

# Banco de Dados

## Laboratório 01

Prof<sup>a</sup> Cristina Verçosa Pérez Barrios de Souza

[cristina.souza@pucpr.br](mailto:cristina.souza@pucpr.br)





# Tópicos

- › SGBD
- › SQL
- › MySQL
- › Laboratório Prática Formativa
- › Laboratório Prática Somativa 1 (entrega em equipe)

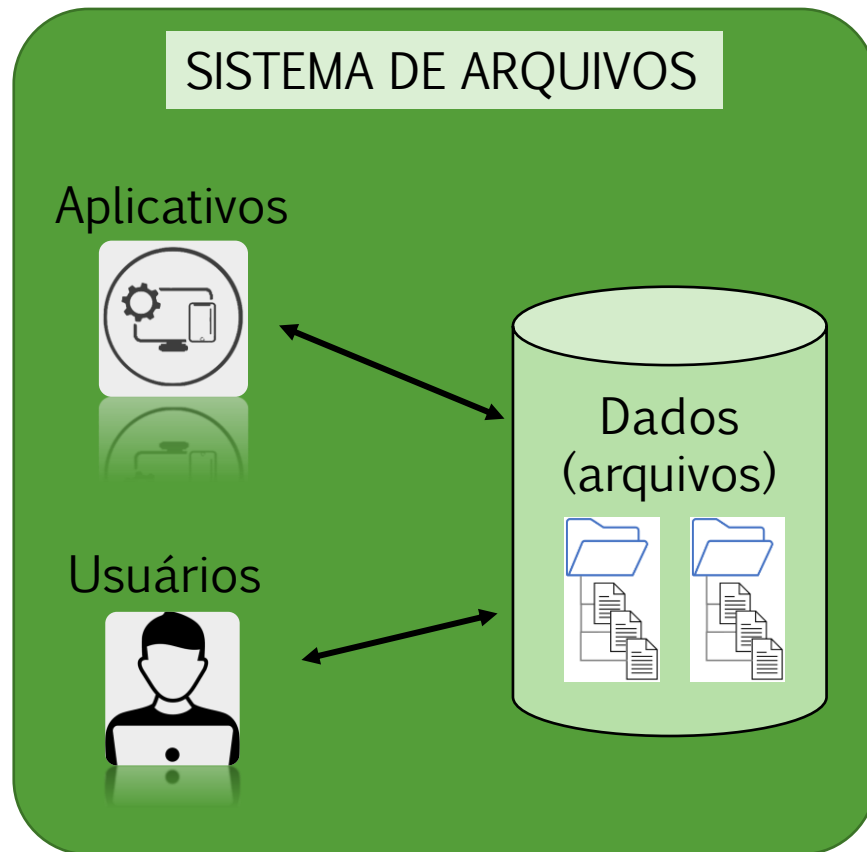
# SGBD

Sistema de Gerenciamento de  
Banco de Dados

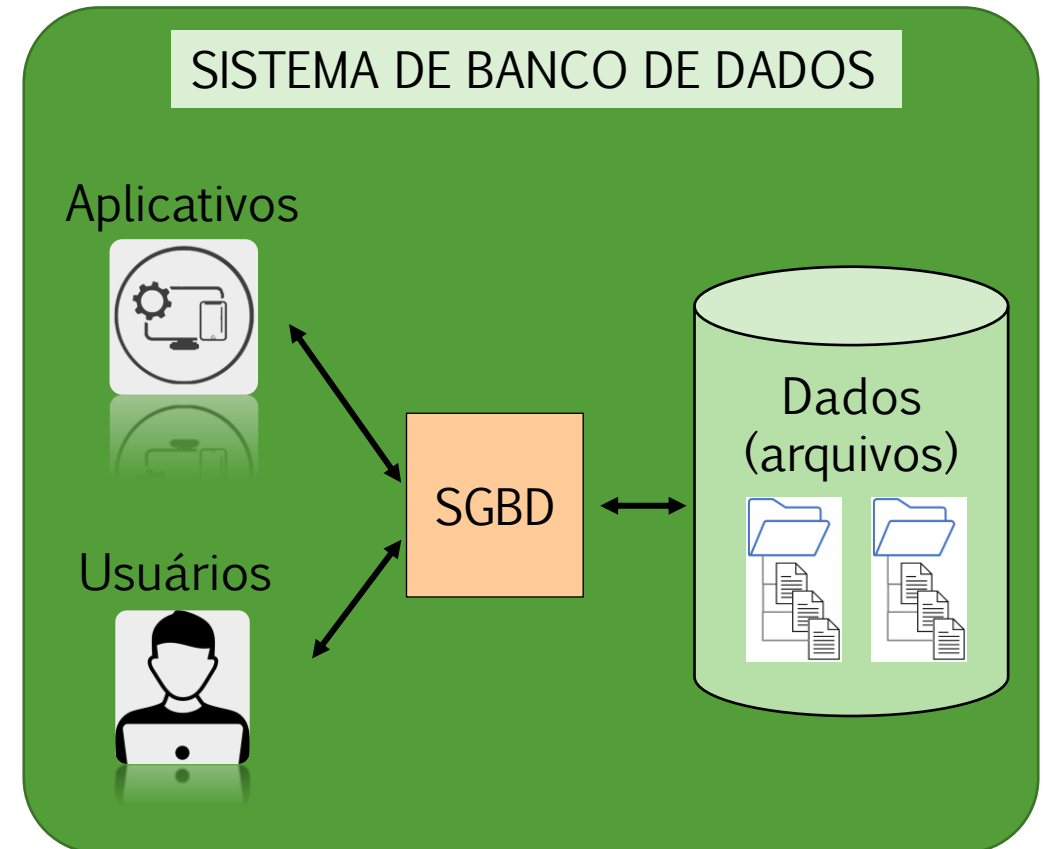




# SGBD vs. Sistema de Arquivos



O acesso/gerenciamento aos/dos dados é feito **diretamente** pelos aplicativos e/ou usuários

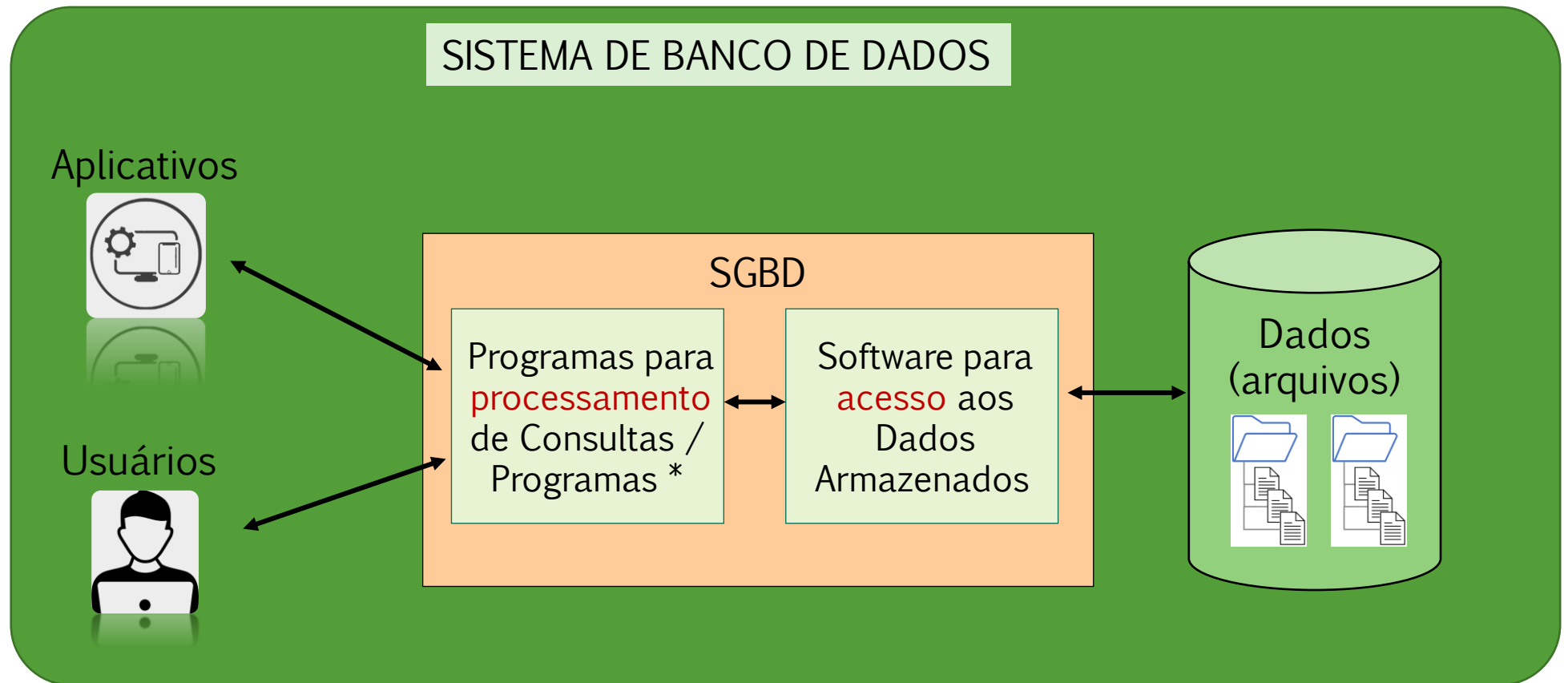


O acesso/gerenciamento aos/dos dados é feito pelo SGBD.

O **SGBD** faz a **intermediação** entre dados e usuários/aplicativos.



# SGBD – Configuração Simplificada



\* Programas em um SGBD podem ser:  
Procedimentos Armazenados (**Stored Procedures**) ou Funções (**Functions**)



# SGBD – Modelo de Dados

- › Conjunto de **ferramentas conceituais** usadas para
  - descrição de dados
  - relacionamento de dados
  - semânticas
  - regras de consistência
  
- › Dentre os diferentes modelos de dados, destaca-se o **Banco de Dados Relacional**
  - Baseado em **registros** (linhas ou tuplas)
  - Usado para **especificar a estrutura lógica do BD** e sua **implementação**



# SGBD – Modelo Relacional

› Representação dos dados em formas de tabelas, como conjunto de linhas e colunas:

- **Tuplas** = linhas ou registros
- **Atributos** = colunas ou campos
- **Domínios** = conjunto de valores válidos para um atributo

Nome_Cliente	Seguro_Social	Rua_Cliente	Cidade_Cliente	Número_Conta
Johnson	192-83-7465	Alma	Palo Alto	A-101
Smith	019-28-3746	North	Rye	A-125
Turner	182-173-6091	Putnam	Stamford	A-305

Diagram illustrating the relational model components:

- Atributo** (Attribute): Points to the column headers (Nome\_Cliente, Seguro\_Social, Rua\_Cliente, Cidade\_Cliente, Número\_Conta).
- Tupla** (Tuple): Points to the rows of data (Johnson, Smith, Turner).
- Domínio** (Domain): Points to the values within a specific attribute column (Seguro\_Social).

# SCRIPT SQL

*Structured Query Language*







# SQL – Structured Query Language

- › **Linguagem comercial** mais utilizada para **Banco de Dados**
  - Faz **consulta, definição, modificação** de dados e **restrições de segurança**
  - Desenvolvida pela IBM
  - Em 1986, torna-se padrão pela ANSI (*American National Standards Institute*) e ISSO (*International Organization for Standardization*)
  - Versão em uso ANSI/ISO é a SQL-2016
  - Última versão ANSI/ISO é a SQL-2016
  
- › Todo produto **SGDB** implementa a **SQL**



# SQL – Linguagem de Banco de Dados

- › **DQL** - *Data Query Language*

- Comando para consulta de dados: **SELECT**

- › **DDL** – *Data Definition Language*

- Comandos da SQL para criar (**CREATE**), alterar (**ALTER**) e eliminar (**DROP**) a estrutura de dados

- › **DML** – *Data Manipulation Language*

- Comandos SQL para realizar inserção (**INSERT**), remoção (**DELETE**) e modificação (**UPDATE**) de dados

- › Referência: [https://www.w3schools.com/mysql/mysql\\_drop\\_db.asp](https://www.w3schools.com/mysql/mysql_drop_db.asp)



# SQL – Sintaxe Resumida

- › Criação de base de dados (DDL)

**CREATE DATABASE** nome\_database ;

- › Eliminação da estrutura completa da base de dados (DDL)

**DROP DATABASE** nome\_database;

**DDL = *Data Definition Language***



# SQL – Sintaxe Resumida

- › Criação de tabela(**DDL**)

```
CREATE TABLE nome_tabela (  
    col1        TIPO_DADO,  
    col2        TIPO_DADO,  
    col3        TIPO_DADO,  
);
```

- › Eliminação da estrutura completa da base de dados (**DDL**)

```
DROP TABLE nome_tabela ;
```

**DDL = Data Definition Language**



# SQL – Sintaxe Resumida

## › Seleção de Dados (**DML - Data Manipulation Language**)

-- Retorna todas as linhas e colunas da tabela

```
SELECT * FROM tabela ;
```

-- Retorna todas as linhas das colunas col1 e col2 que atendem à condição1  
-- da tabela, ordenadas alfabeticamente, de acordo com a col1.

```
SELECT    col1, col2  
FROM      tabela  
WHERE     condição1  
ORDER BY col1;
```



# SQL – Sintaxe Resumida

› Inserção de Dados (**DML**)

```
INSERT INTO tabela (col1, col2)  
VALUES (valor_col1, valor_col2);
```

*DML - Data Manipulation Language*



# SQL – Sintaxe Resumida

› Remoção de Dados (**DML**)

**DELETE FROM** tabela  
**WHERE** condição;



# SQL – Sintaxe Resumida

› Modificação de Dados (**DML**)

```
UPDATE tabela  
SET col1 = novo_valor  
WHERE condição;
```





# SQL – Sintaxe Resumida

## › Resumo SQL **DML** :

- Criação = **INSERT** → *Create*
- Consulta = **SELECT** → *Read*
- Atualização = **UPDATE** → *Update*
- Destruição = **DELETE** → *Delete*



# SERVIDOR MYSQL

Sistema de gerenciamento de  
banco de dados





# MySQL

- › SGBD Relacional
- › Criado na **Suécia** pelos suecos **David Axmark** e **Allan Larsson**, e pelo finlandês **Michael "Monty" Widenius** para a **MySQL AB**
- › Em 2008, a **Sun Microsystems** comprou a **MySQL AB**
- › Em 2009, a **Oracle Corporation** comprou a **Sun Microsystems**, após autorização da Comissão Europeia sobre a suspeita de formação de monopólios no mercado de SGBDs
- › O **sucesso do MySQL** deve-se em grande parte à **fácil integração com o PHP**, incluído, quase que obrigatoriamente, nos pacotes de hospedagem de sites da Internet oferecidos atualmente.
  - Empresas como **Yahoo! Finance**, **MP3.com**, **Motorola**, **NASA**, **Silicon Graphics** e **Texas Instruments** usam o **MySQL** em aplicações de **missão crítica**.
  - A **Wikipédia** é um exemplo de utilização do **MySQL** em sites de **grande audiência**.



# MySQL: Declarações SQL

## › Referência:

<https://dev.mysql.com/doc/refman/8.0/en/sql-statements.html>

- Declaração para Definição de Dados (DDL)
- Declaração para Manipulação de Dados (DML)

## › Referência:

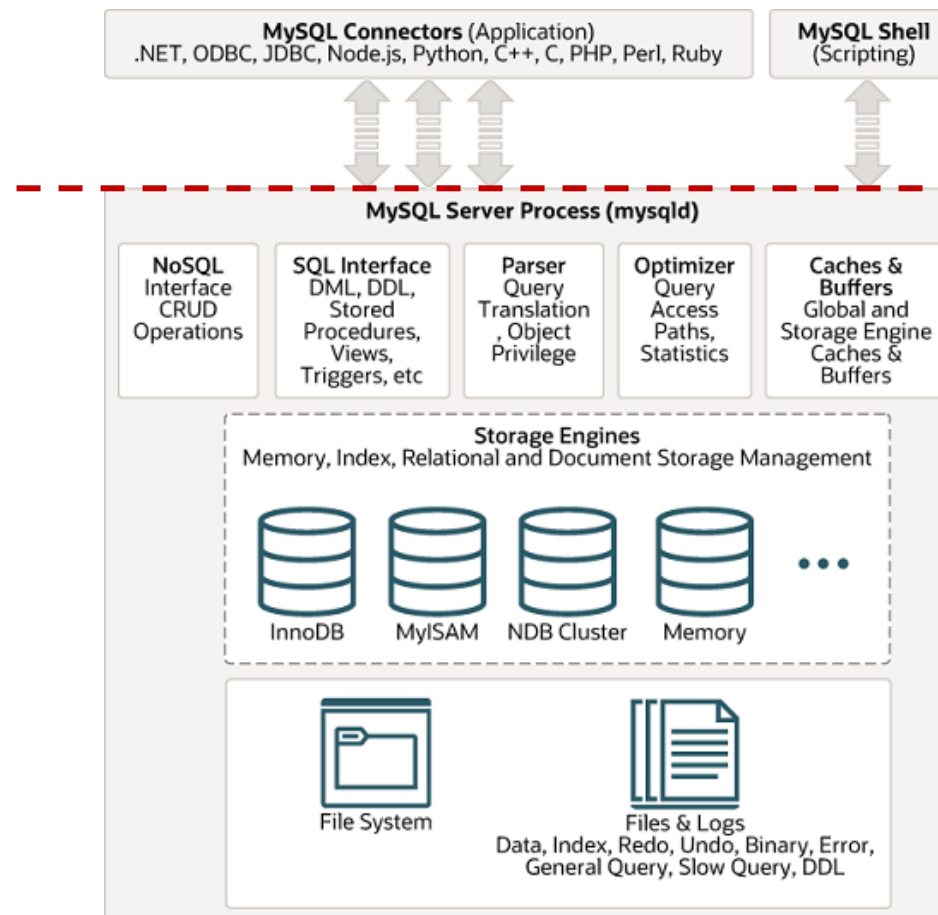
[https://www.w3schools.com/mysql/mysql\\_drop\\_db.asp](https://www.w3schools.com/mysql/mysql_drop_db.asp)



# MySQL Storage Engine Architecture

- › Arquitetura do mecanismo de armazenamento

- › Ref.: <https://dev.mysql.com/doc/refman/8.0/en/pluggable-storage-overview.html>





**NEW SKILLS**



Loading...

# Prática FORMATIVA

Trabalho INDIVIDUAL: realize os exercícios indicados.

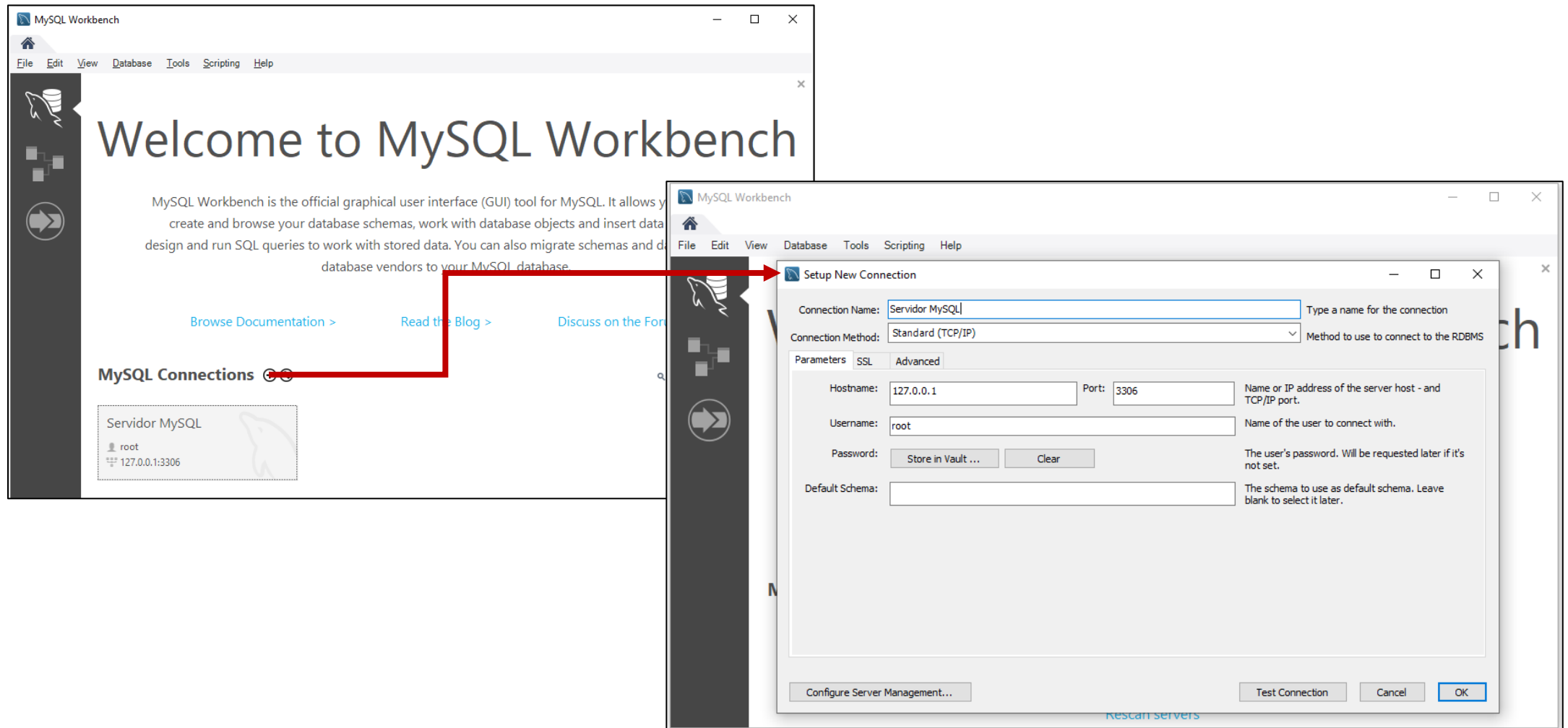


# MySQL Workbench

- › O **MySQL Workbench** é uma aplicação parte do banco de dados MySQL; é uma **interface administrativa completa** para executar consultas SQL e gerenciamento do MySQL.
- › Uma alternativa é o **phpMyAdmin**, também é uma **ferramenta administrativa** do MySQL, que é baseada na Web (executa em um navegador / browser).



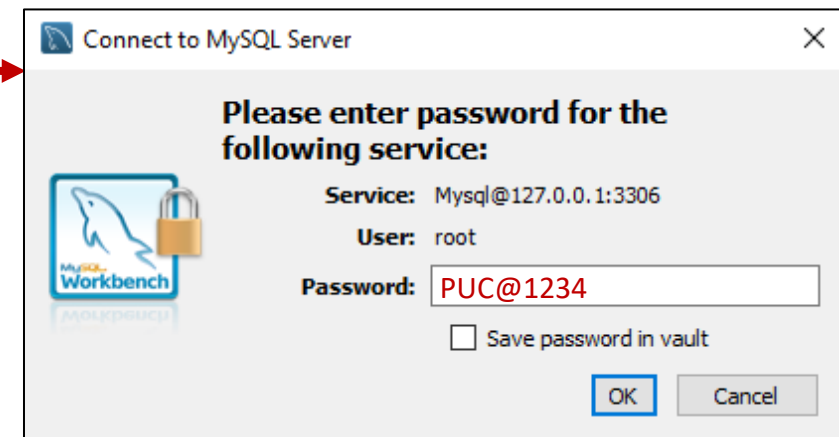
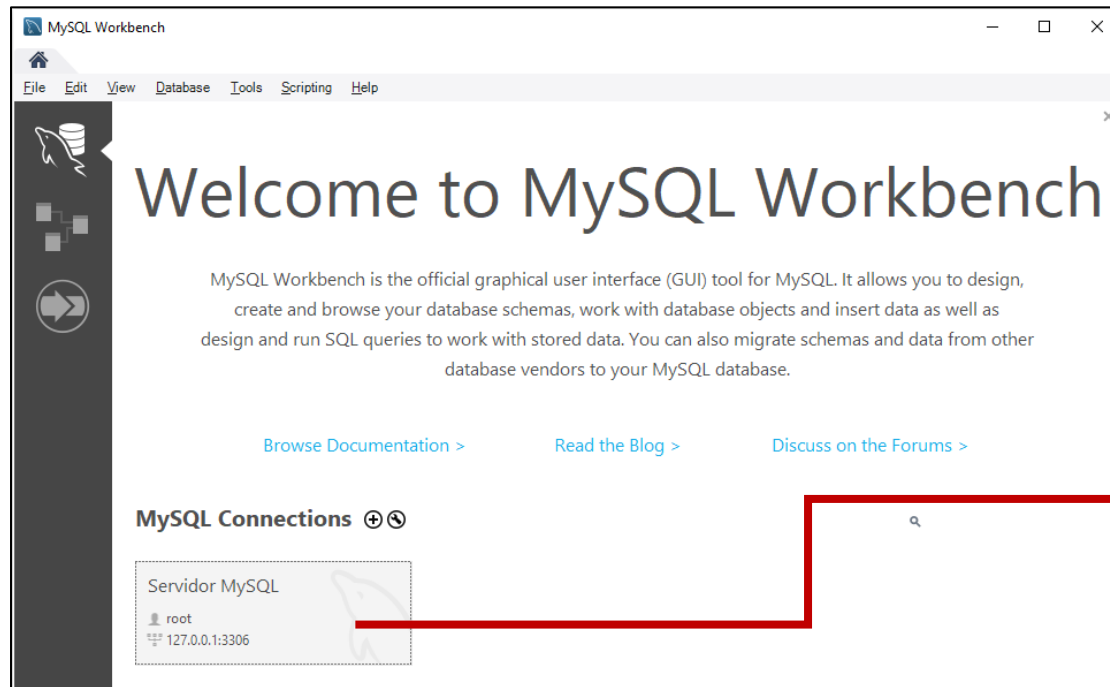
# MySQL Workbench





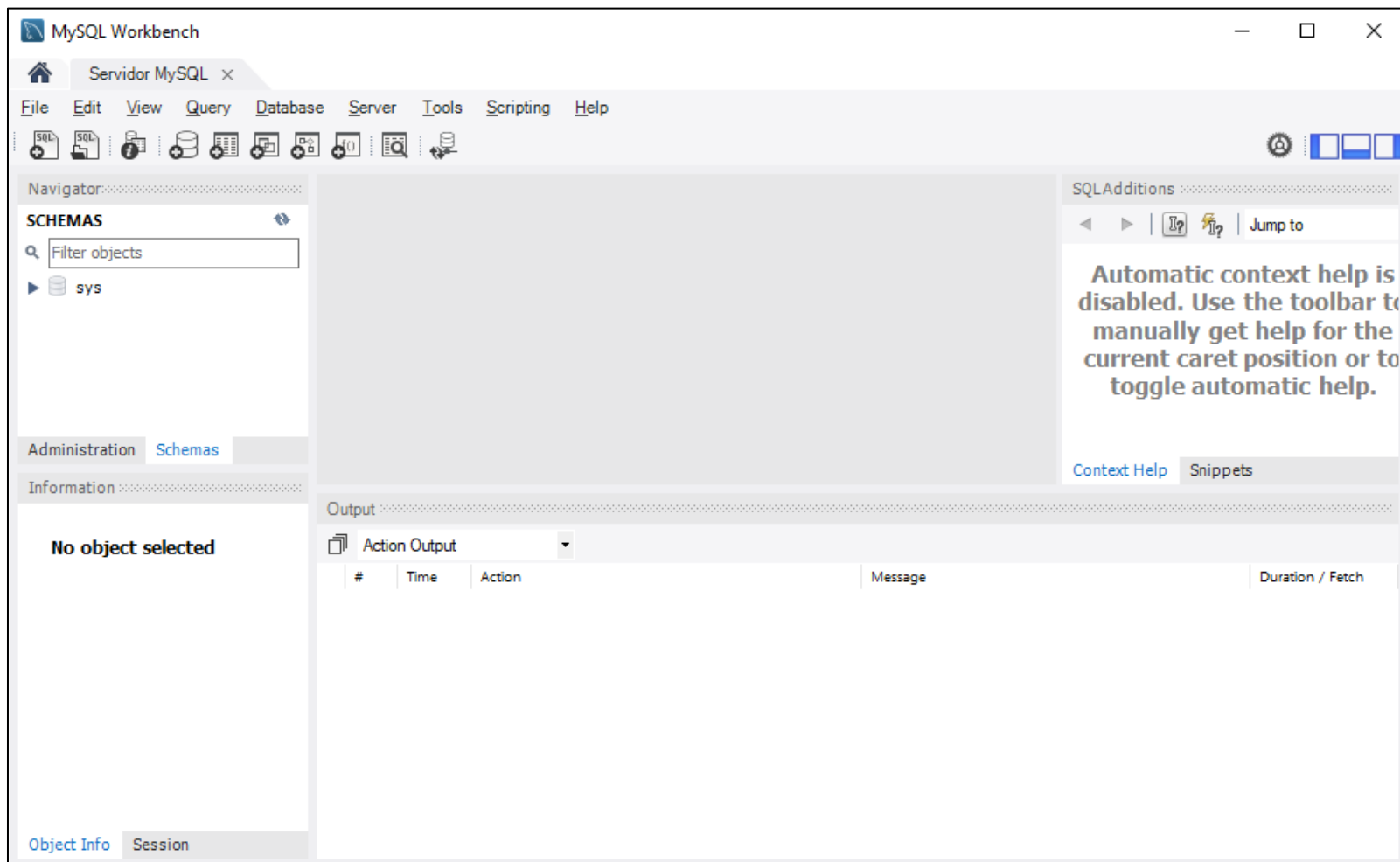


# MySQL Workbench





# MySQL Workbench





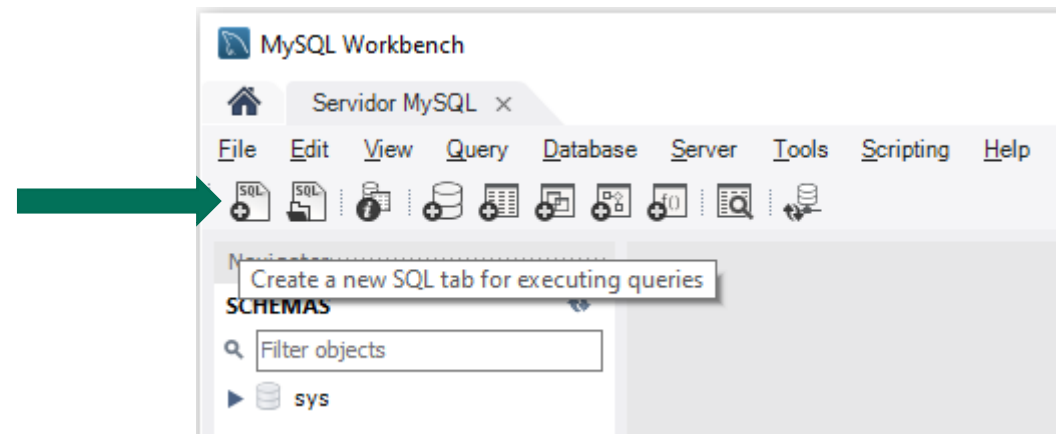
# Formativa: Criação de Database

## › CRIAÇÃO DE DATABASE

- A criação de uma base de dados e suas tabelas é, muitas vezes, um dos primeiros passos da parte prática do desenvolvimento de um novo projeto de software

## › UTILIZANDO O SQL

- Com o **MySQL Workbench** aberto, clique no botão **Nova Consulta (Create new SQL ...)**:





# Formativa: Criação de Database

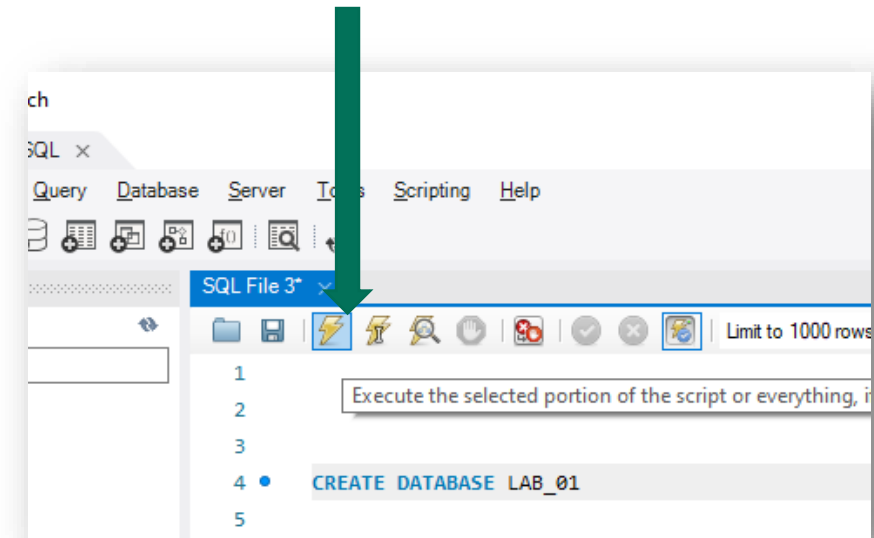
## CRIAÇÃO DE DATABASE

- › Em seguida, na parte central da interface aparecerá uma tela em branco, na qual você poderá digitar os códigos para criar a base de dados, tabelas, efetuar consultas, etc.
- › Use o código abaixo:

```
CREATE DATABASE LAB_01;
```

## EXECUTE O CÓDIGO

- › Clique em **Execute ...**, como mostra a Figura:





# Formativa: Criação de Tabela

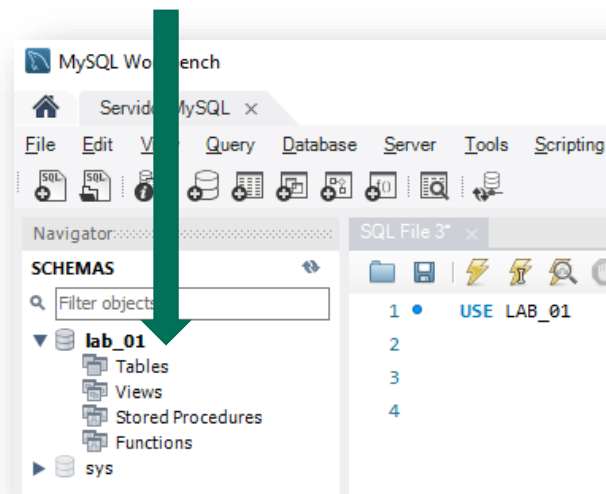
## MUDANDO O CONTEXTO

- › Com a base de dados criada, precisamos entrar no **contexto da nova base** para poder executar novos scripts nela.
- › Para isso, execute o seguinte comando:

```
USE LAB_01;
```

## VERIFICAÇÃO DE NOVO CONTEXTO

- › Verifique a utilização da base de dados escolhida no **MySQL Workbench**:





# Formativa: Criação de Tabela

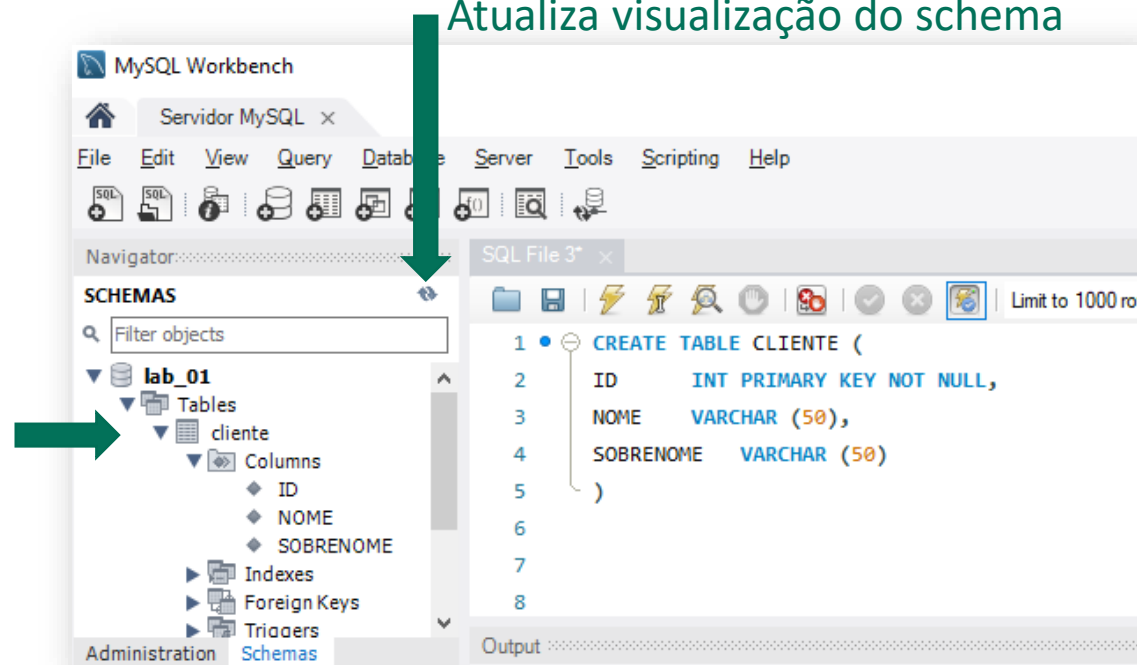
## SCRIPT PARA CRIAR TABELA

- › O próximo passo é criar a tabela.
- › Para isso, execute o seguinte comando:

```
CREATE TABLE CLIENTE (  
    ID INT PRIMARY KEY NOT NULL,  
    NOME VARCHAR (50),  
    SOBRENOME VARCHAR (50)  
);
```

## VERIFICAÇÃO DE NOVA TABELA

Atualiza visualização do schema





# Formativa: Inserção e Seleção de dados em Tabela

## SCRIPT

- › Comando para **Inserção** de dados:

```
INSERT INTO CLIENTE (ID, NOME, SOBRENOME)  
VALUES (1, 'José', 'Silva');
```

- › Comando para **Seleção** de dados:

```
SELECT * FROM CLIENTE;
```

## VERIFICAÇÃO

```
SQL File 3* x  
1 • INSERT INTO CLIENTE (ID, NOME, SOBRENOME)  
2 VALUES (1, 'José', 'Silva');  
3  
4 • SELECT * FROM CLIENTE  
5
```

ID	NOME	SOBRENOME
1	José	Silva
NULL	NULL	NULL



# Formativa: Múltipla Inserção de dados em Tabela

## SCRIPT

- › Comando para **Múltipla Inserção**:

```
INSERT INTO CLIENTE (ID, NOME,
SOBRENOME)
VALUES
(2, 'Maria', 'Alves'),
(3, 'Cláudia', 'Soares'),
(4, 'Pedro', 'Tomé'),
(5, 'Lucas', 'Antunes')
;

SELECT * FROM CLIENTE
```

## VERIFICAÇÃO

```
SQL File 3* x
Limit to 1000 rows
1 • INSERT INTO CLIENTE (ID, NOME, SOBRENOME)
2   VALUES
3     (2, 'Maria', 'Alves'),
4     (3, 'Cláudia', 'Soares'),
5     (4, 'Pedro', 'Tomé'),
6     (5, 'Lucas', 'Antunes') ;
7
8 • SELECT * FROM CLIENTE
9
```

	ID	NOME	SOBRENOME
▶	1	José	Silva
	2	Maria	Alves
	3	Cláudia	Soares
	4	Pedro	Tomé
	5	Lucas	Antunes
*	NULL	NULL	NULL





# Formativa: Alteração de dados em Tabela

## SCRIPT

- › Comando para **Alteração** de dados:

```
UPDATE CLIENTE
SET     NOME      = 'Selma',
        SOBRENOME = 'Silva'
WHERE ID = 3;

SELECT * FROM CLIENTE
```

## VERIFICAÇÃO

```
SQL File 3* x
1 • UPDATE CLIENTE
2   SET     NOME      = 'Selma',
3         SOBRENOME = 'Silva'
4   WHERE ID = 3;
5
6 • SELECT * FROM CLIENTE
```

	ID	NOME	SOBRENOME
▶	1	José	Silva
	2	Maria	Alves
	3	Selma	Silva
	4	Pedro	Tomé
	5	Lucas	Antunes
*	NULL	NULL	NULL



# Formativa: Exclusão de dados em Tabela

## SCRIPT

- › Comando para **Exclusão** de dados:

```
DELETE
  FROM CLIENTE
  WHERE ID = 3;

SELECT * FROM CLIENTE
```

## VERIFICAÇÃO

```
SQL File 3* x
1 • DELETE
2   FROM CLIENTE
3   WHERE ID = 3;
4
5 • SELECT * FROM CLIENTE
6
```

	ID	NOME	SOBRENOME
▶	1	José	Silva
	2	Maria	Alves
	4	Pedro	Tomé
	5	Lucas	Antunes
*	NULL	NULL	NULL



# RESUMO:

- › No **MySQL Workbench**, primeiro, aprendemos a
  - Criar Base de Dados e
  - Criar Tabela
- › Após, realizamos as 4 operações do **CRUD**:
  - Criação = INSERT → *Create*
  - Consulta = SELECT → *Read*
  - Atualização = UPDATE → *Update*
  - Destruição = DELETE → *Delete*





# Prática SOMATIVA

Trabalho EM EQUIPE :

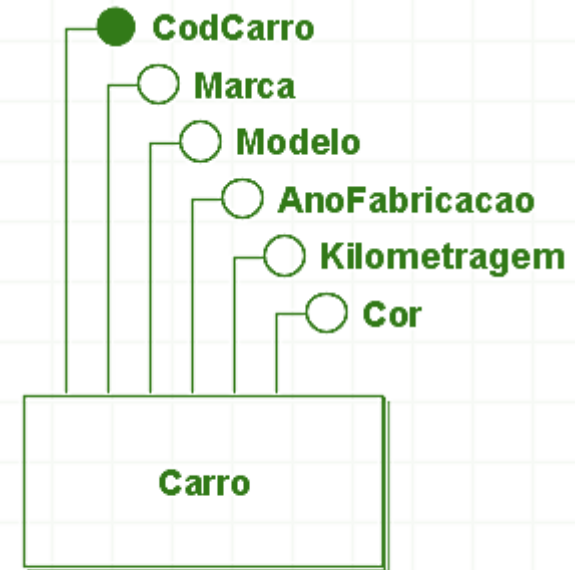
1. Realize os **exercícios indicados**.
2. Salve o **número** do exercício e seu **resultado** (**imagens** da prática realizada) em um arquivo **Word**.
3. Após todos os exercícios, **salve os exercícios em arquivo PDF**.
4. Entregue o **PDF**.



## Prática 1.1

- Na ferramenta **brModelo**, construa o seguinte **modelo entidade-relacionamento (MER)**, para armazenar um conjunto de **carros** em uma loja de carros usados.
- Apresente a **imagem** do seu modelo.

Diagrama **Conceitual** = MER





## Prática 1.2

- a) Ainda na ferramenta **brModelo**, transforme o **modelo entidade-relacionamento (MER)** para seu respectivo **modelo relacional**.
- b) Apresente a **imagem** do seu modelo.

Diagrama **Lógico** = Modelo Relacional

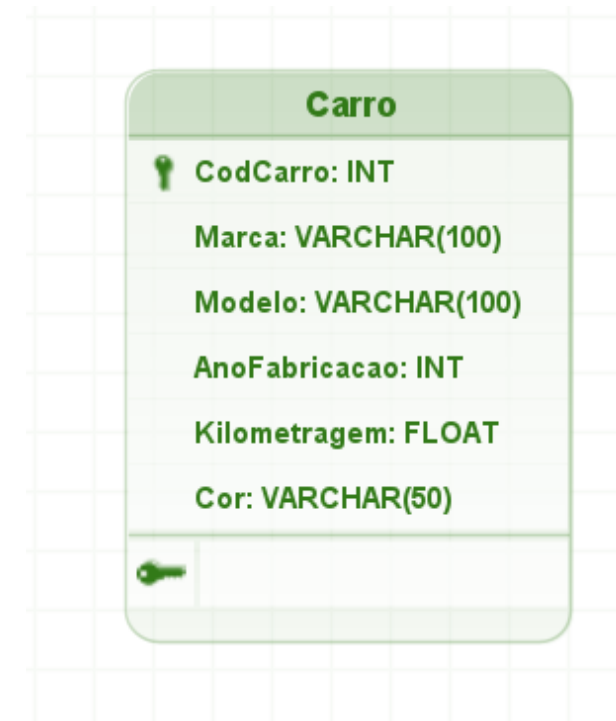




## Prática 1.3

- a) Defina os **tipos de dados** para cada atributo da entidade Carro.
- b) Apresente a **imagem** do seu modelo.

Diagrama **Lógico** = Modelo Relacional





## Prática 1.4

- a) Converta o **modelo relacional** no seu correspondente **modelo físico** (escreva os comandos **SQL DDL**).
- b) Apresente o **código SQL** de criação do seu modelo.

Modelo **Físico** = Comandos **SQL DDL**

```
CREATE TABLE .... (  
    CAMPO 1      TIPO DADO PRIMARY KEY,  
    CAMPO 1      TIPO DADO,  
    CAMPO 1      TIPO DADO,  
    ....  
);
```





## Prática 1.5

- a) Utilize o database **LAB\_01**.
- b) Crie a tabela do **Carro**, com o comando **SQL DDL** obtido no brModelo.
- c) Crie **um único comando** para realizar a **INSERÇÃO** de 5 carros.
- d) Execute o comando para a **INSERÇÃO** e exiba em uma **imagem** os **registros (linhas)** inseridos na tabela Carro com um comando **SQL SELECT**.



## Prática 1.6

- a) Crie **um comando** para realizar a **ALTERAÇÃO** da **cor** de um dos seus carros.
- b) Crie **outro comando** para realizar a **ALTERAÇÃO** do **ano de fabricação** de um outro carro.
- c) Execute os comandos para as **ALTERAÇÕES** e exiba em uma **imagem** os **registros (linhas)** alterados na tabela Carro com um comando **SQL SELECT**.



## Prática 1.7

- a) Crie **um comando** para realizar a **EXCLUSÃO** de um dos seus carros.
- b) Execute o comando para a **EXCLUSÃO** e exiba em uma **imagem** a tabela Carro **após** a **EXCLUSÃO**, com um comando **SQL SELECT**.



# RESUMO:

- › No **MySQL Workbench**, primeiro, aprendemos a
  - Criar Base de Dados e
  - Criar Tabela
- › Após, realizamos as 4 operações do **CRUD**:
  - Criação = INSERT → *Create*
  - Consulta = SELECT → *Read*
  - Atualização = UPDATE → *Update*
  - Destruição = DELETE → *Delete*





# Referência Bibliográfica

- › Sistema de Banco de Dados
  - Abraham Silberschatz, Henry F. Korth, S. Sudaarshan
  
- › Referência do SQL
  - Chapter 13 SQL Statements:  
<https://dev.mysql.com/doc/refman/8.0/en/sql-statements.html>
  - W3Schools: [https://www.w3schools.com/mysql/mysql\\_drop\\_db.asp](https://www.w3schools.com/mysql/mysql_drop_db.asp)
  
- › Documentação Técnica do MySQL
  - MySQL 8.0 Reference Manual
  - <https://docs.microsoft.com/pt-br/sql/sql-server/?view=sql-server-ver16>