

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ИНСТИТУТ НЕПРЕРЫВНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

ОЦЕНКА

ПРЕПОДАВАТЕЛЬ

старший преподаватель		Н.А. Соловьева
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Серверные сценарии. Работа с базой данных.

по дисциплине: Web-технологии

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №	Z1431		М.Д.Быстров
	номер группы	подпись, дата	инициалы, фамилия

Студенческий билет №	2021/3572
----------------------	-----------

Санкт-Петербург 2025

Оглавление

Оглавление	2
Цель работы	3
Задание	4
Структура таблиц БД	5
Доказательство нахождения таблиц в 3 НФ.....	6
Данные в таблицах	7
HTML код.....	8
Код серверных скриптов, используемых на сайте.....	17
Скриншоты.....	25
Вывод.....	26

Цель работы

Цель работы: изучение трехуровневой архитектуры веб-приложений (Клиент – Веб-сервер – Сервер БД) и языка написания серверного сценария php для построения динамического контента web-сайтов.

Задание

Таблица 1 Тема в соответствии с вариантом

№ варианта	Тема
1	Язык Котлин

Базовое задание

Установить компоненты 3-х уровневой архитектуры. В качестве звеньев 3-х уровневой архитектуры разрешается использовать Apache+PHP+MySQL (пакет программ WAMP – XAMPP, Денвер и т.п.). Язык серверных сценариев - PHP.

Подготовить базу данных, состоящую из 2 таблиц, связанные между собой отношением «один-ко-многим». Каждая таблица должна находиться в 3 нормальной форме и содержать не менее 5 полей. Содержимое полей таблицы должно соответствовать теме сайта. Все данные из таблиц должны отображаться на странице, включенной в разрабатываемый сайт. Для формирования веб-страницы использовать команду echo.

Расширенное задание

- 1.. не отображать служебные поля (первичные и вторичные ключи).
- 2.. все поля отобразить в одной таблице на веб-странице
- 3.. добавить возможность ввода новой строки в таблицу базы на веб-странице
- 4.. добавить возможность редактировать строку в таблицу базы на веб-странице
- 5.. добавить возможность удаления строки из таблицы базы на веб-странице
- 6.. написать серверный сценарий на php, выполняющий задание с матрицами из работы № 3.
- 7.. применить оформление через css
- 8.. несколько раз для формирования веб-страницы использовать функцию printf()

Структура таблиц БД

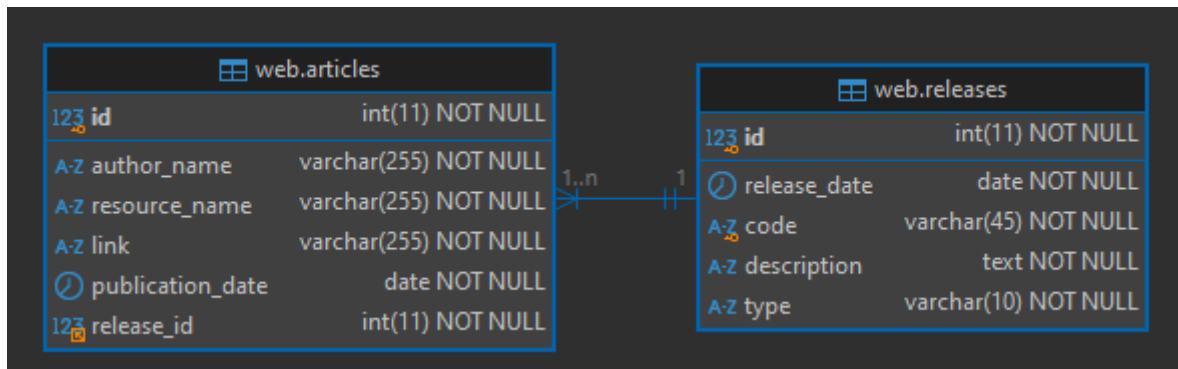


Рисунок 1 Схема БД

Доказательство нахождения таблиц в 3 НФ

Требование третьей нормальной формы (3NF) заключается в том, чтобы в таблицах отсутствовала транзитивная зависимость.

Транзитивная зависимость – это когда неключевые столбцы зависят от значений других неключевых столбцов.

Чтобы нормализовать базу данных до третьей нормальной формы, необходимо сделать так, чтобы в таблицах отсутствовали неключевые столбцы, которые зависят от других неключевых столбцов.

Иными словами, неключевые столбцы не должны пытаться играть роль ключа в таблице, т.е. они действительно должны быть неключевыми столбцами, такие столбцы не дают возможности получить данные из других столбцов, они дают возможность посмотреть на информацию, которая в них содержится, так как в этом их назначение.

Каждое из неключевых полей таблиц, приведенных на рис. 1, не зависит ни от какого другого неключевого поля; данные в этих полях исключительно описывают атрибуты сущности, однозначно идентифицируемой ключевыми полями.

Таким образом, база данных, схема которой приведена на рисунке 1, находится в третьей нормальной форме.

Данные в таблицах

releases Введите SQL выражение чтобы отфильтровать результаты						
Таблица	123 id	release_date	A-Z code	A-Z description	A-Z type	
1	1	2024-11-27	2.1.0	A language release introducing new language featuress.	language	
2	4	2024-05-21	2.0.0	A language release with the Stable Kotlin K2 compiler.	language	
3	6	2022-12-28	1.8.0	A feature release with improved kotlin-reflect performance,	feature	
4	7	2024-10-10	2.0.21	A bug fix release for Kotlin 2.0.20	bugfix	
5	8	2024-08-22	2.0.20	A tooling release for Kotlin 2.0.0 containing performance im	tooling	
Текст						

Рисунок 2 Данные в таблице releases

articles Введите SQL выражение чтобы отфильтровать результаты							
Таблица	123 id	A-Z author_name	A-Z resource_name	A-Z link	publication_date	123 release_id	
1	1	danil-pavlov	What's new in Kotlin 2.1.0	https://kotlinlang.org/docs/whatsnew21.html	2024-12-19	1	↗
2	3	Quattro8	What's new in Kotlin 1.8.0	https://kotlinlang.org/docs/whatsnew18.html	2024-09-25	6	↗
3	4	sarahhaggarty	What's new in Kotlin 2.0.20	https://kotlinlang.org/docs/whatsnew2020.html	2024-10-29	8	↗

Рисунок 3 Данные в таблице articles

HTML код

Здесь будет приведен код одной из страниц создаваемого сайта, в остальных страницах код остался без изменений с выполнения ЛР №3.

1. "4.php"

```
<?php
require "repo.php";
?>

<html>

<head>
    <title>Язык Kotlin</title>
    <link rel="shortcut icon" type="image/x-icon" href="img/kotlin_logo.png" />
    <link rel="stylesheet" type="text/css" href="4.css">

    <meta charset="UTF-8">
    <!-- meta - дополнительное задание -->
    <meta name="description" content="Kotlin programming language">
    <meta name="keywords" content="Kotlin, Java, JVM, JDK">
    <meta name="author" content="Maxim Bystrov">
</head>

<!-- дополнительное задание - использование события окна браузера -->

<body>
    <div id="container" class="container">

        <!-- шапка - обязательный элемент сайта -->
        <div id="header">
            <div id="logo"></div>

            <h2 id="sitename">Язык Kotlin</h2>

            <h3 id="pagename">Статьи</h3>

        </div>

        <!-- навигация между страницами - обязательный элемент сайта -->
        <div id="navigation" class="container">
            <h2><a href="1.html">История</a>
                <a href="2.html">Документация</a>
                <a href="3.html">Источники</a>
```



```

        <a href="4.php">Статьи</a>
    </h2>
</div>

</p>

<!-- боковая панель -->
<div id="sidebar">
    <h2>Новости</h2>
    <h3><a href="https://kotlinlang.org/docs/releases.html">Release
Notes</a></h3>
    <h3><a href="https://blog.jetbrains.com/kotlin/2024/11/kotlin-
roundup-kodee-s-top-picks/">Kotlin Roundup</a>
    </h3>
    <h3><a href="https://blog.jetbrains.com/kotlin/2024/10/ktor-3-
0/">Вышел Ktor 3.0</a></h3>
</div>

<!-- ОСНОВНОЙ КОНТЕНТ -->
<div id="content">

    <h2>Выпуски</h2>

    <?php
    $rows = get_releases();

    for ($i = 0; $i <= sizeof($rows); $i++)
    {
        if ($i < sizeof($rows))
        {
            $row = $rows[$i];
            $id = $row["id"];
        }
        else
        {
            $id = 0;
        }

        // формы для обновления / удаления каждой строки
        echo "<form id=\"releaseDelete\" . $id . \"\" action=\"delete.php\"
method=\"post\"></form>";
        echo "<form id=\"releaseUpdate\" . $id . \"\" action=\"update.php\"
method=\"post\"></form>";

        // данные об идентификаторах
        echo "<input form=\"releaseDelete\" . $id . \"\" type=\"hidden\"
name=\"id\" value=\"\" . $id . \"\" />";
        echo "<input form=\"releaseUpdate\" . $id . \"\" type=\"hidden\"
name=\"id\" value=\"\" . $id . \"\" />";
    }
}

```

```

        // данные об объекте обновления
        echo "<input form=\"releaseDelete\" . $id . \"\" type=\"hidden\"
name=\"table\" value=\"releases\" />";
        echo "<input form=\"releaseUpdate\" . $id . \"\" type=\"hidden\"
name=\"table\" value=\"releases\" />";
    }
    ?>

<table>
    <td>Код</td>
    <td>Описание</td>
    <td>Дата</td>
    <td>Тип</td>

<?php

for ($i = 0; $i <= sizeof($rows); $i++)
{
    if ($i < sizeof($rows))
    {
        $row = $rows[$i];
        $id = $row["id"];
    } else {
        $row = array();
        $id = 0;
    }

    echo "<tr>";

        echo "<td><input name=\"code\" form=\"releaseUpdate\" . $id .
\"\" value=\"\" . (isset($row['code']) ? $row['code'] : \"\") . \"\"></td>";
        echo "<td><textarea name=\"description\"
form=\"releaseUpdate\" . $id . \"\" value=\"\">\" . (isset($row['description']) ?
$row['description'] : \"\") . \"</textarea></td>";
        echo "<td><input name=\"release_date\" form=\"releaseUpdate\"
. $id . \"\" value=\"\" . (isset($row['release_date']) ? $row['release_date'] : \"\")
. \"\"></td>";

        echo "<td><input name=\"type\" form=\"releaseUpdate\" . $id .
\"\" value=\"\" . (isset($row['type']) ? $row['type'] : \"\") . \"\"></td>";
        echo "<td><input type=\"submit\" form=\"releaseUpdate\" . $id
. \"\" value=\"\" . ($id != 0 ? \"Обновить\" : \"Добавить\") . \"\"></td>";

        if ($i < sizeof($rows))
        {
            echo "<td><input type=\"submit\" form=\"releaseDelete\" .
$id . \"\" value=\"Удалить\"></td>";
        }

        echo "</form>";
        echo "</tr>";

```

```

    }
    ?>

</table>

<h2>Статьи</h2>

<?php
$rows = get_articles();

for ($i = 0; $i <= sizeof($rows); $i++)
{
    if ($i < sizeof($rows))
    {
        $row = $rows[$i];
        $id = $row["id"];
    }
    else
    {
        $id = 0;
    }

    // формы для обновления / удаления каждой строки
    echo "<form id=\"articleDelete\" . $id . \"\" action=\"delete.php\"
method=\"post\"></form>";
    echo "<form id=\"articleUpdate\" . $id . \"\" action=\"update.php\"
method=\"post\"></form>";

    // данные об идентификаторах
    echo "<input form=\"articleDelete\" . $id . \"\" type=\"hidden\"
name=\"id\" value=\"\" . $id . \"\" />";
    echo "<input form=\"articleUpdate\" . $id . \"\" type=\"hidden\"
name=\"id\" value=\"\" . $id . \"\" />";

    // данные об объекте обновления
    echo "<input form=\"articleDelete\" . $id . \"\" type=\"hidden\"
name=\"table\" value=\"articles\" />";
    echo "<input form=\"articleUpdate\" . $id . \"\" type=\"hidden\"
name=\"table\" value=\"articles\" />";
}
?>

<table>
    <td>Выпуск</td>
    <td>Автор</td>
    <td>Описание</td>
    <td>Ссылка</td>
    <td>Дата</td>

<?php

```

```

for ($i = 0; $i <= sizeof($rows); $i++)
{
    if ($i < sizeof($rows))
    {
        $row = $rows[$i];
        $id = $row["id"];
    }
    else
    {
        $row = array();
        $id = 0;
    }

    echo "<tr>";

    echo "<td><input name=\"release_code\" form=\"articleUpdate\"
. $id . \"\" value=\"\" . (isset($row['release_code']) ? $row['release_code'] : \"\")
. \"\"></td>";

    echo "<td><input name=\"author_name\" form=\"articleUpdate\" .
$id . \"\" value=\"\" . (isset($row['author_name']) ? $row['author_name'] : \"\") .
\"\"></td>";

    echo "<td><input name=\"resource_name\" form=\"articleUpdate\"
. $id . \"\" value=\"\" . (isset($row['resource_name']) ? $row['resource_name'] :
\"\") . \"\"></td>";

    echo "<td><input name=\"link\" form=\"articleUpdate\" . $id .
\"\" value=\"\" . (isset($row['link']) ? $row['link'] : \"\") . \"\"></td>";
    echo "<td><input name=\"publication_date\"
form=\"articleUpdate\" . $id . \"\" value=\"\" . (isset($row['publication_date']) ?
$row['publication_date'] : \"\") . \"\"></td>";
    echo "<td><input type=\"submit\" form=\"articleUpdate\" . $id
. \"\" value=\"\" . ($id != 0 ? \"Обновить\" : \"Добавить\") . \"\"></td>";

    if ($i < sizeof($rows))
    {
        echo "<td><input type=\"submit\" form=\"articleDelete\" .
$id . \"\" value=\"Удалить\"></td>";
    }

    echo "</form>";
    echo "</tr>";
}
?>

</table>

</div>

<div id="clear"> </div>

```

```

        <!-- подвал - обязательный элемент сайта -->
        <div id="footer">
            <h2>Контакты</h2>
            <p>
                <a href="https://github.com/maxi7665">GitHub</a>
            </p>
            <a href="https://github.com/maxi7665/web-technologies">Исходный код
сайта</a>
        </p>
    </div>
</div>
</body>

</html>

```

2. “4.css”

```

body {
    background: #f1f1f1;
    color: #000;
    font-family: Arial, sans-serif;
    font-size: 14px;
    padding: 10px;
}

/* body {
    font-family: Arial;
    padding: 10px;
    background: #f1f1f1;
} */

#header {
    background: #DCDCDC;
    width: 100%;
    height: 100px;
    display: flex;
    align-items: center;
}

#header h2 {
    text-align: left;
    /* padding: 10px; */
}

#sitename {
    margin-left: auto;

```

```

}

#pagename{
    margin-left: auto;
    margin-right: auto;
    font-size: 30;
}

#logo {
    height: 100%;
    width: 16%;
    text-align: left;
}

#logo img{
    height: 100%;
    width: auto;
}

#navigation {
    background: #333;
    width: 100%;
    height: 50px;
}

#sidebar {
    background: #DCDCDC;
    float: left;
    width: 20%;
    /* height: 100%; */
}

#content {
    background: #DCDCDC;
    float: right;
    width: 79%;
    /* height: 100%; */
}

#content p {
    text-align: left;
    padding-left: 10px;
    padding-right: 10px;
}

#clear {
    clear: both;
}

#footer {

```

```

        background: #333;
        width: 100%;
        /* height: 40px; */
    }

    #footer h2 {
        color: #f2f2f2;
    }

    #footer a {
        color: #f2f2f2;
        text-decoration: none;
    }

    #footer p {
        padding: 3px;
    }

    #container {
        background: #AAA;
        margin: auto auto;
        text-align: center;
        width: 80%;
        /* height: 800px; */
        padding: 10px;
    }

    #navigation a {
        float: left;
        display: block;
        color: #f2f2f2;
        text-align: center;
        padding: 14px 16px;
        text-decoration: none;
    }

    /* Change color on hover */
    #navigation a:hover {
        background-color: #ddd;
        color: black;
    }

    #list{
        text-align: left;
    }

    table {
        margin: 10px;
        border: 2px solid black;
        width: 97%;
    }

```

```

        overflow-x: scroll;
    }

    input {
        width: 100%;
        height: 100%;
    }

    textarea {
        width: 100%;
        height: 100%;
    }

    tr, td {
        border: 1px solid black;
        width: auto;
    }

    #content picture img {
        width: 80%;
    }

    #content pre{
        text-align: left;
    }

    abbr
    {
        font-style: italic;
        color: chocolate;
    }

    #pagename::first-letter
    {
        font-size: 35px;
    }

    #clockdiv {
        text-align: left;
        padding: 10px;
    }

```


Код серверных скриптов, используемых на сайте

1. “update.php”

```
<?php

require 'repo.php';

//var_dump($_POST);
//die();

function prepare_release_row()
{
    $row = array();

    $row['id'] = $_POST['id'];
    $row['code'] = $_POST['code'];
    $row['description'] = $_POST['description'];
    $row['release_date'] = $_POST['release_date'];
    $row['type'] = $_POST['type'];

    return $row;
}

function prepare_article_row()
{
    $row = array();

    $row['id'] = $_POST['id'];
    $row['author_name'] = $_POST['author_name'];
    $row['resource_name'] = $_POST['resource_name'];
    $row['link'] = $_POST['link'];
    $row['publication_date'] = $_POST['publication_date'];
    $row['release_code'] = $_POST['release_code'];

    return $row;
}

if (!isset($_POST["id"]))
{
    die("id field is not present in query");
}

if (!isset($_POST["table"]))
{
    die("table field is not present in query");
}

$id = $_POST["id"];
```

```

$table = $_POST["table"];

if ($id == 0)
{
    if ($table == 'releases')
    {
        $row = prepare_release_row();

        add_release($row);
    }
    else if ($table == 'articles')
    {
        $row = prepare_article_row();

        add_article($row);
    }
    else
    {
        die('table '.$table.' is not found');
    }
}
else
{
    if ($table == 'releases')
    {
        $row = prepare_release_row();

        update_release($row);
    }
    else if ($table == 'articles')
    {
        $row = prepare_article_row();

        update_article($row);
    }
    else
    {
        die('table '.$table.' is not found');
    }
}

header("Location: 4.php");
die();

?>

```

2. “delete.php”

```

<?php

require "repo.php";

//var_dump($_POST);
//die();

if (!isset($_POST["id"]))
{
    die("id field is not present in query");
}

if (!isset($_POST["table"]))
{
    die("table field is not present in query");
}

$id = $_POST["id"];
$table = $_POST["table"];

if ($id == 0)
{
    die("id is equal 0");
}
else
{
    if ($table == 'releases')
    {
        remove_release($id);
    }
    else if ($table == 'articles')
    {
        remove_article($id);
    }
    else
    {
        die('table '.$table.' is not found');
    }
}

header("Location: 4.php");
die();

?>

```

3. “repo.php”

```

<?php

```

```

function init_conn()
{
    $servername = "localhost";
    $username = "root";
    $password = "qwsxza";
    $dbname = "web";

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

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

    return $conn;
}

function get_releases()
{
    $conn = init_conn();

    $sql = "SELECT * FROM releases order by code desc";
    $result = $conn->query($sql);

    $rows = array();

    $row_num = 0;

    if ($result->num_rows > 0)
    {
        while ($row = $result->fetch_assoc())
        {
            $rows[$row_num] = $row;

            $row_num ++;
        }
    }

    $conn->close();

    return $rows;
}

function get_articles()
{
    $conn = init_conn();

    $sql = "SELECT r.code as release_code, a.* FROM releases as r join articles
as a on a.release_id = r.id order by publication_date desc";

```

```

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

$rows = array();

$row_num = 0;

if ($result->num_rows > 0)
{
    while ($row = $result->fetch_assoc())
    {
        $rows[$row_num] = $row;

        $row_num ++;
    }
}

$conn->close();

return $rows;
}

function get_release_by_code($code)
{
    $conn = init_conn();

    $sql = "SELECT * FROM releases where code='".$code."'";
    $result = $conn->query($sql);

    $release = array();

    $row_num = 0;

    if ($result->num_rows > 0)
    {
        // output data of each row
        while ($row = $result->fetch_assoc())
        {
            $release = $row;
        }
    }

    $conn->close();

    return $release;
}

function add_release($row)
{
    $conn = init_conn();

```

```

$sql = "insert into releases(code, description, release_date, type) values ("
      ."".mysqli_real_escape_string($conn, $row['code'])."', "
      ."".mysqli_real_escape_string($conn, $row['description'])."', "
      ."".mysqli_real_escape_string($conn, $row['release_date'])."', "
      ."".mysqli_real_escape_string($conn, $row['type']).''";

//var_dump($sql);

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

$conn->close();
}

function add_article($row)
{
    if (!isset($row['release_code']))
    {
        die('release code is not present in query');
    }

    $release = get_release_by_code($row['release_code']);

    if (!isset($release['id']))
    {
        die('release with code '.$row['release_code'].' is not found');
    }

    $conn = init_conn();

    $sql = "insert into articles(author_name, resource_name, link,
publication_date, release_id) values ("
      ."".mysqli_real_escape_string($conn, $row['author_name'])."', "
      ."".mysqli_real_escape_string($conn, $row['resource_name'])."', "
      ."".mysqli_real_escape_string($conn, $row['link'])."', "
      ."".mysqli_real_escape_string($conn, $row['publication_date'])."', "
      ."".mysqli_real_escape_string($conn, $release['id']).''";

    //var_dump($sql);

    $result = $conn->query($sql);
    $conn->close();
}

function update_release($release)
{
    update_table($release, 'releases');
}

```

```

function update_article($article)
{
    //var_dump($article);

    if (!isset($article['release_code']))
    {
        die('release code is not present in query');
    }

    $release = get_release_by_code($article['release_code']);

    if (!isset($release['id']))
    {
        die('release with code '.$article['release_code'].' is not found');
    }

    $article['release_id'] = $release['id'];
    unset($article['release_code']);

    update_table($article, 'articles');
}

function update_table($row, $table)
{
    $id = $row['id'];

    $sql = "update ".$table." set ";

    $is_first = 0;

    $conn = init_conn();

    foreach ($row as $key => $value)
    {
        if ($is_first != 0)
        {
            $sql = $sql.", ";
        }

        $sql = $sql." ".$key."='".mysqli_real_escape_string($conn,$value)."'";

        $is_first = 1;
    }

    $sql = $sql." where id=".$id.";";

    //var_dump($sql);

    $result = $conn->query($sql);
}

```

```

        $conn->close();
    }

    function remove_release($id)
    {
        remove_from($id, 'releases');
    }

    function remove_article($id)
    {
        remove_from($id, 'articles');
    }

    function remove_from($id, $table)
    {
        $conn = init_conn();

        $sql = "delete from ".$table." where id=".$id.";";

        var_dump($sql);


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

        $conn->close();
    }

    ?>

```


Скриншоты



Язык Kotlin

Статьи

История Документация Источники Статьи

Новости

[Release Notes](#)

[Kotlin Roundup](#)

[Вышел Ktor 3.0](#)

Выпуски

Код	Описание	Дата	Тип		
2.1.0	A language release introducing new	2024-11-27	language	Обновить	Удалить
2.0.21	A bug fix release for Kotlin 2.0.20	2024-10-10	bugfix	Обновить	Удалить
2.0.20	A tooling release for Kotlin 2.0.0	2024-08-22	tooling	Обновить	Удалить
2.0.0	A language release with the Stable	2024-05-21	language	Обновить	Удалить
1.8.0	A feature release with improved	2022-12-28	feature	Обновить	Удалить
				Добавить	

Статьи

Выпуск	Автор	Описание	Ссылка	Дата		
2.1.0	daniil-pavlov	What's new in Kotlin	https://kotlinlang.org	2024-12-19	Обновить	Удалить
2.0.20	sarahhaggarty	What's new in Kotlin	https://kotlinlang.org	2024-10-29	Обновить	Удалить
1.8.0	Quattro8	What's new in Kotlin	https://kotlinlang.org	2024-09-25	Обновить	Удалить
					Добавить	

Контакты

GitHub

Исходный код сайта

Рисунок 4 Таблицы с данными из БД

Вывод

В ходе выполнения четвертой лабораторной работы была создана база данных с таблицами, содержащими данные в 3НФ, разработаны серверные скрипты на языке PHP для изменения данных в БД, средствами HTML и PHP разработан клиентский вывод информации из БД на сайте, с помощью форм ввода HTML реализован способ обращения с из браузера к серверным скриптам изменения данных для модификации данных.

Выполнены пункты обязательного и дополнительного задания.

Приобретены навыки по программированию на языке PHP, работе с POST-запросами, динамической генерации HTML на стороне сервера, изучена трехуровневая архитектура веб-приложений (Клиент – Веб-сервер – Сервер БД).

Приведены скриншоты, демонстрирующие созданный пользовательский интерфейс, а также исходный код разработанных скриптов.