

PRO3151 – Laboratório de SI

L09 – SQL - PHP

Terceira aula de Banco de Dados

Prof. Dr. Marcelo Schneck de Paula Pessôa

Prof. Dr. Marco Aurélio de Mesquita

Prof. Dr. Mauro de Mesquita Spinola

- Integração do **MySQL** com o **PHP**
- Como fazer o acesso ao banco de dados **MySQL** com **PHP**
- Trata-se de um conjunto de funções preparadas para essa finalidade
- Por exemplo: **mysqli_connect**, **mysqli_fetch_array**

Conexão com outros BDs



- O **PHP** pode se conectar com outros bancos de dados como o **Postgree**
- Para esse caso existem também funções similares que podem ser utilizadas
- Por exemplo: **pg_connect**, **pg_fetch_array**

Atenção



O *MySQL* com PHP possui duas versões:

- ***mysql*** – mais antigo e exatamente o mesmo exercitado no *DOS*
- ***mysqli*** – *i* de “*improved*” onde há pequenas alterações nos comandos

Atenção



O *MySQL* com PHP possui duas versões:

- Ao procurar os manuais ou códigos prontos, é necessário atenção pois, embora parecidos não há compatibilidade.
- Pode-se escolher tanto o *mysql* como o *mysqli*.
- Na disciplina e nos exemplos será adotado o *mysqli*.

Sintaxe padrão de uso do SQL standard line:

- `mysql -h localhost -u username -p;`
- `use (banco de dados);`
- comandos de acesso `SELECT....`

Acesso ao mysql

Seleção do banco de dados

Comandos diversos

Sintaxe padrão de uso do SQL com PHP:

- `mysqli_connect` (servidor, username, password, database);
- `mysqli_diversos` comandos;
- `mysqli_close` (*conexão*);

*Acesso ao mysql e
banco de dados*

Comandos diversos

Fecha a conexão

Acesso com o PHP:

- A diferença é que o acesso a um banco de dados é uma **conexão**
- Para se ter acesso a dois bancos de dados é feita uma conexão para cada banco de dados e o acesso às tabelas deve ser precedido do nome da conexão
- Outra diferença é sempre fechar o acesso ao banco de dados após ter sido feito acesso.

Comandos básicos:

- `mysqli_connect` (*servidor, username, password, database*);
- `mysqli_query`(*conexão, query*);
- `mysqli_fetch_array`(*resultado*);
- `mysqli_close` (*conexão*);

PHP & MySQL



- `mysqli_connect(servidor,username,password,database);`
- `mysqli_close(conexão);`
- Tem por objetivo realizar a conexão do PHP com o banco de dados
- No início do programa (ou da página) deve ser feita a conexão
- Ao final do acesso ao banco de dados deve ser desfeita a conexão

`mysql_query` (*query*)

`mysqli_fetch_array`(*resultado*)

- É um conjunto de comandos que permite o acesso propriamente dito ao banco de dados

query

- Trata de uma série grande de comandos muito poderosos para realizar a extração e inserção de dados

query

Aqui serão estudadas as seguintes queries:

- *INSERT* para inserir registros no banco de dados
- *SELECT* que seleciona dados para leitura através de parâmetros de filtragem
- *UPDATE* para editar campos de um registro
- *DELETE* para apagar um registro

mysql

COMANDOS INICIAIS

Marcelo Pessôa - Marco Mesquita - Mauro Spinola

Create database



```
<?php
```

```
$servername = "localhost";  
$username = "root";  
$password = "";
```

visto na pré-aula

```
// Create connection
```

```
$conn = new mysqli($servername,  
$username, $password);
```

```
// Check connection
```

```
if ($conn->connect_error) {  
die("Connection failed: " . $conn-  
>connect_error);  
}
```

```
// Create database
```

```
$sql = "CREATE DATABASE myDB";  
if ($conn->query($sql) === TRUE) {  
echo "Database created successfully";  
} else {  
echo "Error creating database: " . $conn-  
>error;  
}
```

```
$conn->close();
```

```
?>
```

Create table



```
<?php
```

```
$servername = "localhost";
```

```
$username = "root";
```

```
$password = "";
```

```
$dbname = "myDB";
```

```
// Create connection
```

```
$conn = new mysqli($servername,  
$username, $password, $dbname);
```

```
// Check connection
```

```
if ($conn->connect_error) {
```

```
die("Connection failed: " . $conn-  
>connect_error);
```

```
}
```

visto na pré-aula

```
// Create database
```

```
$sql = "CREATE DATABASE myDB";
```

```
if ($conn->query($sql) === TRUE) {
```

```
echo "Database created successfully";
```

```
} else {
```

```
echo "Error creating database: " . $conn-  
>error;
```

```
}
```

```
$conn->close();
```

```
?>
```

mysql

INSERÇÃO DE DADOS

Marcelo Pessôa - Marco Mesquita - Mauro Spinola

Inserção de dados



```
<?php
```

```
$servername = "localhost";  
$username = "root";  
$password = "";  
$dbname = "myDB";
```

visto na pré-aula

```
// Create connection
```

```
$conn = new mysqli($servername,  
$username, $password, $dbname);
```

```
// Check connection
```

```
if ($conn->connect_error) {
```

```
    die("Connection failed: "
```

```
    $conn->connect_error);
```

```
}
```

Essa parte é sempre igual

```
// Insert Records
```

```
if (isset($_POST['submit'])) {
```

```
    $firstname = $_POST['firstname'];
```

```
    $lastname = $_POST['lastname'];
```

```
    $email = $_POST['email'];
```

```
$sql = "INSERT INTO MyGuests (firstname, lastname, email)  
VALUES ('$firstname', '$lastname', '$email)";
```

```
If ($conn->query($sql) === TRUE) {
```

```
    echo "New record created successfully.";
```

```
}else{echo "Error:". $sql . "<br>". $conn->error;
```

```
}
```

```
$conn->close();
```

Inserção de dados



```
$sql = "INSERT INTO MyGuests (firstname, lastname, email) VALUES ('$firstname', '$lastname', '$email')";
```

- \$sql é uma variável PHP
- O que está entre aspas é, para o PHP, um string, um conjunto de caracteres
- O PHP não sabe o significado desse texto
- Envia para o MySQL esses strings que lá são executados

Inserção de dados



```
$sql = "INSERT INTO MyGuests (firstname, lastname, email) VALUES ('$firstname', '$lastname', '$email')";
```

- Atenção quanto às aspas “ ” para o PHP significa inicio e fim do string
- No comando SQL também tem aspas e deve-se usar ‘ ’
- Exceto algumas funções criadas no PHP, os comandos SQL são os mesmos que foram usados no PHPMyAdmin
- Se houver dúvidas quanto à sintaxe fazer a query no PHPMyAdmin e copiar para o programa PHP

mysql

LEITURA DE DADOS

Marcelo Pessôa - Marco Mesquita - Mauro Spinola

Leitura de dados



```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername,
$username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " .
$conn->connect_error);
}
```

visto na pré-aula

```
// View Records
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
// Output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
$row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
```

*Observar
a conexão*

Leitura de dados



```
$sql = "SELECT id, firstname, lastname FROM MyGuests";  
$result = $conn->query($sql);  
$result = mysqli_query($conn, $sql);
```

- Existe uma segunda sintaxe possível para esses comandos
- A primeira é denominada sintaxe de orientação a objeto
- A segunda é denominada notação procedural.

Comando *SELECT*



- A sintaxe completa do comando *SELECT* é:

SELECT [opções] *campos*

[*INTO detalhes_arquivo*]

FROM *tabelas*

[*WHERE condições*]

[*GROUP BY tipo de agrupamento*]

[*HAVING definições de having*]

[*ORDER BY tipo de ordenação*]

[*LIMIT critério de limitação*]

[*PROCEDURE nome da procedure(argumentos)*]

Comando SELECT



- Os itens entre colchetes são opcionais
- As cláusulas mais usadas são:
 - opções nome dos campos da tabela
 - WHERE estabelece as condições de filtragem
 - ORDER BY ascendente ou descendente
- Podem ser selecionados dados de mais de uma tabela

**INTERRUPÇÃO PARA
CONTEXTUALIZAÇÃO DOS
EXERCÍCIOS**

mysql

APLICAÇÃO PARA OS EXERCÍCIOS

Marcelo Pessôa - Marco Mesquita - Mauro Spinola

Aplicação para os exercícios



Estoque

Home

Operação do estoque:
Arquivos CSV - leitura

- [Lê arquivo CSV descrição](#)
- [Lê movimento CSV](#)

Arquivos PHP

- [Lê tabela de movimento de estoque](#)
- [Edita tabela de movimento de estoque](#)
- [Ler movimento ordenado](#)
- [Altera OP - ordem de produção](#)
- [Apaga registro](#)

Copyright © 2011-2023 Fábrika de Software - eLabTI - PRO3151 - Laboratório de Sistemas de Informação.
Professores Marcelo Schneck de Paula Pessoa e Mauro de Mesquita Spinola.
Este é um conjunto de componentes didáticos de software.
Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é permitida desde que citada a fonte.

Marcelo Pessoa - Marco Mesquita - Mauro Spinola

Aplicação para os exercícios

- Fazer o download dos arquivos referentes aos exercícios que serão feitos nesta aula
- Criar um banco de dados denominado Estoque2
- São dois arquivos:
 - Aplicacao.zip – contém o código PHP
 - Estoque2.sql – contém o banco de dados
- Instalar em sua máquina

Aplicação para os exercícios



- Verificar o bom funcionamento da aplicação
- Para isso verificar se os arquivos abrem sem erro

Marcelo Pessôa - Marco Mesquita - Mauro Spinola

Esse código tem duas utilidades:

- Servir como base para a realização dos exercícios do laboratório
- Servir de exemplo de código funcionando completo com CSS, PHP e SQL integrados e funcionando
 - No seu projeto pode usar esse código como orientação para o seu projeto específico

Apresentação do código – alguns detalhes

require "ca/00-canvas.php";

- *trata-se de um truque de programação para trazer o fundo da tela com a parte fixa pronta em um arquivo separado*
- **require** é um comando para trazer um arquivo e colar no ponto do código onde ele foi colocado
- *Evita copiar e colar código*

Apresentação do código – alguns detalhes

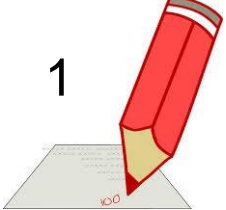
`require("ca/00-configBD.php");`

- *Trata-se de um arquivo que contém os comandos de acesso ao banco de dados*
- *Como esse acesso é sempre o mesmo código, colocar em um arquivo separado*
- ***ca** é um diretório onde ficam todos os arquivos de apoio à aplicação*

Apresentação do código – alguns detalhes

Nome	Data de modificaç...	Tipo	Tamanho
arquivos			
ca			
estilo_arquivos			
preaula_alunos			
000-documentacao			
00-fecha			
00-login			
02-Pagina_abertura			
100-le_arquivocsv			
200-le_movimento			
300-le_movimento			
400-edita_mov_estoque	30/05/2023 00:36	Arquivo PHP	3 KB
401-edita_mov_estoque_edita	30/05/2023 00:38	Arquivo PHP	4 KB
402-edita_mov_estoque_grava	30/05/2023 00:38	Arquivo PHP	3 KB

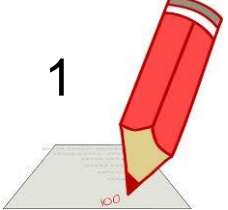
- *Observar a organização e nomeação dos arquivos*
- *Sempre numerar – e colocar um nome mnemônico*
- *Fazer organizado por centenas ou dezenas*
- *Colocar sequencial em um conjunto de arquivos de uma mesma função (veja o 400)*
- ***Nunca usar caracteres especiais e espaço***



Selecionando e ordenando registros



- Criar um arquivo novo denominado **500-movimento_ord.php** que possua as seguintes colunas:
 - ID (identificador do registro de movimento)
 - Data
 - Código (do material que foi movimentado)
 - Descrição
 - Grupo 2
 - Grupo 3
 - Quantidade
 - Unidade
- A página de abertura já tem uma espera para esse arquivo



Selecionando e ordenando registros



- Apresentar ordenado por ordem crescente de código
- A leitura pode ser de todos os movimentos sem filtragem nenhuma
- Usar como base o arquivo existente de leitura da tabela *300-le_movimento.php*
- Dica: as alterações são nos nomes das colunas e no select – verificar os nomes dos campos no phpmyadmin
- Problema: os campos solicitados são de 2 tabelas

mysql

ALTERAÇÃO DE DADOS

Marcelo Pessôa - Marco Mesquita - Mauro Spinola

Alteração de Registro – no exemplo

- Trata-se de um conjunto de arquivos que fazem:
- a leitura da tabela e deixam o campo ID sensível para abertura de somente um registro com o conteúdo atual (400)
 - O formulário de entrada de dados com os dados antigos exibidos para alteração (401)
 - Gravação das alterações feitas no registro específico (402)



Alteração de Registro – leitura da tabela - 400

```
//leitura de todos os campos
$select
$sele="SELECT * FROM stok_moviment";
$query=mysqli_query($db,$sele);
if(!$query) // testa se houve erro ao fazer a query
{
    error_log("Query $query falhou!".mysql_error(),3,"erros.txt");
    echo "deu erro na query!";
}
$i=0;
while ($registro=mysqli_fetch_array($query)) // extração
{
    echo "<tr>";
    // a primeira coluna é com referência para abrir o registro a ser selecionado para edição
    // nesta coluna é colocado o nome do novo arquivo para edição
    $j=0;
    echo("<td align='left'><a href='401-edita mov estoque edita.php?ref=$registro[$j]' >".$registro[$j]."</a></td>");
    for ($j=1; $j<10 ; $j++)
    { // loop para imprimir uma linha
        echo("<td align='center'>".$registro[$j]."</td>");
    }
    $i++;
    echo "</tr>";
}

mysqli_close($db);
```

select

extração

Link no campo 0 (ID)

demais campos 1 a 9

Marcelo Pessôa - Marco Mesquita - Mauro Spinola

Alteração de Registro – extração do registro – 401

```
$sele="SELECT * FROM stok_moviment WHERE $campo[0] = '$ref' ";  
$query=mysqli_query($db,$sele);  
if(!$query) // testa se houve erro ao fazer a query  
{  
    error_log("Query $query falhou!".mysql_error(),3,"erros.txt");  
    echo "deu erro na query!";  
}  
$i=0;
```

extraí o campo
ID selecionado

Alteração de Registro – prepara o form 401

```
echo "<form action='402-edita_mov_estoque_grava.php' method ='post'>";
echo "<input name= 'var0' type='hidden' value='\$ref' />";

while ($registro=mysqli_fetch_array($query))
{
    // a primeira coluna é com referência para abrir o registro a ser selecionado para edição
    // nesta coluna é colocado o nome do novo arquivo para edição      echo "<tr>";
    echo "<th>ID</th>";
    echo ("<td align='center'>".$ref."</td>");
    echo "</tr>";          echo "<tr>";
        echo "<th>codigo</th>";
        echo ("<td align='center'>".$registro[1] "<input name='var1' type='text' value='\$registro[1]' /></td>");
    echo "</tr>";
        echo "<tr>";
            echo "<tr>";
                echo "<th>numdoc</th>";
                echo ("<td align='center'>".$registro[9] "<input name='var9' type='text' value='\$registro[9]' /></td>");
            echo "</tr>";
        echo "</form>";
}
```

uma linha para cada campo

Cada campo é uma variável

Valor original do campo

Alteração de Registro – grava a alteração – 402

```
echo "<p><br>";          $dado[0] = $_POST['var0'] ; echo $dado[0] ;
echo "<br>";           $dado[1] = $_POST['var1'] ; echo $dado[1] ;
echo "<br>";           $dado[2] = $_POST['var2'] ; echo $dado[2] ;
echo "<br>";           $dado[3] = $_POST['var3'] ; echo $dado[3] ;
echo "<br>";           $dado[4] = $_POST['var4'] ; echo $dado[4] ;
echo "<br>";           $dado[5] = $_POST['var5'] ; echo $dado[5] ;
echo "<br>";           $dado[6] = $_POST['var6'] ; echo $dado[6] ;
echo "<br>";           $dado[7] = $_POST['var7'] ; echo $dado[7] ;
echo "<br>";           $dado[8] = $_POST['var8'] ; echo $dado[8] ;
echo "<br>";           $dado[9] = $_POST['var9'] ; echo $dado[9] ;
echo "<br>";
```

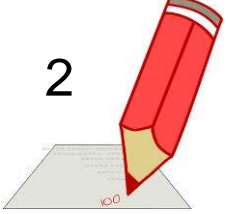
POST para receber os dados

```
$query = " UPDATE stok_moviment SET  codigo='$dado[1]' ,data='$dado[2]' ,qtde='$dado[3]' ,tipo='$dado[4]' ,
deposito='$dado[5]' ,localizacao='$dado[6]' ,transacao='$dado[7]' ,tipodoc='$dado[8]' ,numdoc='$dado[9]'
WHERE ID = '$ponteiro' ";          ## gera2
echo "<p>";
$result = mysqli_query ($db,$query);
if (!$query)
{   echo "Ocorreu um erro no mysqli! O item nao foi gravado!";
    return;
}
ELSE echo ("Gravado com sucesso!");
mysqli_close ($db);
```

especifica qual registro é atualizado

Comando
UPDATE

Marcelo Pessôa - Marco Mesquita - Mauro Spinola



Edição de registro



- Criar um arquivo novo denominado **600-alteraOP.php** que pergunta qual é o número da ordem de produção a ser alterada e abre todos os registros para alteração
- A página de abertura já tem uma espera para esse arquivo
- Não permitir alterar os campos ***tipodoc*** e ***numdoc***

mysql

APAGANDO REGISTROS

Marcelo Pessôa - Marco Mesquita - Mauro Spinola

comando *DELETE*



```
$seleciona = "DELETE FROM tabela WHERE condicao";
```

```
$query=mysqli_query($db,$seleciona);
```

comando *DELETE*



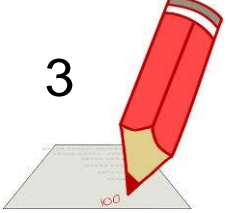
- Esse comando é muito perigoso pois permite eliminar registros de informações que existiam no sistema
- Quando existe dependência de informações entre diferentes tabelas, a eliminação de registros pode criar inconsistência de dados
 - por exemplo em uma nota fiscal eliminar um cabeçalho e não apagar os itens comprados daquela nota

comando *DELETE*



- Uma solução comum em sistemas é ter uma coluna binária (sim/não) que, quando se manda apagar o registro ele passa a não ser mostrado mais, mas continua no banco de dados
- Isso é comum para permitir ***auditoria de sistemas*** para verificar o que foi feito na aplicação

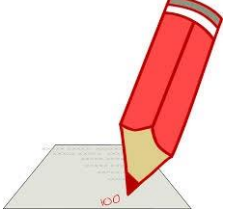
3



Apagando registro



- Criar um arquivo novo denominado 700-apagaR.php que pergunta qual é o número do registro a ser apagado
- A página de abertura já tem uma espera para esse arquivo



Entregas



As entregas devem ser feitas da seguinte forma:

- Fazer um zip de todos os arquivos utilizados com os três exercícios resolvidos e fazer upload desse arquivo no e-disciplinas
- Nomear o arquivo da seguinte forma: L09TxGy.zip onde x e y correspondem aos números de turma e grupo

PRO3151 – Laboratório de SI

L09 – SQL - PHP

Terceira aula de Banco de Dados

Prof. Dr. Marcelo Schneck de Paula Pessoa

Prof. Dr. Marco Aurélio de Mesquita

Prof. Dr. Mauro de Mesquita Spinola