



PROGRAMAÇÃO EM PYTHON

Fernando F. Santos



python



AULA 21

17/04/2023

CONECTANDO O PYTHON COM O BANCO DE DADOS

- MySQL Connector/Python é um driver de banco de dados padronizado para plataformas e desenvolvimento em Python. Nós o usaremos para conectarmos ao banco de dados MySQL com Python. O MySQL Connector/Python pode ser instalado no momento da instalação do MySQL.
- Caso o MySQL Connector/Python não esteja instalado você pode baixar o instalador em:
<https://dev.mysql.com/downloads/connector/python/>

USANDO O MYSQL CONNECTOR/PYTHON

- Após instalação podemos executar nosso primeiro exemplo. Para acessar o MySQL usando o Python podemos usar o módulo "mysql". Veja um exemplo:

```
# Importando o MySQL Connector/Python
import mysql.connector

# Criando o objeto de conexão através do connect() com os parâmetros:
# usuário, senha, endereço do servidor e nome do banco/esquema
# o connect retorna um objeto MySQLConnection
conexao = mysql.connector.connect(user='root', password='1234', host='127.0.0.1', database='agenda')

# Fechando a conexão com o banco de dados
conexao.close()
```

ERROS NA CONEXÃO

- Para lidar com erros de conexão, use a instrução try e capture todos os erros usando a exceção errors.Error (que é uma subclasse de StandardError) conforme exemplo a seguir.

```
import mysql.connector
from mysql.connector import errorcode
try:
    conexao = mysql.connector.connect(user='root', password='1234', host='127.0.0.1', database='agenda')
except mysql.connector.Error as erro:
    if erro.errno == errorcode.ER_ACCESS_DENIED_ERROR:
        print("Usuário ou senha inválidos")
    elif erro.errno == errorcode.ER_BAD_DB_ERROR:
        print("Banco de dados não existe")
    else:
        print(erro)
else:
    conexao.close()
```

INSERINDO REGISTROS

```
# Montando uma String com o comando de insert
inserir_contato = ("insert into contatos (nome, telefone, celular) "
                  "values ('Pateta', '(27)1111-1111', '(27)12345-3333')")
# Executando o comando insert
cursor.execute(inserir_contato)
# Gravando a informação permanentemente
conexao.commit()
# Fechando o cursor
cursor.close()
# Fechando a conexão
conexao.close()
```

ALTERANDO REGISTROS

```
import mysql.connector
conexao = mysql.connector.connect(user='root', password='1234', host='127.0.0.1', database='agenda')
cursor = conexao.cursor()
atualizar_contato = ("update contatos set nome = %s, telefone = %s, celular = %s where id = 1")
contato = ('Pocahontas', '(28)8888-8888', '(28)88888-8888')
cursor.execute(atualizar_contato, contato)
conexao.commit()
cursor.close()
conexao.close()
```

APAGANDO REGISTROS

```
import mysql.connector
conexao = mysql.connector.connect(user='root', password='1234', host='127.0.0.1', database='agenda')
cursor = conexao.cursor()
remover_contato = ("delete from contatos where id = 1")
cursor.execute(remover_contato)
conexao.commit()
cursor.close()
conexao.close()
```

CONSULTANDO REGISTROS

```
import mysql.connector
conexao = mysql.connector.connect(user='root', password='1234', host='127.0.0.1', database='agenda')
cursor = conexao.cursor()
consulta = ("select nome, telefone, celular from contatos where nome like 'M%")
cursor.execute(consulta)
for (nome, telefone, celular) in cursor:
    print(f"Nome: {nome}, telefone: {telefone}, celular: {celular}")
cursor.close()
conexao.close()
```


EXERCÍCIO DE FIXAÇÃO

1. Crie um banco de dados com o nome **empresa**, que contenha a tabela **Funcionarios**(**Codigo**, PrimeiroNome, UltimoNome, DataNasci, CPF, RG, Endereco, CEP, Cidade, Fone, Funcao, Salario)
2. Crie um software em Python que insira, altere e apague os cadastro feitos no banco de dados
3. O software deve apresentar uma consulta que apresente todos os funcionários
4. Faça uma consulta que apresente apenas funcionários com idade menor 30 anos e com salario maior do que R\$1000,00