Министерство образования и науки Российской Федерации

«РЫБИНСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П.А. СОЛОВЬЕВА»

Факультет радиоэлектроники и информатики

Кафедра математического и программного обеспечения электронных вычислительных средств

**КУРСОВОЙ ПРОЕКТ**

по дисциплине

«Базы данных»

на тему:

«Баскетбольный чемпионат»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Студентов группы ИПБ-16 Рахимов К.С.

Руководитель к.ф.м.н, Шаров В.Г.

Рыбинск 2019

Оглавление

[1 Введение3](#_Toc10168447)

[2 Постановка задачи4](#_Toc10168448)

[3 Описание предметной области5](#_Toc10168449)

[4 Инфологическая модель7](#_Toc10168450)

[5 Даталогическая модель9](#_Toc10168451)

[6 Архитектура разрабатываемой системы11](#_Toc10168452)

[7 Функциональные возможности разделов14](#_Toc10168453)

[8 Диаграммы потоков данных15](#_Toc10168454)

[9 Реализация серверной части16](#_Toc10168455)

[10 Реализация клиентской части18](#_Toc10168456)

[11 Примеры запросов30](#_Toc10168470)

[12 Заключение31](#_Toc10168471)

[13 Список литературы32](#_Toc10168472)

1. Введение

**Актуальность разработки**

Актуальность разработки сайта «Баскетбольный чемпионат» заключается в необходимости мониторинга чемпионата по баскетболу, получении информации командах участников, судьях и данные об играх. Использование данной системы значительно облегчит процесс мониторинга чемпионата, как руководителям данной системы, так и зрителям.

**Цель работы и постановка задачи**

Разыгрывается чемпионат по баскетболу. Необходимо структурированно отобразить информацию о матчах, командах, судьях и городах чемпионата, а также иметь возможность изменять ее.

**Технические и программные средства реализации**

Сайт "Баскетбольный чемпионат" работает по клиент-серверной технологии.

Клиент-серверная система характеризуется наличием двух взаимодействующих самостоятельных процессов – клиента и сервера, которые, в общем случае, могут выполняться на разных устройствах, обмениваясь данными по сети.

В клиент-серверной системе функционируют (как минимум) два приложения – клиент и сервер, делящие между собой все функции. Взаимодействием между клиентом и базой данных занимается python фреймворк Flask. Хранением данных занимается база данных SQlite3 и непосредственным манипулированием данными занимается Peewee ORM (объектно-реляционное отображение, или преобразование).

Формированием пользовательского интерфейса занимается клиент, для построения которого можно использовать целый ряд специальных инструментов. Логика обработки данных может выполняться как на клиенте, так и на сервере. Клиент посылает на сервер запросы. Сервер обрабатывает эти запросы и передает клиенту результат.

1. Постановка задачи

**Баскетбольный чемпионат**

Разыгрывается чемпионат по баскетболу. В каждом матче одна из команд является хозяином, а другая гостем.

Матч судит один судья, который не должен жить в городе, который представляет одна из команд.

Даты и время проведения игр определяются заранее составленным расписанием.

Система должна обеспечивать:

- включение в чемпионат команды;

- исключение команды;

- назначение даты игры и судьи;

- исключение игры;

- замена судьи в расписании;

- включение судьи в чемпионат;

- исключение судьи;

- занесение результата игры;

получение справок обо всех играх команды, всех играх в городе, все играх судьи, всех играх заданной даты, всех командах и судьях

1. Описание предметной области

**Общее описание предметной области**

Предметная область задачи – баскетбольный чемпионат. Клиенты просматривают таблицы, исходя из того, что есть базе данных чемпионата. Клиенты могут получить справки об играх команд, судей, играх в определенном городе, обо всех играх по заданной дате, а также просмотреть статистику команд и судей. Администратор имеет право изменять, исключать и добавлять информацию в таблицах.

Система обеспечивает:

* добавление матча;
* изменение матча;
* исключение команды и судьи в выбранном матче;
* добавление команды;
* изменение команды;
* добавление судьи;
* изменение судьи;
* добавление города;
* изменение города;

Система должна обеспечивать получение данных:

* о матчах;
* о командах;
* о судьях;
* о городах;
* **Описание входных документов**

Входные данные заполняются в специальные поля, расположенные в окне программы. При нажатии на кнопки происходит анализ введенных ранее данных и заполняются поля базы данных, либо выводится сообщение о некорректно созданном запросе.

**Описание выходных документов**

Выходные данные представляют собой готовую базу данных с заполненными полями.

**Список ограничений**

Требования к ограничениям:

1. Размер всех строковых полей должен быть менее 200 символов.
2. Должны отсутствовать незаполненные поля при добавлении или изменении записи в БД.
3. Все численные поля в БД имеют диапазон значений от 0 до 2147483647.
4. Имя судей, наименование команд и названия городов не должны повторяться.
5. Инфологическая модель

Сущности и их первичные ключи:

1. Команда (код команды)
2. Матч (код матча)
3. Судья (код судьи)
4. Город (код города)

Бизнес-правила:

1) Команда может являться гостем в разных матчах

2) Команда может не являться гостью ни в одном матче

3) В матче может являться гостем только одна команда

4) В матче обязана являться команда «гость»

5) Команда может являться хозяином в разных матчах

6) Команда может не являться хозяином ни в одном матче

7) В матче может является хозяином только одна команда

8) В матче должна быть команда «хозяин»

9) Судья может быть назначен в разных матчах

10) Судья обязан быть назначенным в матче

11) Матч обязан назначить судью

12) Матч может назначить только одного судью

13) Судья живет не более, чем в одном городе

14) Судья не должен жить в каком-либо городе

15) В городе могут жить несколько судей

16) Судья не обязательно должен жить в каком-либо городе

17) Команда располагается не более, чем в одном городе

18) Команда обязательно располагается в каком-либо городе

19) Город может иметь несколько команд

20) Город обязательно должен иметь команду

**ER-диаграмма**

ER-диаграмма приведена на рисунке 1.

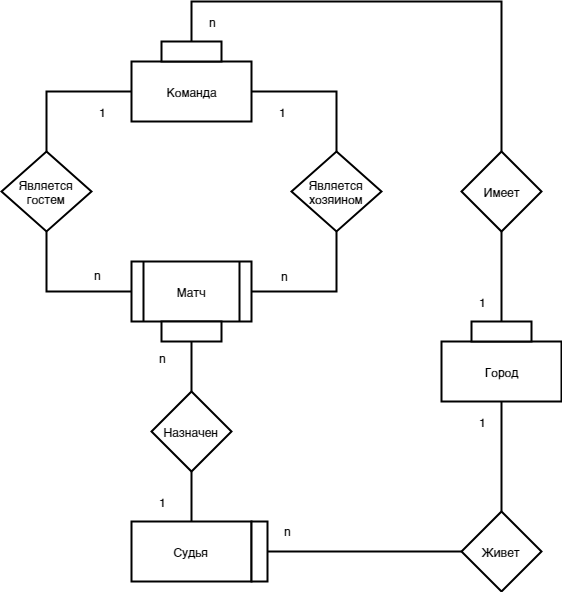


Рисунок 1 – ER-диаграмма

1. Даталогическая модель

Предварительные отношения:

1. Матч ( КодМатча, Хозяин, Гость.. )
2. Команда ( КодКоманды, Наименование.. )
3. Судья ( КодСудьи, Имя.. )
4. Город (КодГорода, Название ..)

Структура отношений базы данных приведена в таблице 1.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование информационного объекта | Наименование реквизита | Имя реквизита в ИЛМ | Тип | Длина | Точность | Признак ключа |
| Match | КодМатча | Kod\_Matcha | I | 20 | - | \* |
| Match | КодКоманды | Kod\_Komanda | I | 20 | - | \* |
| Match | КодКоманды | Kod\_Komanda | I | 20 | - | \* |
| Match | KoдСудьи | Kod\_Sudii | I | 20 | - | \* |
| Match | Дата | Date | date | - | - |  |
| Match | РезультатХозяина | Result\_Hozyain | I | 100 | - |  |
| Match | РезультатГостя | Result\_Gost | I | 100 | - |  |
| Match | Статус | Status | S | 40 | - |  |
| Komanda | КодКоманды | Kod\_Komandi | I | 20 | - | \* |
| Komanda | КодГорода | Kod\_Goroda | I | 20 | - | \* |
| Komanda | Наименование | Naimenovanie | S | 40 | - |  |
| Sudiya | КодСудьи | Kod\_Sudii | I | 20 | - | \* |
| Sudiya | КодГорода | Kod\_Goroda | I | 20 | - | \* |
| Sudiya | Имя | Name | S | 20 | - |  |
| Gorod | КодГорода | Kod\_Goroda | I | 20 | - | \* |
| Gorod | Название | Nazvanie | S | 20 | - |  |

Таблица 1 – структура отношений базы данных

Сводная таблица отношений приведена в таблице 2.

|  |  |
| --- | --- |
| Имя Таблицы | Описание |
| Match | Таблица матча |
| Komanda | Таблица команды |
| Sudiya | Таблица судьи |
| Gorod | Таблица города |

Таблица 2 – сводная таблица отношений

Структура базы данных изображена на рисунке 2.

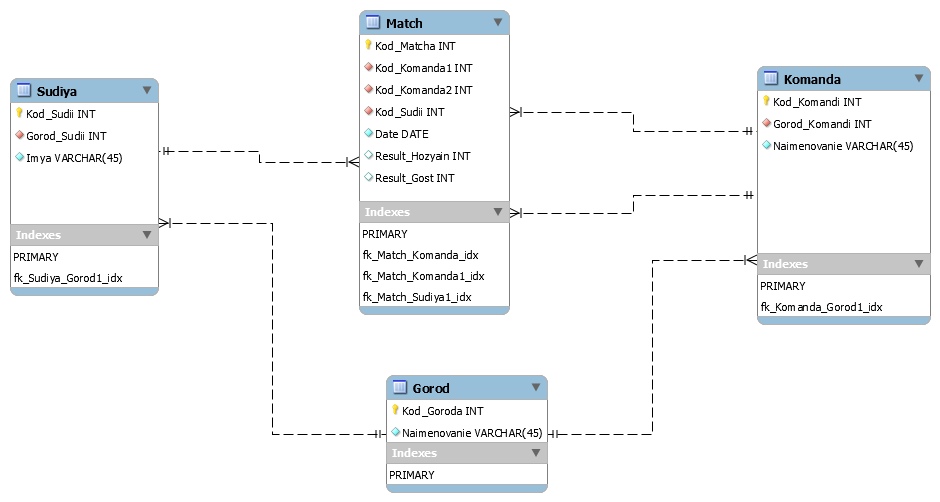


Рисунок 2 - Структура базы данных с использование WorkBench

1. Архитектура разрабатываемой системы

Схема взаимодействия технических средств представлена на рисунке 3.

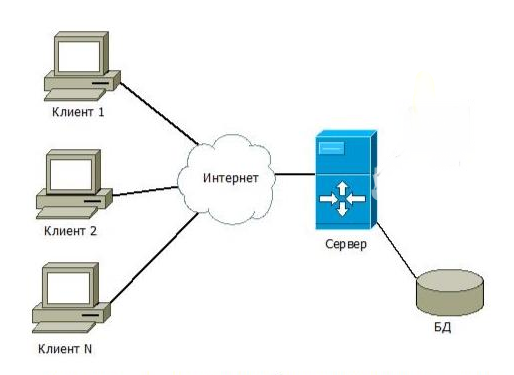


Рисунок 3 – Схема взаимодействия технических средств

Взаимосвязь между пользователем и системой представлена на рисунке 5.

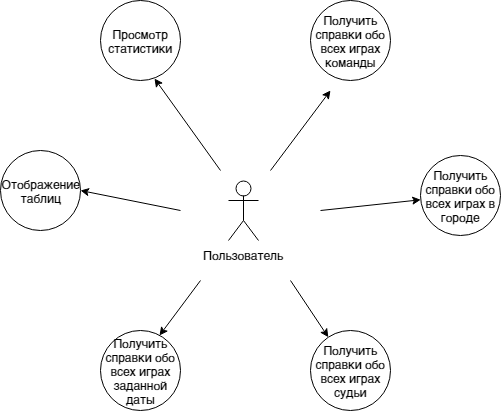


Рисунок 5 – Связь между пользователем и системой

Взаимосвязь между администратором и системой представлена на рисунке 6.

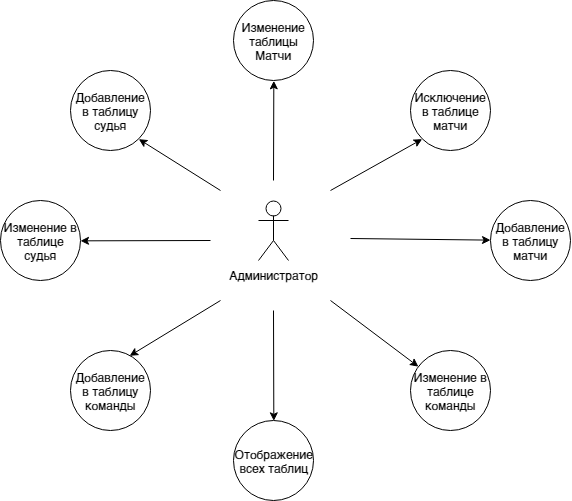


Рисунок 6 – Связь между администратором и системой

1. Функциональные возможности разделов

**Раздел «Матч»**

Данный раздел включает в себя следующие функции:

* получить данные о всех матчах команд;
* добавить матч;
* исключить судью или хозяина матча, или гостя матча;
* изменить судью и результат матча.

**Раздел «Команда»**

Данный раздел включает в себя следующие функции:

* получить данные обо всех командах;
* добавить команду;
* изменить команду;

**Раздел «Судья»**

В данном разделе реализованы функции:

* получить данные обо всех судьях;
* добавить судью;
* изменить судью;

**Раздел «Город»**

В данном разделе реализованы функции:

* получить данные обо всех городах;
* добавить город;
* изменить город;

1. Диаграммы потоков данных

Верхний уровень диаграммы (без декомпозиций) представлен на рисунке 7

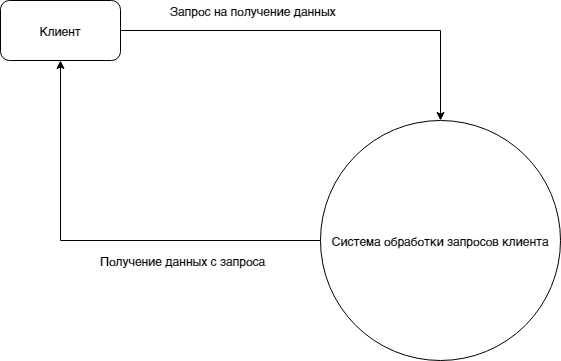


Рисунок 7 – Верхний уровень диаграммы (без декомпозиций)

1. Реализация серверной части

**Пример создания таблиц БД**

CREATE TABLE "match" (

"id" INTEGER NOT NULL PRIMARY KEY,

"NameGorodMatch\_id" VARCHAR(255) NOT NULL,

"NameKomandaHozMatch\_id" VARCHAR(255) NOT NULL,

"NameKomandaGosMatch\_id" VARCHAR(255) NOT NULL,

"NameSudiaMatch\_id" VARCHAR(255),

"DateMatch" DATE NOT NULL,

"ResultHozMatch" INTEGER NOT NULL,

"ResultGosMatch" INTEGER NOT NULL,

"StatusMatch" VARCHAR(255) NOT NULL,

FOREIGN KEY ("NameGorodMatch\_id") REFERENCES "gorod" ("Nazvanie"),

FOREIGN KEY ("NameKomandaHozMatch\_id") REFERENCES "komanda" ("Naimenovanie"),

FOREIGN KEY ("NameKomandaGosMatch\_id") REFERENCES "komanda" ("Naimenovanie"),

FOREIGN KEY ("NameSudiaMatch\_id") REFERENCES "sudiya" ("Name"))

CREATE TABLE "sudiya" (

"id" INTEGER NOT NULL PRIMARY KEY,

"Name" VARCHAR(255),

"NameGorodSud\_id" VARCHAR(255) NOT NULL,

FOREIGN KEY ("NameGorodSud\_id") REFERENCES "gorod" ("Nazvanie"))

CREATE TABLE "komanda" (

"id" INTEGER NOT NULL PRIMARY KEY,

"Naimenovanie" VARCHAR(255) NOT NULL,

"NameGorodKom\_id" VARCHAR(255) NOT NULL,

FOREIGN KEY ("NameGorodKom\_id") REFERENCES "gorod" ("Nazvanie"))

CREATE TABLE "gorod" (

"id" INTEGER NOT NULL PRIMARY KEY,

"Nazvanie" VARCHAR(255) NOT NULL)

1. Реализация клиентской части

**Общая схема работы программы**

При открытии страницы отображается первый блок, в котором отображается ознакомительный текст, а также кнопки для перехода к определённым разделам (Таблицы, Справки, Статистика и кнопка для перехода для авторизации как администратор), Рисунок 8.

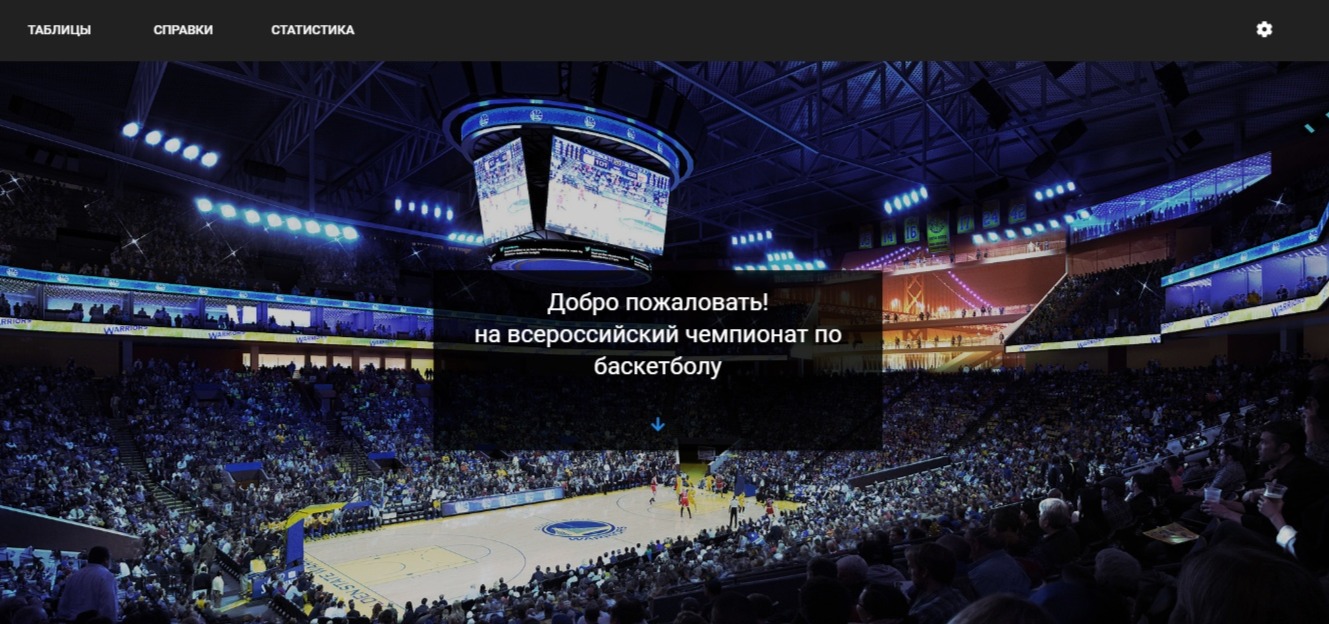


Рисунок 8 – первый блок сайта

В следующем блоке сайта находится раздел «Таблицы», при помощи кнопки Таблицы можно перейти в данный блок. Здесь отображаются все таблицы (Рисунок 9).

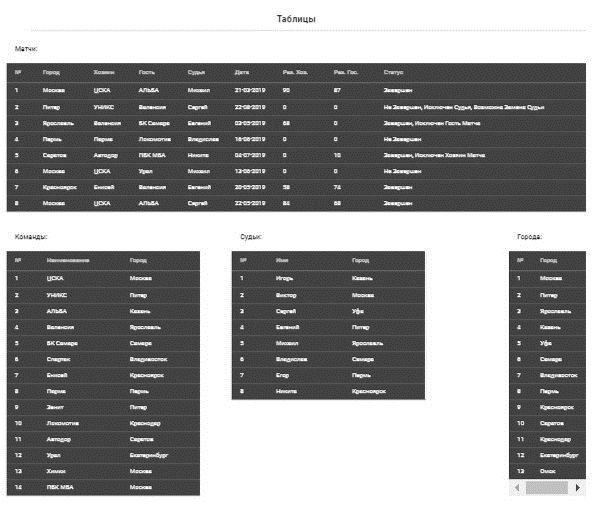
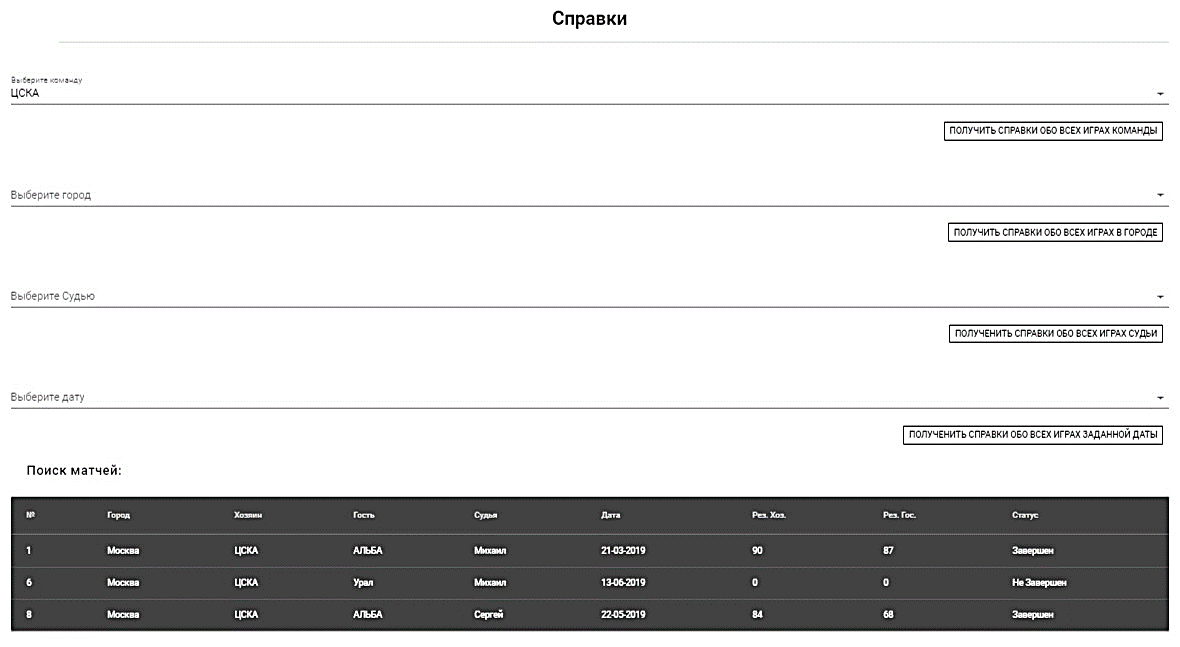


Рисунок 9 – Второй блок сайта (отображение таблиц)

Далее находится третий блок сайта с разделом «Справки», в котором расположены поля для ввода определённых данных, после чего отображение матча в виде таблицы (Рисунок 10).

Рисунок 10 – Получение справок



В последнем, четвертом блоке, находятся графики, а именно статистика команд. Также в самом низу сайта расположена информация и ссылка на Githab, где хранится весь проект по курсовой работе (Рисунок 11).

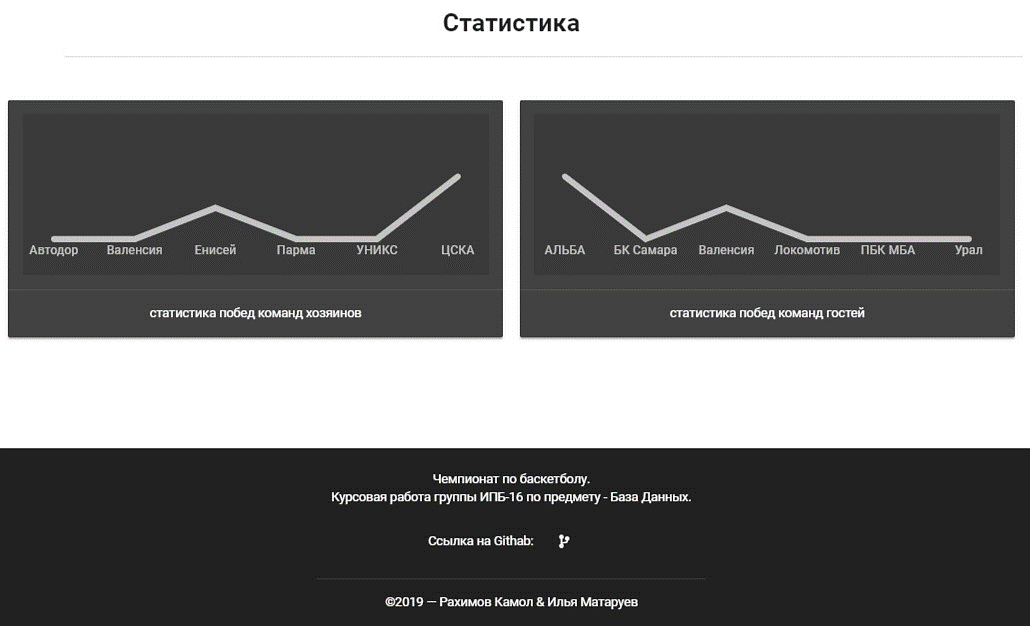


Рисунок 11 – Статистика команд (по победам) и блок с информацией.

Если нажать на шестеренку (кнопку) в первом блоке, то произойдет переход на страницу авторизации (для администраторов), Рисунок 12.

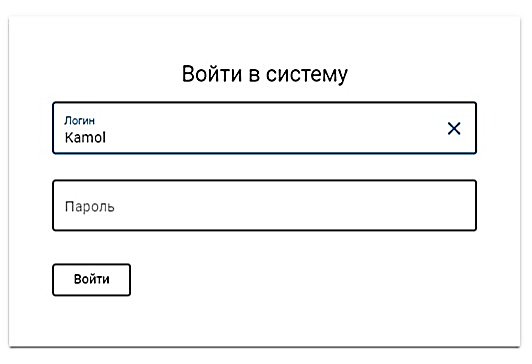


Рисунок 12 – страница авторизации, чтобы попасть в админ панель.

После того как была произведена авторизация, будет выполнен переход на страницу с админ панелью. Страница админ панели имеет следующие разделы:

# Таблица Матчей, возможны добавления и изменения матчей, а также исключить команду и судью (Рисунок 13)

## При нажатии на кнопку «Добавить Матч», появится диалоговое окно, в котором необходимо вести данные о новом матче (Рисунок 14).

## При нажатии на кнопку в виде ручки, появится диалоговое окно, в котором можно изменить только судью и результаты матча (Рисунок 15).

## При нажатии на кнопку в виде круга с восклицательным знаком, появится диалоговое окно, в котором можно исключить только судью и команду (Рисунок 16).

# Таблица Команд, возможны добавления и изменения команд (Рисунок 17)

## При нажатии на кнопку «Добавить Команду», появится диалоговое окно, в котором необходимо ввести данные о новой команде (Рисунок 18).

## При нажатии на кнопку в виде ручки, появится диалоговое окно, в котором можно изменить данные определенной команды (Рисунок 19).

# Таблица Судей, возможны добавления и изменения судей (Рисунок 20)

## При нажатии на кнопку «Добавить Судью», появится диалоговое окно, в котором необходимо ввести данные о новой судье (Рисунок 21).

## При нажатии на кнопку в виде ручки, появится диалоговое окно, в котором можно изменить данные определенного судьи (Рисунок 22).

# Таблица Городов, возможны добавления и изменения городов (Рисунок 23)

## При нажатии на кнопку «Добавить Город», появится диалоговое окно, в котором необходимо ввести данные о новом городе (Рисунок 24).

## При нажатии на кнопку в виде ручки, появится диалоговое окно, в котором можно изменить данные определенного судьи (Рисунок 25).

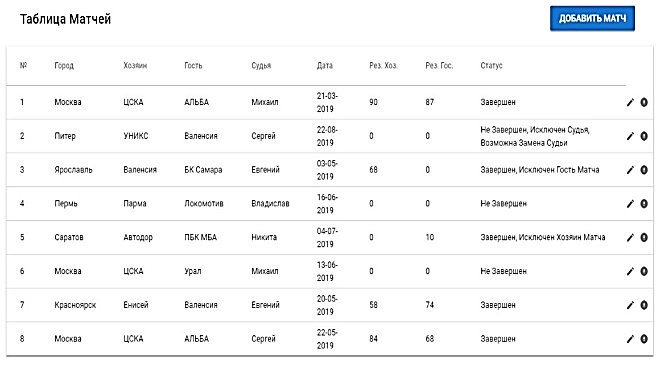


Рисунок 13 – Таблица Матчей



Рисунок 14 – Диалоговое окно «Добавить Матч»



Рисунок 15 – Диалоговое окно «Изменить матч»



Рисунок 16 – Диалоговое окно «Исключение в матче»



Рисунок 17 – Диалоговое окно Таблица Команд



Рисунок 18 – Диалоговое окно «Добавить команду»



Рисунок 20 – Диалоговое окно «Изменить команду»

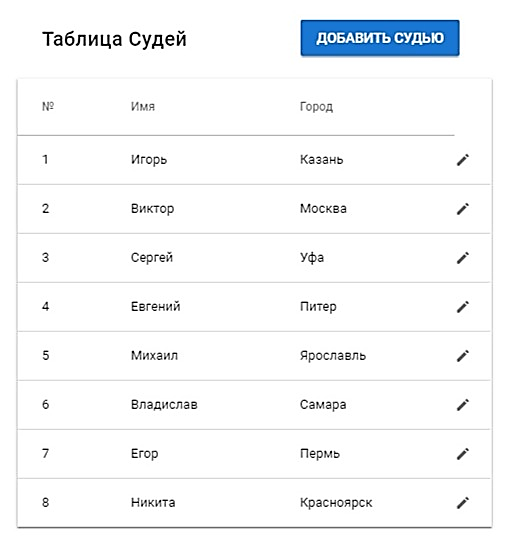


Рисунок 21 – Таблица Судей



Рисунок 22 – Диалоговое окно «Добавить судью»

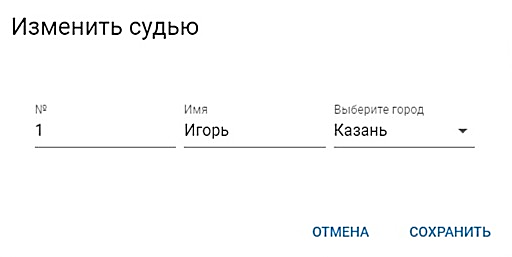


Рисунок 23 – Диалоговое окно «Изменить судью»

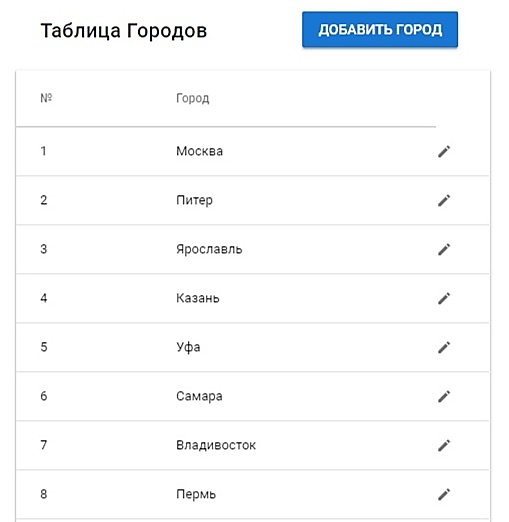


Рисунок 24 – Таблица городов



Рисунок 25 – Диалоговое окно «Добавить город»



Рисунок 26 – Диалоговое окно «Изменить город»

1. Примеры запросов

INSERT INTO Komanda (Naimenovanie, NameGorodKom) values(‘ЦСКА’, ‘Москва’) – добавление новой команды;

INSERT INTO Sudiya (Name, NameGorodSud) values(‘Михаил’, ‘Ярославль’) – добавление нового судьи;

SELECT

"t1"."id",

"t1"."NameGorodMatch\_id",

"t1"."NameKomandaHozMatch\_id",

"t1"."NameKomandaGosMatch\_id",

"t1"."NameSudiaMatch\_id",

"t1"."DateMatch",

"t1"."ResultHozMatch",

"t1"."ResultGosMatch",

"t1"."StatusMatch" FROM "match" AS "t1" GROUP BY

"t1"."id",

"t1"."NameGorodMatch\_id",

"t1"."NameKomandaHozMatch\_id",

"t1"."NameKomandaGosMatch\_id",

"t1"."NameSudiaMatch\_id",

"t1"."DateMatch",

"t1"."ResultHozMatch",

"t1"."ResultGosMatch",

"t1"."StatusMatch"

HAVING ("t1"."StatusMatch" = 'Завершен') – группировка всех записей команды в таблице матч, которые имеют статус «Завершен».

SELECT "t1"."id", "t1"."Nazvanie" FROM "gorod" AS "t1" WHERE ("t1"."Nazvanie" = 'Омск') – выборка записи, которая имеет значение «Омск»

1. Заключение

В результате проделанной работы были созданы база данных и клиентское приложение, реализующие в полной мере требуемый функционал. В процессе разработки мы получили опыт проектирования и создания баз данных, и навыки работы в разработке сайтов - инструментах для баз данных, а также практиковались с использованием языков SQL, Python и Javascript.

1. Список литературы

Марк Лутц Programmin python, 1996

Вильямс Секреты Python. 59 рекомендаций по написанию эффективного кода, 2017

Руководства:

к веб-фреймворку Flask <http://flask.pocoo.org/docs/1.0/>

к библиотеке Vue <https://vuejs.org/v2/guide/>