

BANCO DE DADOS E PHP – PARTE 01

PROF. FÁBIO KRAVETZ

DESENVOLVIMENTO DE SISTEMAS

1º SEMESTRE/2024

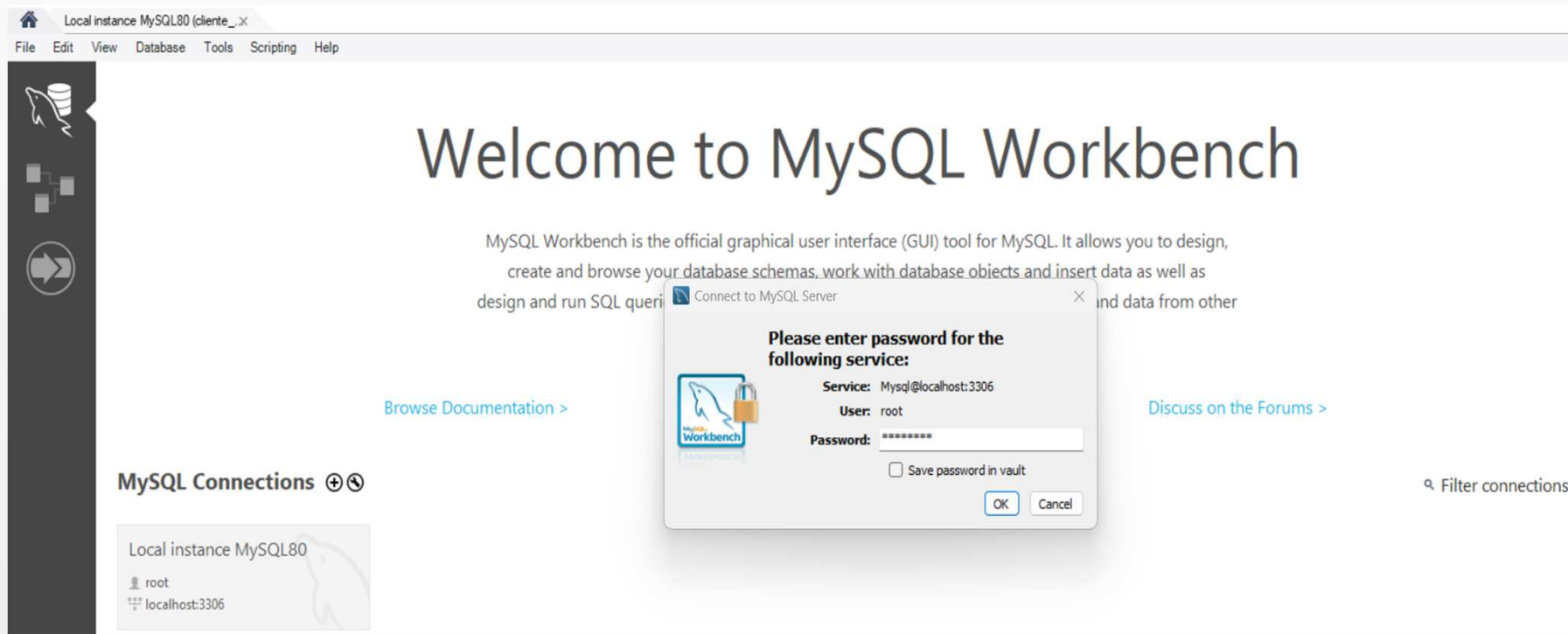


MySQL

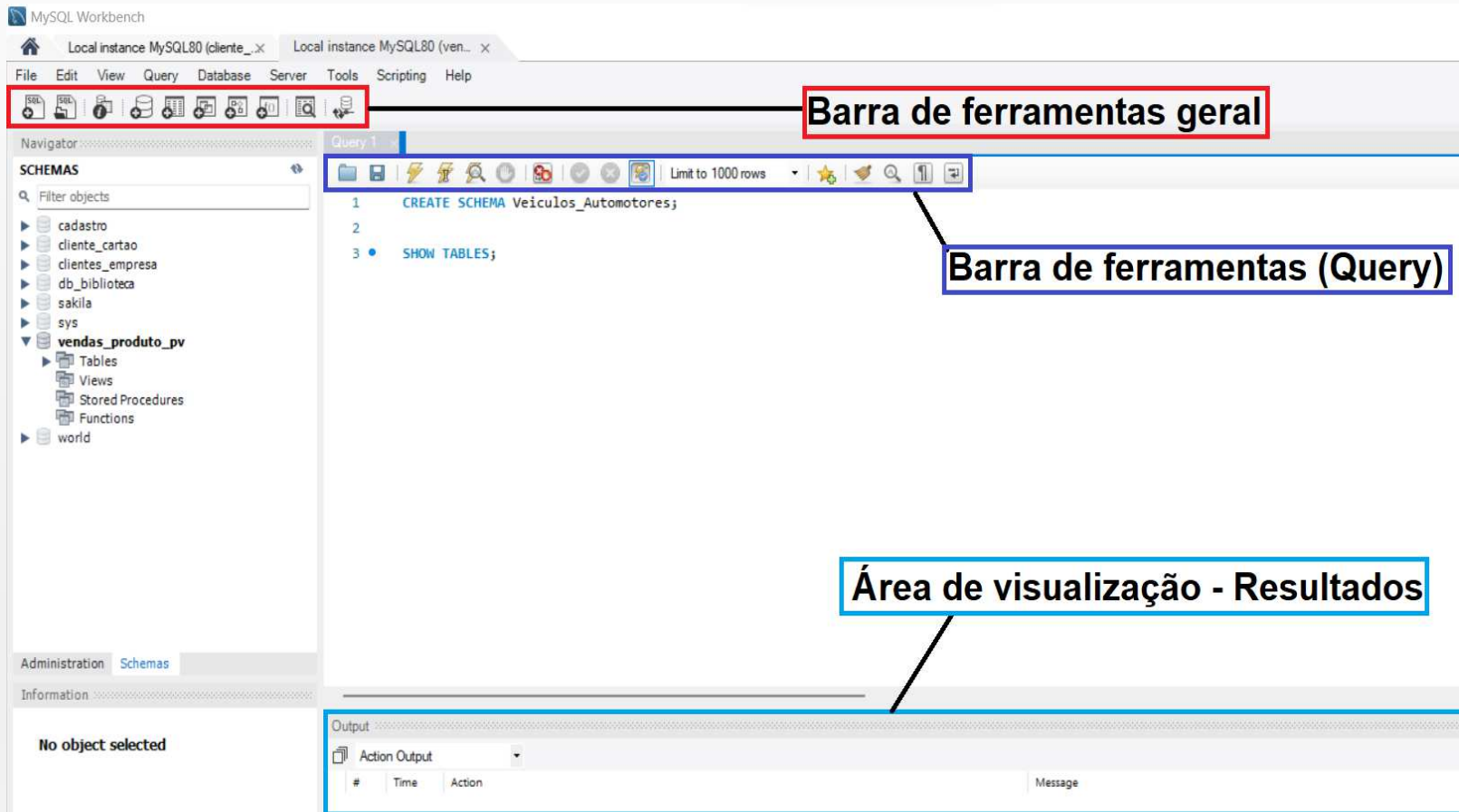
- É um banco de dados de código aberto (qualquer pessoa pode usar e modificar o software);
- Os bancos de dados são um repositório de dados essencial para todas as aplicações de software;
- Um banco de dados relacional armazena dados em tabelas separadas;
- SQL (Structured Query Language) é a linguagem padronizada mais comum utilizada para acessar bancos de dados;
- O MySQL é utilizado em plataformas de Ecommerce (Uber e Booking.com), plataformas de redes sociais (Facebook, X, LinkedIn) e afins.



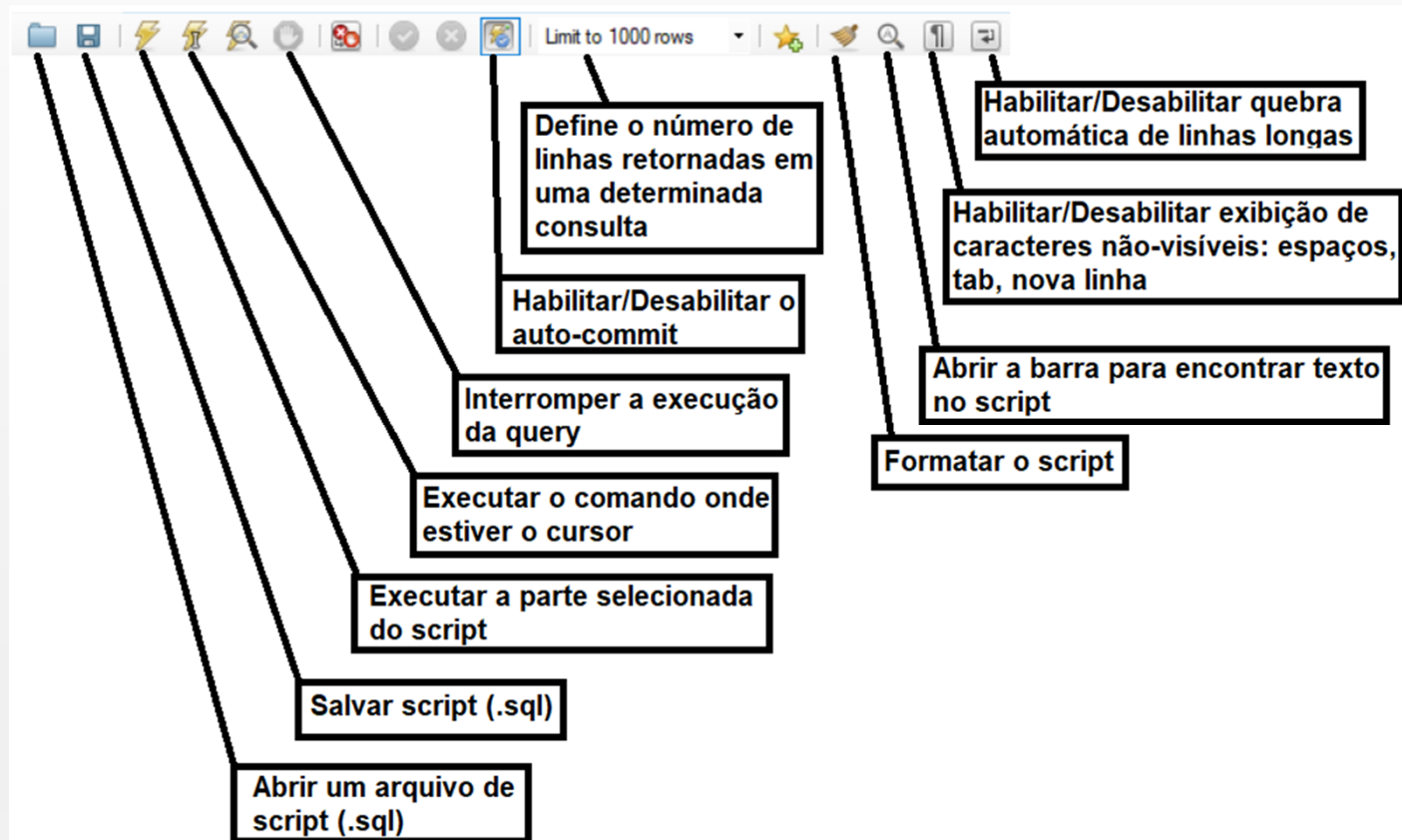
MySQL



MySQL



MySQL – BARRA DE FERRAMENTAS (QUERY)



MySQL – INSERT

- Em sua forma mais simples, a instrução INSERT é usada para acrescentar uma única tupla a uma tabela;
- É possível escrever a instrução INSERT de duas formas:

INSERT INTO *table_name* (*column1*, *column2*, *column3*, ...)
VALUES (*value1*, *value2*, *value3*, ...);

- Se estiver acrescentando valores para todas as colunas da tabela, não será necessário especificar os nomes das colunas na consulta SQL. Todavia, garanta que a ordem dos valores esteja na mesma ordem das colunas da tabela.

INSERT INTO *table_name* **VALUES** (*value1*, *value2*, *value3*, ...);



MySQL – SELECT

- O comando SELECT possibilita a consulta e retorno de um conjunto de registros de uma ou mais tabelas.
- O comando em questão possui diversas cláusulas opcionais que podem ser utilizadas conforme a informação que se deseja obter;
- **SELECT** *COLUMN1, COLUMN2, ...* **FROM** *TABLE_NAME*;
- Os campos **column1**, **column2**, ... são os nomes das colunas que se deseja obter os dados. Caso seja necessário a obtenção de todos os dados de uma determinada tabela, use a seguinte sintaxe:
- **SELECT** * **FROM** *TABLE_NAME*;



MySQL – UPDATE

- A instrução UPDATE é utilizado para modificar valores de atributo(s) de uma ou mais tuplas selecionadas;
- É possível especificar (atualizar) um determinado atributo com valor NULL;
- A sintaxe do comando UPDATE é simples, observe abaixo:

UPDATE nome_da_tabela

SET nome_da_coluna1=valor_da_coluna1 WHERE condição;

- O argumento SET é responsável por especificar as colunas da tabela que serão alteradas/atualizadas;
- Ressalta-se que ao listar o nome das colunas, as mesmas devem sempre ser separadas por vírgulas.



MySQL – DELETE

- A instrução/comando DELETE remove tupla(s) de uma tabela, sendo necessária a cláusula WHERE para selecionar as tuplas a serem excluídas;
- Uma cláusula WHERE inexistente especifica que todas as tuplas na relação deverão ser excluídas, porém, a tabela permanece no BD (TRUNCATE TABLE);

DELETE FROM table_name WHERE condition;

- Destaca-se que esse comando não exclui estruturas do BD, apenas os dados armazenados em determinada tabela.



PhpMyAdmin

- É uma ferramenta que se destina a trabalhar com a administração de um servidor de BD MySQL ou MariaDB;
- Utilizado para executar tarefas de administração, como:
 - Criação de banco de dados;
 - Execução de consultas;
 - Adição de contas de usuários.
- XAMPP possui embutido o MySQL e o Apache. Caso não tenha o Workbench instalado pode-se utilizar o PhpMyAdmin clicando-se no botão Admin;



PhpMyAdmin

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

Modules

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	15328 12008	50249, 50254	Stop Admin Config Logs
<input type="checkbox"/>	MySQL	16188	3307	Stop Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

Config Netstat Shell Explorer Services Help Quit



PhpMyAdmin

- **Servidor:** localhost (instalado na máquina que estamos trabalhando);
- **Usuário:** root (Controla tudo dentro do BD MySQL);
- **Senha:** Na instalação do xampp vem com a senha em branco;
- **Banco:** São os esquemas (schemas) que venham a ser criados;
- **Tabelas:** Dados armazenados.



MySQLi

- A partir da versão PHP 7 o suporte para o PHP 5 foi encerrado;
- Com a descontinuação da extensão original do MySQL (**mysql_***), teve-se que optar por trabalhar com MySQLi ou PDO;
- MySQL é um componente central do XAMPP;
- O MySQLi foi projetado para aproveitar melhor os recursos mais recentes do servidor MySQL;
- O MySQLi funciona somente com banco de dados MySQL, enquanto o PDO é mais flexível;



PDO

- Dentro do PHP existem várias formas de conexão com o banco de dados, uma delas é o PDO;
- PDO é um objeto nativo do PHP, porém não é necessário compreender orientação a objetos para utilizá-lo;
- A extensão PHP Data Objects (PDO) define uma interface leve e comum para acessar bancos de dados em PHP;
- O PDO está disponível a partir da versão 5.1 e dá suporte a vários sistemas gerenciadores de Banco de dados, como:
 - MySQL
 - PostgreSQL;
 - Oracle;
 - SQL Server.



PDO

- Primeiro passo para se trabalhar com o PDO é criar uma conexão com o banco de dados, como:
 - **host:** Local onde o banco de dados se encontra;
 - **dbname:** o nome do banco de dados;
 - **Usuario;**
 - **Senha;**
- O nome da variável que vai receber a conexão não precisa necessariamente ser **\$pdo**, mas como boa prática é recomendável que tal variável tenha a ver com o seu respectivo conteúdo;

```
//Para criar a nossa conexão precisamos armazenar ela em uma variável  
$pdo = new PDO('mysql:host=localhost;dbname=teste', 'root', '1234');
```



PDO

```
<?php
// MySQL
$pdo = new PDO("mysql:host=localhost;dbname=banco",
               "root", "senha");

// PostgreSQL
$pdo = new PDO("pgsql:host=localhost;dbname=banco",
               "root", "senha");

// SQLite
$pdo = new PDO("sqlite:banco.sqlite");
?>
```



PDO

- **PDO::exec():** Utilizado para executar uma instrução SQL que não retorna resultados diretos (INSERT, UPDATE ou DELETE);

```
$pdo->exec("INSERT INTO teste.tab_usuario  
(nomeUsu, estadoCivil, cpf, profissao) VALUES  
( 'José da Silva', 'Casado', 03322578911, 'Engenheiro' )");
```

- **PDO::query():** Utilizado para executar uma consulta SQL no banco de dados e retornar um objeto que contém os resultados da consulta (SELECT);

```
$result = $pdo->query("SELECT * FROM tab_usuario")->fetchAll();
```



PDO x MySQLi

- Suporte ao banco de dados:
 - O PDO suporta 12 diferentes tipos de banco de dados;
 - MySQLi suporta apenas o MySQL;
- Quando precisar mudar o projeto para usar outro banco de dados, o PDO torna o processo mais simples;

```
$pdo = new PDO('mysql:host=localhost;dbname=database',  
  'usuario', 'senha');  
  
$pdo = new PDO('pgsql:host=localhost;dbname=database',  
  'usuario', 'senha');
```



Dúvidas?

