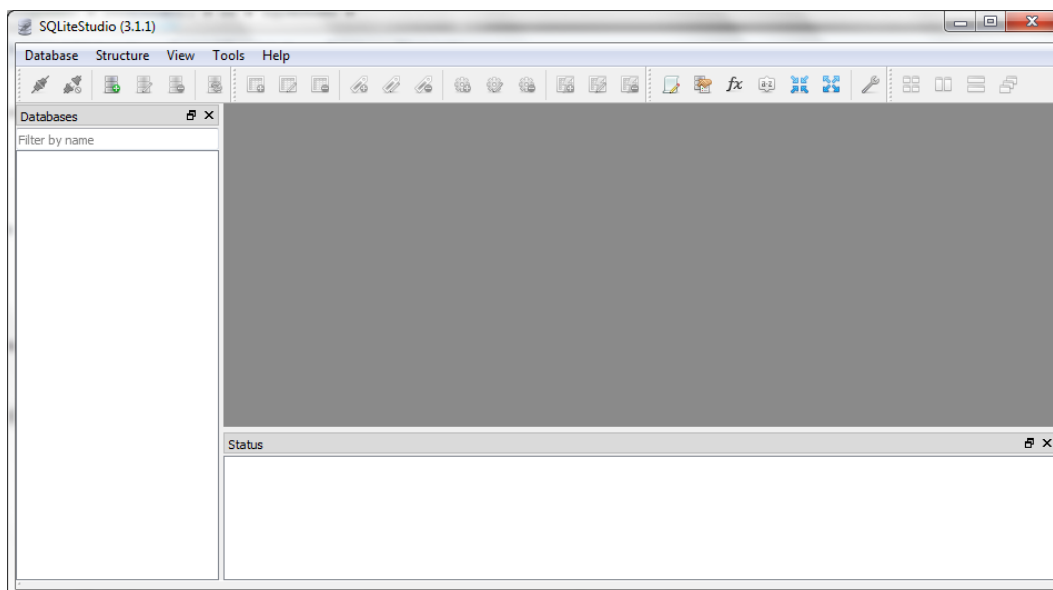


Figura 2. Tela inicial do SQLite Studio

PASSO 2: Criar um Novo Banco de Dados

Acesse a opção de menu **Database / Add a Database**, como mostra a Figura 3 ou clique no terceiro ícone que aparece na barra de ferramentas. Na nova janela que será aberta, clique no botão com o sinal de + (Figura 4). Então selecione a pasta onde deseja criar o banco e digite “Exemplo1.db”. Este será o nome do BD SQLite que será criado e que, futuramente, irá armazenar as tabelas de todos os exemplos apresentados neste capítulo. Após digitar o nome, clique em “Salvar” (Figura 5). Obs.: você pode escolher uma pasta qualquer, tanto em seu computador como em um pen drive, para armazenar o BD “Exemplo1.db”.

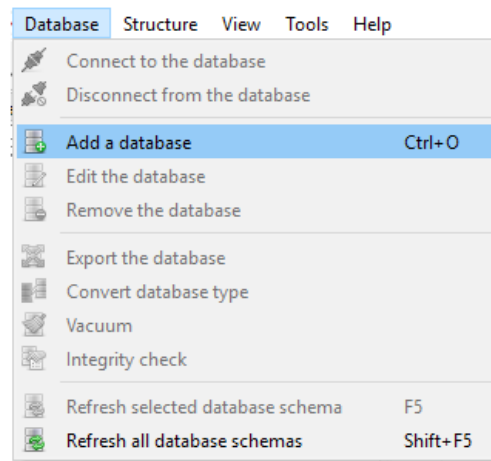


Figura 3. Criando um novo banco de dados no SQLite Studio (parte 1)

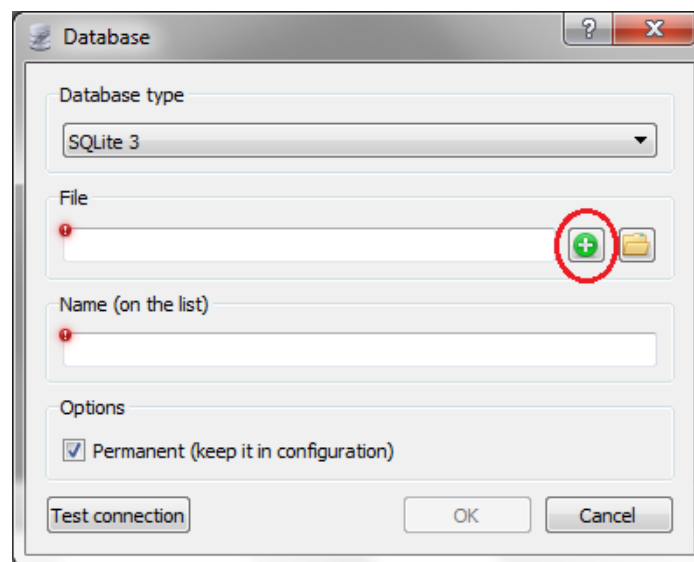


Figura 4. Criando um novo banco de dados no SQLite Studio (parte 2)

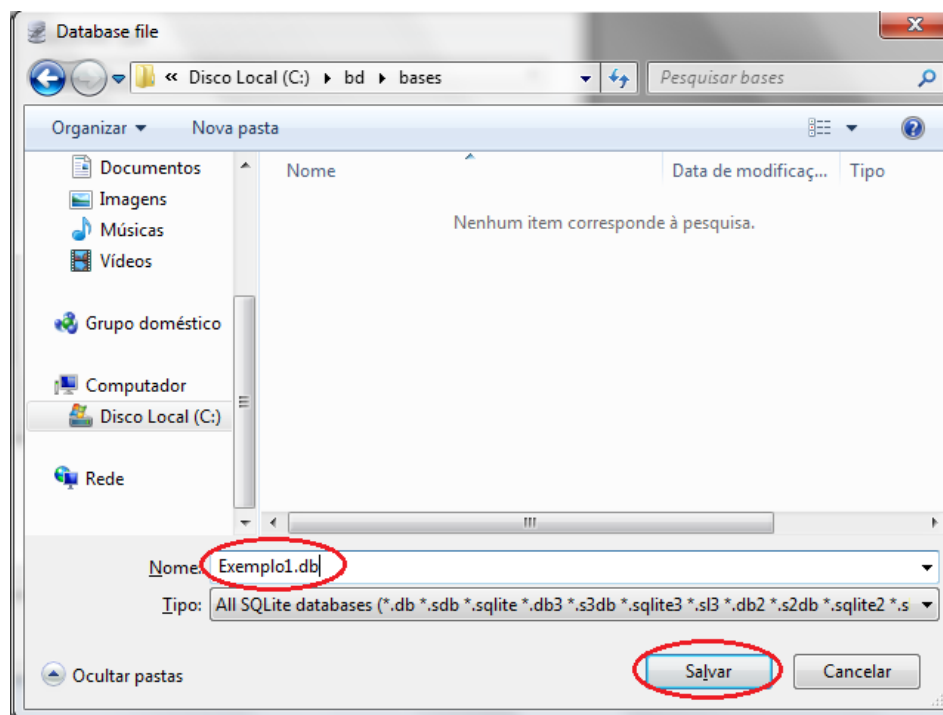


Figura 5. Criando um novo banco de dados no SQLite Studio (parte 3)

A seguir, a tela da Figura 6 será exibida. Mantenha a opção “Permanente (keep it in configuration)” marcada e clique então no botão OK. Pronto, o BD “Exemplo1” está criado!

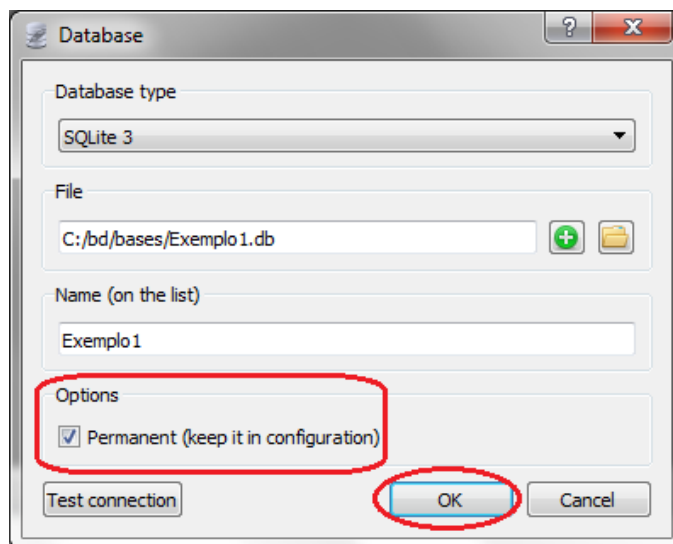


Figura 6. Criando um novo banco de dados no SQLite Studio (parte 4)

PASSO 3: Efetuar a conexão com o BD criado

Para poder atuar como projetista de BD (ou seja, criar tabelas e outros objetos), você precisará agora realizar a conexão com o banco “Exemplo1.db”. Ao contrário dos

SGBDs normais, o SQLite não exige nenhuma informação de login e senha para efetuar a conexão. Isso ocorre porque, conforme já mencionado em nossas aulas, o SQLite não é um SGBD real, mas sim uma biblioteca que emula um SGBD.

Para efetuar a conexão, selecione o BD “Exemplo1” e clique no ícone de conexão (primeiro ícone da barra de tarefas – destacado na Figura 7). Também é possível fazer a conexão acessando o menu **Database / Connect to the Database**.

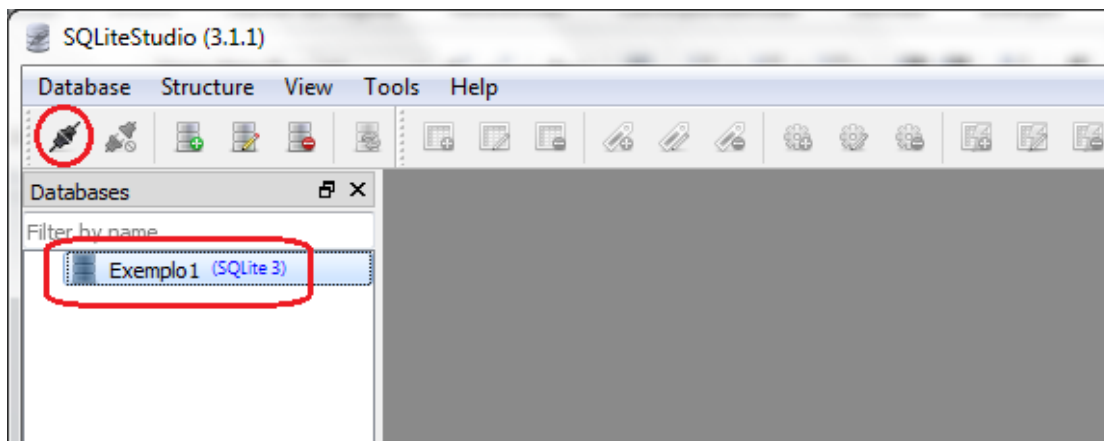


Figura 7. Efetuando a conexão com o banco de dados “Exemplo1”

PASSO 4: Criando uma Tabela

Nosso BD “Exemplo1” ainda está completamente vazio. Que tal criar uma tabela “dentro” dele usando a linguagem SQL? Para isto, clique no botão **Open SQL Editor**, destacado na Figura 8. Ele abrirá o editor de comandos SQL, que, como o próprio nome diz, possibilita a digitação e execução de instruções SQL. O SQL Editor também pode ser aberto através do menu **Tools / Open SQL editor**.

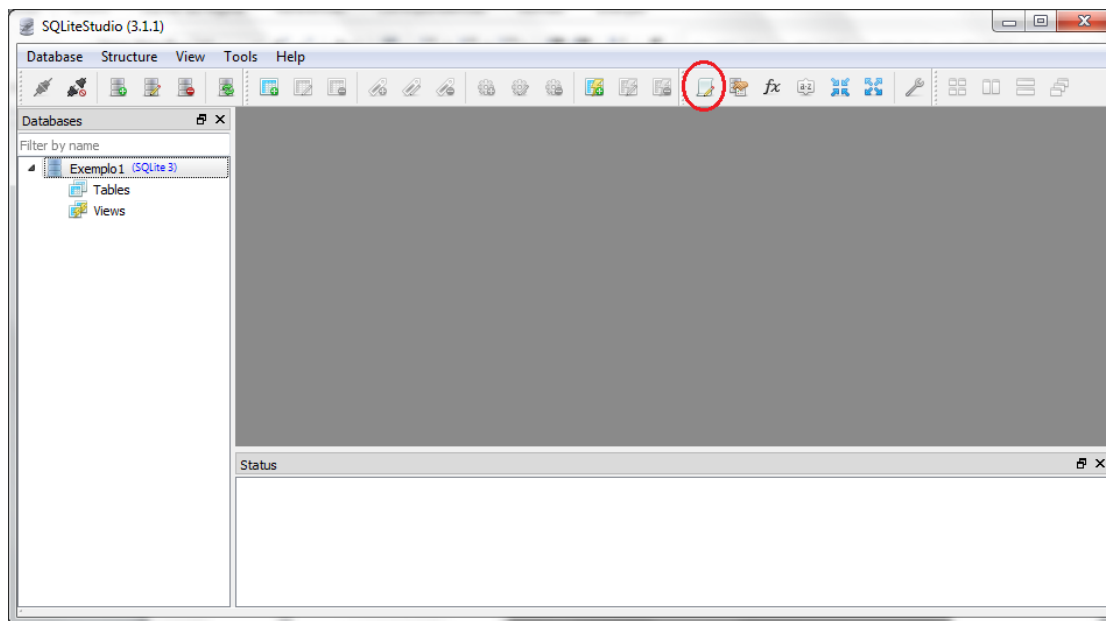


Figura 8. Abrindo o SQL Editor

Na área de edição (aba **Query**), você digitará e executará a sua primeira instrução SQL. Esta instrução será responsável por definir o esquema abaixo dentro do BD “Exemplo1”:

Filme(titulo: alfanumérico,
ano: inteiro,
resumo: alfanumérico,
pais: alfanumérico,
duracao: inteiro,
avaliacao: real)

CREATE TABLE é a instrução utilizada na SQL para a criação de tabelas (ou relações, ambas as palavras são sinônimos no modelo relacional). Em sua forma básica, a instrução é utilizada da seguinte maneira:

- Começa com a expressão CREATE TABLE seguida pelo nome da tabela a ser criada;
- Segue com uma lista de atributos e seus respectivos tipos. Esta especificação deve ser feita entre parênteses, com cada definição de atributo separada por vírgula; No SQLite os tipos básicos são INT (inteiro), NUM (real) e TEXT (alfanumérico).
 - Os atributos que não poderão aceitar valores nulos devem ser especificados com o uso da expressão NOT NULL após a indicação de seu tipo.
 - Os atributos que compõem a chave primária são indicados com o uso da expressão PRIMARY KEY, no final da especificação do comando.

Para criar a tabela *Filme* no SQLite, a seguinte declaração pode ser utilizada:

```
CREATE TABLE Filme (  
    titulo VARCHAR(100),  
    ano INT,  
    resumo VARCHAR(2000),  
    pais CHAR(2),  
    duracao INT,  
    avaliacao NUM,  
    PRIMARY KEY (titulo, ano)  
);
```

Veja que, neste comando, a tabela *Filme* é criada com seis atributos. Dois do tipo inteiro (“ano” e “duracao”); um do tipo real (“avaliacao”) e três atributos alfanuméricos (“titulo”, “resumo” e “pais”). Veja que “titulo” e “ano”, em conjunto, foram escolhidos para formar a PK.

Para criar a tabela através do SQLite Studio, digite o script **CREATE TABLE** no editor SQL e depois clique no botão **Execute query** (destacado na Figura 9) ou aperte a tecla **F9**.

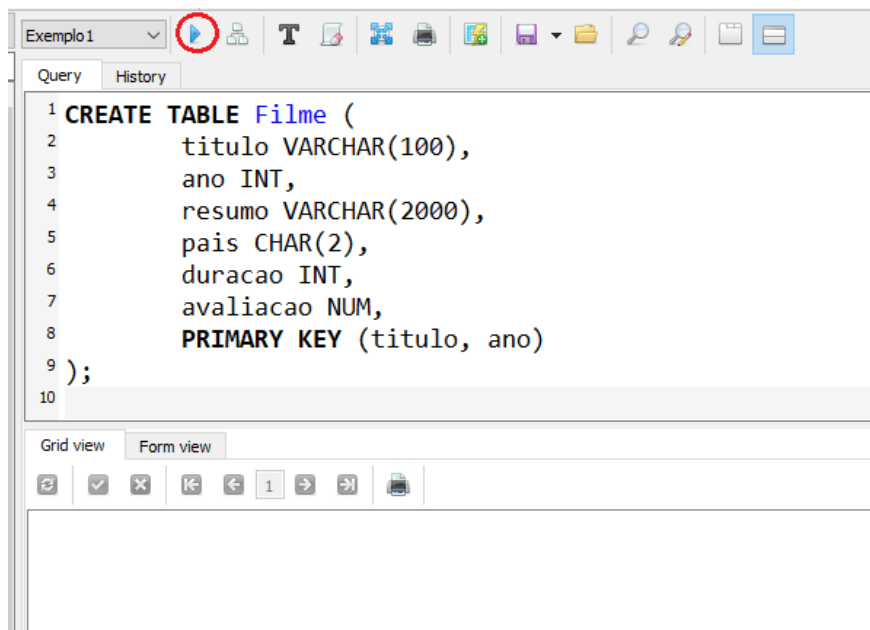


Figura 9. Criando a Tabela *Filme*

A janela status (parte inferior da tela) mostrará se cada operação foi ou não realizada com sucesso. Se tudo tiver corrido bem, uma mensagem similar à apresentada na Figura 10 será apresentada.

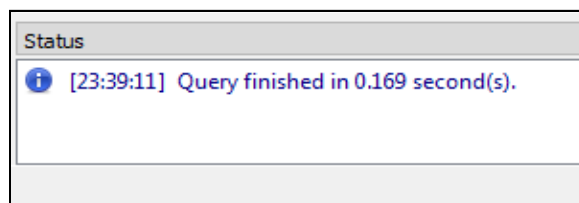


Figura 10. Janela de status apresentando mensagem de sucesso

ATENÇÃO

As instruções SQL não fazem distinção entre letras maiúsculas e minúsculas (ou seja, tanto faz usar “CREATE TABLE” ou “create table”). Muitas vezes, as instruções são definidas em várias linhas com a única finalidade de melhorar a legibilidade, mas isto também não é requerido pela linguagem. O ponto-e-vírgula é utilizado para indicar o fim da instrução.

PASSO 5: Criando as Demais Tabelas

```
CREATE TABLE Pais (  
    sigla CHAR(2),  
    nome VARCHAR(60),  
    PRIMARY KEY (sigla)  
);
```

```
CREATE TABLE Artista (  
    nome VARCHAR(60),  
    sexo CHAR(1),  
    PRIMARY KEY (nome)  
);
```

```
CREATE TABLE Genero (  
    nome VARCHAR(15),  
    PRIMARY KEY (nome)  
);
```

```
CREATE TABLE FilmeGenero (  
    titulofilme VARCHAR(100),  
    ano INT,  
    nomegenero VARCHAR(15),  
    PRIMARY KEY (titulofilme, ano, nomegenero)  
);
```

```
CREATE TABLE FilmeElenco (  
    titulofilme VARCHAR(100),  
    ano INT,  
    nomeartista VARCHAR(60),  
    PRIMARY KEY (titulofilme, ano, nomeartista)  
);
```

No SQLite Studio, você pode criar estas tabelas de diferentes formas:

- Abrindo várias janelas de edição SQL, uma para cada script.
- Digitando e executando um comando CREATE TABLE de cada vez na janela de edição SQL (após digitar clique no botão **Excute Query** ou digite **F9**).

- Digitar o script de todas as tabelas de uma vez na janela de edição SQL, então selecioná-los com o mouse (você pode selecionar todos ou um de cada vez) clicando no botão **Excute Query** ou digitando **F9**.

Com o comando INSERT você pode inserir dados em qualquer tabela. E com o comando SELECT pode consultar os dados de uma tabela. Esses comandos serão apresentados em outras aulas, mas abaixo apresentamos um exemplo envolvendo a tabela país:

```
--Insere o Brasil na tabela de países
INSERT INTO Pais VALUES('BR', 'Brasil');

--mostra o conteúdo da tabela de países
SELECT * FROM Pais;
```

EXERCÍCIO:

A página a seguir apresenta o questionário de uma pesquisa sobre o uso de linguagens de programação entre cientistas de dados e desenvolvedores de apps.

Utilizando o SQLite Studio, crie um novo BD chamado “pesq_linguagens.db”. Após criá-lo, realize a conexão com o mesmo e utilize o comando CREATE TABLE para definir o esquema de **uma única tabela** capaz de armazenar os dados dessa pesquisa.

Nº do Questionário: _____ a) Ano de Nascimento: _____ b) País: _____ c) Salário: _____ d) Qual a sua profissão? () Cientista de Dados () Desenvolvedor de Apps	e). Qual a sua escolaridade? 1. Nível médio 2. Graduação (em andamento) 3. Graduação (completa) 4. Mestrado 5. Doutorado f) Quais as 3 linguagens de programação que você mais utiliza? 1. _____ 2. _____ 3. _____
---	---

Se você quiser avançar, tente inserir alguns registros em sua tabela com o uso do comando INSERT. Se tudo der certo, você vai poder visualizar estes registros com o uso do comando SELECT.