

# Abrindo conexões com Banco de Dados em Python

## 1. Banco de Dados MySQL

Bancos de dados em SQL, utilizam sempre a linguagem natural SQL para comunicação entre o banco de dados e o software, seja lá qual for a linguagem utilizada. O funcionamento padrão de um banco de dados é:

**DATABASE** → um diretório reservado para agrupar tabelas e por sua vez organizá-los, a função de um banco de dados é servir como espaço para agrupar os dados de forma sistêmica utilizando diversas tabelas salvas em si.

**TABELA** → uma tabela é um modelo conhecido pelo Banco de Dados, que assume as características de um determinado tipo de informação, anotando por sua vez os seus dados (nome, endereço, senha, horários etc.) e que tipo de dado ele assume (se ele é texto, data ou numérico por exemplo).

**USUÁRIO** → uma credencial no banco, que deve ser fornecido para reconhecimento e controle, todo usuário deve possuir senha e por sua vez suas autorizações que são quais tarefas ele pode realizar, quais bancos e quais tabelas ele tem acesso.

## 2. Criando um Banco de Dados e Tabela para testes

Para criarmos um banco de dados, vamos assumir as seguintes características:

SERVIDOR MYSQL: **127.0.01** → ip local da máquina

USUÁRIO: **root** → usuário mestre, tem acesso total ao banco de dados

**Não possui senha.**

BANCO DE DADOS: **dbpython**

---

Abra o console do MySQL utilizando o WampServer instalado em sua máquina, ele possuirá o servidor MySQL instalado e configurado. Ao abrir faça os comandos abaixo:

```
-- irá apagar qualquer banco de dados existente de mesmo nome
DROP DATABASE IF EXISTS dbpython;
-- cria um novo banco de dados chamado 'dbpython'
CREATE DATABASE dbpython;
-- cria uma tabela 'contatos' sobre este banco de dados
CREATE TABLE dbpython.contatos(
    id INT PRIMARY KEY AUTO_INCREMENT,
    nome VARCHAR(255),
    email VARCHAR(255),
    telefone VARCHAR(255)
) ENGINE = InnoDB;

-- dentre parenteses estão separados por virgula todas
-- as características, que serão os dados a serem salvos
-- de cada novo contato cadastrado no banco de dados.
```

O resultado deve ser o seguinte:

```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.27 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> DROP DATABASE IF EXISTS dbpython;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> CREATE DATABASE dbpython;
Query OK, 1 row affected (0.00 sec)

mysql> CREATE TABLE dbpython.contatos(
  -> id INT PRIMARY KEY AUTO_INCREMENT,
  -> nome VARCHAR(255),
  -> email VARCHAR(255),
  -> telefone VARCHAR(255)
  -> ) ENGINE = InnoDB;
Query OK, 0 rows affected (0.03 sec)

mysql>
```

### 3. Abrindo uma conexão com o banco de dados pelo Python

Abra um novo script em Python, e faça o seguinte código:

```
# importando a biblioteca que baixamos pelo PIP
import mysql.connector as mysql

try:
    # esta é a função de conexão com o banco
    # devem ser fornecidos as configurações de conexão
    conexao = mysql.connect(
        host = "127.0.0.1", # ip do servidor
        user = "root", # usuario
        password = "", # senha do usuario
        database = "dbpython"
    )
    print("Objeto: ", conexao)
    print("Classe do objeto: ", type(conexao))

except mysql.Error as e:
    # capturando possiveis erros de conexao ou SQL com TRY CATCH
    print(e.msg)
```