

DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES EM PYTHON

Tema 3 : Aula 08 - Python com Banco
de dados (Parte IV).



Prof. Ronaldo Candido
ronaldo.candido@estacio.br




Estácio

2023.2

1

DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES EM PYTHON


 **Estácio**

Objetivos da aula

- **Empregar as funcionalidades para recuperação de registros em tabelas.**

Conteúdo Programático


1. Situação-problema.
2. Seleção de registros em uma tabela.
3. Seleção de registros usando uma junção.
4. Seleção de registros relacionados.
5. Atividade verificadora de aprendizagem.
6. Aprenda+.



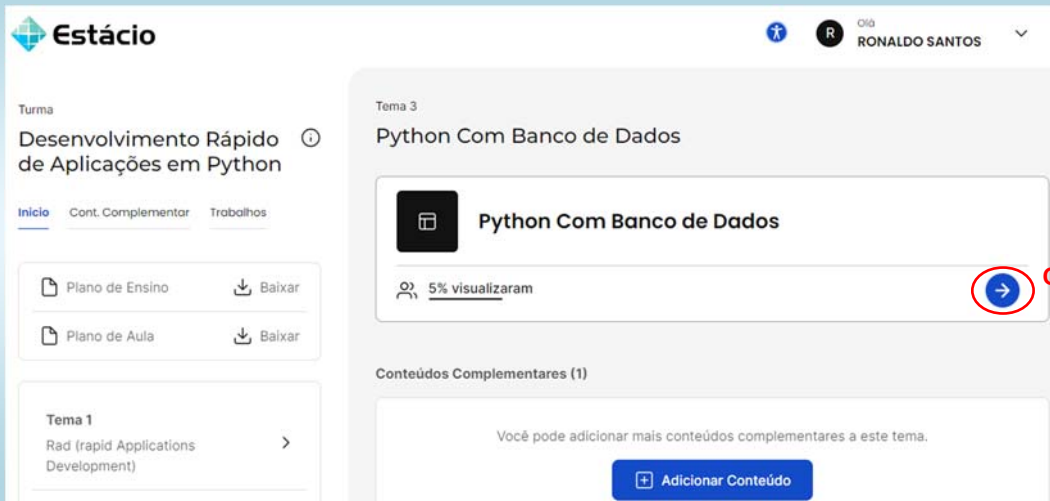
Prof. Ronaldo Candido

AULA 08

2


DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES EM PYTHON 

Sala de Aula Virtual (SAVA)




Prof. Ronaldo Candido AULA 08

3

DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES EM PYTHON 

Situação-problema

- Na aula anterior vimos como inserir, modificar e excluir dados de Banco de Dados usando Python. Entretanto uma aplicação que faz uso de banco de dados, a operação que é usada com maior frequência é a consulta aos dados.
- Como realizar operações de consulta ao Banco de Dados usando Python?

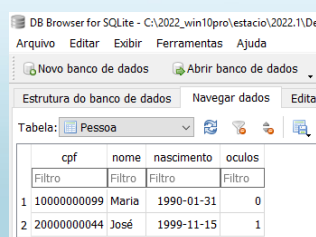


Prof. Ronaldo Candido AULA 08

4

Seleção de registros em uma tabela

- Para selecionar registros em uma tabela, utilizamos o comando **SELECT**, do **SQL**.
Sintaxe:
`SELECT coluna1, coluna2, ... FROM tabela1,tabela2 WHERE [condição];`
- Utilizaremos um exemplo para selecionar todos os registros na tabela Pessoa :
`SELECT * FROM Pessoa ;`
- Assim como nos outros comandos, podemos utilizar uma **string** sem delimitadores, uma **string** com o delimitador "?" ou uma **string** com argumentos nomeados para a condição da cláusula **WHERE**.



	cpf	nome	nascimento	oculos
1	10000000099	Maria	1990-01-31	0
2	20000000044	José	1999-11-15	1

Prof. Ronaldo Candido

AULA 08

5

Exemplo de código no SQLite

```
#Aula08_exemplo01
import sqlite3 as conector #apelido
def listar_tabela():
    try: #abertura da conexão
        conexao = conector.connect('meu_banco.db')
        cursor = conexao.cursor()
        #execução de comandos SQL
        sql = 'SELECT * FROM Pessoa'
        cursor.execute(sql)
        #recuperação dos dados
        registros = cursor.fetchall()
        print('Tipo retornado:', type(registros))
```

```
#saída de dados sem formatação
for reg in registros: #reg é uma tupla
    cpf,nome,nascimento,oculos = reg
    print('CPF:',cpf,'Nome:',nome,'Nasc.:',
nascimento,'Óculos?',oculos)
except conector.DatabaseError as err:
    print('Erro de banco de dados',err)
finally:
    #fechamento das conexões
    if(conexao):
        cursor.close()
        conexao.close()
listar_tabela() #executar a função
```

Prof. Ronaldo Candido

AULA 08

6

Seleção de registros usando uma junção

- Para selecionar registros de tabelas relacionadas usamos o **SELECT** com junção. Sintaxe:

```
SELECT tab1.col1, tab1.col2, tab2.col1... FROM tab1 JOIN tab2 ON tab1.colN = tab2.colM;
```

- Primeiro, definimos quais colunas serão retornadas utilizando a sintaxe nome_tabela.nome_coluna, depois indicamos as tabelas que desejamos juntar e, por último, indicamos como alinhar os registros de cada tabela, ou seja, quais são os atributos que devem ser iguais (**colN** e **colM**). Exemplo :

```
SELECT Veiculo.placa, Veiculo.ano, Veiculo.cor, Veiculo.motor, Veiculo.proprietario,
Marca.nome FROM Veiculo JOIN Marca ON Marca.id = Veiculo.marca;
```

- As tuplas retornadas serão similares à seguinte:

```
('AAA0001', 2001, 'Prata', 1.0, 10000000099, 'Marca A')
```

Os elementos de uma tupla ficam entre parenteses

tuple = ("a", "b", "c")

Virgulas são usadas para separar os elementos da tupla

Seleção de registros usando uma junção (continuação)

```
#Aula08_exemplo02.py
```

```
def listar_tabelas_join():
```

```
    try:
```

```
        conexao = conector.connect('meu_banco.db')
```

```
        cursor = conexao.cursor()
```

```
        sql = "SELECT Veiculo.placa, Veiculo.ano, Veiculo.cor,
Veiculo.motor, Veiculo.proprietario, Marca.nome FROM Veiculo
JOIN Marca ON Marca.id = Veiculo.marca;"
```

```
        cursor.execute(sql)
```

```
        registros = cursor.fetchall()
```

```
        for reg in registros:
```

```
            placa,ano,cor,motor,cpf,marca = reg
```

```
            print('Placa:',placa,'Ano:',ano,'Cor:',cor,
```

```
                  'Motor:',motor,'CPF:',cpf,'Marca:',marca)
```

```
#continua...
```

É necessário que o banco de dados já tenha sido criado e inserido com as três tabelas da aula anterior !!!

```
except conector.DatabaseError as err:
    print('Erro de banco de dados',err)
finally:
    if(conexao):
        cursor.close()
        conexao.close()
    listar_tabelas_join()
```

Seleção de registros relacionados

- Para finalizar, vamos recuperar todas as pessoas, com seus respectivos veículos e marcas. Usaremos funções para executar os scripts:

```
import sqlite3 as conector #apelido
def listar_tabelas_relacionadas():
    try:
        conexao = conector.connect('meu_banco.db')
        cursor = conexao.cursor()
        sql = 'SELECT * FROM Pessoa'
        cursor.execute(sql)
        pessoas = []
        reg_pessoas = cursor.fetchall()
        for reg_pessoa in reg_pessoas:
            cpf,nome,nascimento,oculos = reg_pessoa

            print('Proprietário:',nome)
            veiculos = recuperar_veiculos(conexao,cpf)
            pessoas.append(veiculos)
    except conector.DatabaseError as err:
        print('Erro de banco de dados',err)
    finally:
        if(conexao):
            cursor.close()
            conexao.close()
```

Seleção de registros relacionados (continuação)

```
def recuperar_veiculos(conexao, cpf):
    try:
        cursor = conexao.cursor()
        sql = 'SELECT * FROM Veiculo JOIN Marca ON Marca.id=Veiculo.marca WHERE Veiculo.proprietario=?;'
        cursor.execute(sql,(cpf,))
        veiculos = []
        registros = cursor.fetchall()
        for registro in registros:
            veiculos.append(registro)
            print(registro)
    except conector.DatabaseError as err:
        print('Erro de banco de dados',err)

    finally:
        cursor.close()
        return (veiculos)

#executar função
listar_tabelas_relacionadas()

#encerrando
print("Fim do programa")
```


Atividade verificadora de aprendizagem

- Fazer o Avaliando o Aprendizado 1 em aula (presencial) ou Tarefa no Teams.
- Voltando ao cenário que trata de um sistema de registro de notas de alunos em uma pequena instituição de ensino, implementar funcionalidades de consulta ao Banco de Dados:
 - Buscar os dados do aluno ao solicitar sua matrícula.
 - Listar todos os alunos de um determinado curso.
 - Mostrar as notas de todos os alunos, ordenados pelo nome.



Aprenda +

- Assistir o vídeo: SENTDEX. "Read from (SELECT) Database table SQLite3 with Python 3 part 3". Disponível em <<https://www.youtube.com/watch?v=NCc5r7Wr7gg>>. (Ative as legendas e a tradução para português). Acesso em: 20 abr. 2023.
- Assistir o vídeo: STACK. "Trabalhando com Banco de Dados SQL usando Python". Disponível em <<https://www.youtube.com/watch?v=WKoeBgKC0o0>>. Acesso em: 20 abr. 2023.
- Atividade Autônoma Aura:

Questão 1) A linguagem de programação Python tem enorme facilidade e flexibilidade para a realização de consultas nos mais variados banco de dados. Qual comando abaixo é usado para rodar uma consulta:

 - a) execute()
 - b) rodar()
 - c) executar()
 - d) run()
 - e) show()

Aprenda + (continuação)

Questão 2) A linguagem de programação Python tem enorme facilidade e flexibilidade para a realização de consultas nos mais variados banco de dados. Após executar a solicitação da consulta deve processar o resultado através dos métodos:

- a) fetchone, fetchall e fetchmany.
- b) fetread, fetchall e fetchmany.
- c) fetchone, fetchall e fetchmany.
- d) fetchone, fetchall e fetchmuch.
- e) fetchone, fetchall e fetchwrite.




Para a próxima aula...

- Leitura da documentação “**tkinter** — Interface Python para Tcl/Tk”. Disponível em: <<https://docs.python.org/pt-br/3/library/tkinter.html>>. Acesso em 20 abr. 2023.
- Conteúdo digital da disciplina, Tema 4 - “Interface Gráfica Com Python”, Módulo 1.
- Estudar o Tema 4 - Aula 09 de **DESENVOLVIMENTO RÁPIDO EM PYTHON** no SAVA previamente.



DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES EM PYTHON



 **Estácio**

Referências

BANIN, S. L. Python 3 Conceitos e Aplicações. Uma Abordagem Didática. 1a. ed. São Paulo: Érica, 2018. Páginas 329-363. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788536530253/>. Acesso em: 16 fev. 2023.

ESTÁCIO. Conteúdo digital da disciplina, Tema 3 - "Python com banco de dados", Módulo 4: "Seleção de registros de uma tabela". Disponível na Sala de aula virtual da disciplina.

PERKOVIC, L. Introdução à Computação Usando Python - Um Foco no Desenvolvimento de Aplicações. 1a. ed. Rio de Janeiro: LTC, 2016. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788521630937/>. Acesso em: 16 fev. 2023.

Prof. Ronaldo Candido

AULA 08

15

DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES EM PYTHON

Dúvidas, sugestões ou análises ???


 **Estácio**

Prof. Ronaldo Candido
ronaldo.candido@estacio.br

16