



- Pré-requisitos
- Preparação de Terreno
- Inserindo no BD
- Listagem do BD
- Exclusão do BD
- Atualização do BD
- Resumo das Operações do BD

Python

Lógica de Programação

Conectando Python com o Banco de Dados MySQL

01 Pré-Requisitos

- Neste *cheat sheet*, aprenderemos como interagir com Banco de Dados MySQL usando o Python
- Os pré-requisitos são os seguintes:
 - Instalação do Python 3
 - <https://realpython.com/installing-python/> ou
 - <https://www.python.org/downloads/>
 - Instalação do Conector do MySQL
 - No *prompt*, digite o seguinte comando:
 - `python3 -m pip install mysql-connector-python`
 - Instalação do MySQL e Workbench
 - <https://www.alura.com.br/artigos/mysql-do-download-e-instalacao-ate-sua-primeira-tabela>

02 Preparação do Terreno

- O que precisamos agora é criar as nossas tabelas que vamos usar. Para isto, precisamos realizar os seguintes passos:

Comandos (Procedimentos)	Descrição
<code>CREATE DATABASE ouvidoria;</code>	Cria o BD ouvidoria
<code>USE ouvidoria;</code>	Seleciona o BD ouvidoria para uso
<pre> 1 CREATE TABLE usuarios (2 id INT AUTO_INCREMENT, 3 nome VARCHAR(100), 4 usuario VARCHAR(30), 5 senha VARCHAR(20), 6 PRIMARY KEY(id) 7); </pre>	Cria a tabela usuarios com os campos ID, nome, usuário e senha.

03 Inserindo no BD

- No código a seguir, realizamos uma inserção de elementos no BD.
 - Linha 1: Importamos a biblioteca do MySQL
 - Linha 3 a 8: Criamos uma conexão com BD, que está funcionando na mesma máquina (localhost), com usuário root, senha 2424 e o nome do BD é ouvidoria.
 - Poderia ter feito em uma linha só, mas quebramos em várias para ficar mais entendível

```

1 import mysql.connector
2
3 connection = mysql.connector.connect(
4   host="localhost",
5   user="root",
6   password="2424",
7   database="ouvidoria"
8 )
9
10 cursor = connection.cursor()
11
12 sql = "INSERT INTO usuarios (nome, usuario, senha) VALUES (%s, %s, %s)"
13 data = ('Daniel Abella', 'dabella', '12345')
14
15 cursor.execute(sql, data)
16 connection.commit()
17
18 userid = cursor.lastrowid
19
20 cursor.close()
21 connection.close()
22
23 print("ID do novo usuário:", userid)
                    
```

- **Linha 10:** usamos a conexão criada para criar um cursor, que é quem realiza as consultas no BD.
- **Linha 12:** criamos uma String com a consulta que faremos
 - Note que, os 3 valores de um novo usuário usamos %s
- **Linha 13:** criamos uma tupla com 3 elementos, que são os 3 valores de um novo usuário
- **Linha 15:** O cursor executa o SQL com os valores da tupla
 - A inserção foi feita no BD? Ainda não ☹️ Precisamos chamar a linha 16, que é o commit
- **Linha 16:** O *commit* é quem é o cara! É agora que a inserção solicitada na linha 15 é realizada!

Por que precisa do *commit* da linha 16? No exemplo acima, temos apenas uma operação de *execute* (linha 15). Mas, antes do *commit*, poderíamos ter inúmeras operações *execute* (além da linha 15). E, todas essas operações só seriam enviada para o BD após o *commit*.

- **Linha 18:** Pega o ID gerado automaticamente na inserção
- **Linha 20:** Terminou de usar o cursor? Feche com *close*.
- **Linha 21:** Terminou de usar a conexão? Feche com *close*.
- **Linha 23:** Imprime o ID Gerado na inserção.

Em resumo:

- `Execute()` envia o comando ao BD
- `Commit()` efetiva todos os comandos enviados pelos `Execute()` executados antes do `commit`

04 Listagem do BD

- Nesta seção apresentaremos como realizar um `SELECT` na base de dados.
 - Linha 12: Select que será realizado
 - Linha 14: Executa o comando de Select
 - Linha 15: Executa um comando de listar tudo chamado `fetchall`
 - A variável `listaUsuarios` recebe uma lista com tuplas que representa os resultados
 - Linha 20: Apresenta a `listaUsuarios` sem segredo
 - Linha 21: Apresentamos a tupla com os dados de cada usuário
 - Linha 22: Maneira usual de apresentar os dados.
 - Só apresentei os 2 primeiros campos :P

```
1 import mysql.connector
2
3 connection = mysql.connector.connect(
4     host="localhost",
5     user="root",
6     password="2424",
7     database="ouvidoria"
8 )
9
10 cursor = connection.cursor()
11
12 sql = "SELECT * FROM usuarios"
13
14 cursor.execute(sql)
15 listaUsuarios = cursor.fetchall()
16
17 cursor.close()
18 connection.close()
19
20 for usuario in listaUsuarios:
21     print(usuario)
22     print(usuario[0], usuario[1])
```

05 Exclusão do BD

- Linha 12: Consulta para exclusão (exclui se o ID for igual a 1)
- Linha 13: Informa o valor do ID (neste exemplo, 1) Usamos uma tupla
- Linha 18: Obtemos a quantidade de linhas afetadas pela consulta.
- Linha 23: Imprime a quantidade de linhas afetadas.

```
1 import mysql.connector
2
3 connection = mysql.connector.connect(
4     host="localhost",
5     user="root",
6     password="2424",
7     database="ouvidoria"
8 )
9
10 cursor = connection.cursor()
11
12 sql = "DELETE FROM usuarios WHERE id = %s"
13 data = (2,)
14
15 cursor.execute(sql, data)
16 connection.commit()
17
18 recordsaffected = cursor.rowcount
19
20 cursor.close()
21 connection.close()
22
23 print('linhas afetadas ', recordsaffected)
```

06 Atualização do BD

- Para finalizar, a seguir usamos o comando `UPDATE`. Creio que, nenhuma linha abaixo é novidade, dado os exemplos apresentados anteriormente.

```
1 import mysql.connector
2
3 connection = mysql.connector.connect(
4     host="localhost",
5     user="root",
6     password="2424",
7     database="ouvidoria"
8 )
9
10 cursor = connection.cursor()
11
12 sql = "UPDATE usuarios SET nome = %s, usuario = %s WHERE id = %s"
13 data = ('Abella', 'abellad', 1)
14
15 cursor.execute(sql, data)
16 connection.commit()
17
18 recordsaffected = cursor.rowcount
19
20 cursor.close()
21 connection.close()
22
23 print('linhas afetadas ', recordsaffected)
```

07 Resumo das Operações com BD

- `SELECT ... FROM ... WHERE ...`
- `INSERT INTO ... VALUES ...`
- `UPDATE ... SET ... WHERE ...`
- `DELETE FROM ... WHERE ...`

SQL INSERT,
UPDATE, DELETE?

