Министерство образования и науки РФ

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра Информационные технологии и автоматизированные системы

Базы данных

Лабораторная работа № 4

Тема: «Основные команды SQL: DDL, DML.»

Выполнил: студент группы Мокрушин Н.Д.

Проверил: доцент кафедры ИТАС

Петренко А.А.

г. Пермь – 2024

Оглавление

Цель работы 3

Задачи работы 4

Этапы выполнения 5

Заключение 6

Список используемой литературы 7

# Цель работы

Вывести результаты демонстрации команд DDL и DML + INNER JOIN на страницу браузера используя бесплатное ПО OpenServer

# Задачи работы

# 1. изучить команды DDL;

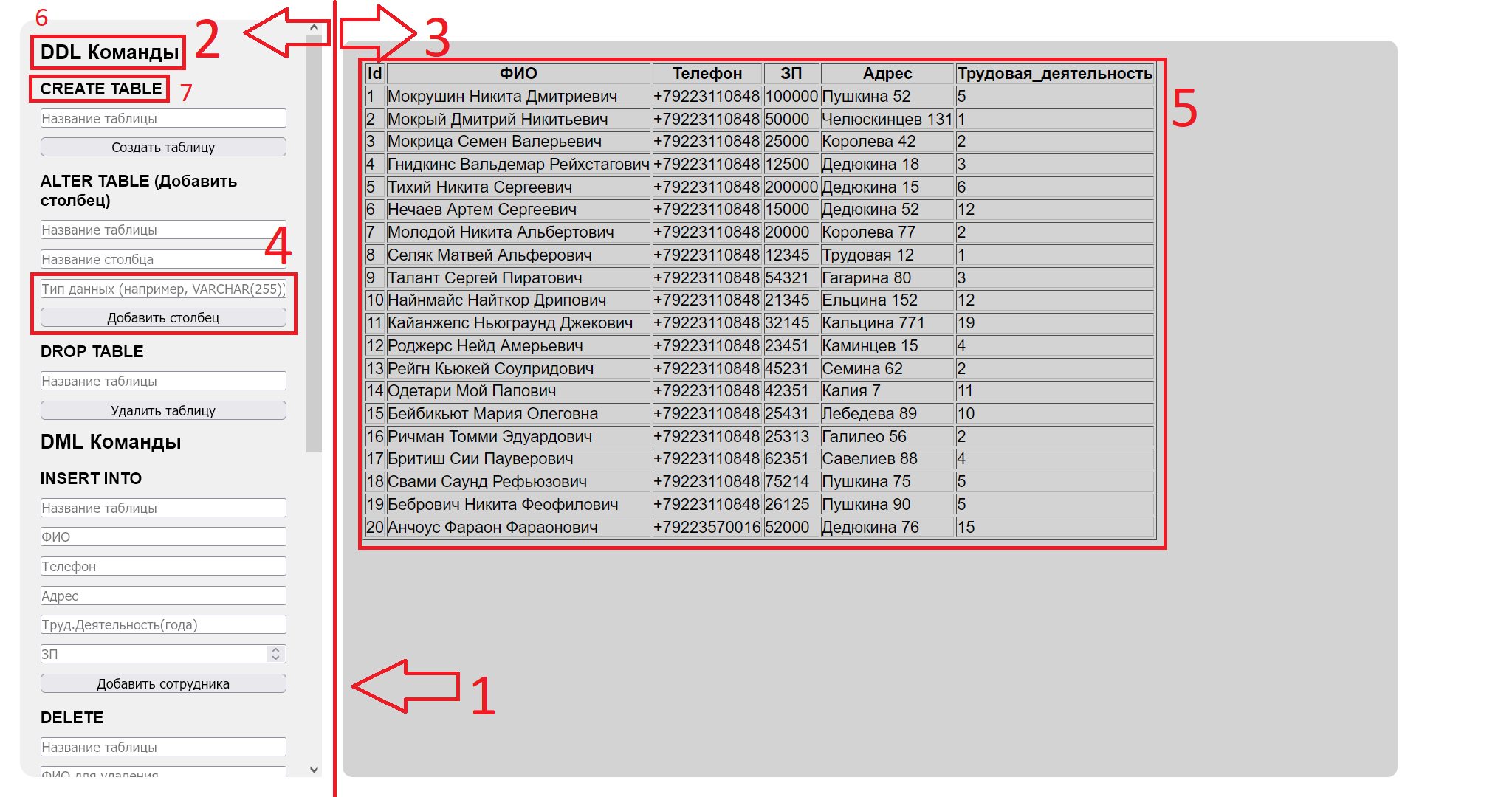
# 2. изучить команды DML;

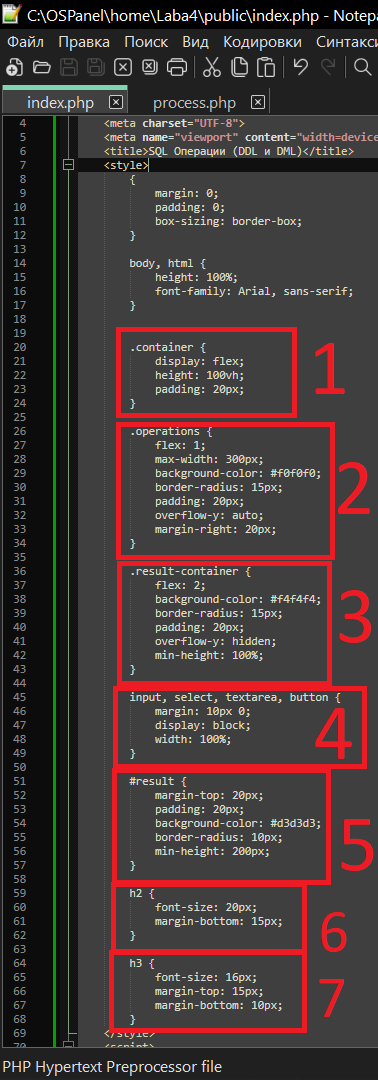
# 3. разработать страницу, демонстрирующую все команды DDL и DML + INNER JOIN;

# 4. вывести результат работы на страницу браузера;

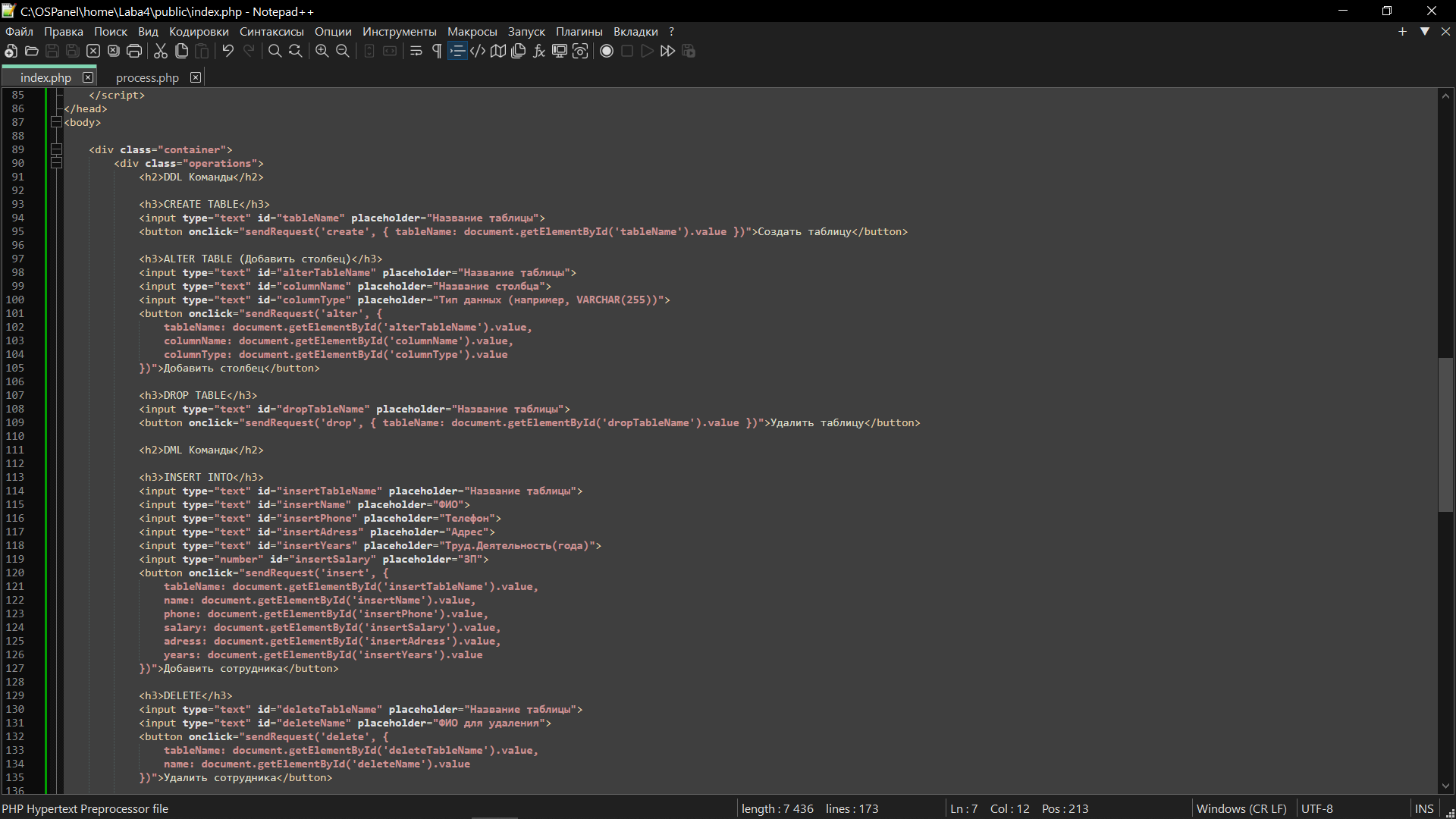
# Этапы выполнения

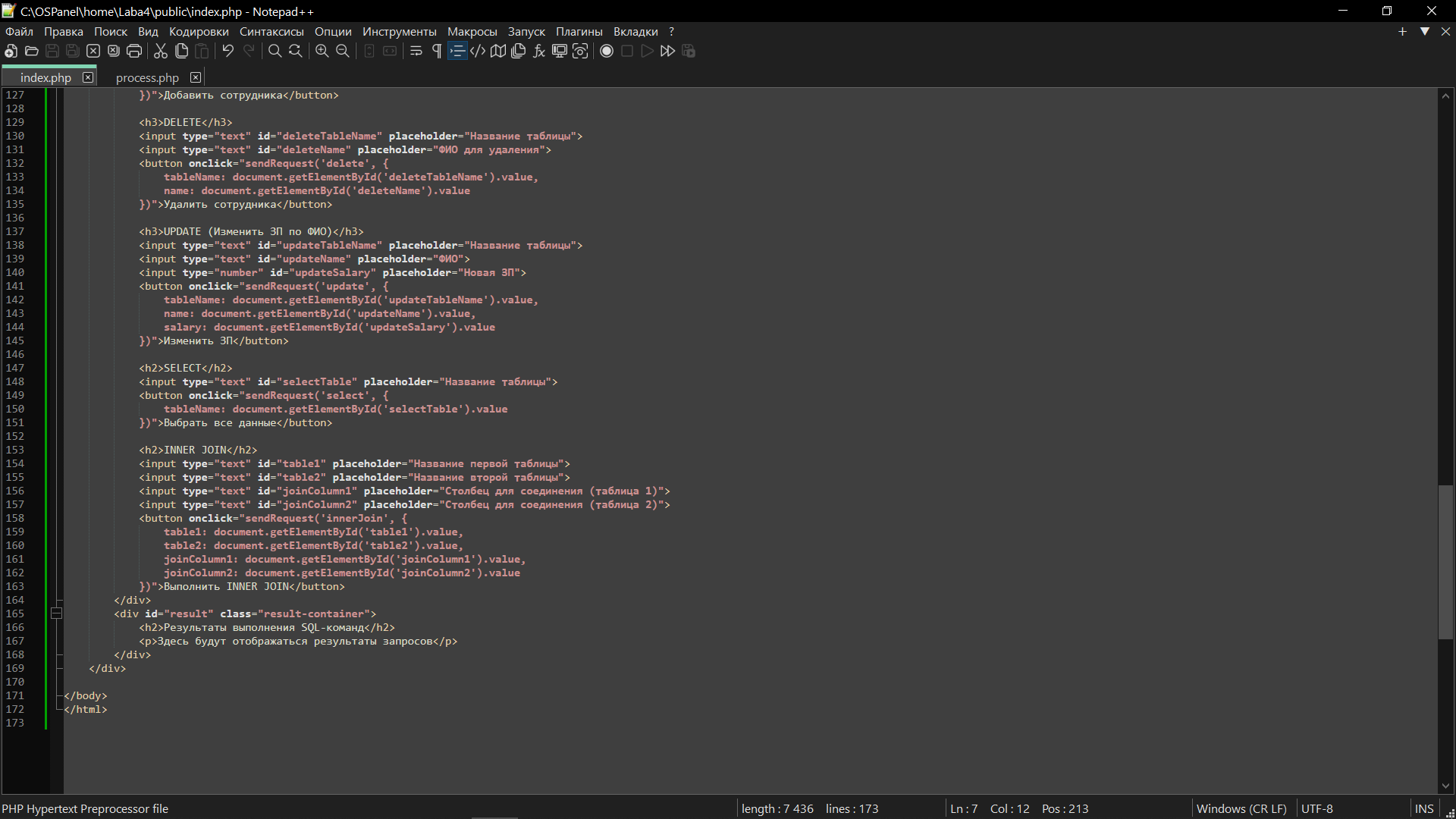
1. **Оформление страницы**



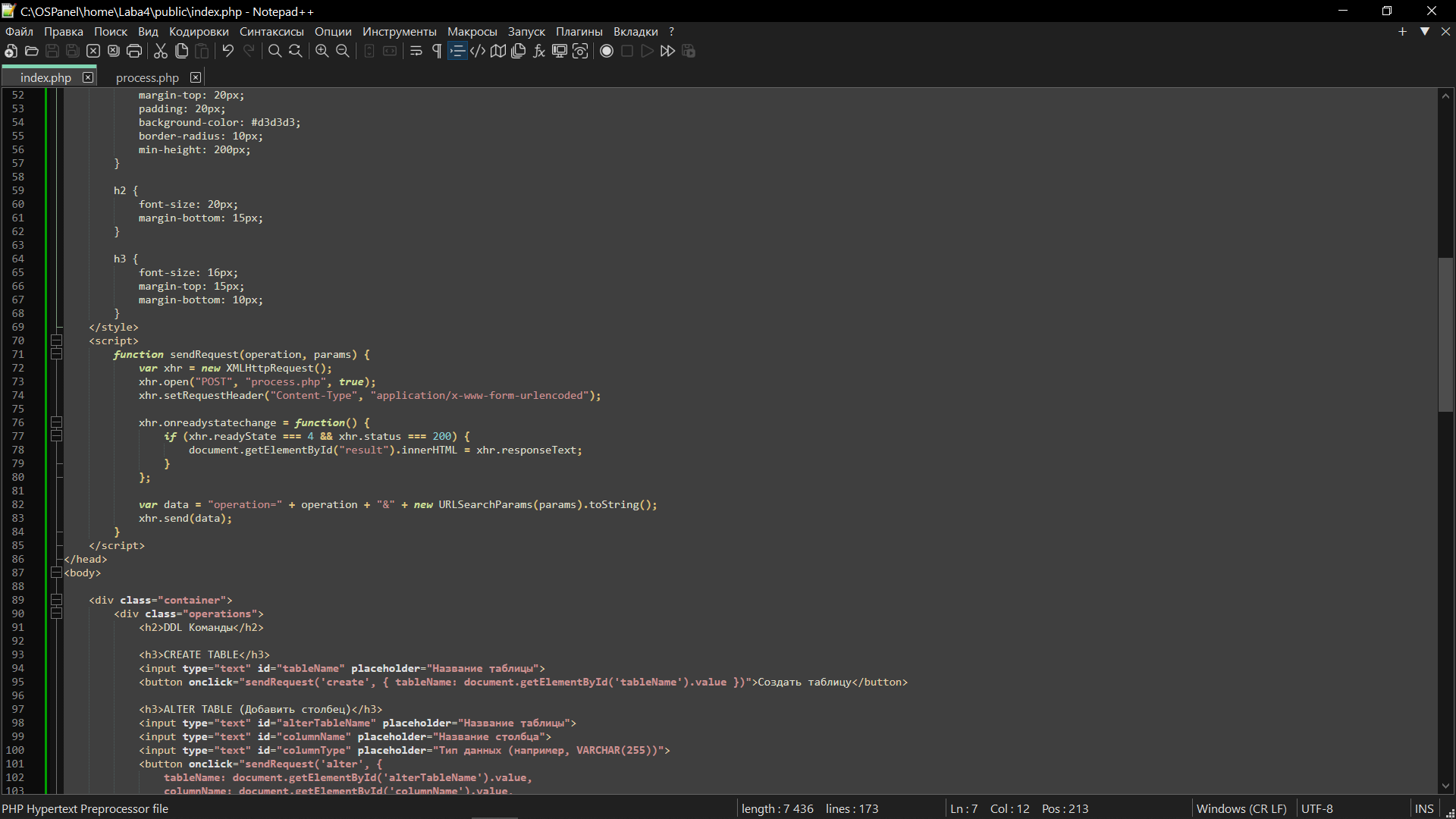


1. Стиль для класса контейнер flex разметки, который разделяет полотно страницы на две части: operations(2) и result-container(3)
2. Стиль для класса operations с локальным ползунком(overflow-y: auto;), в котором будут находиться кнопки, инпут-поля для реализации операций
3. Стиль для класса result-container со статичным положением на странице(overflow-y: hidden), на котором будут находиться результаты вывода операций(либо сообщение о успешном выполнении операции, либо вывод таблицы)
4. Стиль для классов инпут-полей и кнопок
5. Стиль для вывода результата конкретной операции
6. Стиль для класса заголовка
7. Стиль для класса подзаголовка
8. **Размещение классов на полотне страницы**





1. **JavaScript функция для отправления POST запроса в php код**



В параметры функции **sendRequest** передаются 2 переменные: тип операции и параметры для выполнения операции.

Создается объект XMLHttpRequest (**xhr**), который используется для отправки HTTP-запроса к серверу.

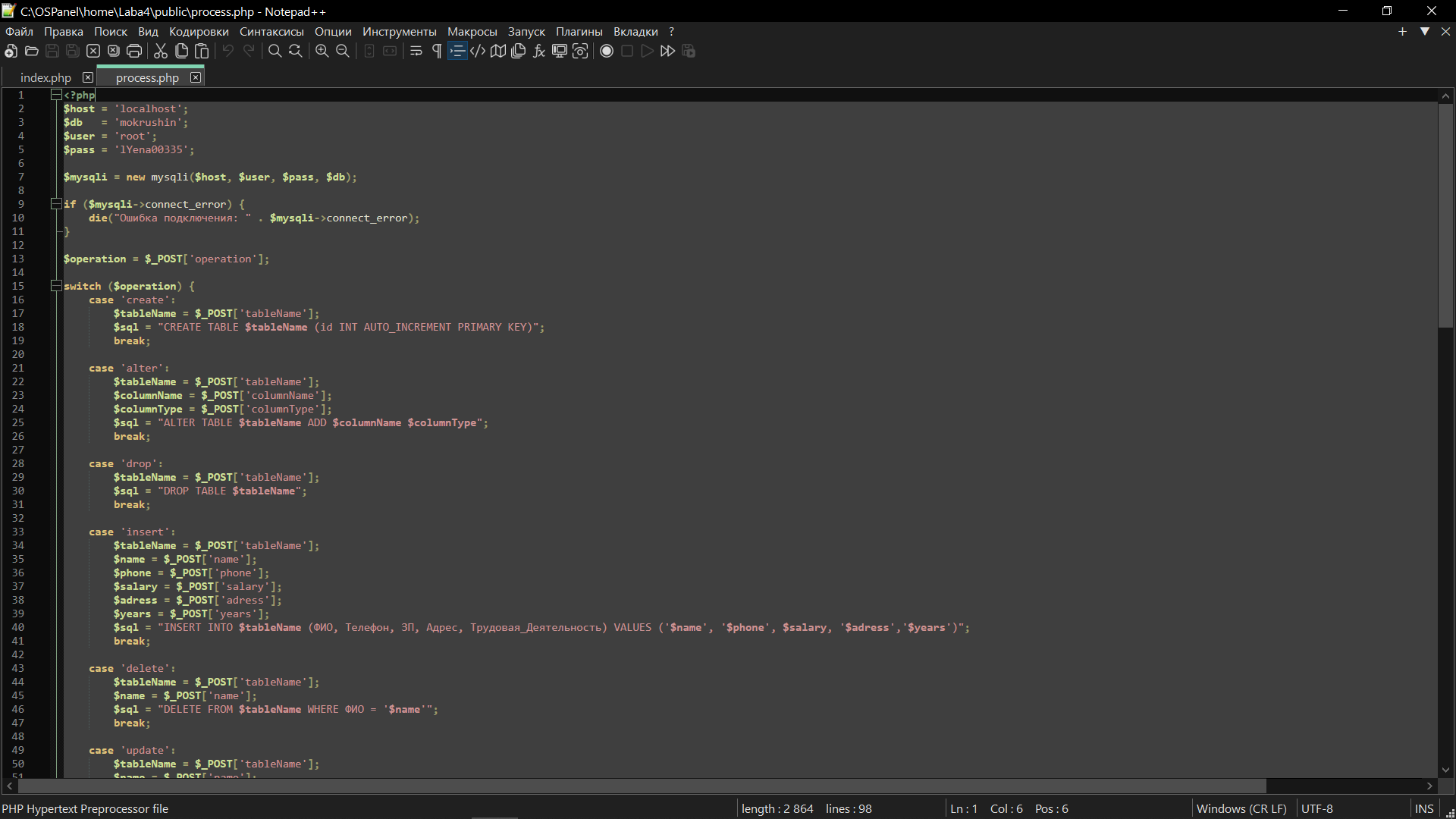
**xhr.open** — создает новый POST-запрос к файлу process.php (обработчик на сервере).

**xhr.setRequestHeader** — устанавливает тип передаваемых данных (закодированный в формате application/x-www-form-urlencoded).

**xhr.onreadystatechange** — назначается функция для обработки ответа сервера. Если запрос успешно завершен (readyState === 4 и status === 200), результат выводится в элемент с ID **result**.

**xhr.send(data)** — отправляет данные на сервер. Здесь формируется строка запроса, состоящая из операции и параметров.

1. Разбор php кода
2. Подключение к БД



1. Получение типа операции из POST запроса и конструкция Switch для определения дальнейших действий

2.1) Create – получает из POST запроса имя таблицы, и создает по умолчанию в этой таблице ключевое поле **id** типа int с авто инкрементацией и выполняет sql команду на основе данных, на странице браузера появляется надпись «операция успешно выполнена»

2.2) Alter – получает из POST запроса имя таблицы, имя колонки и тип данных, после выполняет sql запрос на основе данных, на странице браузера появляется надпись «операция успешно выполнена»

2.3) Drop – получает из POST запроса имя таблицы и выполняет команду на основе данных, на странице браузера появляется надпись «операция успешно выполнена»

2.4) Insert – получает из POST запроса имя таблицы, ФИО, телефон, зп, адрес, года стажировки и выполняет команду, на странице браузера появляется надпись «операция успешно выполнена» (**INSERT INTO очень чувствительная к структуре таблицы команда, если в таблице будет на одно поле больше/меньше или какое-то поле будет отличаться от запроса в консоль будет выводиться ошибка, поэтому было принято решение сделать команду для конкретной таблицы но все еще давать возможность пользователю выбрать имя таблицы**)

2.5) Delete – получает из POST запроса имя таблицы и ФИО для удаления строки и выполняет команду на основе данных, на странице браузера появляется надпись «операция успешно выполнена»

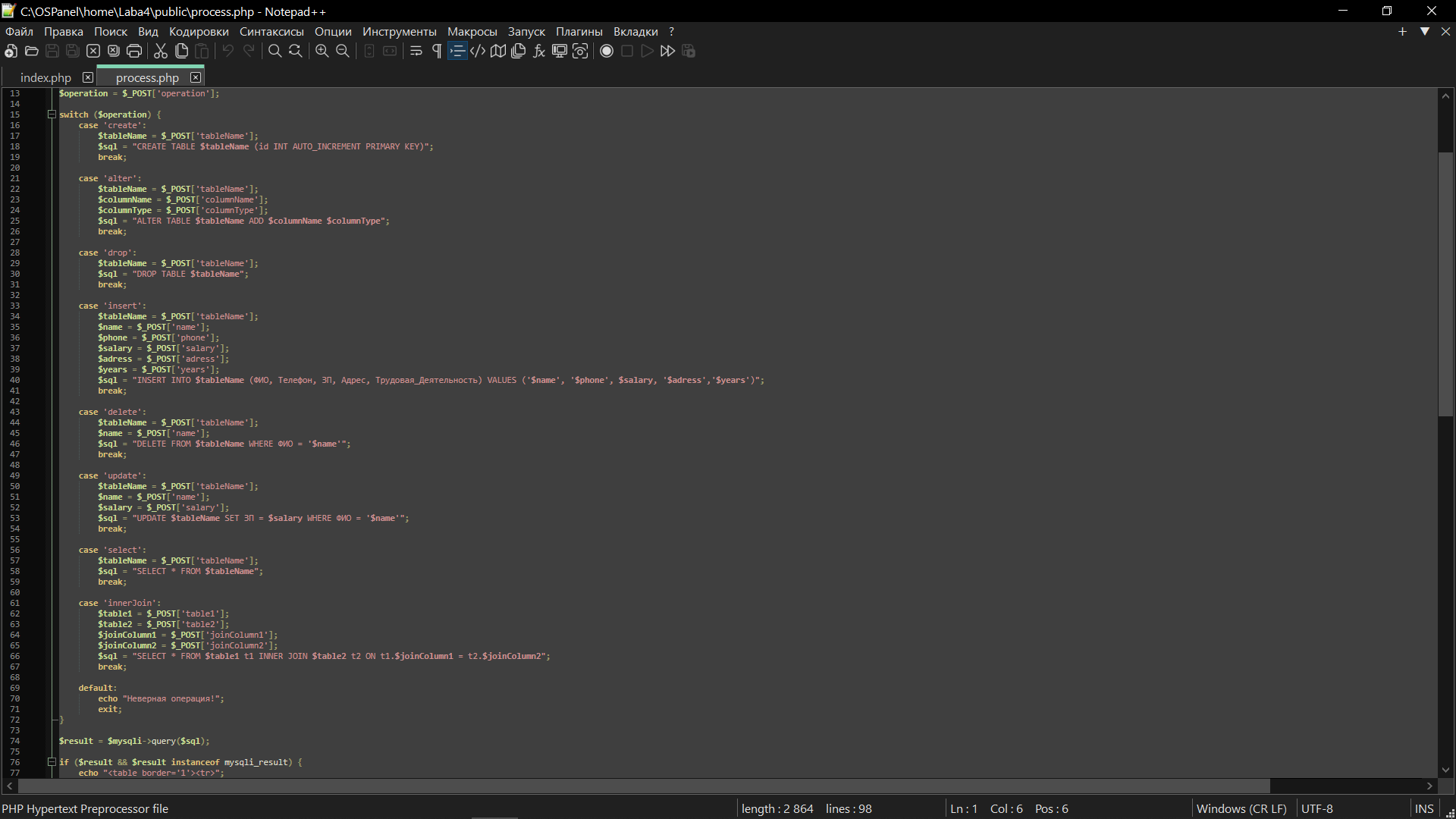
2.6) Update – получает из POST запроса имя таблицы ФИО и новое значение ЗП, для замены старого значения ЗП и выполняет команду на основе данных, на странице браузера появляется надпись «операция успешно выполнена»

2.7) Select – получает из POST запроса имя таблицы и выводит полную таблицу на страницу браузера, на странице браузера появляется таблица

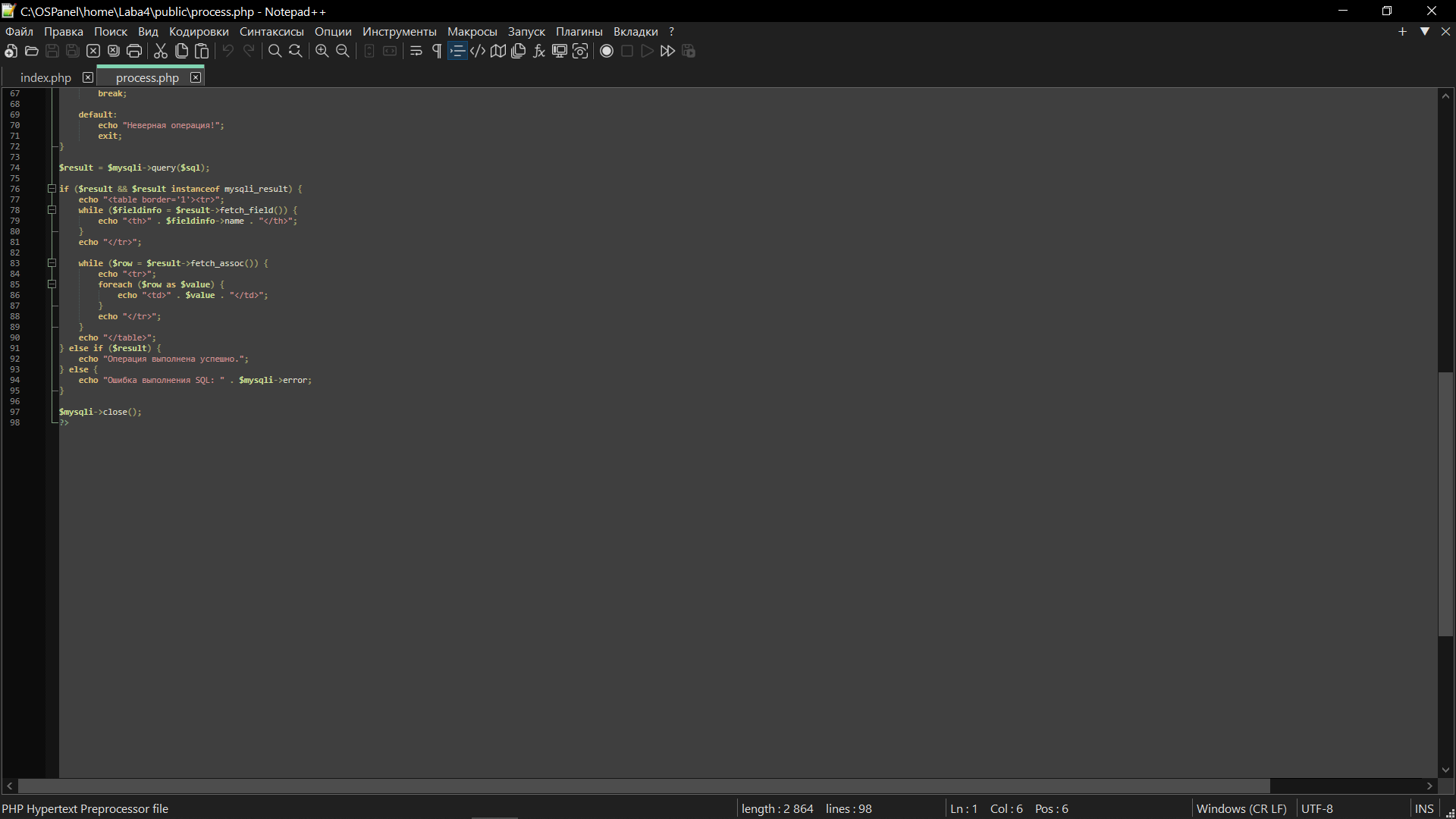
2.8) Inner Join – получает из POST запроса два имени таблицы, и две колонки для объединения, выполняет команду, на странице браузера появляется таблица

2.9) Default – если из POST запроса получена некорректная/неизвестная/несуществующая операция на странице браузера будет выведено «неверная операция»

2.10) Truncate – получает из POST запроса имя таблицы, выполняет команду, на странице браузера появляется надпись «операция успешно выполнена»



1. Построение таблицы



Выполняется SQL-запрос, который хранится в переменной **$sql,** результат запроса сохраняется в переменную **$result.**

Создаем HTML-таблицу и проходим по полям результата (fetch\_field), чтобы вывести заголовки столбцов.

Затем перебираем строки (fetch\_assoc), и для каждой строки создаем <tr> с ячейками <td>, где отображаются значения полей.

В конце скрипта мы закрываем соединение с базой данных, чтобы освободить ресурсы.

1. **Демонстрация работы команд**
2. Create



1. Alter

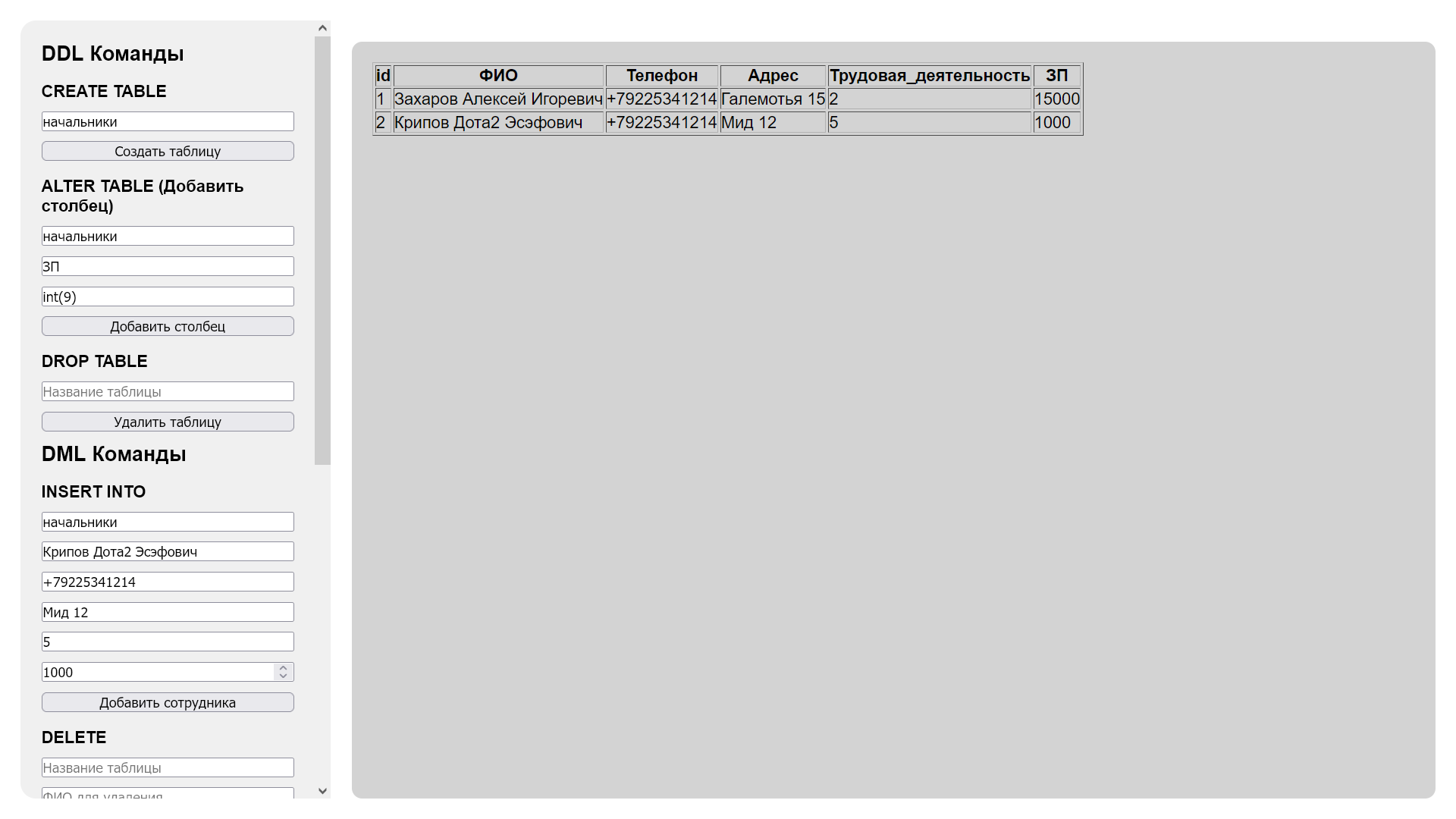


2.1) Заполняем структуру таблицы



1. Truncate

До:



После:

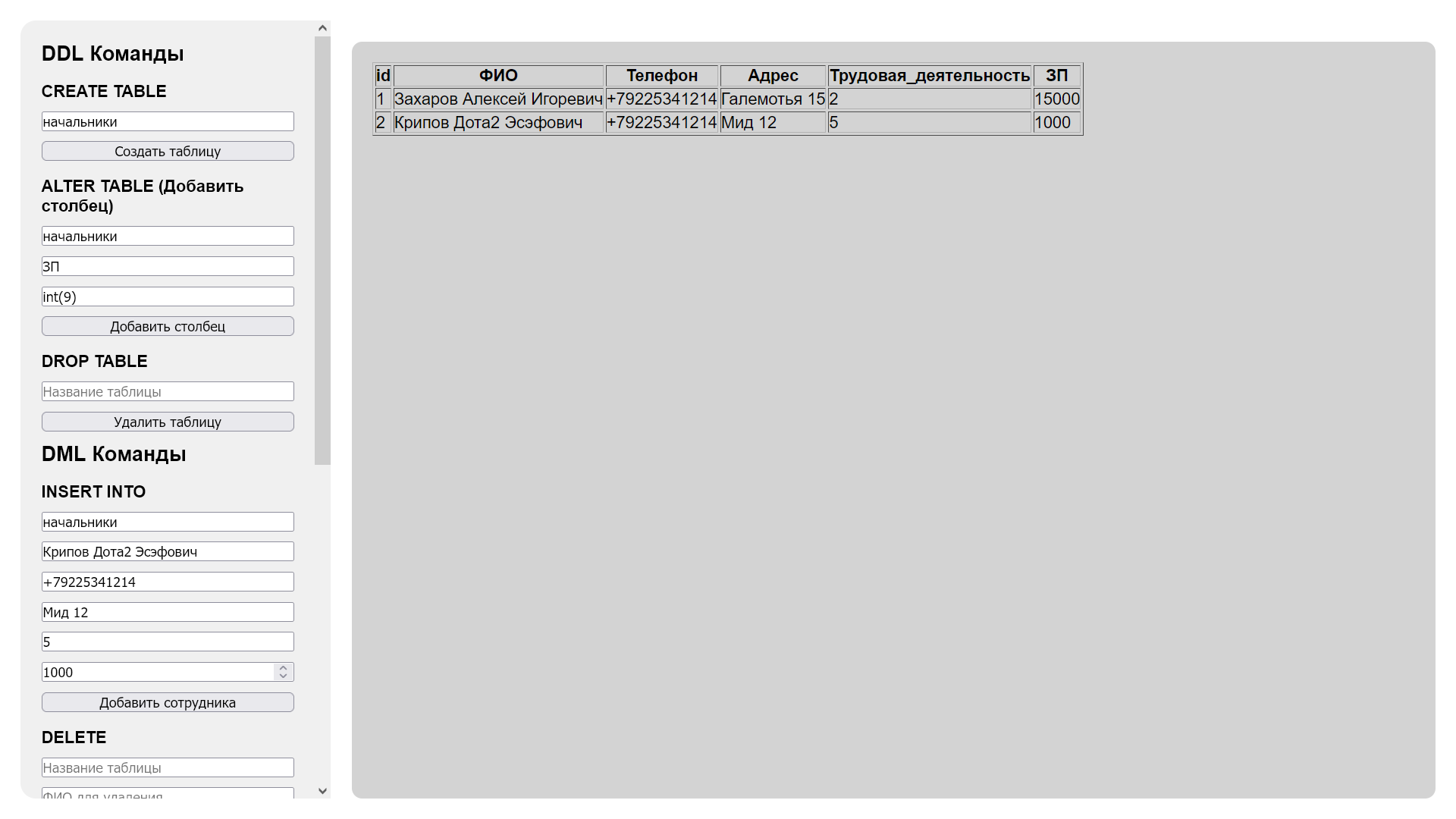


1. Пока что пропустим команду Drop Table, и перейдем к Insert Into

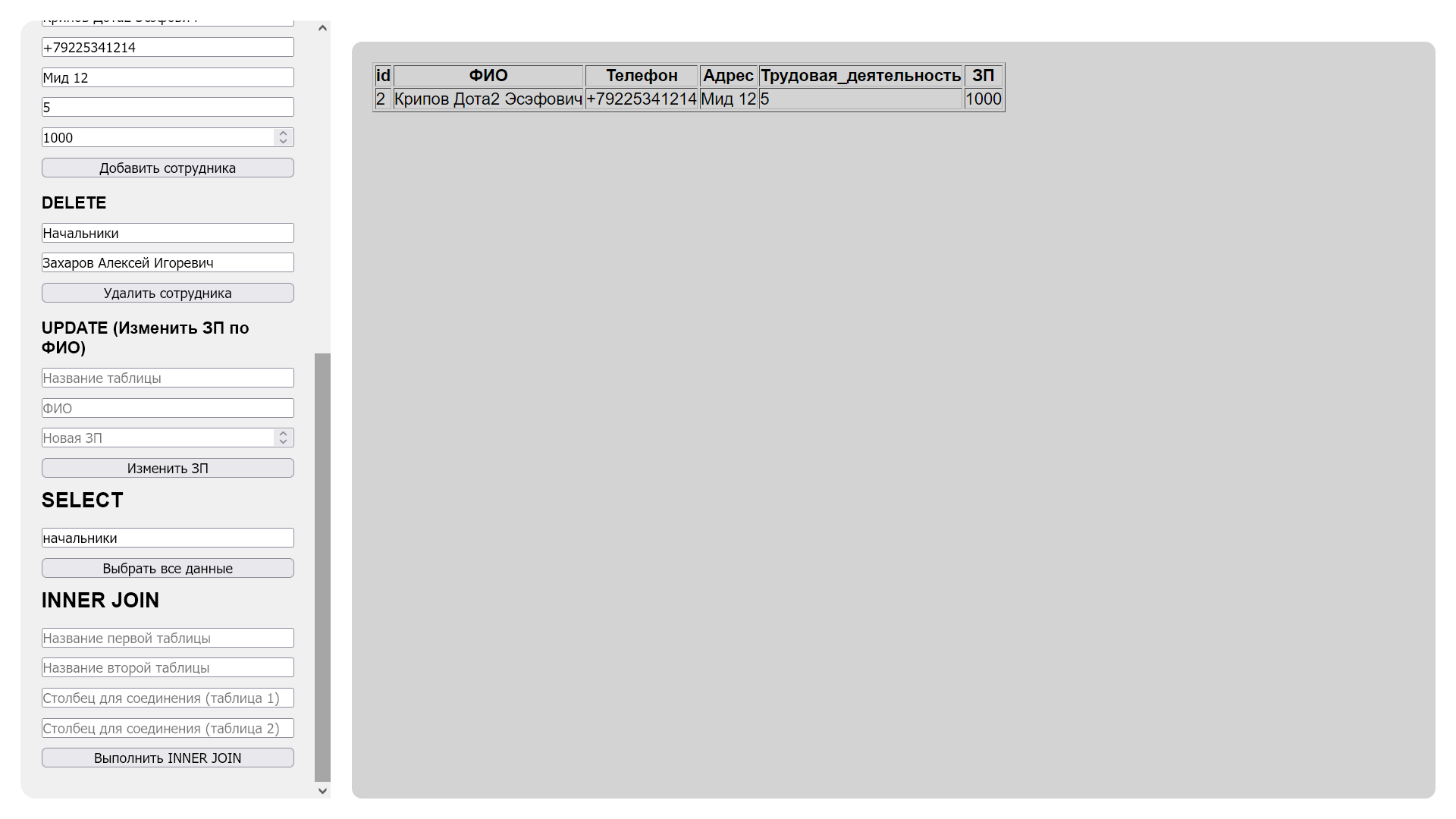


1. Delete

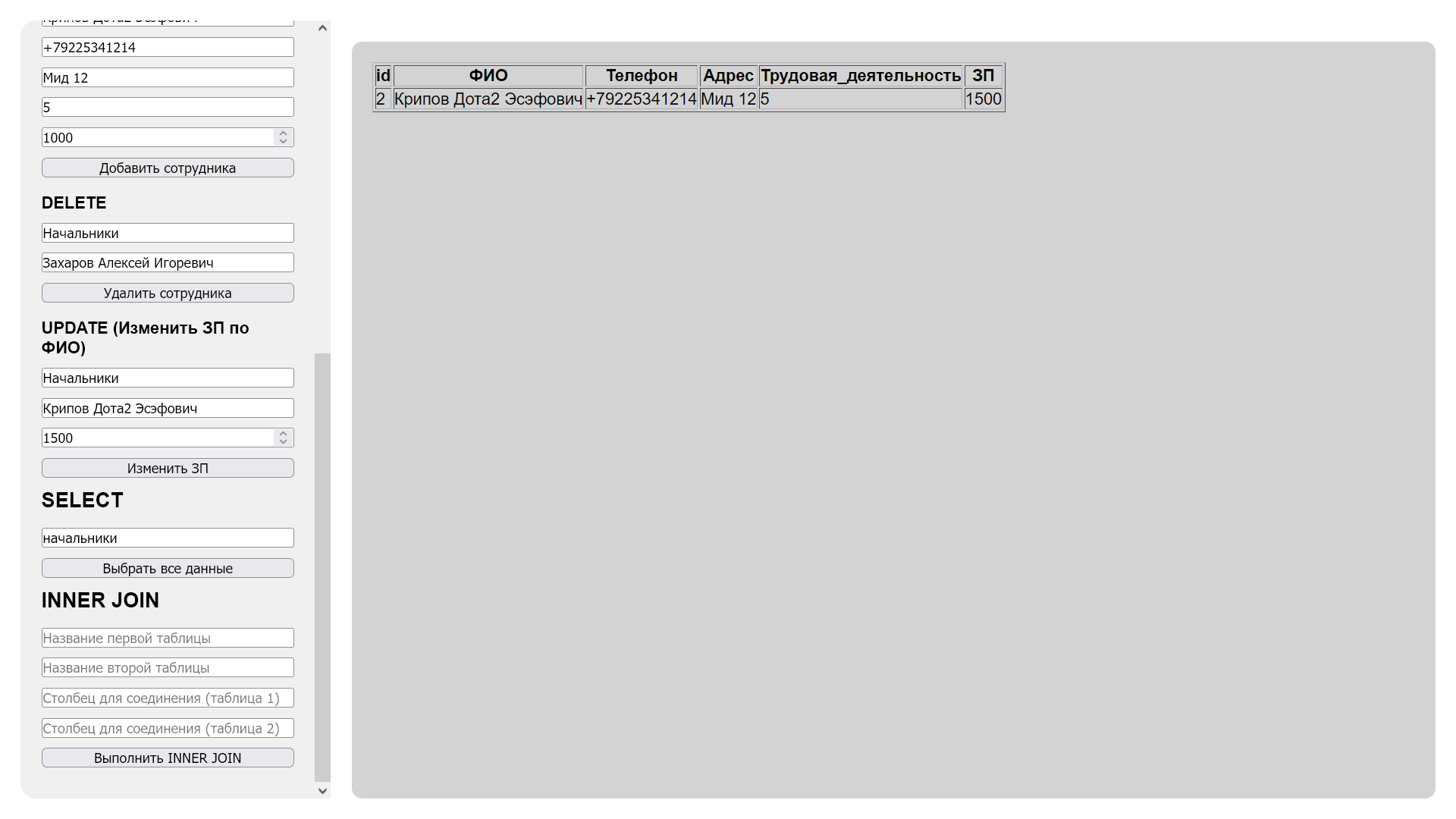
До:



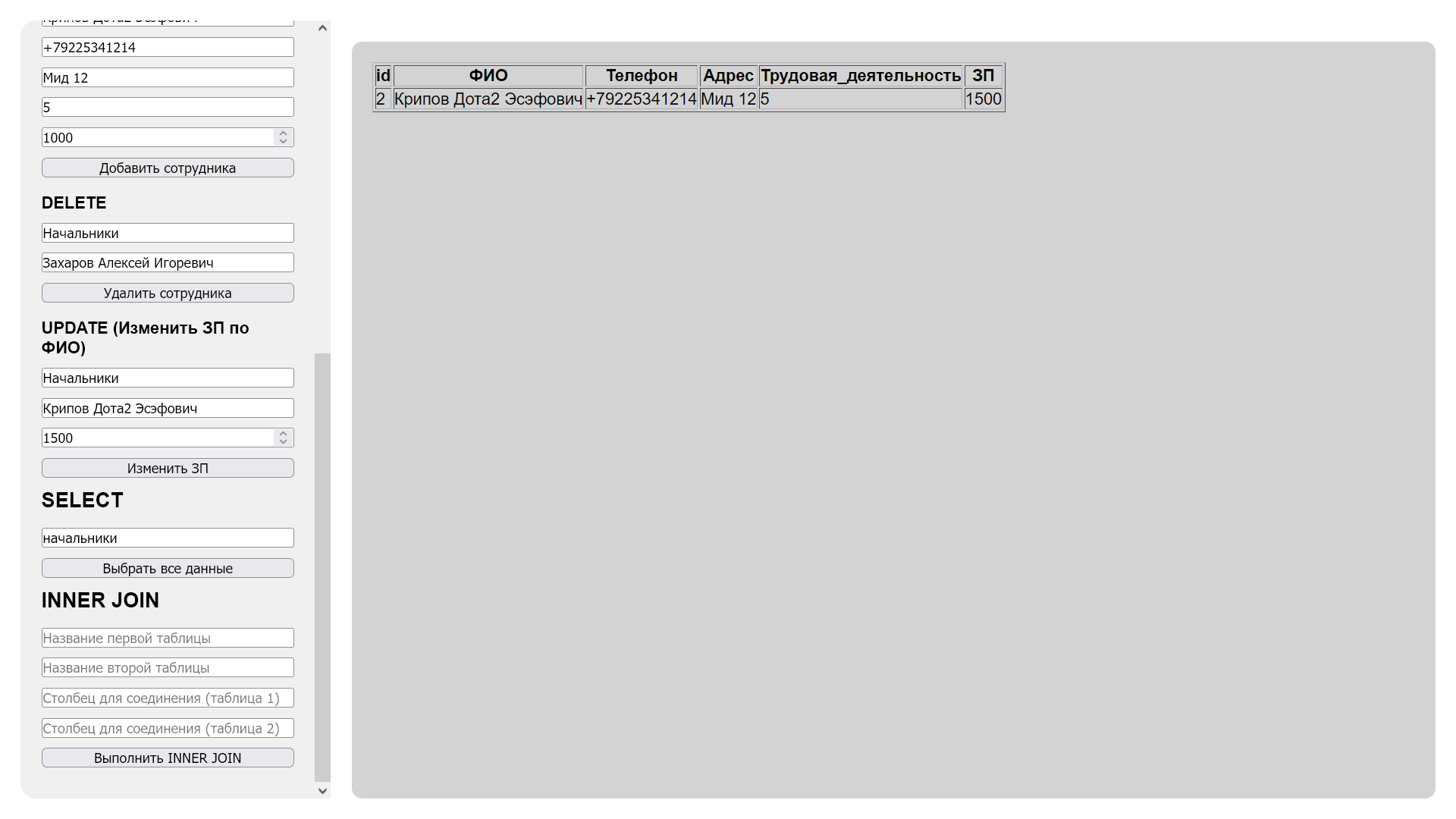
После:



1. Update

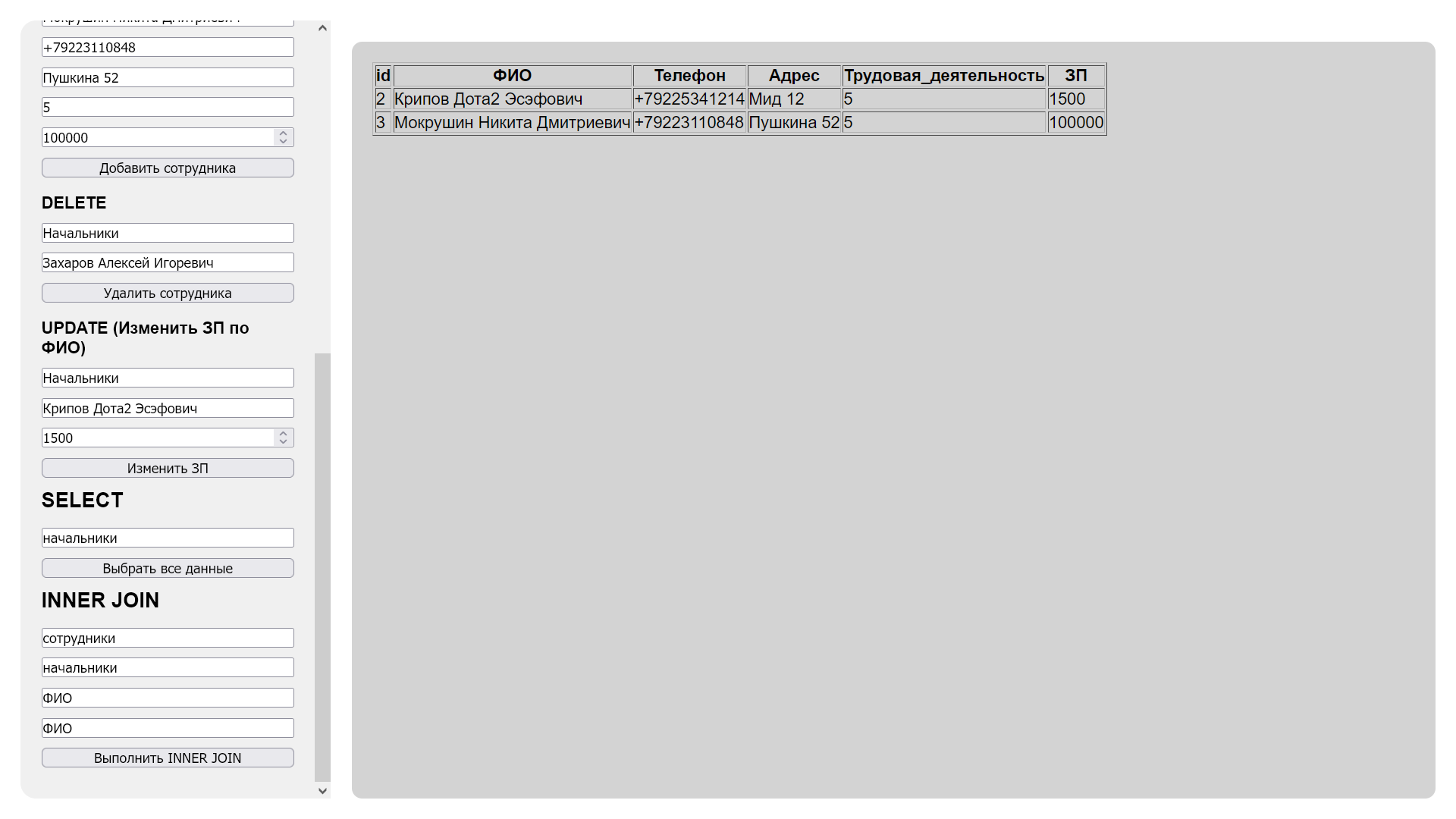


1. Select

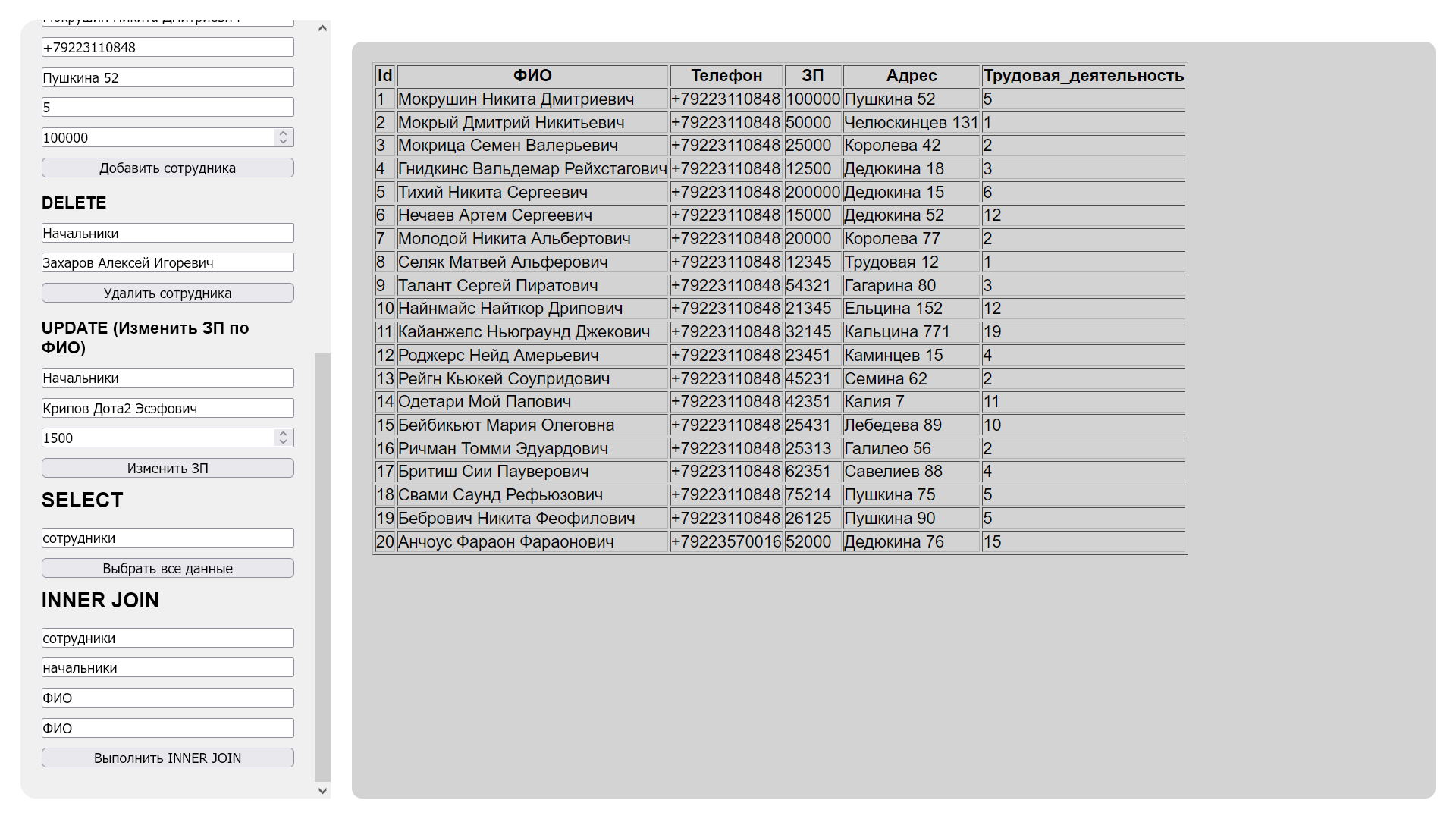


1. Inner Join

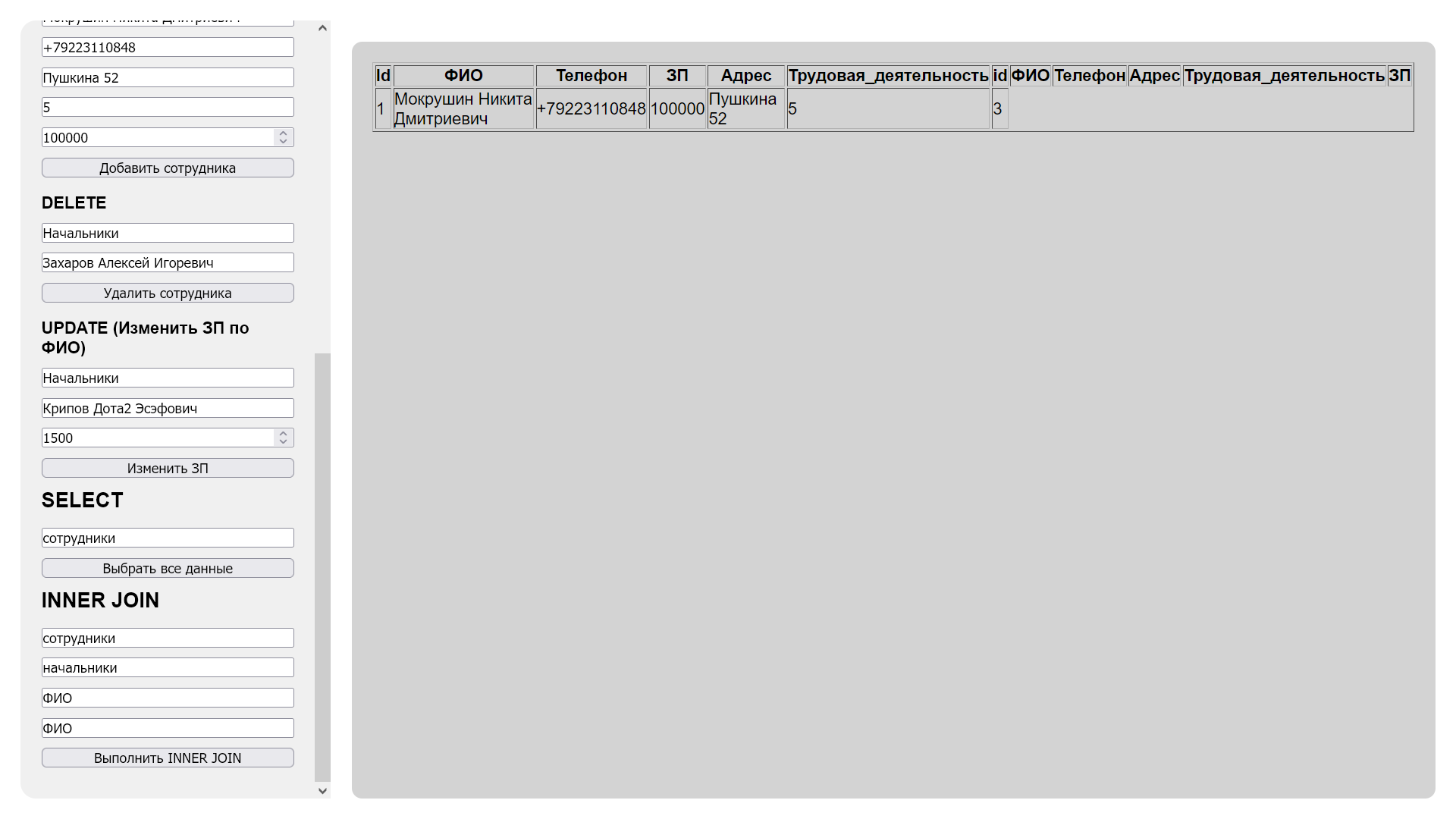
Начальники:



Сотрудники:



Inner Join:



1. Drop Table



После удаления выполним Select для удаленной таблицы:



Как можно увидеть, данной таблицы больше не существует и на страницу выводится ошибка

# 

Код HTML:

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>SQL Операции (DDL и DML)</title>

<style>

{

margin: 0;

padding: 0;

box-sizing: border-box;

}

body, html {

height: 100%;

font-family: Arial, sans-serif;

}

.container {

display: flex;

height: 100vh;

padding: 20px;

}

.operations {

flex: 1;

max-width: 300px;

background-color: #f0f0f0;

border-radius: 15px;

padding: 20px;

overflow-y: auto;

margin-right: 20px;

}

.result-container {

flex: 2;

background-color: #f4f4f4;

border-radius: 15px;

padding: 20px;

overflow-y: hidden;

min-height: 100%;

}

input, select, textarea, button {

margin: 10px 0;

display: block;

width: 100%;

}

#result {

margin-top: 20px;

padding: 20px;

background-color: #d3d3d3;

border-radius: 10px;

min-height: 200px;

}

h2 {

font-size: 20px;

margin-bottom: 15px;

}

h3 {

font-size: 16px;

margin-top: 15px;

margin-bottom: 10px;

}

</style>

<script>

function sendRequest(operation, params) {

var xhr = new XMLHttpRequest();

xhr.open("POST", "process.php", true);

xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");

xhr.onreadystatechange = function() {

if (xhr.readyState === 4 && xhr.status === 200) {

document.getElementById("result").innerHTML = xhr.responseText;

}

};

var data = "operation=" + operation + "&" + new URLSearchParams(params).toString();

xhr.send(data);

}

</script>

</head>

<body>

<div class="container">

<div class="operations">

<h2>DDL Команды</h2>

<h3>CREATE TABLE</h3>

<input type="text" id="tableName" placeholder="Название таблицы">

<button onclick="sendRequest('create', { tableName: document.getElementById('tableName').value })">Создать таблицу</button>

<h3>ALTER TABLE (Добавить столбец)</h3>

<input type="text" id="alterTableName" placeholder="Название таблицы">

<input type="text" id="columnName" placeholder="Название столбца">

<input type="text" id="columnType" placeholder="Тип данных (например, VARCHAR(255))">

<button onclick="sendRequest('alter', {

tableName: document.getElementById('alterTableName').value,

columnName: document.getElementById('columnName').value,

columnType: document.getElementById('columnType').value

})">Добавить столбец</button>

<h3>DROP TABLE</h3>

<input type="text" id="dropTableName" placeholder="Название таблицы">

<button onclick="sendRequest('drop', { tableName: document.getElementById('dropTableName').value })">Удалить таблицу</button>

<h3>TRUNCATE TABLE</h3>

<input type="text" id="truncateTableName" placeholder="Название таблицы">

<button onclick="sendRequest('truncate', { tableName: document.getElementById('truncateTableName').value })">Очистить таблицу</button>

<h2>DML Команды</h2>

<h3>INSERT INTO</h3>

<input type="text" id="insertTableName" placeholder="Название таблицы">

<input type="text" id="insertName" placeholder="ФИО">

<input type="text" id="insertPhone" placeholder="Телефон">

<input type="text" id="insertAdress" placeholder="Адрес">

<input type="text" id="insertYears" placeholder="Труд.Деятельность(года)">

<input type="number" id="insertSalary" placeholder="ЗП">

<button onclick="sendRequest('insert', {

tableName: document.getElementById('insertTableName').value,

name: document.getElementById('insertName').value,

phone: document.getElementById('insertPhone').value,

salary: document.getElementById('insertSalary').value,

adress: document.getElementById('insertAdress').value,

years: document.getElementById('insertYears').value

})">Добавить сотрудника</button>

<h3>DELETE</h3>

<input type="text" id="deleteTableName" placeholder="Название таблицы">

<input type="text" id="deleteName" placeholder="ФИО для удаления">

<button onclick="sendRequest('delete', {

tableName: document.getElementById('deleteTableName').value,

name: document.getElementById('deleteName').value

})">Удалить сотрудника</button>

<h3>UPDATE (Изменить ЗП по ФИО)</h3>

<input type="text" id="updateTableName" placeholder="Название таблицы">

<input type="text" id="updateName" placeholder="ФИО">

<input type="number" id="updateSalary" placeholder="Новая ЗП">

<button onclick="sendRequest('update', {

tableName: document.getElementById('updateTableName').value,

name: document.getElementById('updateName').value,

salary: document.getElementById('updateSalary').value

})">Изменить ЗП</button>

<h2>SELECT</h2>

<input type="text" id="selectTable" placeholder="Название таблицы">

<button onclick="sendRequest('select', {

tableName: document.getElementById('selectTable').value

})">Выбрать все данные</button>

<h2>INNER JOIN</h2>

<input type="text" id="table1" placeholder="Название первой таблицы">

<input type="text" id="table2" placeholder="Название второй таблицы">

<input type="text" id="joinColumn1" placeholder="Столбец для соединения (таблица 1)">

<input type="text" id="joinColumn2" placeholder="Столбец для соединения (таблица 2)">

<button onclick="sendRequest('innerJoin', {

table1: document.getElementById('table1').value,

table2: document.getElementById('table2').value,

joinColumn1: document.getElementById('joinColumn1').value,

joinColumn2: document.getElementById('joinColumn2').value

})">Выполнить INNER JOIN</button>

</div>

<div id="result" class="result-container">

<h2>Результаты выполнения SQL-команд</h2>

<p>Здесь будут отображаться результаты запросов</p>

</div>

</div>

</body>

</html>

# Код PHP:

<?php

$host = 'localhost';

$db = 'mokrushin';

$user = 'root';

$pass = 'lYena00335';

$mysqli = new mysqli($host, $user, $pass, $db);

if ($mysqli->connect\_error) {

die("Ошибка подключения: " . $mysqli->connect\_error);

}

$operation = $\_POST['operation'];

switch ($operation) {

case 'create':

$tableName = $\_POST['tableName'];

$sql = "CREATE TABLE $tableName (id INT AUTO\_INCREMENT PRIMARY KEY)";

break;

case 'alter':

$tableName = $\_POST['tableName'];

$columnName = $\_POST['columnName'];

$columnType = $\_POST['columnType'];

$sql = "ALTER TABLE $tableName ADD $columnName $columnType";

break;

case 'drop':

$tableName = $\_POST['tableName'];

$sql = "DROP TABLE $tableName";

break;

case 'truncate':

$tableName = $\_POST['tableName'];

$sql = "TRUNCATE TABLE $tableName";

break;

case 'insert':

$tableName = $\_POST['tableName'];

$name = $\_POST['name'];

$phone = $\_POST['phone'];

$salary = $\_POST['salary'];

$adress = $\_POST['adress'];

$years = $\_POST['years'];

$sql = "INSERT INTO $tableName (ФИО, Телефон, ЗП, Адрес, Трудовая\_Деятельность) VALUES ('$name', '$phone', $salary, '$adress','$years')";

break;

case 'delete':

$tableName = $\_POST['tableName'];

$name = $\_POST['name'];

$sql = "DELETE FROM $tableName WHERE ФИО = '$name'";

break;

case 'update':

$tableName = $\_POST['tableName'];

$name = $\_POST['name'];

$salary = $\_POST['salary'];

$sql = "UPDATE $tableName SET ЗП = $salary WHERE ФИО = '$name'";

break;

case 'select':

$tableName = $\_POST['tableName'];

$sql = "SELECT \* FROM $tableName";

break;

case 'innerJoin':

$table1 = $\_POST['table1'];

$table2 = $\_POST['table2'];

$joinColumn1 = $\_POST['joinColumn1'];

$joinColumn2 = $\_POST['joinColumn2'];

$sql = "SELECT \* FROM $table1 t1 INNER JOIN $table2 t2 ON t1.$joinColumn1 = t2.$joinColumn2";

break;

default:

echo "Неверная операция!";

exit;

}

$result = $mysqli->query($sql);

if ($result && $result instanceof mysqli\_result) {

echo "<table border='1'><tr>";

while ($fieldinfo = $result->fetch\_field()) {

echo "<th>" . $fieldinfo->name . "</th>";

}

echo "</tr>";

while ($row = $result->fetch\_assoc()) {

echo "<tr>";

foreach ($row as $value) {

echo "<td>" . $value . "</td>";

}

echo "</tr>";

}

echo "</table>";

} else if ($result) {

echo "Операция выполнена успешно.";

} else {

echo "Ошибка выполнения SQL: " . $mysqli->error;

}

$mysqli->close();

?>

# Заключение

В ходе выполнения лабораторной работы были успешно достигнуты все поставленные цели и задачи. Были изучены команды DDL и DML + INNER JOIN, так же были продемонстрирован вывод всех команд на страницу браузера.

# Список используемой литературы

* Welling, L., & Thomson, L. (2009). *PHP и MySQL. Разработка web-приложений*. Символ-Плюс.
* DDL, DML, DCL и TCL в языке SQL - <https://sky.pro/media/gruppy-operatorov-sql/#title1>
* Горшенин, А. (2017). *Основы создания веб-приложений с использованием PHP и MySQL*. Питер.