



# INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 10 - SQL I

# O QUE IREMOS APRENDER

- 01** CONTEXTUALIZAÇÃO DA AULA DE HOJE
- 02** BANCO DE DADOS
- 03** SQL
- 04** TIPOS DE BANCO DE DADOS
- 05** SGBDS
- 06** MYSQL
- 07** CRIANDO UM BANCO DE DADOS
- 08** COMANDOS SQL (CREATE, DROP, TRUNCATE)

# Contextualização da aula de hoje

Usar um banco de dados é essencial por várias razões em ambientes onde é necessário **armazenar** e **gerenciar** dados de forma eficiente. Os bancos de dados desempenham um papel crítico em organizações e aplicativos modernos, pois fornecem uma estrutura robusta para armazenar, acessar e proteger os dados.

Eles facilitam a tomada de decisões informadas, o compartilhamento de informações, garantem a segurança dos dados e promovem a eficiência operacional.

# Banco de Dados

Basicamente, um banco de dados é uma planilha, como a do excel, gerenciada por um software, entretanto, além do software. há outras diferenças:

- Como os dados são armazenados e manipulados
  - Quem pode acessar os dados
- Quantos dados podem ser armazenados



## Banco de Dados

# Banco de Dados

As planilhas foram originalmente projetadas para serem usadas por um único usuário, e suas características refletem esse propósito.

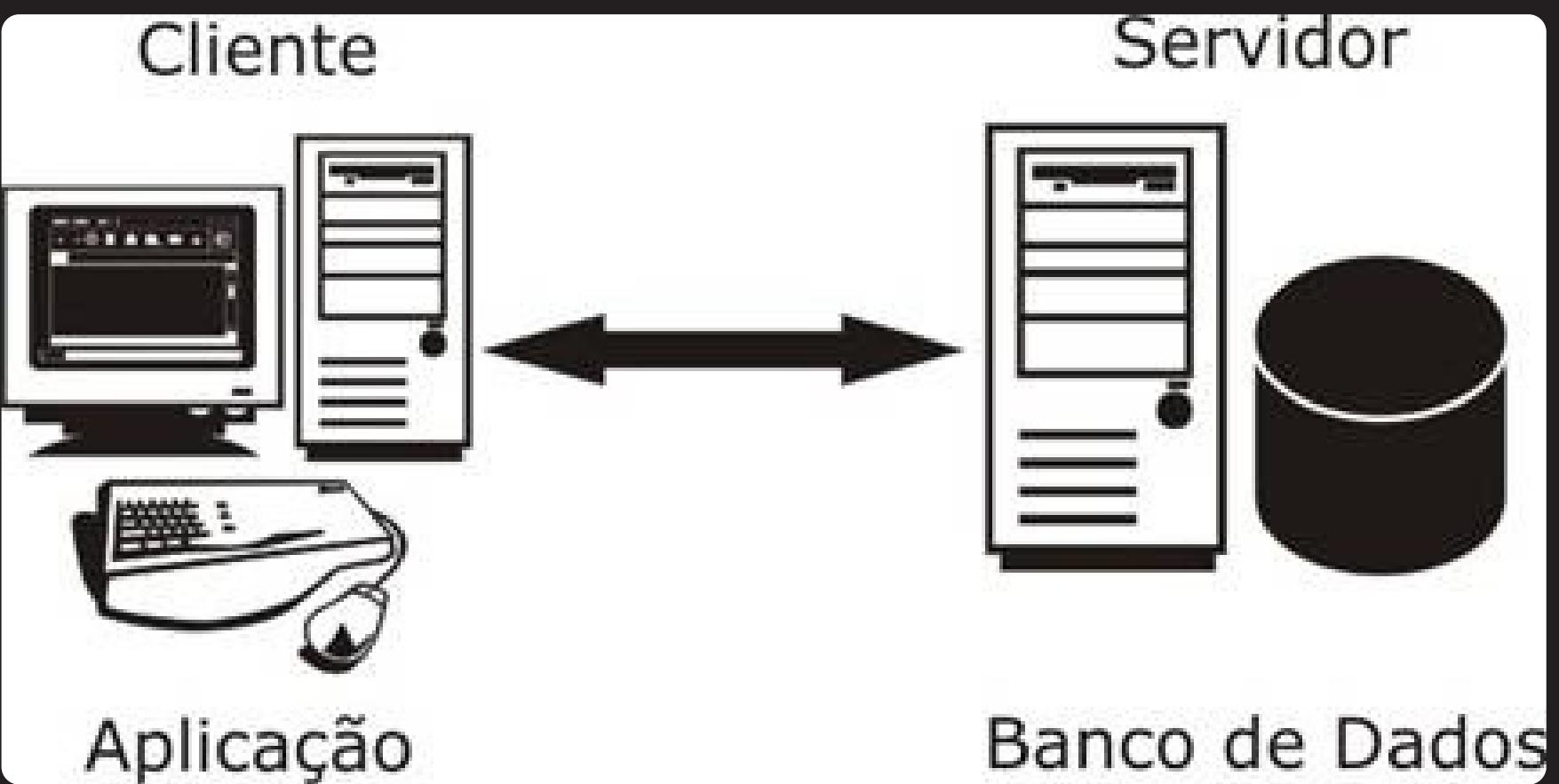
Elas são ótimas para um único usuário ou para um pequeno número de usuários que não precisam realizar manipulações de dados muito complicadas



# Banco de Dados

**Um banco de dados é uma coleção organizada de informações, ou dados, estruturados, geralmente armazenados eletronicamente em um sistema de computador.**

**Normalmente, um banco de dados é controlado por um Sistema de Gerenciamento de Banco de Dados (SGBD).**



# SQL

O SQL em Python refere-se à integração da linguagem de consulta estruturada (SQL) com a linguagem de programação Python. SQL é uma linguagem utilizada para gerenciar bancos de dados, permitindo a criação, modificação e consulta de dados armazenados em um banco de dados relacional. Ao combinar SQL com Python, os desenvolvedores podem aproveitar a capacidade do Python para manipular dados e a flexibilidade do SQL para realizar operações específicas em um banco de dados.

# SQL

Os dados nos tipos mais comuns de bancos de dados em operação atualmente são modelados em **linhas e colunas** em uma série de **tabelas** para tornar o processamento e a consulta de dados eficientes. Os dados podem ser facilmente acessados, gerenciados, modificados, atualizados, controlados e organizados.

A maioria dos bancos de dados usa a linguagem de consulta estruturada (SQL) para escrever e consultar dados.

# Tipos de Banco de Dados

01

## Bancos de dados relacionais

Os itens são organizados como um conjunto de tabelas com colunas e linhas.

02

## Bancos de dados orientados a objetos

As informações são representadas na forma de objetos, como na programação orientada a objetos.

03

## Bancos de dados distribuídos

Consiste em dois ou mais arquivos localizados em sites diferentes. Pode ser armazenado em vários computadores, localizados no mesmo local físico ou espalhados por diferentes redes.

04

## Data warehouses

Um repositório central de dados, um data warehouse é um tipo de banco de dados projetado para consultas e análises rápidas.

05

## Bancos de dados autônomos

São baseados em nuvem e usam machine learning para automatizar tarefas de gerenciamento de rotina executadas por administradores de banco de dados.

06

## Bancos de dados NoSQL

Banco de dados não relacional permite que dados não estruturados e semiestruturados sejam armazenados e manipulados (em contraste com um banco de dados relacional, que define como todos os dados inseridos no banco de dados devem ser compostos).



# Tipos de Banco de Dados

07

## Bancos de dados gráficos

Os bancos de dados gráficos armazenam estruturas de dados complexas, que seriam incompatíveis em uma base tradicional. É ideal para lidar com dados altamente interconectados.

09

## Banco de dados de documentos/JSON

Projetado para armazenamento, recuperação e gerenciamento de informações orientadas a documentos, os bancos de dados de documentos são uma maneira moderna de armazenar dados no formato JSON, em vez de linhas e colunas.

08

## Banco de dados multimodelo

Combinam diferentes tipos de modelos de banco de dados em um back-end único e integrado. Isso significa que eles podem acomodar vários tipos de dados.

10

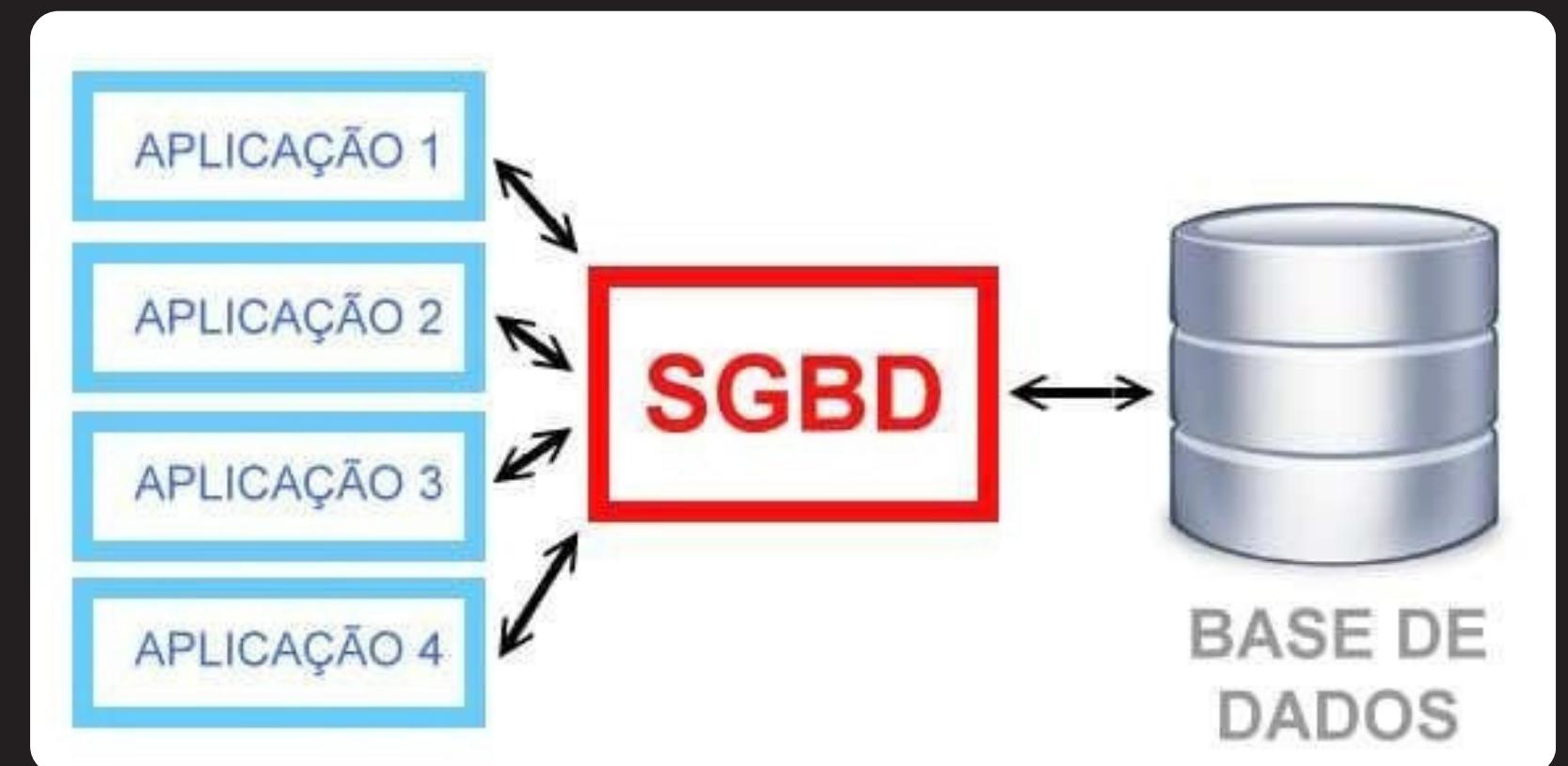
## Bancos de dados em nuvem

É uma coleção de dados, estruturados ou não estruturados, que residem em uma plataforma de computação em nuvem privada, pública ou híbrida.



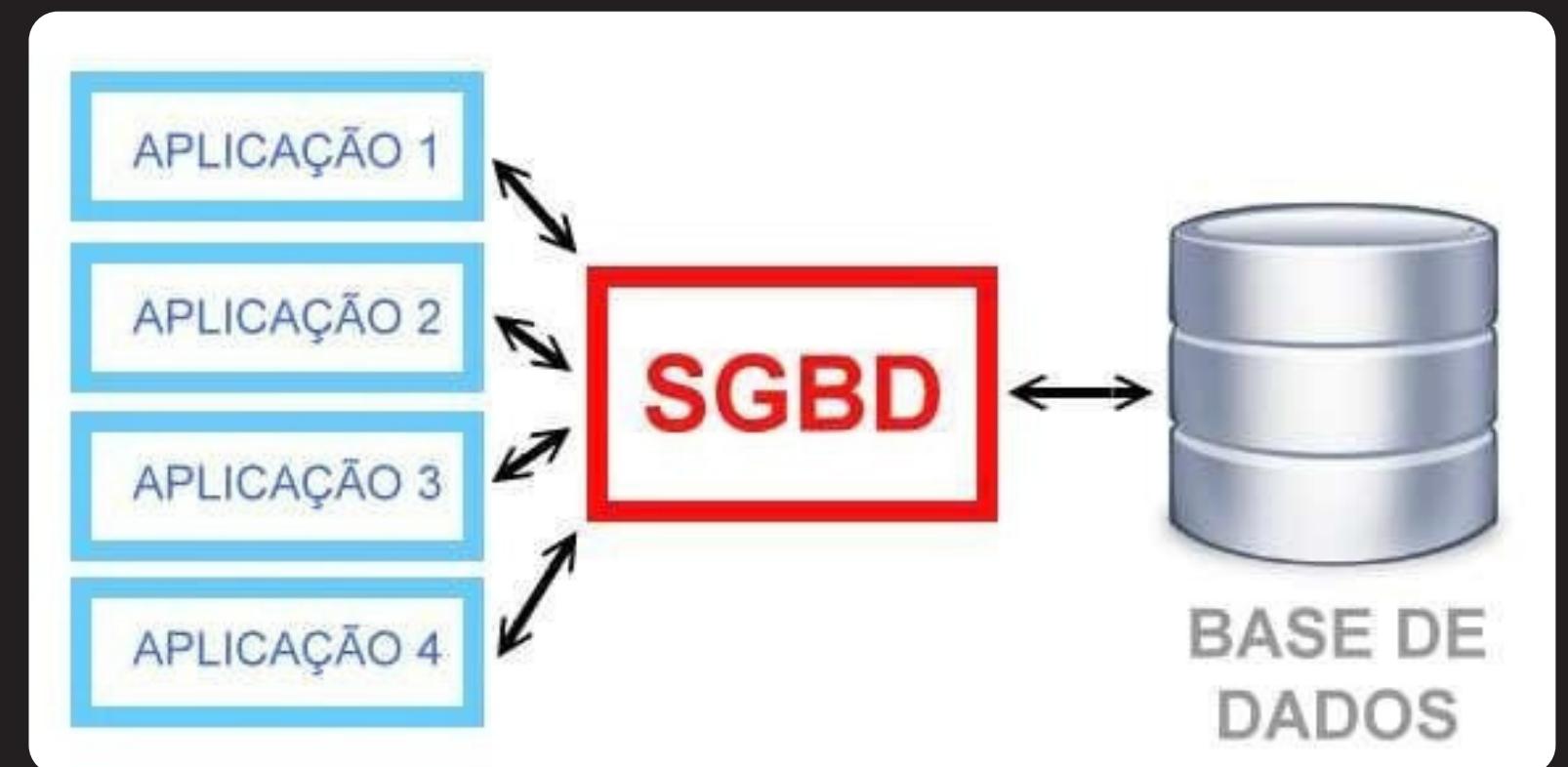
# SGBDS

Um banco de dados normalmente requer um programa abrangente de banco de dados, conhecido como sistema de gerenciamento de banco de dados (SGBD). Um SGBD serve como uma interface entre o banco de dados e seus usuários finais ou programas, permitindo que os usuários recuperem, atualizem e gerenciem como as informações são organizadas e otimizadas.



# SGBDS

Um SGBD também facilita a supervisão e o controle de bancos de dados, permitindo uma variedade de operações administrativas, como monitoramento de desempenho, ajuste e backup e recuperação. Alguns exemplos de softwares de bancos de dados populares ou SGBD incluem MySQL, Microsoft Access, Microsoft SQL Server, FileMaker Pro, Oracle Database e dBASE.



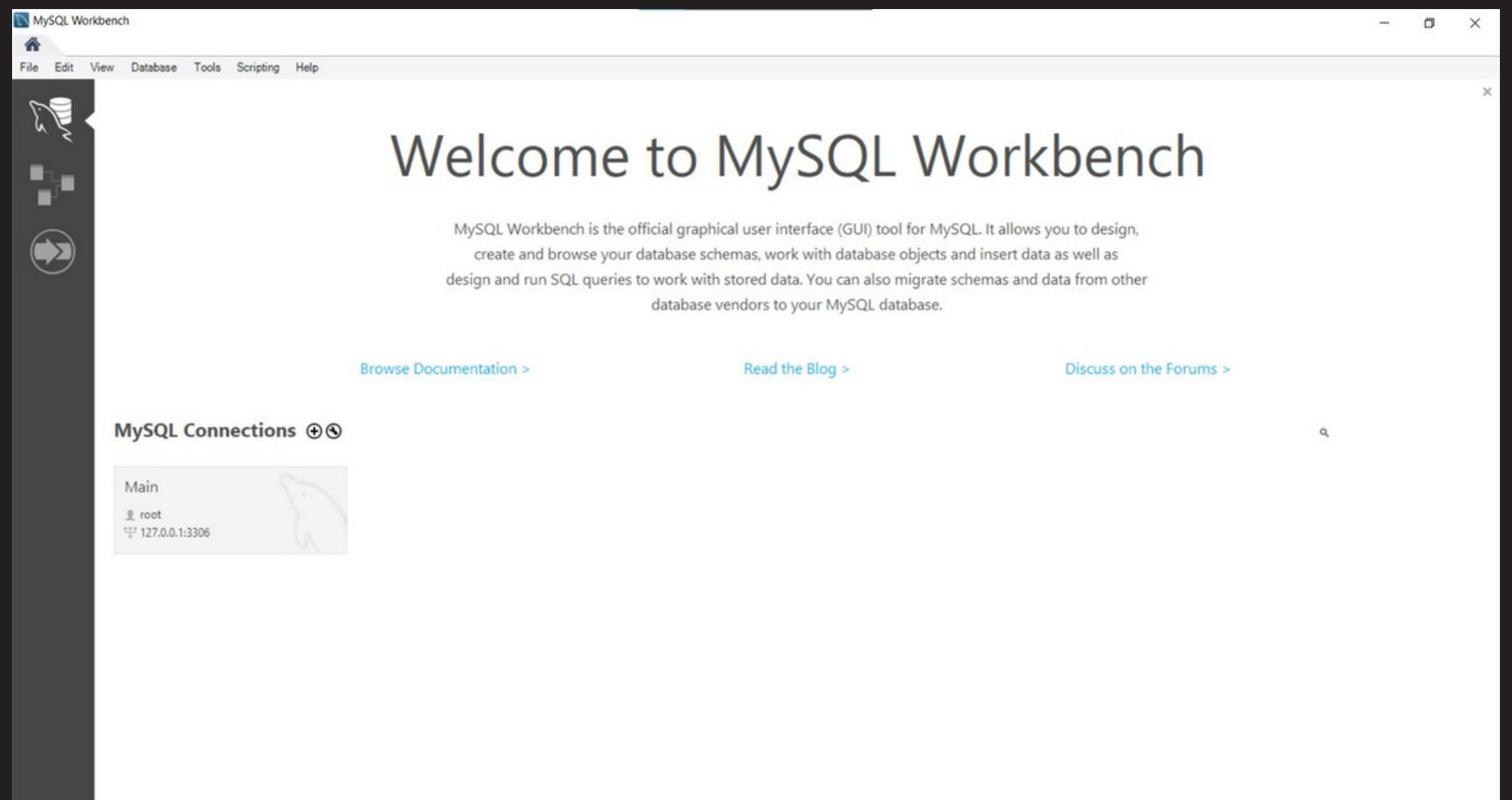
# MYSQL

O MySQL é o SGBD por trás de alguns dos principais sites e aplicativos baseados na web do mundo, incluindo Uber, LinkedIn, Facebook, Twitter e YouTube. O MySQL Workbench é uma ferramenta visual de design de banco de dados que integra desenvolvimento , administração , design, criação e manutenção de banco de dados SQL em um único ambiente de desenvolvimento integrado para o sistema de banco de dados MySQL.



# MYSQL

O MySQL Workbench é uma ferramenta gráfica para desenvolvedores e administradores de bancos de dados MySQL. Ele permite criar, gerenciar e consultar bancos de dados, além de oferecer recursos para modelagem de dados, administração de usuários e migração de dados. Em resumo, é uma ferramenta abrangente para trabalhar com MySQL.



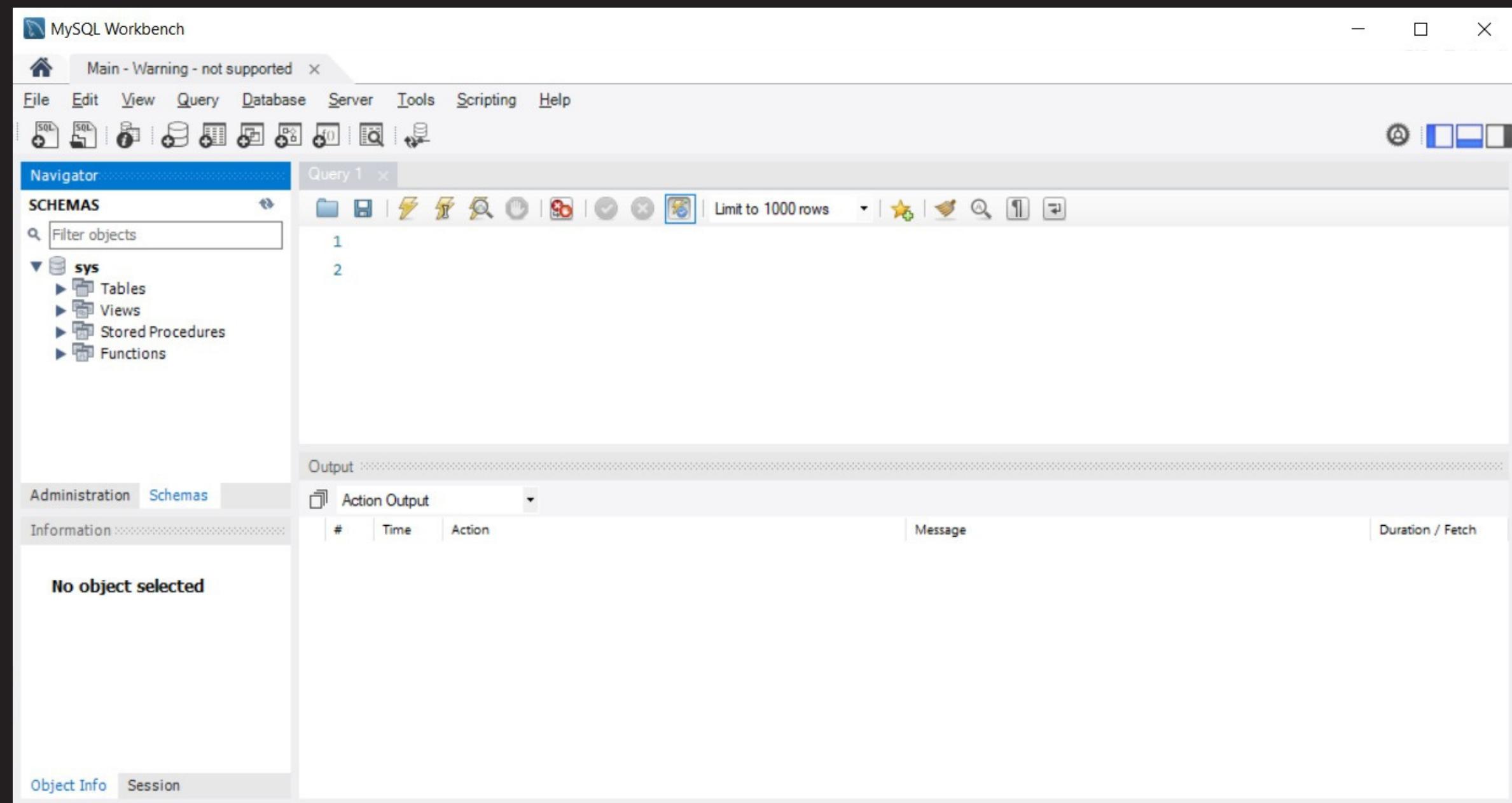
# MYSQL

O workbench configura um servidor na nossa máquina onde podemos criar bancos de dados, executar códigos SQL, entre outras funcionalidades. Essa conexão é chamada de "Local Instance".

O usuário por padrão é o usuário root.

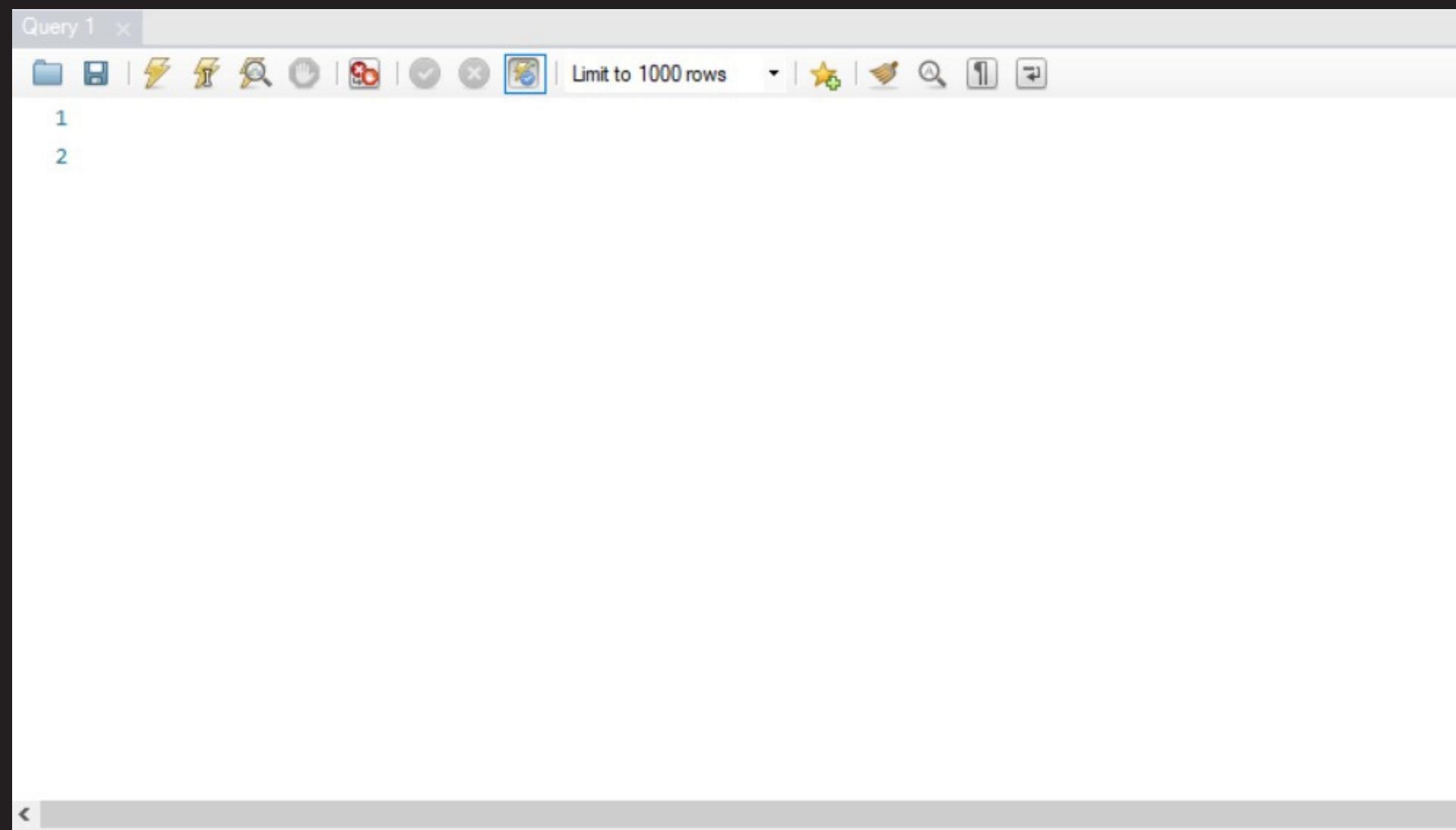
Esse servidor roda no nosso localhost, na porta 3306 por padrão.

# MYSQL



Ao clicar nessa conexão e logar pelo usuário root, vemos essa tela

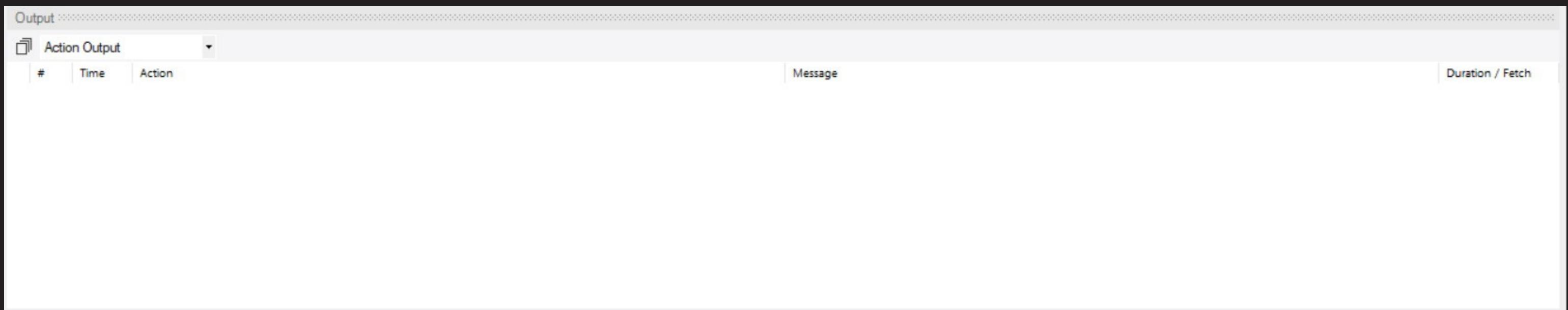
# MYSQL



Na parte central da tela temos o ambiente onde escreveremos os código SQL, salvaremos e executaremos

# MYSQL

Na parte inferior temos as saídas, podemos visualizar o resultado da execução dos códigos SQL por exemplo.

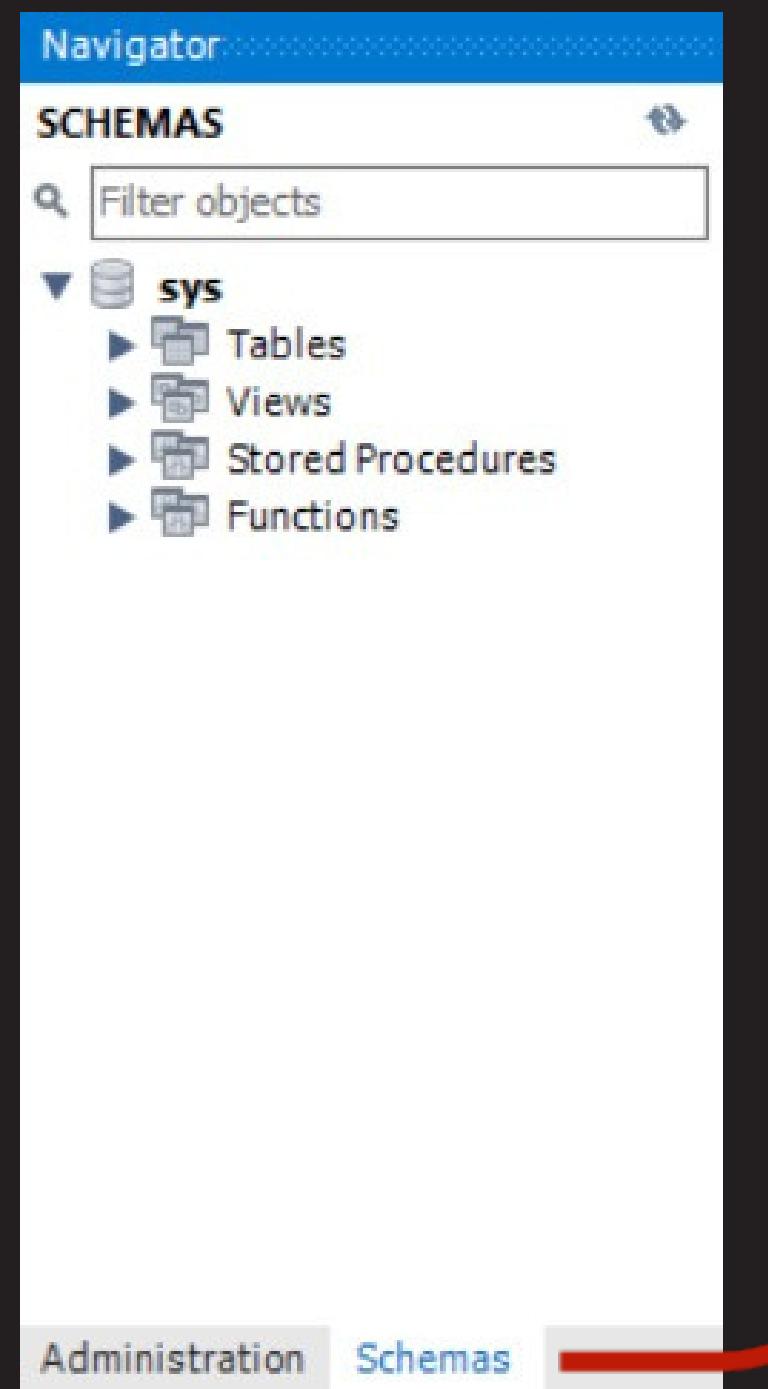


# MYSQL



No lado esquerdo, temos um menu de navegação com algumas opções importantes para o gerenciamento, performance e configuração. Na parte inferior termos duas abas, "Administration" e "Schemas".

# MYSQL

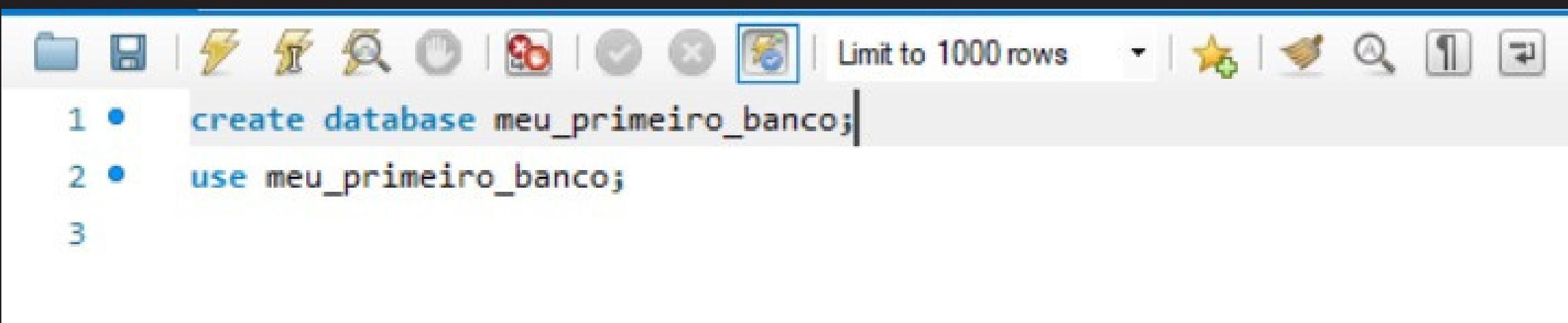


Na aba "Schemas" temos os bancos de dados criados. Dentro de cada Schema conseguimos ver as tabelas, as views, as stored Procedures e as Functions.

# Criando um Banco de Dados

## Primeiro comando SQL

Para criar nosso primeiro banco de dado (schema), utilizamos o comando:  
**create database nome\_do\_banco.**

A screenshot of the MySQL Workbench interface. The toolbar at the top has various icons for database management. Below the toolbar, there is a query editor window. The query editor contains three numbered lines of SQL code:

```
1 • create database meu_primeiro_banco;
2 • use meu_primeiro_banco;
3
```

The first line, 'create database meu\_primeiro\_banco;', is highlighted with a blue selection bar. The second line, 'use meu\_primeiro\_banco;', is also partially visible. The third line is empty.

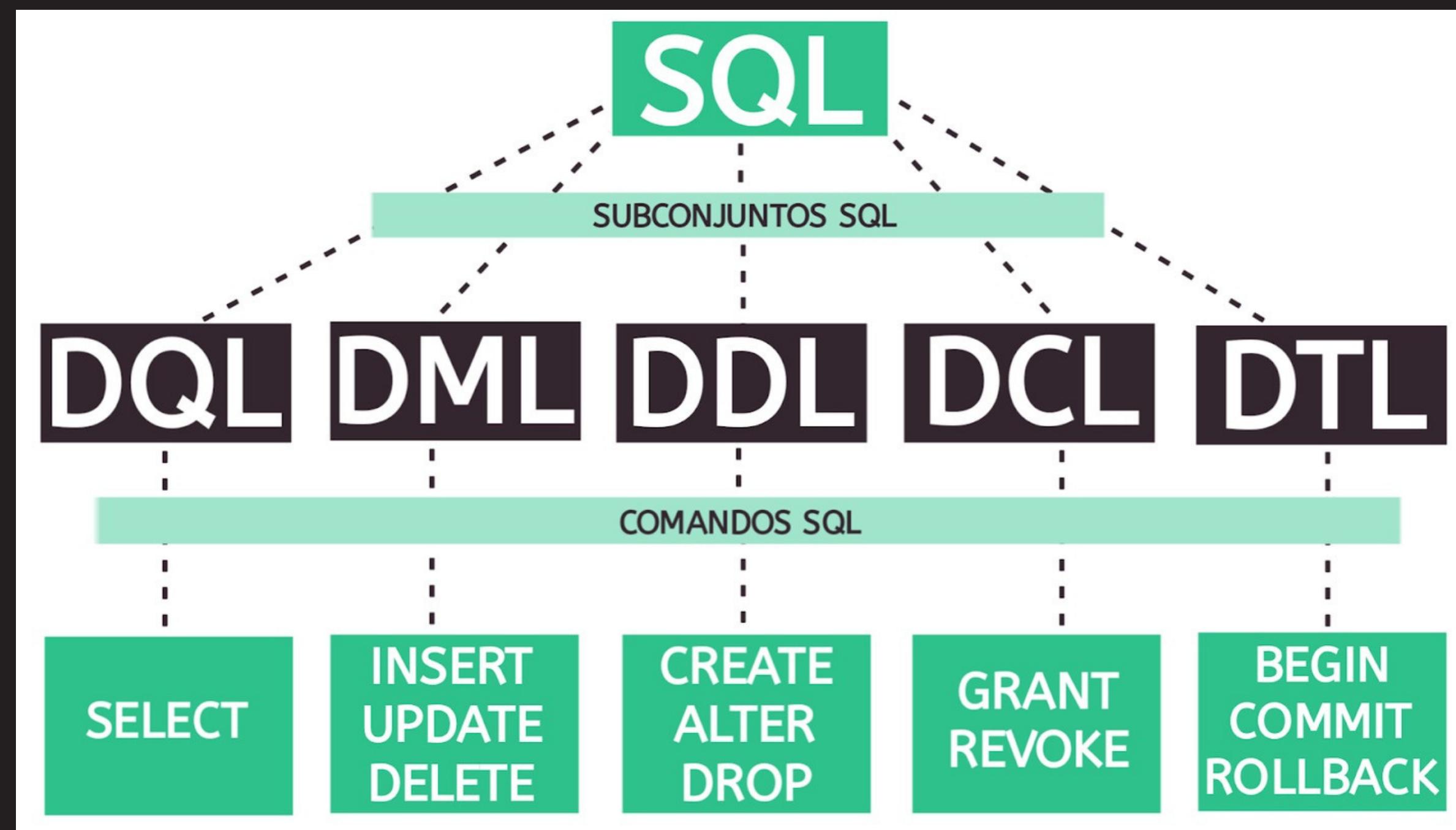
Lembrando que esse comando irá apenas criar o banco de dados, para utilizarmos um banco de dados criado, utilizamos: **use nome\_do\_banco**

# Comandos SQL

o SQL é uma linguagem para se trabalhar com banco de dados relacionais.

Com o SQL podemos gerenciar os nossos dados em um banco de dados através das querys, ou seja, através das requisições. Vamos entender que temos diferentes grupos de comandos para fazer diferentes requisições ao SGBD. Cada grupo de comando também será responsável por uma letra do **CRUD**

# Comandos SQL



IN

# Tipos de Comandos

01

## **CREATE**

Criar ou adicionar novas entradas

02

## **READ (RETRIEVE)**

Ler, recuperar ou ver entradas existentes

03

## **UPDATE**

Atualizar ou editar entradas existentes

04

## **Delete (Destroy)**

Remover entradas existentes

IN

# Comandos SQL

Os comandos DDL (Data Definition Language) são estruturais. Eles servem para definir aspectos da estrutura do banco de dados, como criar tabelas, excluir tabelas, alterar tabelas e mais. São mais focados em um aspecto amplo, e menos nos dados necessariamente.

Ou seja, esses comandos servirão para a estrutura do banco de dados e **não para os dados**.

# Comandos SQL

Os comandos DDL normalmente não são usados por um usuário geral, pois, nesse cenário, um usuário seria capaz de mexer na estrutura do banco de dados.

Fazem parte dos comandos DDL: **CREATE, DROP, ALTER, TRUNCATE, COMMENT, RENAME**

# Comandos SQL - CREATE



```
1 CREATE DATABASE IF NOT EXISTS name_db  
2  
3 -- Sintaxe Básica  
4  
5 -- O banco será criado se não existir um  
6 -- banco de dados com o mesmo nome
```

Existem duas instruções CREATE disponíveis no SQL:

- CREATE DATABASE
- CREATE TABLE

O comando CREATE DATABASE irá criar um novo banco de dados

# Comandos SQL - CREATE

Com o banco de dados criado, precisamos criar as tabelas, para armazenar os dados.

O comando **CREATE TABLE** é usado para criar uma tabela. Sabemos que uma tabela é composta por linhas e colunas, portanto, ao criar tabelas, precisamos fornecer todas as informações ao SQL sobre os nomes das colunas, tipos de dados a serem armazenados nas colunas, tamanho dos dados, etc.



```
1 CREATE TABLE nome_tabela(  
2     coluna_01 tipo_dado(tamanho)  
3     coluna_02 tipo_dado(tamanho)  
4     coluna_03 tipo_dado(tamanho)  
5 )  
6 -- Sintaxe Básica  
7  
8 CREATE TABLE Usuario(  
9     id int,  
10    nome varchar(40) -- Tamanho máximo  
11    esta_ativo boolean  
12 )  
13  
14 -- Exemplo
```

# Comandos SQL - DROP

O comando DROP é utilizado para excluir um banco de dados inteiro ou apenas uma tabela.

A instrução DROP destrói os objetos como um banco de dados, tabela, índice ou visão.



- 1 `DROP TABLE nome_tabela`
- 2 `DROP DATABASE nome_bd`

# Comandos SQL - TRUNCATE

A instrução TRUNCATE que é usada para remover os dados de uma tabela para fins de deslocação (vazia para reutilização).

O resultado dessa operação remove rapidamente todos os dados de uma tabela, geralmente ignorando vários mecanismos de imposição de integridade.

A instrução TRUNCATE TABLE é logicamente equivalente à instrução DELETE FROM (sem a cláusula WHERE).



```
1 TRUNCATE TABLE nome_tabela
2
3 -- Ao executar a comando acima,
4 -- a tabela será truncada ou seja,
5 -- os dados serão excluídos, mas a
6 -- estrutura permanecerá na memória
7 -- para operações posteriores
```

# Comandos SQL - TRUNCATE X DROP

Truncar preserva a estrutura da tabela para uso futuro, ao contrário de DROP TABLE onde a tabela é excluída com sua estrutura completa.

A exclusão da tabela ou banco de dados usando a instrução DROP não pode ser revertida, portanto, deve ser usada com cautela.

# Resumo da aula

Um **banco de dados** é uma coleção organizada de informações armazenadas eletronicamente. Ele oferece a capacidade de armazenar, recuperar, atualizar e gerenciar dados de maneira eficiente. São usados em muitos aplicativos, para armazenar e acessar informações de forma confiável e segura.

**SQL (Structured Query Language)** é uma linguagem de programação usada para gerenciar, consultar e manipular bancos de dados relacionais. Ela oferece comandos para criar, recuperar, atualizar e excluir dados, tornando-se uma ferramenta essencial para trabalhar com sistemas de gerenciamento de banco de dados.

# Resumo da aula

MySQL é um sistema de gerenciamento de banco de dados relacional (RDBMS) de código aberto amplamente utilizado. Ele permite armazenar, gerenciar e recuperar dados de maneira eficiente. Principais características incluem suporte a SQL, escalabilidade, alta velocidade e robustez.

Os comandos SQL (Structured Query Language) servem para interagir com bancos de dados relacionais. Eles desempenham um papel fundamental na manipulação e gestão de dados em sistemas de gerenciamento de banco de dados (SGBD).

# ATIVIDADE PRÁTICA 1

Crie um banco de dados chamado "escola" e as seguintes tabelas:

Tabela "alunos" com colunas: id\_aluno, nome, idade.

Tabela "cursos" com colunas: id\_curso, nome\_curso, carga\_horaria. Tabela "matriculas" com colunas: id\_matricula , id\_aluno id\_curso, data\_matricula.

# ATIVIDADE PRÁTICA 2

Usando o comando TRUNCATE TABLE, exclua todos os dados da tabela "matriculas" sem excluir a estrutura da tabela.

# ATIVIDADE PRÁTICA 3

Usando o comando DROP DATABASE, exclua o banco de dados "escola". Certifique-se de que você tenha feito um backup dos dados, pois essa ação apagará todo o banco de dados.

# DESAFIO PRÁTICO

## Sistema de uma escola

Crie um banco de dados para um sistema de uma escola, esse banco de dados ficará responsável por persistir os dados sobre alunos, professores, turmas e disciplinas.

Para os alunos é importante que contenha um número de matrícula, o nome, a idade, e o endereço.

# DESAFIO PRÁTICO

## **Sistema de uma escola**

Para os professores, deverá conter um número de matrícula, nome, especialidade e endereço.

Para a turma deverá conter um identificador, horário de início e dia de semana.

Para disciplina é importante que contenha um identificador, nome e quantidade de aulas.

# Material Complementar

- Exploração: Não tenha medo de explorar e testar diferentes códigos. A experimentação é uma grande aliada da aprendizagem.
- Perguntas: Faça perguntas, seja curioso! Entender o "porquê" das coisas ajuda a consolidar o conhecimento.
- Revisão: Revise o que aprendeu, tente explicar para si mesmo ou para outras pessoas. Ensinar é uma ótima forma de aprender.

Prática: A prática leva à perfeição. Quanto mais exercícios fizer, mais fácil será lembrar e entender os conceitos.



# SE LIGA NO CONTEÚDO DA PRÓXIMA AULA!

AULA 11 DE PYTHON:  
SQL II



**INFINITY SCHOOL**  
VISUAL ART CREATIVE CENTER

# Comandos SQL - INSERT

Usado para adicionar novos registros a uma tabela. Você que fornece os valores deseja inserir nas colunas correspondentes.

```
1 INSERT INTO alunos (nome, idade) VALUES ("Maria Silva", 22);
2
3 INSERT INTO alunos (nome, idade) VALUES
4 ("Maria Silva", 22),
5 ("Ana Rodriges", 21),
6 ("Pedro Lima", 23);
7
8 INSERT INTO alunos (nome, idade)
9 SELECT nome, idade FROM alunos_antigos WHERE idade > 20;
10
```

Este comando insere dados na tabela "alunos" com base em uma consulta SELECT em outra tabela "alunos\_antigos" WHERE - onde, a idade é maior que 20.



# INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 10 - SQL I