Министерство образования и науки РФ

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра Информационные технологии и автоматизированные системы

Базы данных

Лабораторная работа № 5

Тема: «Триггеры в СУБД»

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 работы триггера на страницу браузера из таблицы LOG используя бесплатное ПО OpenServer

# Задачи работы

1. Изучить понятие триггер;

2. Создать триггеры (на добавление, изменение и удаление) в базе данных

лабораторной работы № 5, создав в БД дополнительную таблицу LOG;

3. Реализовать демонстрацию триггера на сайте.

# Этапы выполнения

1. **Создание таблицы LOG**

CREATE TABLE LOG (

id INT AUTO\_INCREMENT PRIMARY KEY,

action\_time TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

action\_type VARCHAR(10),

table\_name VARCHAR(50),

old\_data TEXT,

new\_data TEXT

);

Используя данный SQL-код, создадим таблицу LOG, где:

**Id** – Первичный ключ с автоинкрементном,

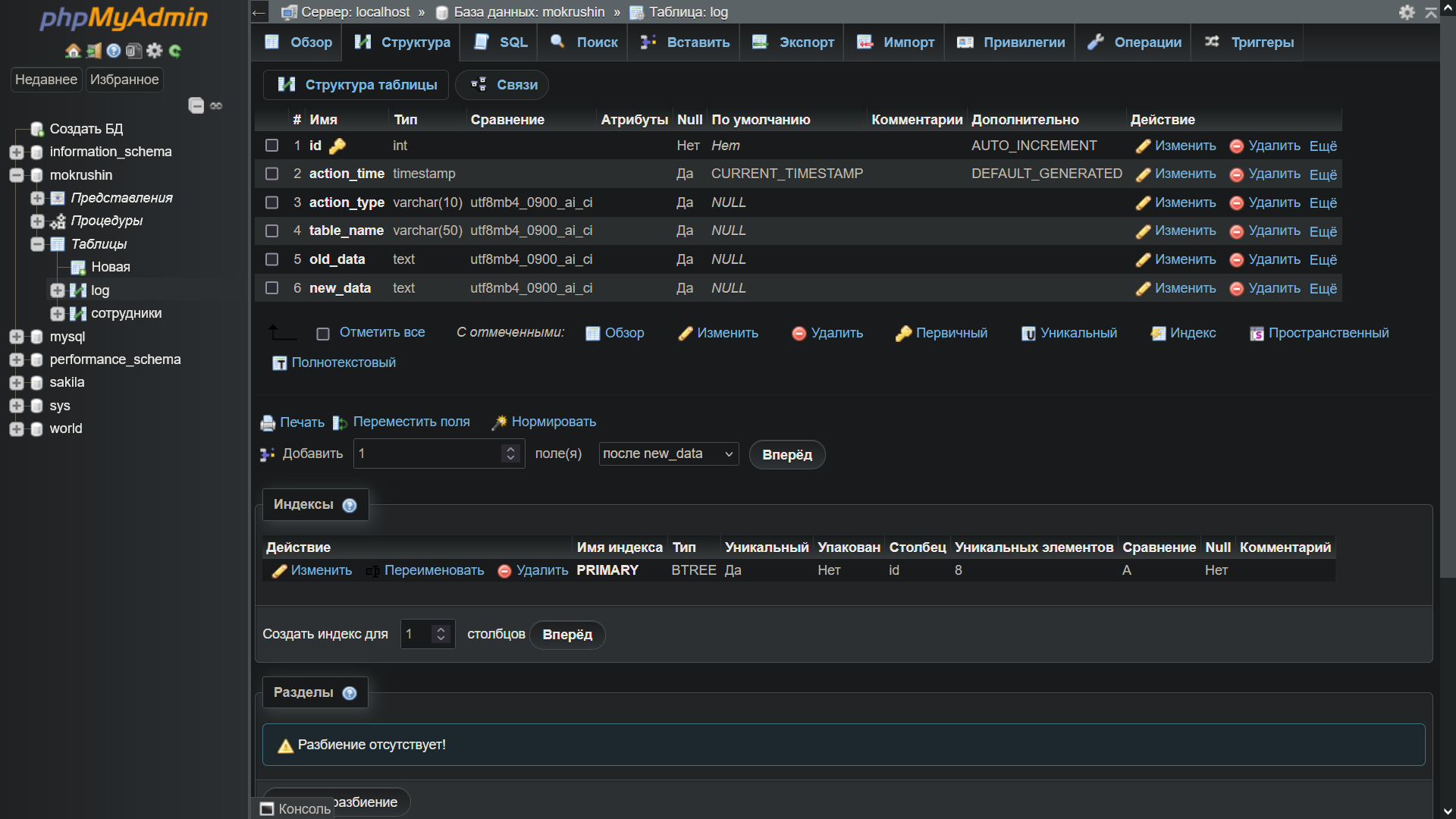
**action\_time** – Атрибут, хранящий в себе время и дату(тип timestamp, default current\_timestamp - Устанавливает текущее время по умолчанию при добавлении новой записи.)

**action\_type** – Тип операции(INSERT, DELETE, UPDATE)

**table\_name** – Имя таблицы, в которой была произведена одна из операций

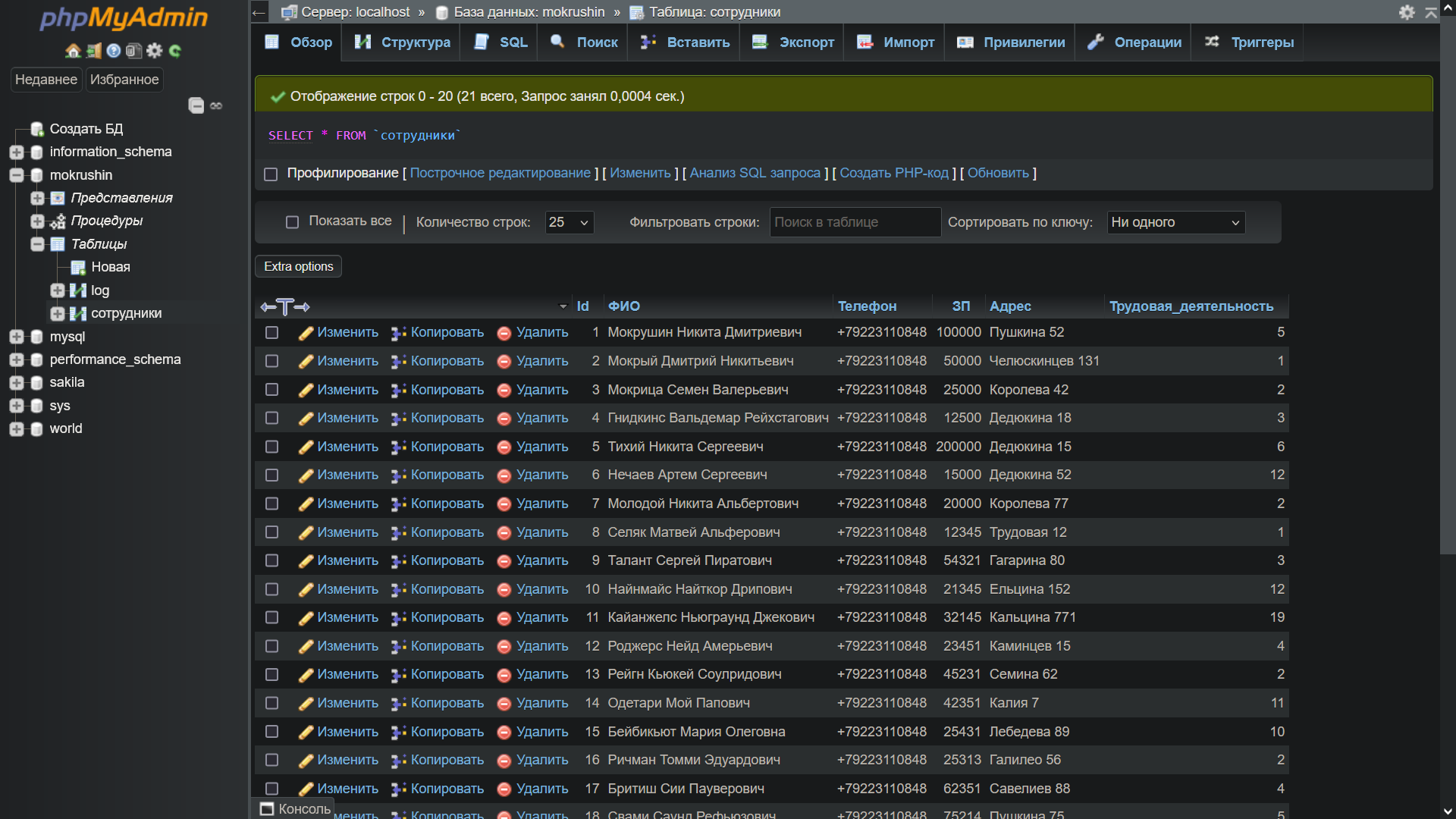
**old\_data –** Старые данные(если таковые имелись)

**new\_data –** Новые данные(если такие данные будут введены)

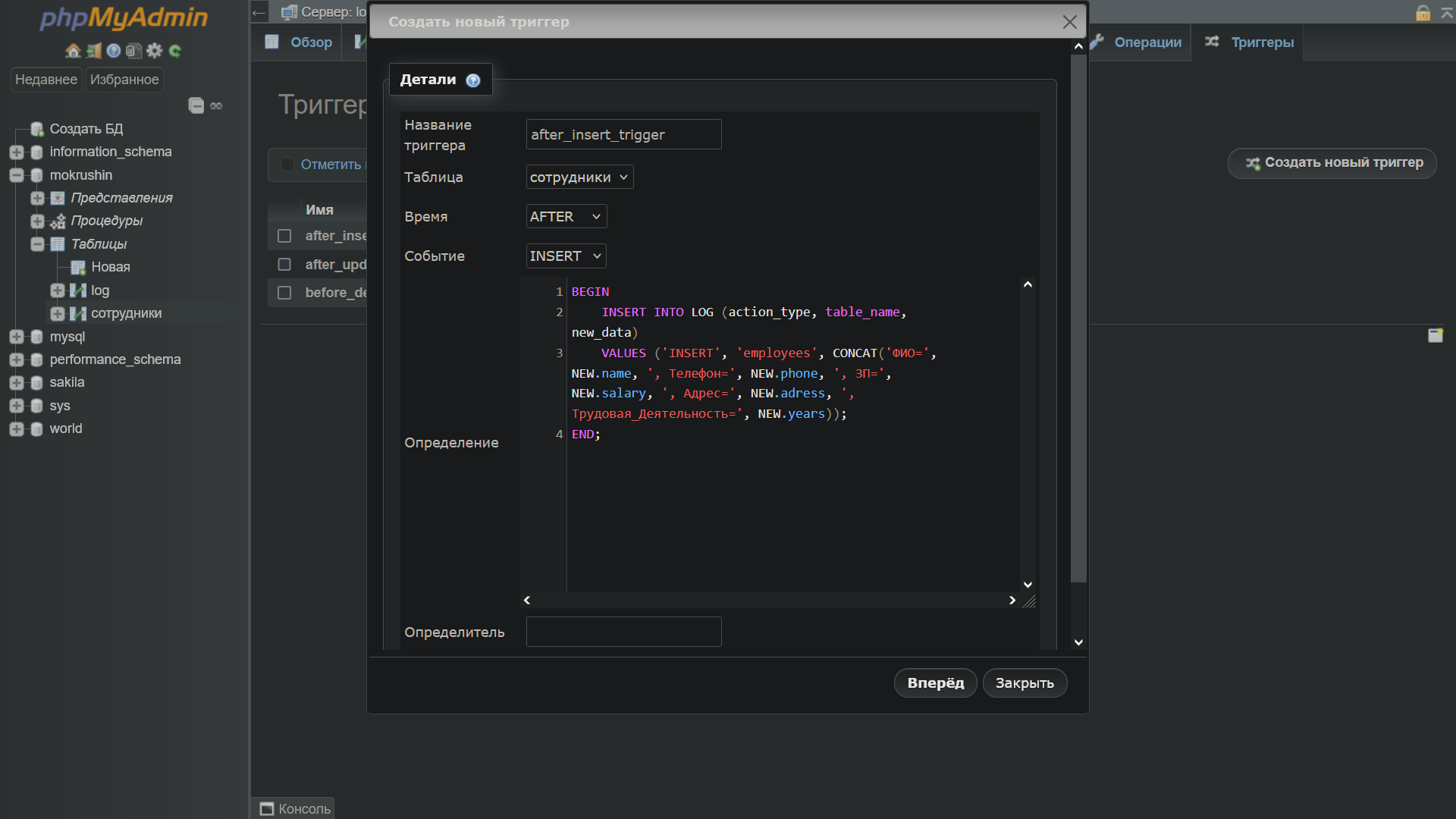


1. **Создание триггеров**

На панели созданной нами таблицы нужно найти кнопку «триггеры» и нажать на неё, вы увидите перед собой меню и необходимо следующим шагом нажать на «создать новый триггер»



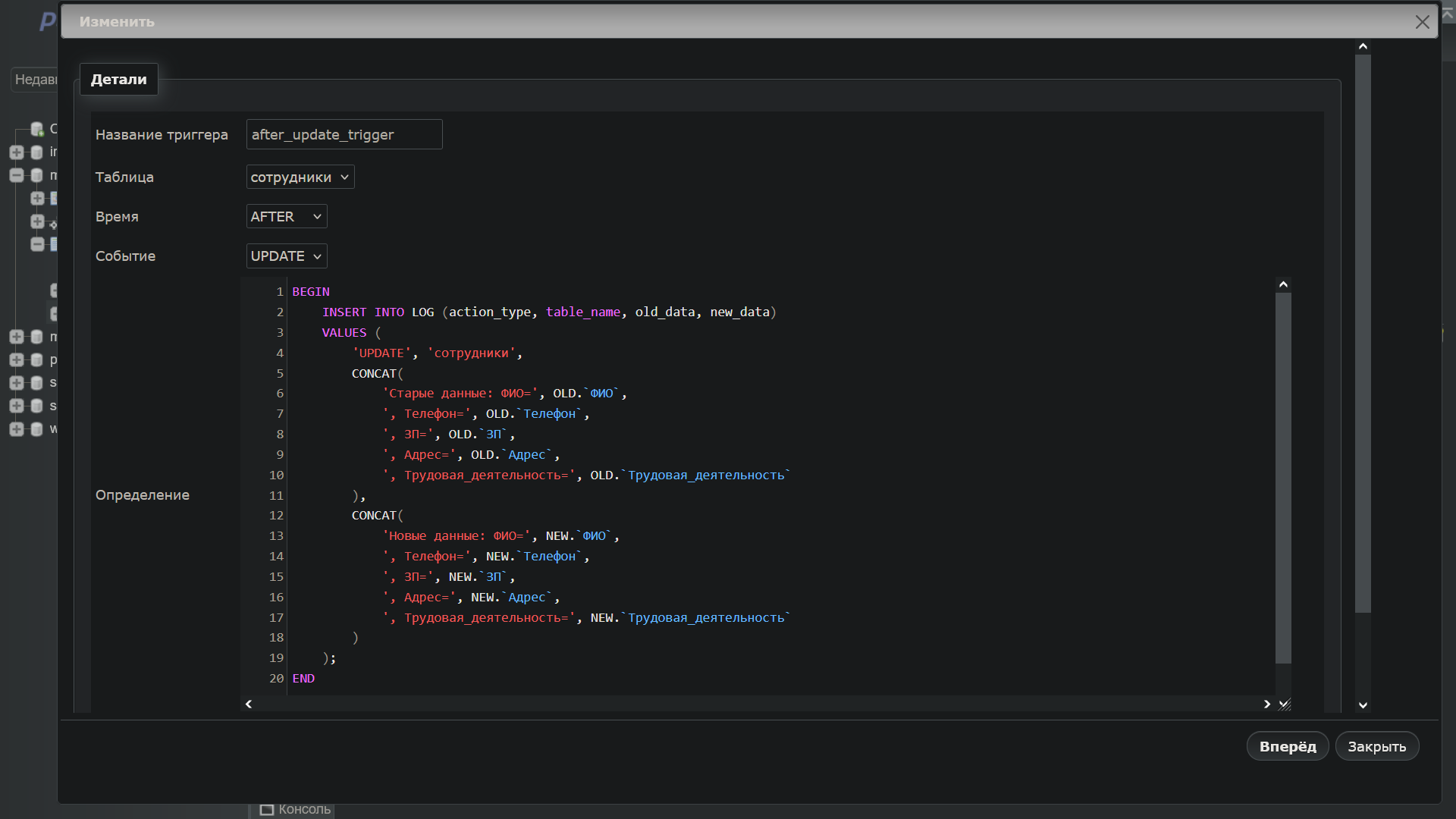
Создадим первый триггер, который будет срабатывать после команды INSERT:

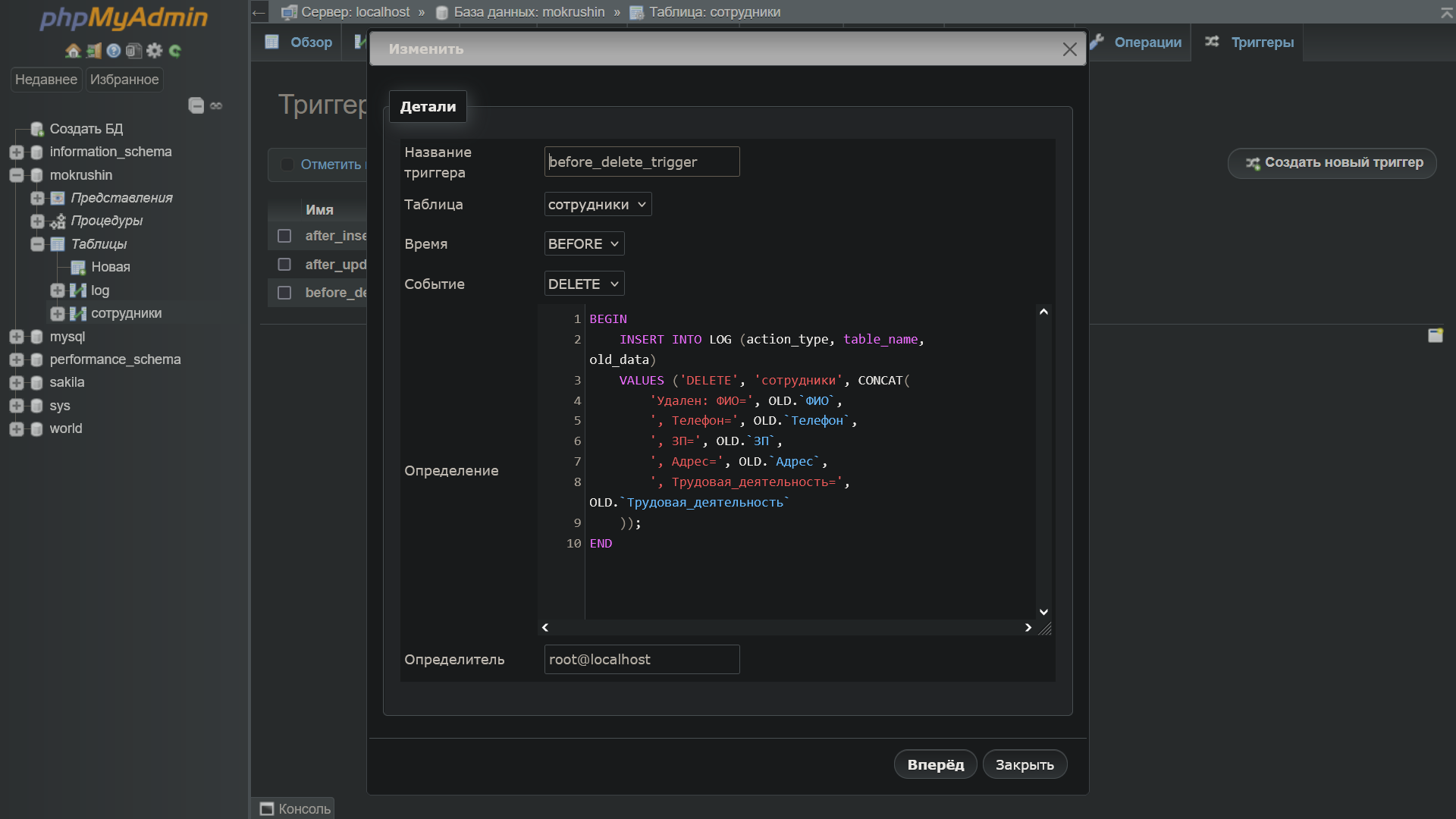


Данный код делает следующее:

Создает триггер с названием after\_insert\_trigger в таблицу «сотрудники», который срабатывает после команды INSERT, после срабатывания триггера, код будет делать INSERT в таблицу LOG с данными action\_type, table\_name и new\_data в которых будет «INSERT», название таблицы и строка с новыми данными.

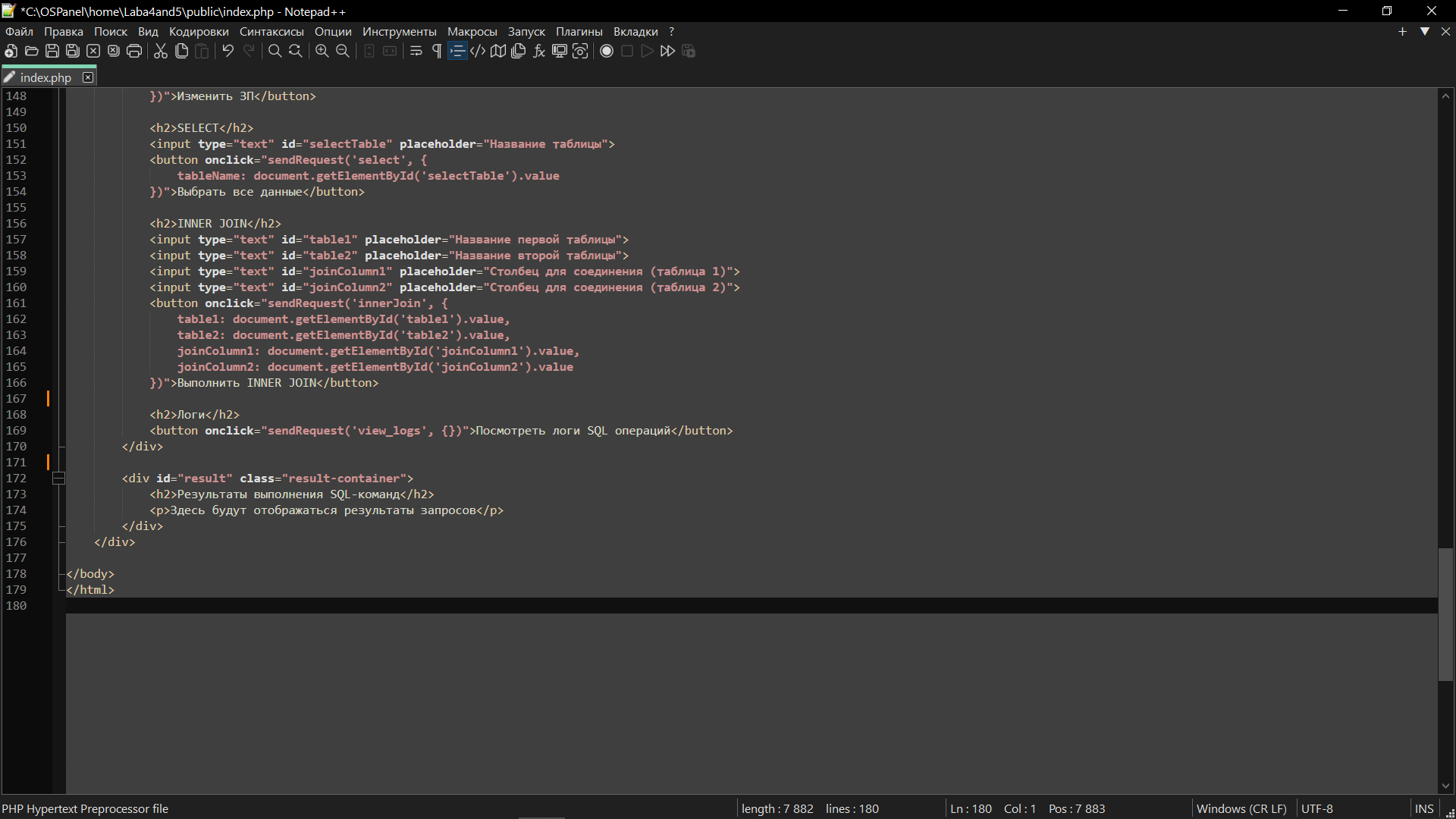
Так же поступим с последующими триггерами:



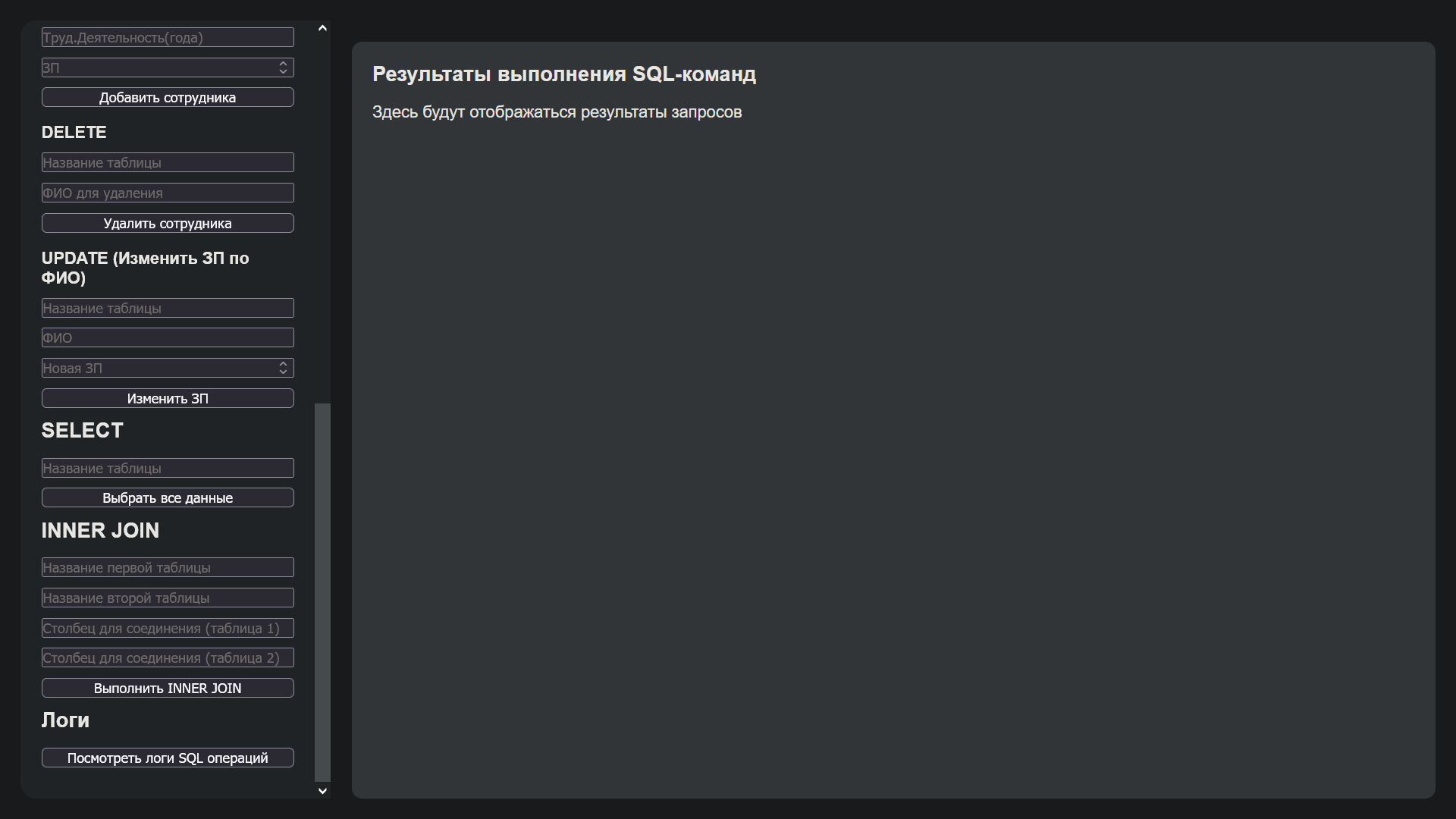


1. Демонстрация на сайте

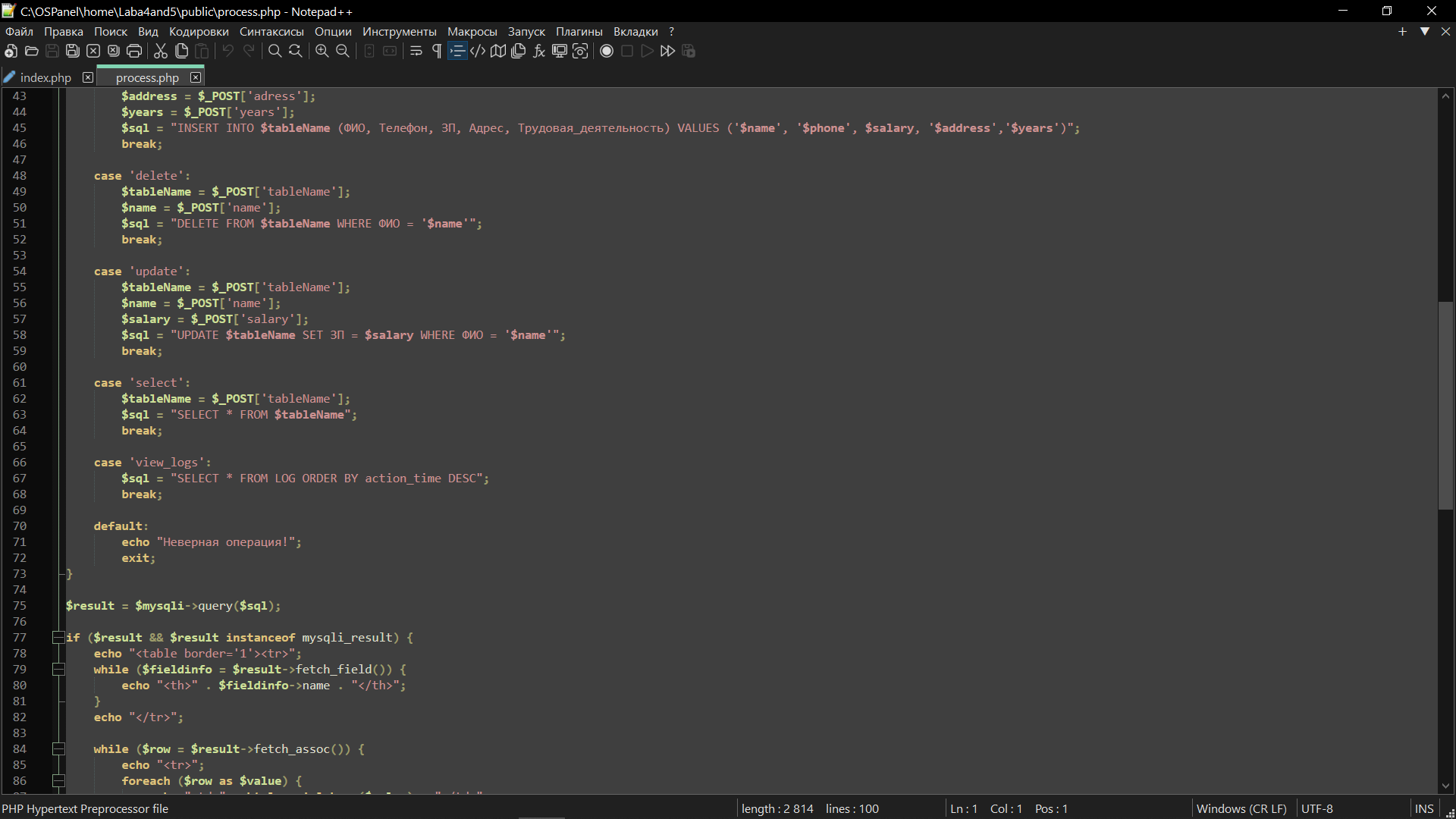
Используем сайт из лабораторной работы №4, лишь изменив некоторую часть кода, например:



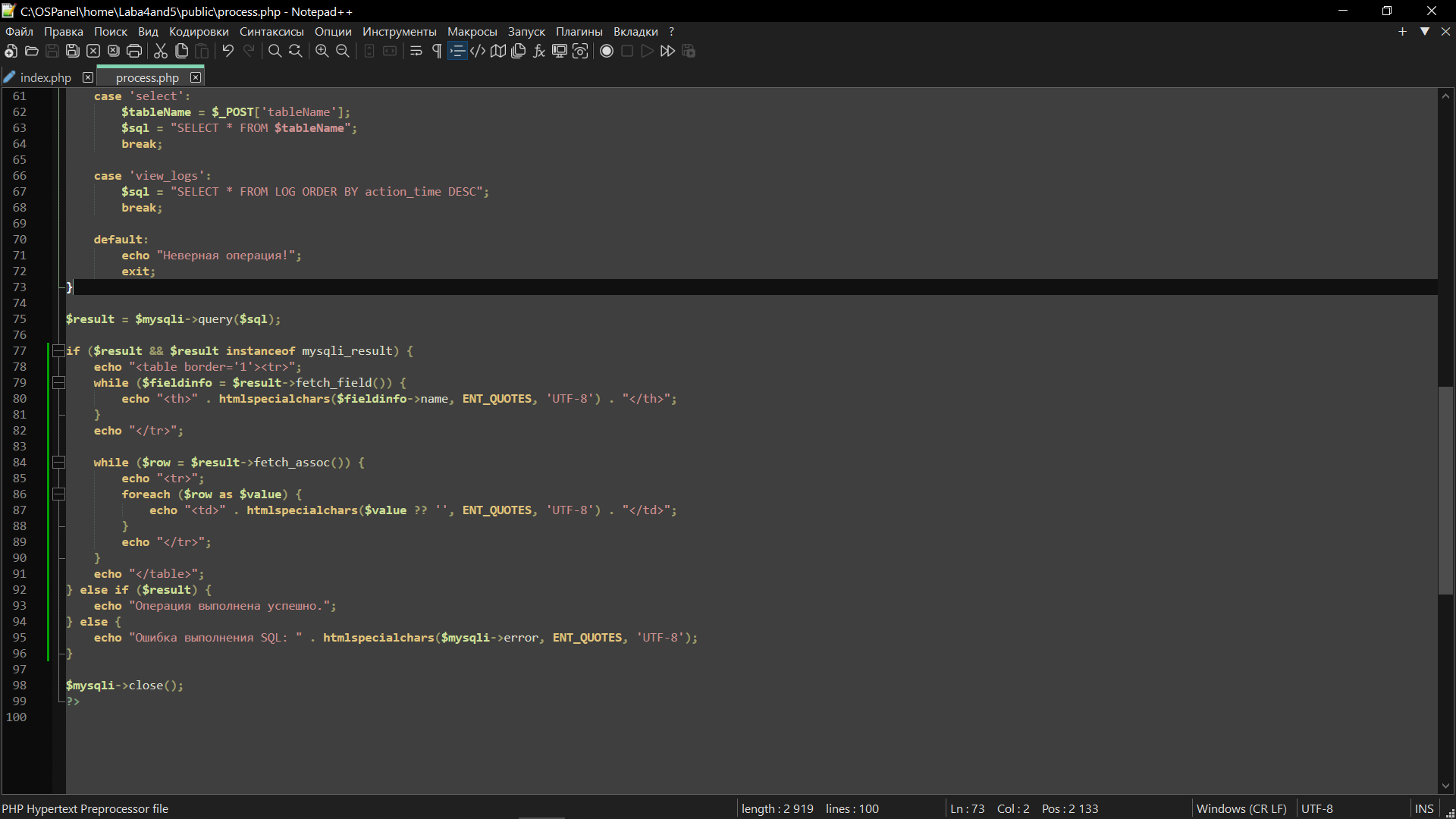
В конец контейнера с операциями создадим кнопку, которая при нажатии будет выводить таблицу логов с помощью SELECT на панель вывода результатов



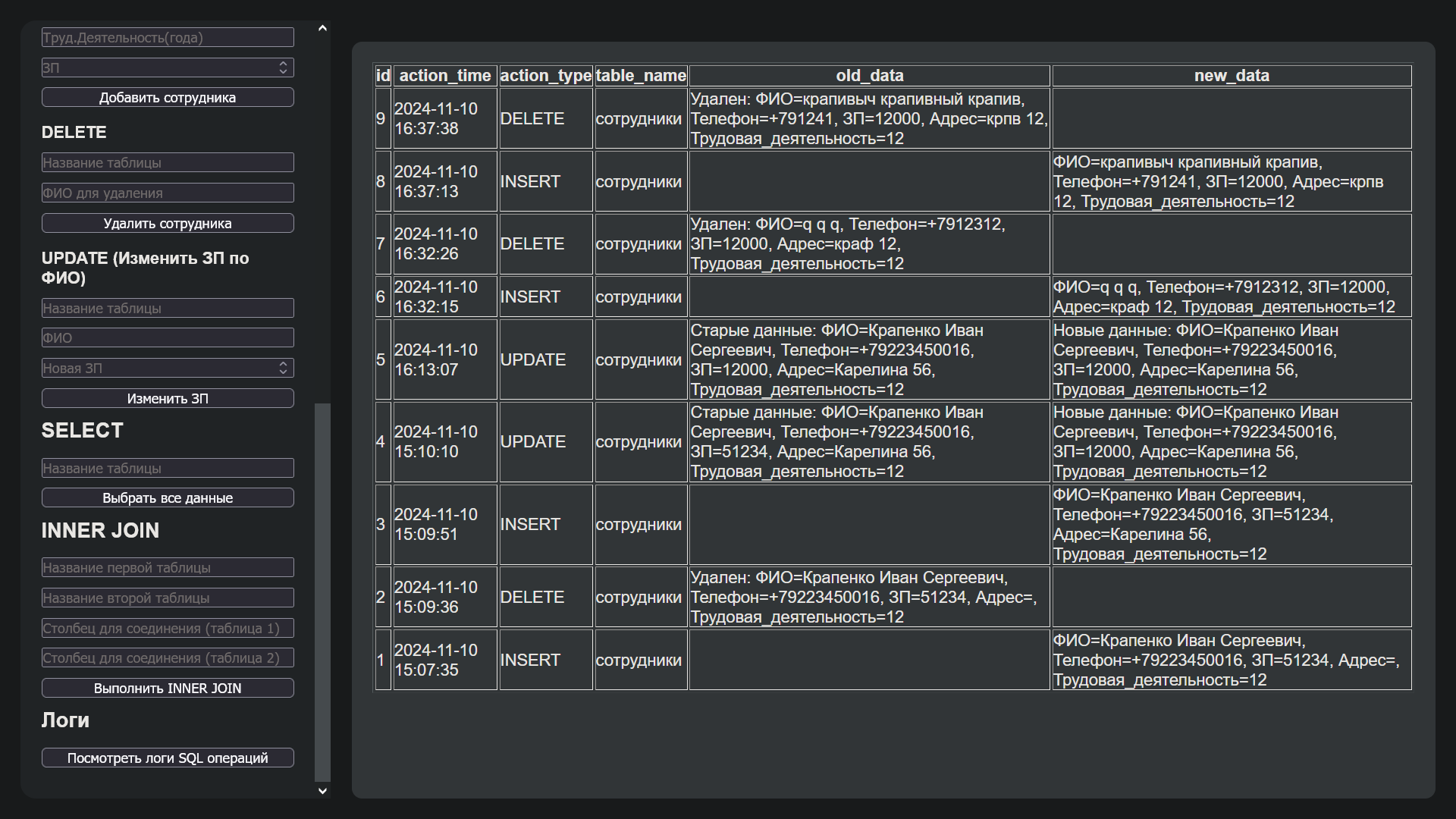
Так же, так как в кнопку мы добавили новую операцию в наш POST запрос, который ранее не был реализован, стоит изменить PHP-код:



В таблице могут появляться NULL строки, поэтому необходимо переделать вывод таблицы в php коде:



И вот что по итогу получилось:



1. **Получившийся код**

HTML-код:

<!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>

\*, \*::before, \*::after {

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>

<h2>Логи</h2>

<button onclick="sendRequest('view\_logs', {})">Посмотреть логи SQL операций</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>

PHP- код:

<?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'];

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

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

$sql = "INSERT INTO $tableName (ФИО, Телефон, ЗП, Адрес, Трудовая\_деятельность) VALUES ('$name', '$phone', $salary, '$address','$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 'view\_logs':

$sql = "SELECT \* FROM LOG ORDER BY action\_time DESC";

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>" . htmlspecialchars($fieldinfo->name, ENT\_QUOTES, 'UTF-8') . "</th>";

}

echo "</tr>";

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

echo "<tr>";

foreach ($row as $value) {

echo "<td>" . htmlspecialchars($value ?? '', ENT\_QUOTES, 'UTF-8') . "</td>";

}

echo "</tr>";

}

echo "</table>";

} else if ($result) {

echo "Операция выполнена успешно.";

} else {

echo "Ошибка выполнения SQL: " . htmlspecialchars($mysqli->error, ENT\_QUOTES, 'UTF-8');

}

$mysqli->close();

?>

# Заключение

В ходе выполнения лабораторной работы были успешно достигнуты все поставленные цели и задачи. Были изучены триггеры, так же были продемонстрирован вывод логов на страницу браузера.

# Список используемой литературы

* Триггеры MySQL - https://habr.com/ru/articles/37693/
* Горшенин, А. (2017). *Основы создания веб-приложений с использованием PHP и MySQL*. Питер.