

TEXT

 TEXT

## VIDEO • 8 MIN

## VIDEO • 9 MIN

## VIDEO • 5 MIN

## VIDEO • 11 MIN

 TEXT

 TEXT

 QUIZ · 2 PERGUNTAS · ESSE É PRÉ REQUISITO

Módulo 6 - Visualização de 0/15 ▾

## 4 COMENTÁRIOS/DÚVIDAS

Vamos colocar em prática nossos aprendizados sobre banco de dados, assim como a instrutora fez na aula anterior 😊

### Parte 1:

The screenshot shows the Google Colab web interface. At the top, there are tabs for '+ Código' and '+ Texto', and a status bar indicating 'RAM' and 'Disco'. The main heading of the notebook is 'Conectando SQL com pandas'. Below the heading, there is a code cell with the text '1 Comece a programar ou gere código com IA.' and a small icon of a person.

```
1 import sqlite3
```

```
1 sqlite3.connect()
```

Analise dados.ipynb

Arquivo Editar Ver Inserir Ambiente de ex: Comentário

Arquivos

Disco

Disponível: 80.64 GB

0s conclusão: 15:28

```
[123] 1 import sqlite3
```

```
1 sqlite3.connect('/content/drive/MyDrive/programaria/status_brasil')
```

```
1 conexao = sqlite3.connect('/content/drive/MyDrive/program
```

```
'SELECT * FROM Municipios_Brasileiros WHERE Cidade='Itaquaquecetuba'
```

Faça a consulta:

```
1 pd.read_sql(query, con=conexao)
```

	Cidade	Estado	Regiao	municipio_ID
0	Itaquaquecetuba	SP	Sudeste	5043

Parte 2:

Vamos acompanhar a média da renda por estado.

Copie o JOIN que fizemos no DBeaver, na tabela status\_brasil:

```
--ALTER TABLE Municipios_Brasileiros DROP COLUMN pais ;
--SELECT * FROM Municipios_Brasileiros WHERE Cidade LIKE 'Itaquã';
--SELECT * FROM Municipio_Status WHERE populacao_residente>50000;
--SELECT Municipios_Brasileiros.Cidade, Municipio_Status.populacao_residente FROM Municipios_Brasileiros
--INNER JOIN Municipio_Status ON Municipios_Brasileiros.municipio_ID = Municipio_Status.municipio_ID
--SELECT Estado, COUNT(Cidade) FROM Municipios_Brasileiros GROUP BY Estado ORDER BY 2 DESC;
--SELECT SUM(pessoas_branças), SUM(pessoas_pretas_pardas) FROM Gerencia_Regiao ;
--SELECT Regiao, MIN(pessoas_pretas_pardas) FROM Gerencia_Regiao;
SELECT Regiao FROM Gerencia_Regiao WHERE gerencia_branca>gerencia_preta_parda ;
```

Regiao
Sudeste
Sul
Centro-Oeste

Volte no notebook e cole:

```
1 iros.Cidade, Municipio_Status.populacao_residente FROM Municipios_Br
2 Municipios_Brasileiros.municipio_ID = Municipio_Status.municipio_ID
```

Organize: adicione três aspas simples no início e final do texto; mude alinhe as duas linhas do texto; mude no texto de **cidade para estado** e de **populacao.residente para renda**

```
1 query = '''SELECT Municipios_Brasileiros.Cidade, Municipio_Status.po
2         INNER JOIN Municipio_Status ON Municipios_Brasileiros.municipi
```

Inclua AVG e adicione Municipio\_Status.renda dentro dos parênteses:

```
1 Municipios_Brasileiros.Estado, AVG(Municipio_Status.renda) FROM Municipi
2 municio_Status ON Municipios_Brasileiros.municipio_ID = Municipio_Stat
```

Adicione GROUP BY ao final do texto e Municipios\_Brasileiros.Estado:

```
1 query = '''SELECT Municipios_Brasileiros.Estado, AVG(Municipio_Status.
2         INNER JOIN Municipio_Status ON Municipios_Brasileiros.municipi
3         GROUP BY Municipios_Brasileiros.Estado;'''
```

Execute a consulta novamente:

```
1 pd.read_sql(query, con=conexao)
```

Agora vamos fazer a leitura do nosso arquivo usando pandas.

Adicione dentro das aspas o caminho do arquivo que já havíamos salvo em aulas anteriores:

```
1 dados = pd.read_csv(r'/content/drive/MyDrive/progra
```

The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with files named 'analise\_dados.csv', 'analise\_dados.ipynb', and 'arquivo\_final.csv'. The code editor shows the following code:

```
1 dados = pd.read_csv(r'/content/drive/MyDrive/progra
```

The interface also shows a 'Comentário' (Comment) button and a 'Comp' (Compile) button. The status bar at the bottom indicates '0s' and 'conclusão: 15:38'.

```
[131] 1 dados = pd.read_csv('/content/drive/MyDrive/programaria/analise')
```

Deixe em formato de lista as informações:

```
1 list(dados['UF ONDE MORA'].unique())
```

Crie uma variável que vai receber esses resultados:

```
1 lista_estados = list(dados['UF ONDE MORA'].unique())
```

Parte 3:

Um exemplo sobre strings, acompanhe na videoaula [❤️](#)

Parte 4:

Volte na query que realizamos anteriormente, adicione **WHERE Municipios\_Brasileiros.Estado IN (())**

```
1 query = '''SELECT Municipios_Brasileiros.Estado
2         INNER JOIN Municipio_Status ON Munic
3         WHERE Municipios_Brasileiros.Estado I
4         GROUP BY Municipios_Brasileiros.Estado
```

Em seguida, adicione `.format(', '.join(['?' for _ in lista_estados]))` ao final do texto

```
1 Brasileiros.Estado, AVG(Municipio_Status.renda) FROM Municipios_Bras
2 Status ON Municipios_Brasileiros.municipio_ID = Municipio_Status.mun
3 ileiros.Estado IN
4 rasilieiros.Estado;''' .format(', '.join(['?' for _ in lista_estados]))
```

Acompanhe como fica:

```
1 Brasileiros.Estado, AVG(Municipio_Status.renda) FROM Municipios_Bras
2 Status ON Municipios_Brasileiros.municipio_ID = Municipio_Status.mun
3 ileiros.Estado IN (())
4 rasilieiros.Estado;''' .format(', '.join(['?' for _ in lista_estados]))
```

Execute:

```
1 pd.read_sql(query, con=conexao; params=lista_estados)
```

Agora, vamos adicionar dentro de uma variável:

```
[158] 1 estados_renda = pd.read_sql(query, con=conexao, params=lista_estados)
```

Mude o nome da coluna:

```
1 dados.rename(columns={'UF ONDE MORA': 'Estado'}, inplace=True)
```

Em seguida, faça o merge:

```
1 dados.merge(estados_renda, on="Estado", how="left")
```

Atribua o resultado para nossos dados:

```
[164] 1 dados = dados.merge(estados_renda, on="Estado", how="left")
```

Faça a correlação:

```
1 correlacao_renda_salario = dados['SALARIO'].corr(dados[''])
```

Adicione dentro das aspas simples `AVG(Municipio_Status.renda)` e execute:

```
1 lario = dados['SALARIO'].corr(dados['AVG(Municipio_Status.renda)'])
+ Código + Texto
[166] 1 correlacao_renda_salario = dados['SALARIO'].corr(dados['AVG(Municipio
2 correlacao_renda_salario
0.12083384881429463
```

