**Explicação Detalhada do Script PHP para Cadastro e Consulta de Alunos**

**1. Verificação do Método de Requisição (if ($\_SERVER["REQUEST\_METHOD"] == "POST")):**

* Este bloco verifica se o formulário foi enviado utilizando o método POST. O método POST é geralmente utilizado para enviar dados do formulário para o servidor.
* Se o método for POST, o script prossegue para a próxima etapa, que é capturar os dados do usuário.

**2. Capturando Dados do Usuário ($\_POST["nome"] e $\_POST["comentario"]):**

* As variáveis globais $\_POST["nome"] e $\_POST["comentario"] capturam os valores dos campos nome e comentario do formulário, respectivamente.
* Estes valores são armazenados nas variáveis $nomeAluno e $comentario para serem utilizados posteriormente.

**3. Validação Básica (opcional - if (empty($nomeAluno) || empty($comentario)))**:

* Este bloco de código realiza uma validação básica para verificar se os campos nome e comentario foram preenchidos pelo usuário.
* Se qualquer um dos campos estiver vazio, uma mensagem de erro é exibida e o script é encerrado.
* A validação básica é importante para evitar que dados incorretos sejam armazenados no banco de dados.

**4. Conexão com o Banco de Dados (mysqli\_connect)**:

* As variáveis $hostname, $username, $password e $database armazenam as credenciais de acesso ao banco de dados.
* A função mysqli\_connect é utilizada para estabelecer uma conexão com o banco de dados MySQL.
* Se a conexão for bem-sucedida, a variável $conn conterá um objeto que representa a conexão com o banco de dados.
* Se a conexão falhar, uma mensagem de erro é exibida utilizando a função die.

**5. Inserindo Dados no Banco de Dados (mysqli\_prepare, mysqli\_stmt\_bind\_param, mysqli\_stmt\_execute)**:

* A função mysqli\_prepare é utilizada para preparar uma consulta SQL com segurança contra ataques de injection.
* A consulta SQL (INSERT INTO alunos (nome, comentario) VALUES (?, ?)) insere os valores das variáveis $nomeAluno e $comentario na tabela alunos do banco de dados.
* A função mysqli\_stmt\_bind\_param associa os valores das variáveis $nomeAluno e $comentario aos parâmetros ? na consulta SQL.
* A função mysqli\_stmt\_execute executa a consulta SQL preparada.
* Se a consulta for executada com sucesso, uma mensagem de confirmação é exibida.
* Caso contrário, uma mensagem de erro é exibida com detalhes do erro fornecidos pela função mysqli\_error.
* A função mysqli\_stmt\_close é utilizada para fechar a consulta preparada.

**6. Fechando a Conexão com o Banco de Dados (mysqli\_close)**:

* A função mysqli\_close é utilizada para fechar a conexão com o banco de dados.
* É importante fechar a conexão ao final do script para liberar recursos e evitar problemas de desempenho.

**7. Consulta de Alunos (if (isset($\_GET["nome"])))**:

* Este bloco de código verifica se a variável $\_GET["nome"] está definida na URL.
* A variável $\_GET["nome"] contém o valor do campo nome do formulário de consulta.
* Se a variável estiver definida, significa que o usuário está realizando uma consulta de aluno.

**8. Conexão com o Banco de Dados para Consulta (mysqli\_connect)**:

* A conexão com o banco de dados é estabelecida da mesma forma que no bloco de cadastro.
* Se a conexão for bem-sucedida, a variável $conn conterá um objeto que representa a conexão com o banco de dados.

**9. Consulta SQL para Buscar Aluno Específico ($sql, mysqli\_query)**:

* A consulta SQL (SELECT nome, comentario FROM alunos WHERE nome LIKE '%$nomeConsulta%') busca os dados do aluno com nome correspondente ao valor da variável $nomeConsulta.
* O operador LIKE permite pesquisas por nome que contenham o termo informado, proporcionando mais flexibilidade na busca.

**Explicação Continuada do Script PHP para Cadastro e Consulta de Alunos**

**10. Exibindo Resultados da Consulta (continuação):**

* Se a consulta encontrou resultados (mysqli\_num\_rows($result) > 0), o script exibe os dados do aluno:
  + Um título <h2>Resultados da Consulta para '$nomeConsulta':</h2> é exibido para informar o usuário sobre os resultados da busca.
  + Uma tabela HTML é criada para apresentar os dados de forma organizada:
    - O cabeçalho (<thead>) da tabela define as colunas Nome e Comentário.
    - O corpo (<tbody>) da tabela itera sobre cada linha de resultado (while ($row = mysqli\_fetch\_assoc($result))) utilizando um loop while.
      * Para cada linha, cria-se uma linha (<tr>) na tabela.
      * As colunas (<td>) da tabela exibem o valor do nome ($row['nome']) e comentário ($row['comentario']) obtidos do resultado da consulta.
      * A função nl2br($row['comentario']) converte quebras de linha presentes no comentário para o formato HTML <br>, garantindo a exibição correta na tabela.
  + A tabela HTML é fechada (</table>).
* Se a consulta não encontrou resultados (else), uma mensagem informando que nenhum aluno foi encontrado é exibida.

**11. Fechando a Conexão com o Banco de Dados (mysqli\_close)**:

* A conexão com o banco de dados é fechada da mesma forma que no bloco de cadastro, liberando recursos e evitando problemas de desempenho.

**Observações:**

* Este script utiliza prepared statements para realizar consultas SQL com segurança contra ataques de injection.
* O script realiza validação básica para evitar dados incorretos no banco de dados.
* Você pode personalizar o script para atender às suas necessidades específicas, como adicionar novos campos ao formulário e à tabela de resultados.
* Lembre-se de substituir as credenciais de acesso ao banco de dados ($hostname, $username, $password, $database)

**Criação do banco de dados**

**Explicação detalhada do comando SQL CREATE TABLE para o script de alunos**

Este trecho de código é um comando MySQL utilizado para criar uma tabela chamada "alunos" em um banco de dados. Vamos analisá-lo passo a passo:

1. **CREATE TABLE**: Esta palavra-chave especifica que queremos criar uma nova tabela no banco de dados.
2. **alunos**: Este é o nome que você está dando à tabela. Você pode escolher qualquer nome relevante que reflita o propósito da tabela (por exemplo, estudantes, usuários, produtos, etc.).
3. **( )**: Os parênteses delimitam a definição das colunas da tabela, que especificam os dados que serão armazenados em cada linha da tabela.
4. **id INT AUTO\_INCREMENT PRIMARY KEY**: Esta linha define a primeira coluna da tabela. Aqui está uma breakdown:
   * **id**: Este é o nome da coluna. É recomendável usar nomes claros e descritivos para colunas.
   * **INT**: Isso especifica o tipo de dado da coluna. Nesse caso, INT representa um inteiro (número inteiro) usado para armazenar valores numéricos.
   * **AUTO\_INCREMENT**: Esta palavra-chave informa ao banco de dados para incrementar automaticamente o valor em 1 para cada novo registro inserido na tabela. Isso garante que cada registro tenha um identificador único.
   * **PRIMARY KEY**: Esta palavra-chave define a coluna "id" como a chave primária da tabela. Uma chave primária é um identificador único para cada linha na tabela e não pode ser nula (conter nenhum valor).
5. **nome VARCHAR(255) NOT NULL**: Esta linha define a segunda coluna da tabela:
   * **nome**: Este é o nome da coluna, provavelmente armazenando o nome do aluno.
   * **VARCHAR(255)**: Isso especifica o tipo de dado como VARCHAR, que pode armazenar strings de texto de tamanho variável. O número (255) indica o número máximo de caracteres permitidos na coluna "nome". Você pode ajustar esse valor com base nos comprimentos de nome esperados.
   * **NOT NULL**: Esta palavra-chave garante que a coluna "nome" não possa estar vazia para nenhum registro. Deve conter um valor.
6. **comentario TEXT NOT NULL**: Esta linha define a terceira coluna da tabela:
   * **comentario**: Este é o nome da coluna, provavelmente armazenando o comentário do aluno.
   * **TEXT**: Este tipo de dado permite o armazenamento de grandes quantidades de texto em comparação com VARCHAR. É adequado para armazenar comentários ou descrições que podem ser mais longas do que um nome.
   * **NOT NULL**: Semelhante a "nome", isso garante que a coluna "comentario" não possa estar vazia e deva conter um valor.
7. **;**: O ponto e vírgula no final encerra a instrução SQL.