

5.1 - O que é um banco de dados: diferença de bancos relacionais e não relacionais

VIDEO - 11 MIN

5.2 - Entidades e Relacionamentos

VIDEO - 14 MIN

Exercício de Revisão | Entidades e relacionamentos

QUIZ - 1 PERGUNTA

5.3 - Conhecendo SQL (parte 1)

VIDEO - 13 MIN

5.3 - Conhecendo SQL (parte 2)

VIDEO - 6 MIN

Exercício de Código | Conhecendo SQL

TEXT

5.4 - Fazendo manipulações: DML

VIDEO - 19 MIN

Exercício de Código | Fazendo manipulação DML

TEXT

Exercício de Revisão | Fazendo manipulação: DML

QUIZ - 1 PERGUNTA

5.5 - Fazendo consultas em SQL

VIDEO - 17 MIN

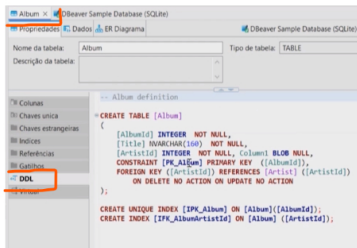
### Exercício de Código

Vamos começar a explorar o DBeaver? Então chegou a sua vez de seguir o que a instrutora Carol fez na aula, agora na prática, e criar as suas tabelas!

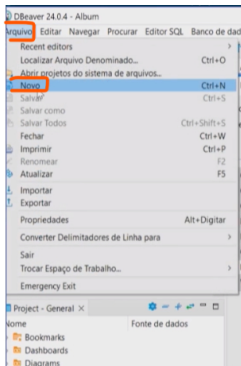
E se precisar, pode acompanhar o gabarito ✓

Parte 1:

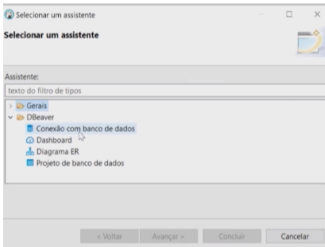
Em **álbum** > **DDL**:



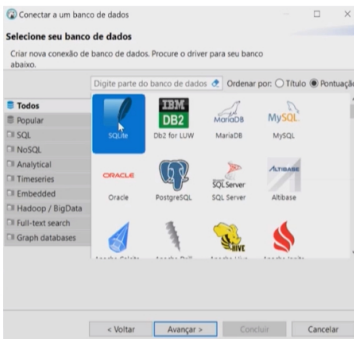
Clique em **arquivo** (no canto superior esquerdo) > **novos**:



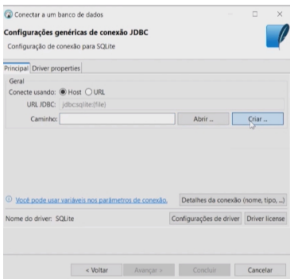
Na janela que abriu, clique em **Gerais** > **Conexão com banco de dados** > **Avançar**:



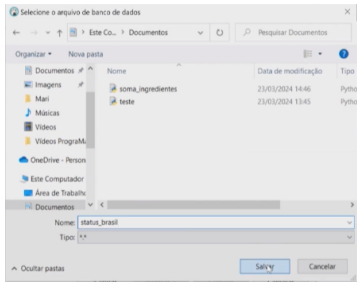
Selecione a opção **SQLite** > **Avançar**



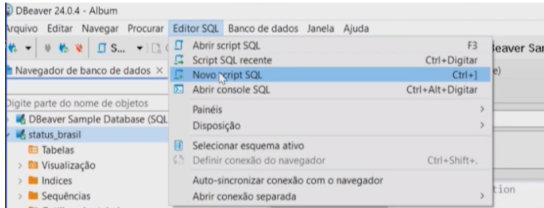
Clique em **criar** e selecione uma pasta que preferir:



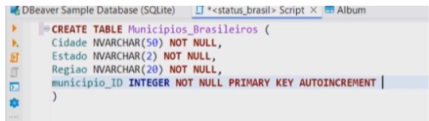
Optamos por nomear como **status\_brasil > salvar > concluir:**



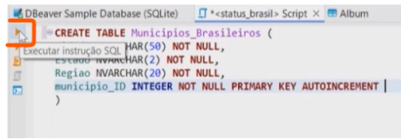
Clique em **Editor SLQ > Novo script SQL:**



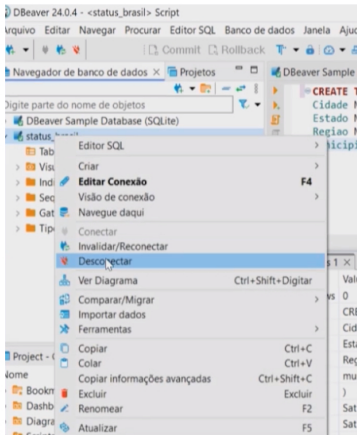
Faça a estrutura:



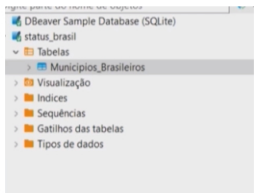
Crie a tabela clicando na opção **Executar instrução SQL** do lado esquerdo do script:



Conecte novamente o banco de dados clicando em **status\_brasil** (com o botão direito do mouse) > **Desconectar > Sim** ou ir direto em **Atualizar** ao final das opções.



Em seguida acompanhe a tabela criada:



Parte 2:

Crie a estrutura da tabela de **Municipio\_Status:**

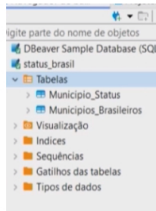


```
CREATE TABLE Municipio_Status (
    status_ID INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    populacao_residente INTEGER NOT NULL,
    IDHPI_rank INTEGER NOT NULL,
    educacao INTEGER NOT NULL,
    renda INTEGER NOT NULL,
    municipio_ID INTEGER NOT NULL,
    CONSTRAINT fk_municipio FOREIGN KEY (municipio_ID) REFERENCES Municipios_Brasileiros(municipio_ID)
);
```

Execute o script:

```
EXECUTAR INSTRUÇÃO SQL
--CREATE TABLE Municipio_Status (
--    status_ID INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
--    populacao_residente INTEGER NOT NULL,
--    IDHPI_rank INTEGER NOT NULL,
--    educacao INTEGER NOT NULL,
--    renda INTEGER NOT NULL,
--    municipio_ID INTEGER NOT NULL,
--    CONSTRAINT fk_municipio FOREIGN KEY (municipio_ID) REFERENCES Municipios_Brasileiros(municipio_ID)
--);
```

Conecte novamente o banco de dados como feito anteriormente e acompanhe a tabela:



Crie a estrutura da tabela de Gerencia\_Regiao:

```
CREATE TABLE Gerencia_Regiao (
    gerencia_ID INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    Regiao TEXT(10) NOT NULL,
    pessoas_brancas INTEGER NOT NULL,
    pessoas_pretas_pardas INTEGER NOT NULL,
    gerencia_branca INTEGER NOT NULL,
    gerencia_preta_parda INTEGER NOT NULL,
    CONSTRAINT fk_regiao FOREIGN KEY (Regiao) REFERENCES Municipios_Brasileiros (Regiao)
);
```

Execute o script:

```
EXECUTAR INSTRUÇÃO SQL
--CREATE TABLE Gerencia_Regiao (
--    gerencia_ID INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
--    Regiao TEXT(10) NOT NULL,
--    pessoas_brancas INTEGER NOT NULL,
--    pessoas_pretas_pardas INTEGER NOT NULL,
--    gerencia_branca INTEGER NOT NULL,
--    gerencia_preta_parda INTEGER NOT NULL,
--    CONSTRAINT fk_regiao FOREIGN KEY (Regiao) REFERENCES Municipios_Brasileiros (Regiao)
--);
```

Conecte novamente o banco de dados como feito anteriormente e acompanhe a tabela:

