

Aula_12

RafaelaB

Aula/Desafio 12

```
Sys.time()
```

```
[1] "2025-10-09 09:43:16 -03"
```

```
getwd()
```

```
[1] "H:/Documentos/me315"
```

Importando bibliotecas

```
import polars as pl
import sqlite3
```

```
conn = sqlite3.connect('H:/Documentos/me315/disco.db')
cursor = conn.cursor()
```

Criação manual de uma tabela

- O método `execute` é empregado para executar comandos dentro do banco SQL.

```
cursor.execute('''CREATE TABLE IF NOT EXISTS vendas (
    id INTEGER PRIMARY KEY,
    vendedor TEXT,
    produto TEXT,
    valor REAL,
    data_venda DATE)''')
```

```
<sqlite3.Cursor object at 0x0000026B869CE640>
```

Inserção de dados em uma tabela

- Ao realizar uma inserção, deve-se executar o `commit`, que fará a confirmação da operação.

```
cursor.execute('''INSERT INTO vendas (vendedor, produto, valor, data_venda)
VALUES
    ('Ana', 'Produto A', 120.5, '2024-09-01'),
    ('Carlos', 'Produto B', 200.0, '2024-10-02'),
    ('Ana', 'Produto C', 150.0, '2024-09-03'),
    ('Bruno', 'Produto A', 300.0, '2024-11-04'),
    ('Carlos', 'Produto C', 100.0, '2024-10-05');''')
```

```
<sqlite3.Cursor object at 0x0000026B869CE640>
```

```
conn.commit()
```

Consulta simples no SQLite

```
cursor.execute("SELECT * FROM vendas")
```

```
<sqlite3.Cursor object at 0x0000026B869CE640>
```

```
rows = cursor.fetchall()
for row in rows:
    print(row)
```

```
(1, 'Ana', 'Produto A', 120.5, '2024-09-01')
(2, 'Carlos', 'Produto B', 200.0, '2024-10-02')
(3, 'Ana', 'Produto C', 150.0, '2024-09-03')
(4, 'Bruno', 'Produto A', 300.0, '2024-11-04')
(5, 'Carlos', 'Produto C', 100.0, '2024-10-05')
(6, 'Ana', 'Produto A', 120.5, '2024-09-01')
(7, 'Carlos', 'Produto B', 200.0, '2024-10-02')
(8, 'Ana', 'Produto C', 150.0, '2024-09-03')
(9, 'Bruno', 'Produto A', 300.0, '2024-11-04')
(10, 'Carlos', 'Produto C', 100.0, '2024-10-05')
```

```
#Estamos consultando as linhas da tabela vendas que acabamos de criar
```

Integração com Polars

```
dados = pl.read_database("SELECT * FROM vendas", conn)
print(dados)
```

shape: (10, 5)

id	vendedor	produto	valor	data_venda
---	---	---	---	---
i64	str	str	f64	str
1	Ana	Produto A	120.5	2024-09-01
2	Carlos	Produto B	200.0	2024-10-02
3	Ana	Produto C	150.0	2024-09-03
4	Bruno	Produto A	300.0	2024-11-04
5	Carlos	Produto C	100.0	2024-10-05
6	Ana	Produto A	120.5	2024-09-01
7	Carlos	Produto B	200.0	2024-10-02
8	Ana	Produto C	150.0	2024-09-03
9	Bruno	Produto A	300.0	2024-11-04
10	Carlos	Produto C	100.0	2024-10-05

Exemplos

1. Qual é o total de vendas por vendedor?

```
vendas_total = pl.read_database('''
    SELECT vendedor, SUM(valor) as total_vendas
    FROM vendas
    GROUP BY vendedor;
''', conn)
print(vendas_total)
```

shape: (3, 2)

vendedor	total_vendas
---	---
str	f64
Ana	541.0

Bruno	600.0
Carlos	600.0

2. Qual é o valor médio de venda por vendedor?

```
vendas_medias = pl.read_database('''
    SELECT vendedor, AVG(valor) as total_vendas
    FROM vendas
    GROUP BY vendedor;
''', conn)
print(vendas_medias)
```

shape: (3, 2)

vendedor	total_vendas
---	---
str	f64
Ana	135.25
Bruno	300.0
Carlos	150.0

#AVG -> expressão que calcula o valor médio de um conjunto de linhas

3. Crie uma tabela contendo o *nome do vendedor*, o *número de vendas* realizadas, o *total vendido* e o *valor médio por venda*.

```
vendas_comb = pl.read_database("""
SELECT vendedor,
    COUNT(*) as numero_vendas,
    SUM(valor) as total_vendas,
    AVG(valor) as media_vendas
FROM vendas
GROUP BY vendedor;
""", conn)
print(vendas_comb)
```

shape: (3, 4)

vendedor	numero_vendas	total_vendas	media_vendas
---	---	---	---

str	i64	f64	f64
Ana	4	541.0	135.25
Bruno	2	600.0	300.0
Carlos	4	600.0	150.0

4. Quais foram as vendas de pelo menos 200,00?

```
ticket_alto = pl.read_database("""
SELECT * FROM vendas WHERE valor >= 200
""", conn)
print(ticket_alto)
```

shape: (4, 5)

id	vendedor	produto	valor	data_venda
---	---	---	---	---
i64	str	str	f64	str
2	Carlos	Produto B	200.0	2024-10-02
4	Bruno	Produto A	300.0	2024-11-04
7	Carlos	Produto B	200.0	2024-10-02
9	Bruno	Produto A	300.0	2024-11-04

Operações com Datas em SQLite

1. Qual foi o volume total de vendas?

```
vendas_mensais = pl.read_database("""
SELECT strftime('%Y-%m', data_venda) AS mes, SUM(valor) AS total_vendas
FROM vendas GROUP BY mes ORDER BY mes
""", conn)
print(vendas_mensais)
```

shape: (3, 2)

mes	total_vendas
---	---
str	f64
2024-09	541.0

```
2024-10    600.0
2024-11    600.0
```

```
#Ele apenas "seleciona" o ano e o mês e chama a coluna de "mês"
```

Criando a tabela de produtos

```
cursor.execute('''
CREATE TABLE IF NOT EXISTS produtos (
    id INTEGER PRIMARY KEY,
    nome TEXT NOT NULL,
    categoria TEXT NOT NULL,
    preco REAL NOT NULL,
    estoque INTEGER NOT NULL
);
''')
```

```
<sqlite3.Cursor object at 0x0000026B869CE640>
```

```
cursor.execute('''
INSERT INTO produtos (nome, categoria, preco, estoque) VALUES
    ('Produto A', 'Categoria 1', 100.0, 50),
    ('Produto B', 'Categoria 2', 150.0, 30),
    ('Produto C', 'Categoria 1', 200.0, 20),
    ('Produto D', 'Categoria 2', 250.0, 10),
    ('Produto E', 'Categoria 3', 300.0, 0);
''')
```

```
<sqlite3.Cursor object at 0x0000026B869CE640>
```

```
conn.commit()
```

Consultando a tabela de produtos

```
prods = pl.read_database("SELECT * FROM produtos", conn)
print(prods)
```

shape: (10, 5)

id	nome	categoria	preco	estoque
---	---	---	---	---
i64	str	str	f64	i64
1	Produto A	Categoria 1	100.0	50
2	Produto B	Categoria 2	150.0	30
3	Produto C	Categoria 1	200.0	20
4	Produto D	Categoria 2	250.0	10
5	Produto E	Categoria 3	300.0	0
6	Produto A	Categoria 1	100.0	50
7	Produto B	Categoria 2	150.0	30
8	Produto C	Categoria 1	200.0	20
9	Produto D	Categoria 2	250.0	10
10	Produto E	Categoria 3	300.0	0

- Apresente uma tabela com o nome do produto, seu valor de compra e venda, além do lucro no momento da venda.

```
lucros = pl.read_database("""
SELECT produto, valor AS compra, preco AS venda, preco-valor AS lucro
FROM vendas
INNER JOIN produtos ON vendas.produto = produtos.nome
""", conn)
print(lucros)
```

shape: (20, 4)

produto	compra	venda	lucro
---	---	---	---
str	f64	f64	f64
Produto A	120.5	100.0	-20.5
Produto A	120.5	100.0	-20.5
Produto B	200.0	150.0	-50.0
Produto B	200.0	150.0	-50.0
Produto C	150.0	200.0	50.0
...
Produto C	150.0	200.0	50.0
Produto A	300.0	100.0	-200.0
Produto A	300.0	100.0	-200.0
Produto C	100.0	200.0	100.0

Produto C	100.0	200.0	100.0
-----------	-------	-------	-------

Exemplos

1. Qual foi o valor médio por vendedor?

```
lucro_medio = pl.read_database("""
SELECT vendedor, produto, AVG(preco-valor) AS lucro_medio
FROM vendas
INNER JOIN produtos ON vendas.produto = produtos.nome
GROUP BY vendedor
""", conn)
print(lucro_medio)
```

shape: (3, 3)

vendedor	produto	lucro_medio
---	---	---
str	str	f64
Ana	Produto A	14.75
Bruno	Produto A	-200.0
Carlos	Produto B	25.0

Desconectando banco de dados

```
conn.close()
```