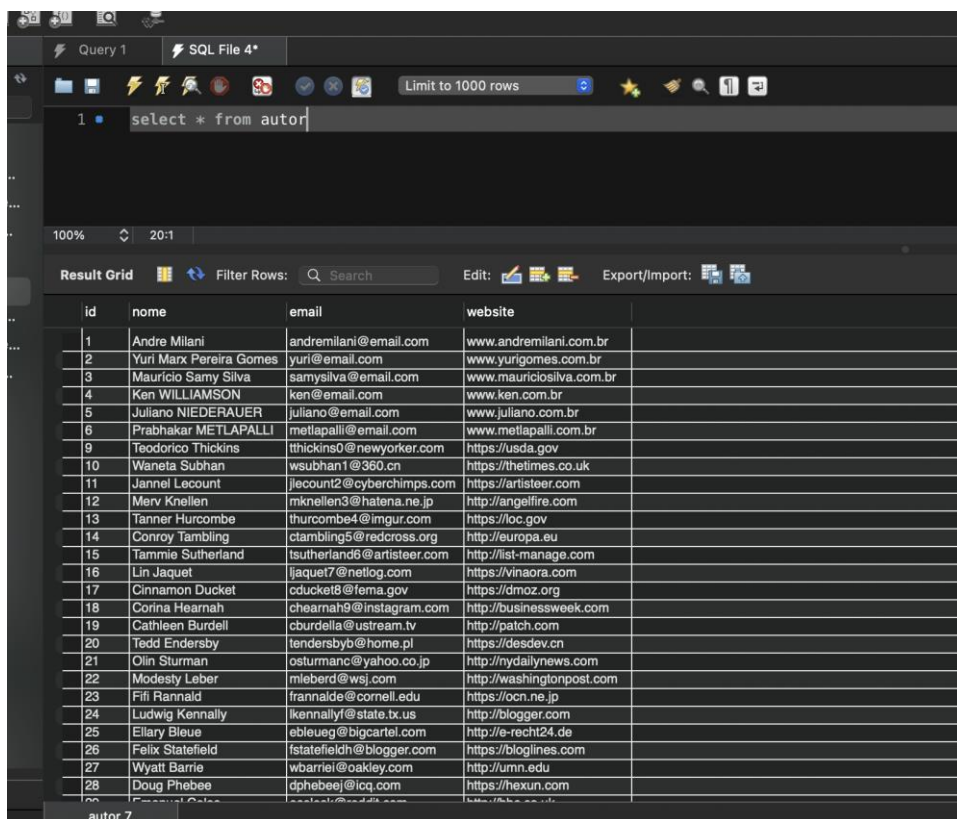


Nome: Lucas Stein

Alterações feitas.

1. Retirei a senha 123 do arquivo de conexão, pois meu banco de casa não possui senha.

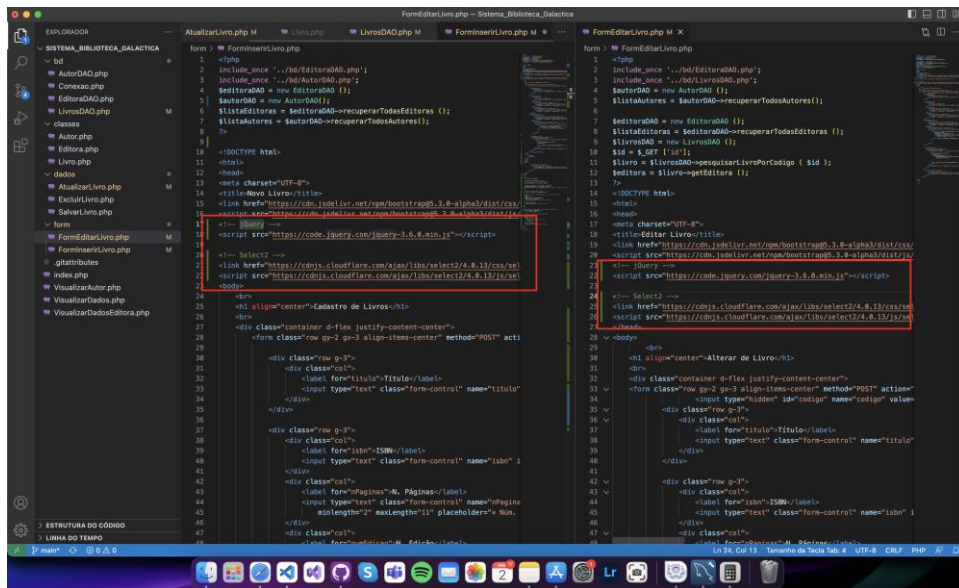
```
bd > Conexao.php
1  <?php
2  class Conexao {
3      protected $conn;
4      function __construct() {
5          $this->conn = new PDO ( "mysql:host=localhost;dbname=estante", "root", "" );
6      }
7      public function obterConexao() {
8          return $this->conn;
9      }
10 }
11 ?>
12
```



The screenshot shows a database management tool interface. At the top, there's a toolbar with icons for file operations and a search bar. Below the toolbar, a query editor shows the command 'select * from autor'. The results are displayed in a table with columns: id, nome, email, and website. The table contains 28 rows of data, each representing an author. The interface also includes a 'Result Grid' section with options for filtering, editing, and exporting data.

id	nome	email	website
1	Andre Milani	andremilani@email.com	www.andremilani.com.br
2	Yuri Marx Pereira Gomes	yuri@email.com	www.yurigomes.com.br
3	Mauricio Samy Silva	samysilva@email.com	www.mauriciosilva.com.br
4	Ken WILLIAMSON	ken@email.com	www.ken.com.br
5	Juliano NIEDERAUER	juliano@email.com	www.juliano.com.br
6	Prabhakar METLAPALLI	metlapalli@email.com	www.metlapalli.com.br
9	Teodorico Thickins	tthickins0@newyorker.com	https://usda.gov
10	Waneta Subhan	wsuhban1@360.cn	https://thetimes.co.uk
11	Jannel Lecount	jlecount2@cyberchimps.com	https://artisteer.com
12	Merv Knellen	mknellen3@hatena.ne.jp	http://angelfire.com
13	Tanner Hurcombe	thurcombe4@imgur.com	https://loc.gov
14	Conroy Tambling	ctambling5@redcross.org	http://europa.eu
15	Tammie Sutherland	tsutherland6@artisteer.com	http://list-manage.com
16	Lin Jaquet	ljaquet7@netlog.com	https://vinaora.com
17	Cinnamon Duckett	cduckett8@fema.gov	https://dmoz.org
18	Corina Hearnah	chearnah9@instagram.com	http://businessweek.com
19	Cathleen Burdell	cburdella@ustream.tv	http://patch.com
20	Tedd Endersby	tendersbyb@home.pl	https://desdev.cn
21	Olin Sturman	osturmanc@yahoo.co.jp	http://nydailynews.com
22	Modesty Leber	mleberd@wsj.com	http://washingtonpost.com
23	Fifi Rannald	frannalde@cornell.edu	https://ocn.ne.jp
24	Ludwig Kennally	lkennallyf@state.tx.us	http://blogger.com
25	Ellary Bleue	ebleueg@bigcartel.com	http://e-recht24.de
26	Felix Statefield	fstatefieldh@blogger.com	https://bloglines.com
27	Wyatt Barrie	wbarriei@oakley.com	http://umn.edu
28	Doug Phebee	dphebeej@icq.com	https://hexun.com

2. Importei as bibliotecas do jQuery e do Select2 nos arquivos form/FormInserirLivro.php e form/FormEditarLivro.php.



- No arquivo FormInserirLivro, corrija o nome do campo de N Edição para Editora, porque o arquivo original estava com o label errado.
- No arquivo FormInserirLivro, troquei a classe do select de class="form-control form-select" para class="select2" multiple, justamente para identificar posteriormente que esse select seria utilizado pelo framework Select2.
- Adicionei o input de autores na tela de cadastro de livro (copiei do editar e tirei a parte de consultar livros por autor). Cada option é um autor, então retirei o getCodigo do value do autor para exibir o getNome, pois no select 2 só exibia o código do autor. Com o getNome exibe o nome dele.

O próprio componente permite selecionar mais de um autor e depois adicionar todos de uma única vez.

Cadastro de Livros

Título

* Título:

ISBN

* ISBN:

N. Páginas

* Núm. Páginas:

N. Edição

* Núm. Edição:

Ano de publicação

* Ano Publicação:

Editora

Novatec

Autor

Abagael Twiddy

Abbye Batterbee

Abdul McConnachie

Adan Folke

Add

Id

Ações

Salvar

Cancelar

O próprio componente também já valida qual autor já foi selecionado na lista, portanto não permite selecionar o mesmo autor duas vezes (lembro que o professor comentou dessa validação durante a aula).

Autor

<input type="text" value="x Abdul McConnachie"/>	<button>Add</button>
<div><div>Abagael Twiddy</div><div>Abbye Batterbee</div><div>Abdul McConnachie</div><div>Adan Folke</div><div>Adelaida Luto</div><div>Adelbert Scarlet</div></div>	Ações

O próprio componente faz o filtro, sem precisar fazer novas consultas no banco. Como a pesquisa já foi feita uma vez, todos os registros já estão no componente, então ele filtra os resultados já existentes. Isso deixa a tela mais rápida, pois não dispara uma nova consulta sempre que uma letra é preenchida.

Cadastro de Livros

Título * Título:		
ISBN * ISBN:	N. Páginas * Núm. Páginas:	N. Edição * Núm. Edição:
Ano de publicação * Ano Publicação:	Editora Novatec	
Autor		
<input type="text" value="ada"/>	<button>Add</button>	
<div><div>Adan Folke</div></div>	Ações	
<button>Salvar</button> <button>Cancelar</button>		

- Adicionei id no botão Add para o Javascript conseguir encontrar o botão pelo id e pegar o evento de clique nele.

```
<?php } ?>
</select>
<button id="addautor" style="float: left" type="button" class="btn btn-outline-secondary">Add</button>
</div>
</div>
<table id="tableAutor">
```

```

<script>
    $('#select2').select2();

    $('#addautor').click(function() {
        // Obter os valores selecionados do Select2
        var selectedValues = $('#select2').val();
    });

```

7. Criei uma tabela HTML para receber os autores escolhidos no select2. Essa tabela precisa ter um ID para o Ajax identificar ela. Id colocado = "tabelaAutores". Criei as colunas Nome e Ações nela. Coluna Ações tem apenas a opção de excluir o autor.

8. Quando clica no botão Add, o comando javascript pega todos os autores escolhidos do select e coloca em uma variável lista. Para cada autor selecionado no select, uma linha HTML é concatenada por meio do método append(). Essa linha exibe o texto de cada elemento da lista (no caso, o nome do autor). Além do nome, também é adicionado um texto de excluir. Tentei colocar button, mas sempre que clicava para excluir um autor, ele tentava mandar a requisição POST. Com tag "a" funcionou.
9. Depois que o autor era selecionado na lista, a variável lista tem seu valor setado para null, portanto o componente é limpo na tela.

```

127 <script>
128
129     /// Adicionar na lista ///
130
131     $('#select2').select2();
132
133     $('#addautor').click(function() {
134         // Obter os valores selecionados do Select2
135         var selectedValues = $('#select2').val();
136
137         // Adicionar cada valor selecionado à tabela HTML
138         $.each(selectedValues, function(index, value) {
139             $('#tabelaAutores tbody').append('<tr><td>' + value + '</td><td><a class="btn-delete" data-id="'+value+'">Excluir</a></td></tr>');
140         });
141
142         // Limpar o Select2 após a seleção
143         $('#select2').val(null).trigger('change');
144
145         // Enviar os valores selecionados para o servidor usando AJAX
146         $.ajax({
147             type: 'POST',
148             data: { values: selectedValues },
149             success: function(data) {
150                 console.log('Valores enviados com sucesso!');
151             },
152             error: function() {
153                 console.log('Erro ao enviar valores!');
154             }
155         });
156     });
157

```