

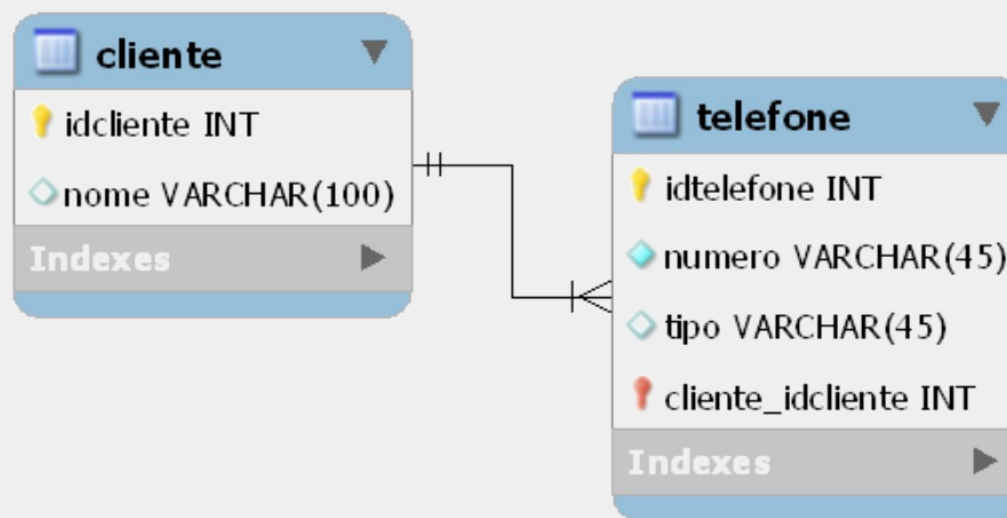


# INTEGRAÇÃO PHP E MYSQL: MYSQLI E PDO

Desenvolvimento de Software para Internet



# Vamos criar um cadastro de cliente e telefone?





# Criando banco de dados – parte 1

```
CREATE DATABASE agenda;  
USE agenda;  
CREATE TABLE cliente (  
    idcliente INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(255) NOT NULL,  
    datanasc DATE,  
    foto VARCHAR(255)  
) Engine=InnoDB;
```



# Criando banco de dados – parte 2

```
CREATE TABLE telefone (  
    idtelefone INT AUTO_INCREMENT PRIMARY KEY,  
    numero VARCHAR(255) NOT NULL,  
    tipo VARCHAR(255),  
    cliente_idcliente INT NOT NULL,  
    CONSTRAINT fk_telefone_cliente  
    FOREIGN KEY (cliente_idcliente)  
    REFERENCES cliente (idcliente)  
    ON DELETE CASCADE  
    ON UPDATE NO ACTION  
)  
Engine=InnoDB;
```



# Tabelas do banco de dados agenda (já com informações)

idcliente	nome	datanasc	foto
1	Carolina	2000-08-22	NULL
2	Adriano	2002-05-20	NULL
3	Maria	2003-03-13	NULL
4	Manuela Andrade	2000-09-23	fotos/2023-10-16_232554foto2.png
5	Maria da Silva	1970-09-23	fotos/2023-10-16_232826foto.png
6	Cristo Redentor	2000-01-01	fotos/2023-10-16_233459Cristo_Redentor_head.jpg
7	Francisco	2323-09-22	fotos/2023-10-16_233555BXK1_DSC00066800.jpg

cliente

telefone

idtelefone	numero	tipo	cliente_idcliente
1	22233-2332	Fixo	1
2	2322-12221	Fixo	2
3	90000-2233	Celular	2
4	93434-2343	Celular	3





# Integração de PHP com MySQL





# Duas formas de integrar PHP e MySQL

- **MySQLi - MySQL Improved** é uma extensão exclusiva para o MySQL que adiciona novos recursos para uma interface de banco de dados MySQL | MySQLi é tanto procedural quanto orientada a objetos
- **PDO: PHP Data Objects** é uma camada de abstração de acesso a dados, o que significa que, independentemente do tipo de banco de dados que você usa (MySQL, Postgree, Oracle), as mesmas funções estão disponíveis para executar consultas e obter resultados | Orientado a objetos





# MySQLi procedural



Função	Objetivo
<code>mysqli_connect('servidor','username','password','database')</code>	Conexão com o banco
<code>mysqli_query('consulta sql')</code>	Fazer consulta ao banco
<code>mysqli_fetch_array('consulta retornada')</code>	Armazena o retorno da consulta em um array
<code>mysqli_fetch_assoc('consulta retornada')</code>	Armazena o retorno da consulta em um array associativo (chave > valor)
<code>mysqli_num_rows('consulta retornada')</code>	Retorna o número de linhas do conjunto de resultados

Mais funções do modo procedural: <https://www.geeksforgeeks.org/mysqli-procedural-functions/>







# MySQLi orientado a objetos

Função	Objetivo
<code>\$mysqli = new mysqli("localhost","my_user","my_password","my_db")</code>	Conexão com o banco
<code>\$mysqli -&gt; connect_errno</code>	Verifica erro de conexão
<code>\$result = \$mysqli -&gt; query("consulta sql");</code>	Fazer consulta ao banco
<code>\$result -&gt; fetch_array('MYSQLI_NUM ou MYSQL_ASSOC')</code>	Armazena o retorno da consulta em um array. Importante indicar o tipo
<code>\$result -&gt; fetch_assoc()</code>	Armazena o retorno da consulta em um array
<code>\$result -&gt; num_rows</code>	Retorna o número de linhas do conjunto de resultados





# Vantagem do PDO sobre MySQLi

- Suporte a bancos de dados:
  - PDO: 12 diferentes tipos de bancos | MySQLi: apenas MySQL
- Quando for necessário mudar o projeto para usar outro banco de dados, o PDO torna o processo mais simples
  - Muda-se parâmetros de conexão e, no máximo, algumas consultas usadas, caso não sejam compatíveis com o





# Como é a conexão do PDO?

```
// PDO
```

```
$pdo = new PDO("mysql:" . "host=localhost;" . "dbname=database",  
'username', 'password');
```

```
// mysqli, object oriented way
```

```
$mysqli = new mysqli('localhost', 'username', 'password', 'database');
```

Crédito: [Slides da Profa Lara Oberderfer | IFSC Chapecó](#)





# Exemplo de consultas diferentes



**INSERT** cliente **VALUES**

('Carolina','2000-08-22');



**INSERT INTO** cliente

(nome,datanasc) **VALUES**

('Carolina','2000-08-22');





# Observação sobre o PDO

- Antes de trabalhar com o PDO é necessário habilitar o driver do PDO e o driver do banco que será utilizado
- Para habilitar o PDO:
  - Abrir arquivo php.ini (no diretório onde o PHP foi instalado)
  - Remover os comentários (;) das linhas ao lado
- Habilitando **PDO** no *Windows*:
  - `extension=php_pdo.dll`
  - `extension=php_pdo_mysql.dll`
- Habilitando **PDO** no *Linux*:
  - `extension=pdo.so`
  - `extension=pdo_mysql.so`

Fonte: [Slides da Profa Lara Oberderfer | IFSC Chapecó](#)





# Manipulação de dados com PDO

- Para trabalhar com operações no banco, o PDO possui um método conhecido como **prepare**. Este método apenas prepara uma operação no banco de dados, sendo necessário a utilização do método **execute** para realmente executar a operação.





# Inserção de dados

Inicia a query

Adiciona valores

Realiza a inserção

```
$nome = "Marcia";  
$datanasc = "1980-09-20";
```

```
$query = $pdo -> prepare("INSERT INTO agenda.cliente (nome,  
datanasc) VALUES (?,?)");
```

```
$query -> bindParam(1, $nome, PDO::PARAM_STR);  
$query -> bindParam(2, $datanasc, PDO::PARAM_STR);  
$query -> execute();
```







# Inserção de dados: INOUT

```
$nome = "Marcia";  
$datanasc = "1980-09-20";
```

```
$query = $pdo -> prepare("INSERT INTO agenda.cliente (nome,  
datanasc) VALUES (?, ?)");
```

```
$query -> bindParam(1, $nome, PDO::PARAM_STR);  
$query -> bindParam(2, $datanasc, PDO::PARAM_STR);  
$query -> execute();
```





# Inserção de dados: espaços reservados nomeados

```
$nome = "Leonardo";  
$datanasc = "1977-01-21";
```

```
$query = $pdo -> prepare("INSERT INTO agenda.cliente (nome,  
datanasc) VALUES (:nome,:datanasc)");
```

```
$query -> bindParam(':nome', $nome, PDO::PARAM_STR);  
$query -> bindParam(':datanasc', $datanasc, PDO::PARAM_STR);  
$query -> execute();
```





# Visualização de dados

```
$resultado = $pdo->query("SELECT nome, datanasc FROM cliente");  
while ($item = $resultado->fetch(PDO::FETCH_OBJ))  
{  
    echo $item->nome."<br>";  
    echo $item->datanasc."<br>";  
    echo "<hr>";  
}
```





# Excluindo dados

```
$id = 4;
```

```
$query = $pdo -> prepare("DELETE FROM cliente WHERE idcliente = :id");  
$query -> bindParam(':id',$id,PDO::PARAM_INT);  
$query -> execute();
```





# Editando dados

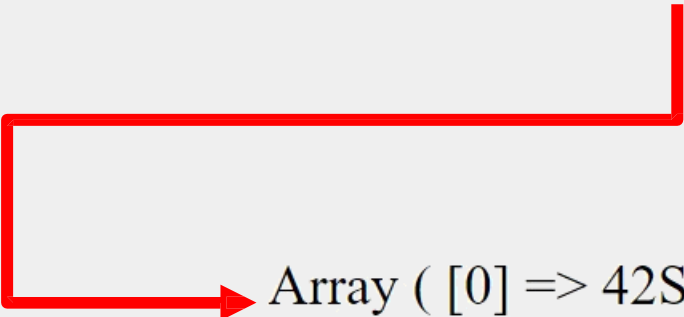
```
$id = 1;  
$nome = "Carolzinha";  
$datanasc = "1992-08-22";  
  
$query = $pdo -> prepare("UPDATE cliente SET nome = ?, datanasc = ? WHERE idcliente = ?");  
  
$query -> bindParam(1,$nome,PDO::PARAM_STR);  
$query -> bindParam(2,$datanasc,PDO::PARAM_STR);  
$query -> bindParam(3,$id,PDO::PARAM_INT);  
$query -> execute();
```





# Como testar se deu certo uma execução?

```
$ok = $query -> execute();  
if(!$ok) {  
    print_r($query->errorInfo());  
}
```




Array ( [0] => 42S22 [1] => 1054 [2] => Unknown column 'nomex' in 'field list' )





# Vídeo para aprofundar: mysqli procedural



**Cadastro Simples com PHP e MySQL - CRUD**

Robson Borges

14 vídeos • 34.417 visualizações • Última atualização em ...

Reproduzir tu... Ordem aleató...

Video Number	Video Title	Views	Age	Duration
1	PHP + MySQL - CRUD Simples - Aula 1	20 mil visualizações	há 4 anos	10:09
2	PHP + MySQL - CRUD Simples - Aula 2	7,9 mil visualizações	há 4 anos	3:07
3	PHP + MySQL - CRUD Simples - Aula 3	7,6 mil visualizações	há 4 anos	4:28
4	PHP + MySQL - CRUD Simples - Aula 4	8,2 mil visualizações	há 4 anos	17:39
5	PHP + MySQL - CRUD Simples - Aula 5	4,4 mil visualizações	há 4 anos	7:12
6	PHP + MySQL - CRUD Simples - Aula 6	4,7 mil visualizações	há 4 anos	18:07
7	PHP + MySQL - CRUD Simples - Aula 7	3,6 mil visualizações	há 4 anos	




[Playlist do Robson Borges](#)





# Vídeo para aprofundar: PDO



- 1  **Curso de PDO - Aula 01 - Programando com PHP7**  
Portal Hugo Cursos • 2,7 mil visualizações • há 3 anos
- 2  **Curso de PDO - Aula 02 - Qual a melhor Estruturação em PHP?**  
Portal Hugo Cursos • 1 mil visualizações • há 3 anos
- 3  **Curso de PDO - Aula 03 - Banco de Dados**  
Portal Hugo Cursos • 884 visualizações • há 3 anos
- 4  **Curso de PDO - Aula 04 - Conexão em PDO**  
Portal Hugo Cursos • 714 visualizações • há 3 anos
- 5  **Curso de PDO - Aula 05 - Testando a conexão**  
Portal Hugo Cursos • 708 visualizações • há 3 anos
- 6  **Curso de PDO - Aula 06 - Inserindo dados com PDO**  
Portal Hugo Cursos • 719 visualizações • há 3 anos
- 7  **Curso de PDO - Aula 07 - Método Prepare**  
Portal Hugo Cursos • 644 visualizações • há 3 anos

[Playlist do Portal Hugo Cursos](#)





# Material sobre MySQLi e PDO

- IFSC – Profa: Lara Popov:  
<https://docente.ifsc.edu.br/lara.popov/web2/AulaMysqliversusPDO.pdf>





# DÚVIDAS?



Créditos:

Profª: Carolina Sacramento  
[Carolina.sacramento@fiocruz.br](mailto:Carolina.sacramento@fiocruz.br)

