МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

РАЗРАБОТКА МНОГОПОЛЬЗОВАТЕЛЬСКОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ГОЛОСОВАНИЯ ЗА ИСХОД СПОРТИВНЫХ СОБЫТИЙ

Курсовое проектирование

по дисциплине «Базы и банки данных»

КР.1-53 01 02.10028578.12.81-01

Исполнитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Остапенко Александр

(подпись) Константинович АСОИ-181

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Мрочек Татьяна Владимировна

(подпись)

Дата допуска к защите \_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_

Могилев 2021

Содержание

[Введение 4](#_Toc71565530)

[1 Техническое задание 5](#_Toc71565531)

[1.1 Анализ предметной области (данные по оргштатной структуре предприятия, информация о принятых технологиях деятельности) 5](#_Toc71565532)

[1.2 Постановка задачи (стратегические направления и перспективы развития предприятия) 5](#_Toc71565533)

[2 Технический проект информационной системы 7](#_Toc71565534)

[2.1 Разработка структурной диаграммы программного модуля и её описание 7](#_Toc71565535)

[2.1.1 Контекстная диаграмма и диаграммы детализации процессов 7](#_Toc71565536)

[2.1.2 Диаграмма дерева узлов 11](#_Toc71565537)

[2.2 Информационная модель 12](#_Toc71565538)

[2.2.1 Идентификация сущностей и связей. ER-диаграмма логического уровня 12](#_Toc71565539)

[2.2.2 ER-диаграмма физического уровня. Ограничения ссылочной целостности. Переопределение триггеров. Индексирование отношений. 14](#_Toc71565540)

[2.2.3 Определение представлений, хранимых процедур серверной компоненты 15](#_Toc71565541)

[2.3 Верификация спроектированной логической модели 17](#_Toc71565542)

[3 Реализация системы 26](#_Toc71565543)

[3.1 SQL-определения регламентированных запросов 26](#_Toc71565544)

[3.2 SQL-определения триггеров 33](#_Toc71565545)

[3.3 SQL-определения хранимых процедур 36](#_Toc71565546)

[3.4 SQL-определения курсоров 42](#_Toc71565547)

[3.5 Распределение прав доступа пользователей 46](#_Toc71565548)

[4 Результаты тестирования информационной системы 49](#_Toc71565549)

[Заключение 56](#_Toc71565550)

[Список использованных источников 57](#_Toc71565551)

Введение

Целью данной работы является проектирование многопользовательской информационной системы. Систему предполагается использовать для организации онлайн-голосования за исходы спортивных событий с возможностью просмотра статистики голосования. Приложение имеет несколько преимуществ по сравнению со своими аналогами: простота в использовании, гибкость, масштабируемость. Данное приложение может быть использовано для создания и проведения голосований и выявления победителя в голосовании. Пояснительная записка к курсовому проектированию содержит 4 раздела:

* Техническое задание – раздел содержит всю необходимую информацию по предметной области, а также постановку задачи.
* Технический проект информационной системы – раздел содержит функциональную и информационную модель информационной системы.
* Реализация системы – содержит SQL-определения, применимые в рамках поставленной задачи.
* Результаты тестирования информационной системы – содержит описание различных тестов, которые проходила информационная система.
* Заключение – содержит обобщение выполненной работы.

1 Техническое задание

1.1 Анализ предметной области

В качестве объекта автоматизации выбрана система голосования.

Система создаётся на основе следующих документов:

* список спортивных команд;
* расписание спортивных событий;
* список прошедших спортивных событий с исходами;
* политика конфиденциальности;
* правила спортивных игр.

1.2 Постановка задачи (стратегические направления и перспективы развития предприятия)

Проектирование многопользовательской информационной системы. Систему предполагается использовать для организации онлайн-голосования за исходы спортивных событий с возможностью просмотра статистики голосования. Пользователи будут иметь возможность проголосовать за исходы в любом количестве матчей, а также просматривать статистику голосования других пользователей за матчи. Организаторы голосования смогут просматривать статистику голосования конкретного пользователя по различным спортивным событиям. Категории пользователей делятся на следующее: пользователь, организатор голосования, модератор, администратор.

Систему предполагается создать с целью упростить управление рабочим процессом для самых различных команд, для улучшения качества обслуживания заказчика.

Критерий оценки достижений целей системы — увеличение числа вовлечённых в голосование пользователей. Так как онлайн-платформа агрегирует в себе большее количество спортивных событий по различным видам спорта и позволяет увеличить охват пользователей.

Система должна удовлетворять следующим требованиям:

* Надежности;
* Безопасности;
* Требования к защите информации от несанкционированного доступа. Должна быть предусмотрена защита от несанкционированного доступа к данным, ввода данных, их удаления;
* Система должна быть доступна с любого компьютера в сети Интернет;
* Информация, хранящаяся в системе, должна быть защищена от аварийных ситуаций, влияния внешних воздействий (радиоэлектронная защита).

Функции, выполняемые подсистемами объектов автоматизации:

* + Выдача списка организованных и проведённых голосований. Без использования автоматизированной системы поиск необходимой информации о спортивных мероприятиях может занимать значительное время. Исключён долгий поиск мероприятий, мероприятия хранятся сколь угодно долго.
  + Добавление новых спортивных мероприятий, организация по ним голосования. При использовании системы добавление новых мероприятий будет происходить легко и быстро.
  + Просмотр статистики голосования. Это позволяет определять фаворитов. Также по итогам голосования легко определить пользователя, угадавшего наибольшее количество исходов и наградить его, что повышает популярность сайта.
  + История проведённых мероприятий и статистика голосования по ним позволяет отследить популярность спортивных команд и изменение популярности во времени, а также причины и момент изменения популярности.

Программное обеспечение системы не должно зависеть от аппаратных средств компьютера. Необходимое программное обеспечение:

* MS SQL Server 2019, .Net Core, Windows 7/10.

2 Технический проект информационной системы

2.1 Разработка структурной диаграммы программного модуля и её описание

2.1.1 Контекстная диаграмма и диаграммы детализации процессов

Контекстная диаграмма представлена на Рисунке 1.

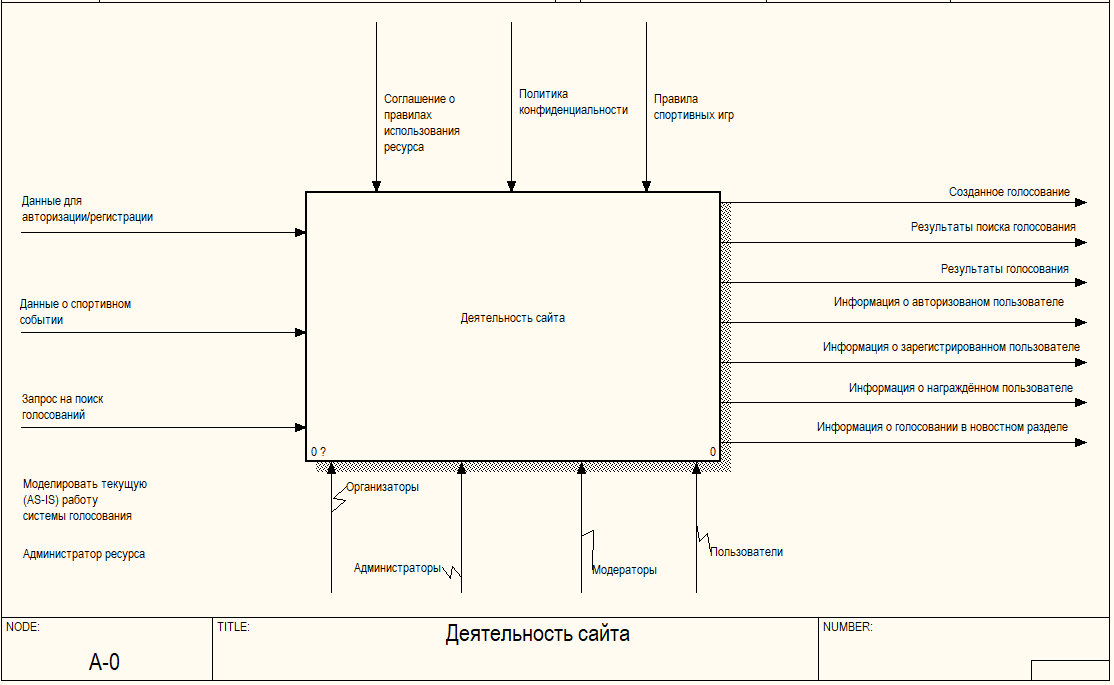


Рисунок 1 – Контекстная диаграмма.

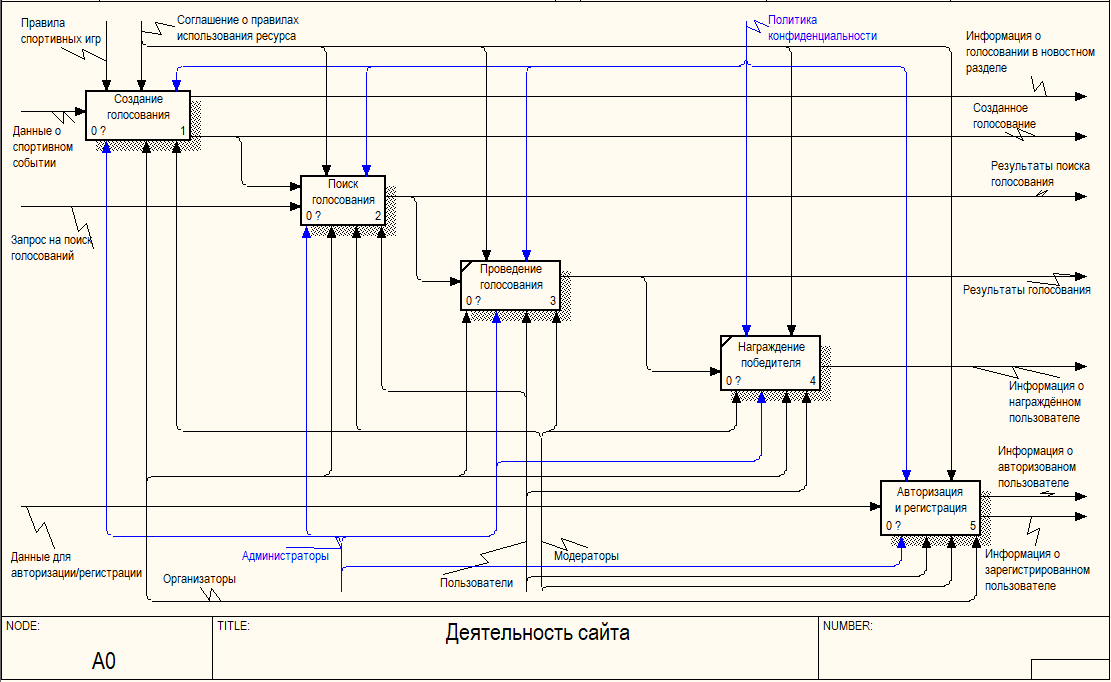


Рисунок 2 – Декомпозиция процесса «Деятельность сайта»

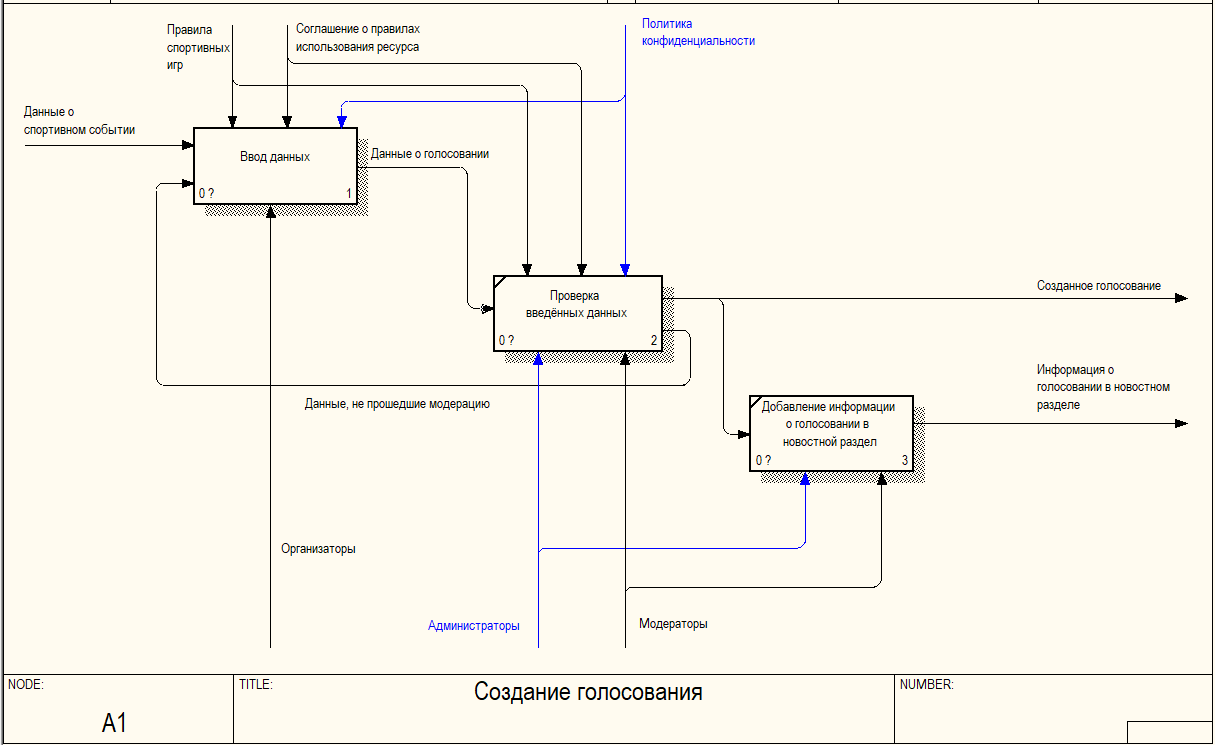


Рисунок 3 – Декомпозиция процесса «Создание голосования»

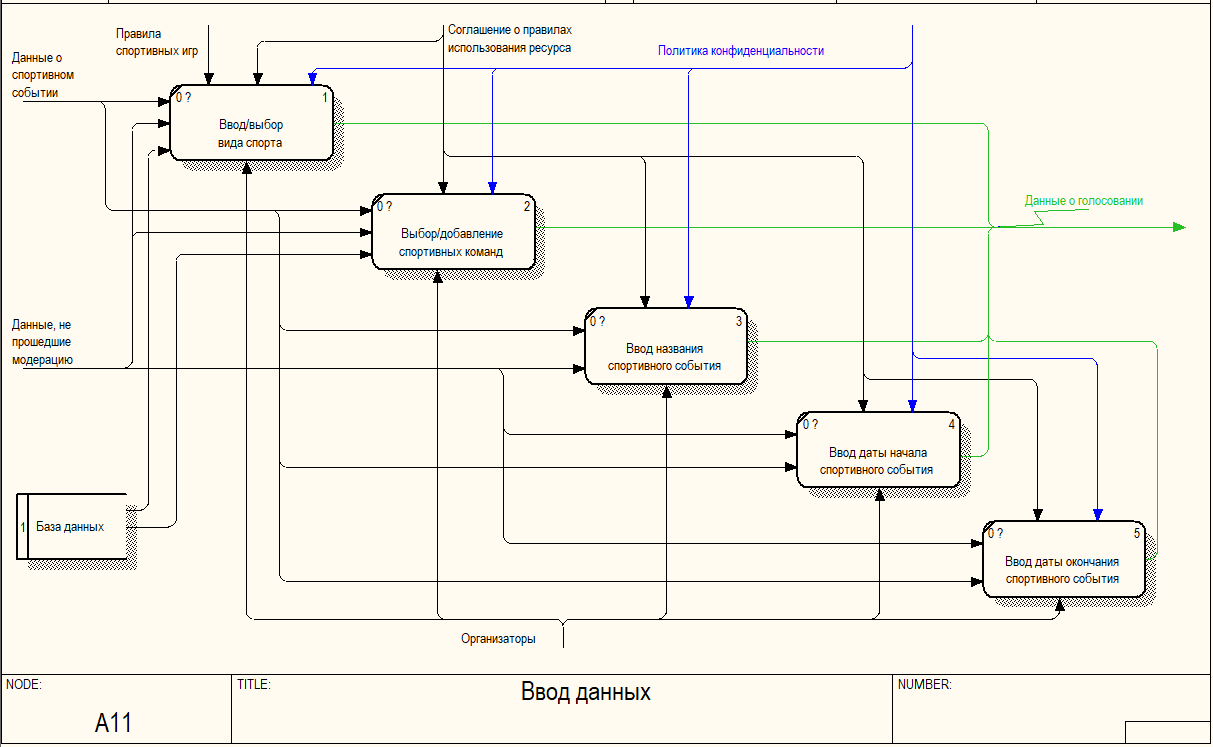


Рисунок 4 – Декомпозиция процесса «Ввод данных»

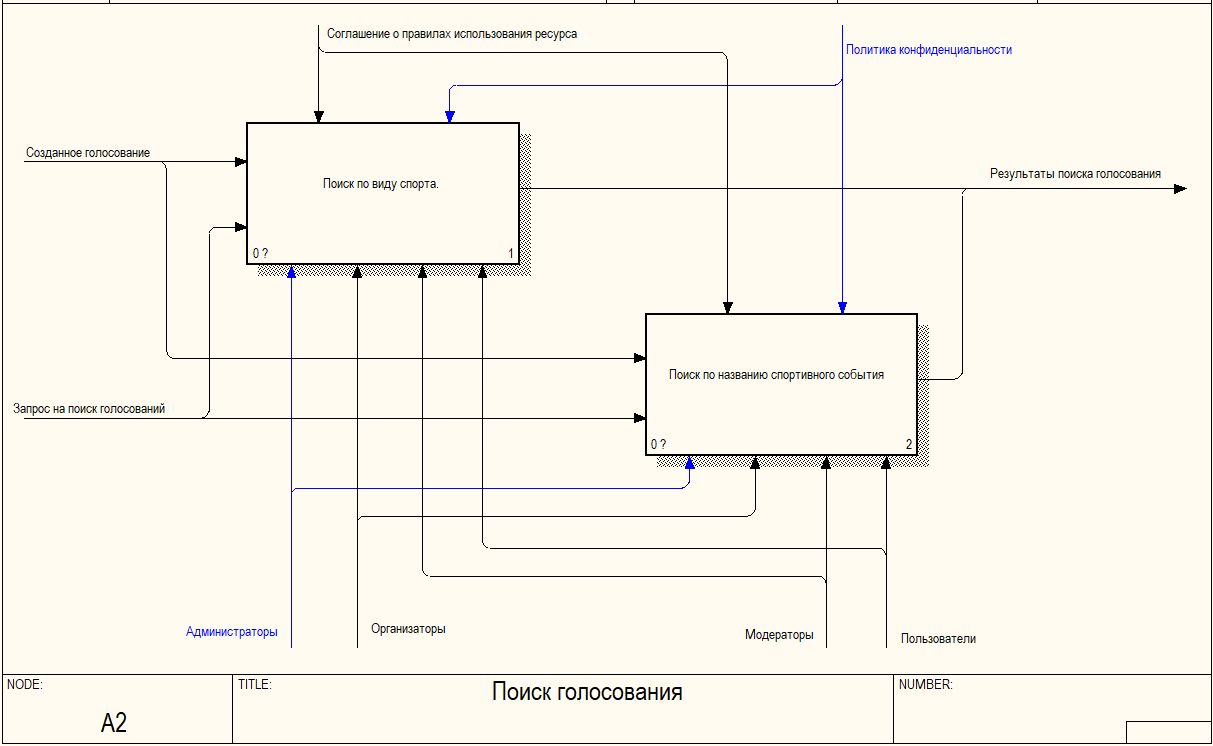


Рисунок 5 – Декомпозиция процесса «Поиск голосования»

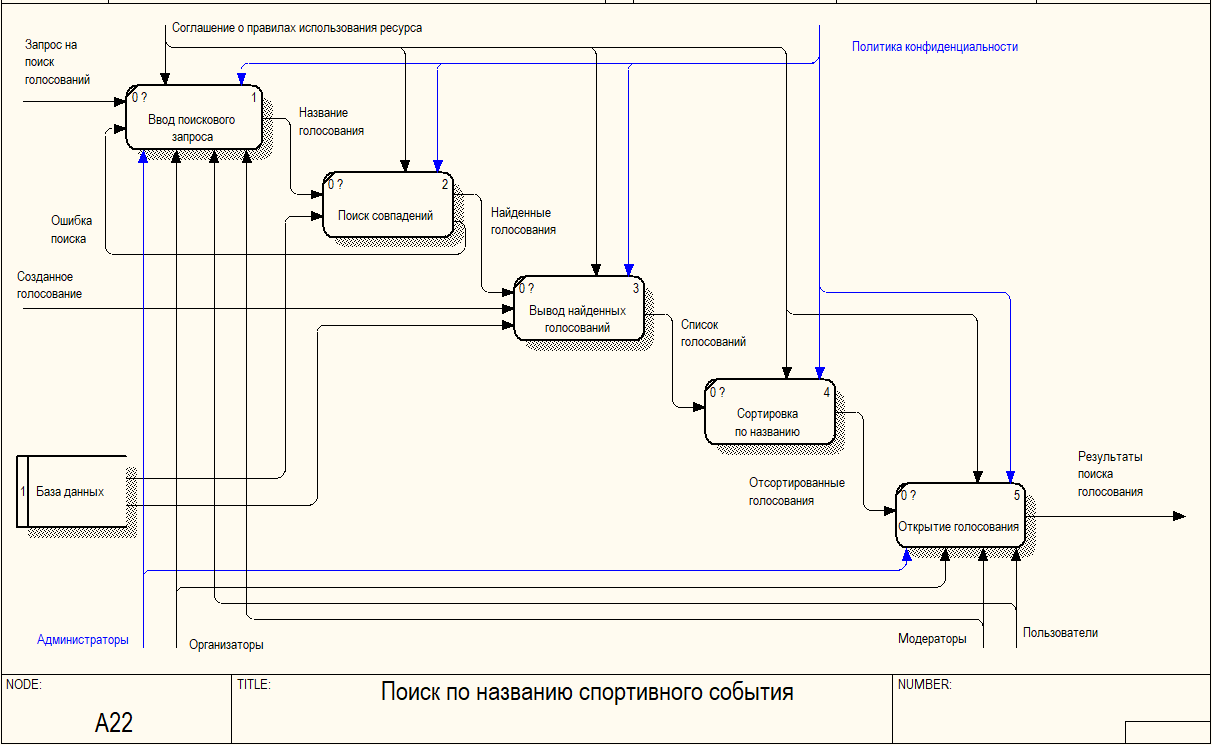


Рисунок 6 – Декомпозиция процесса «Поиск по названию спортивного события»

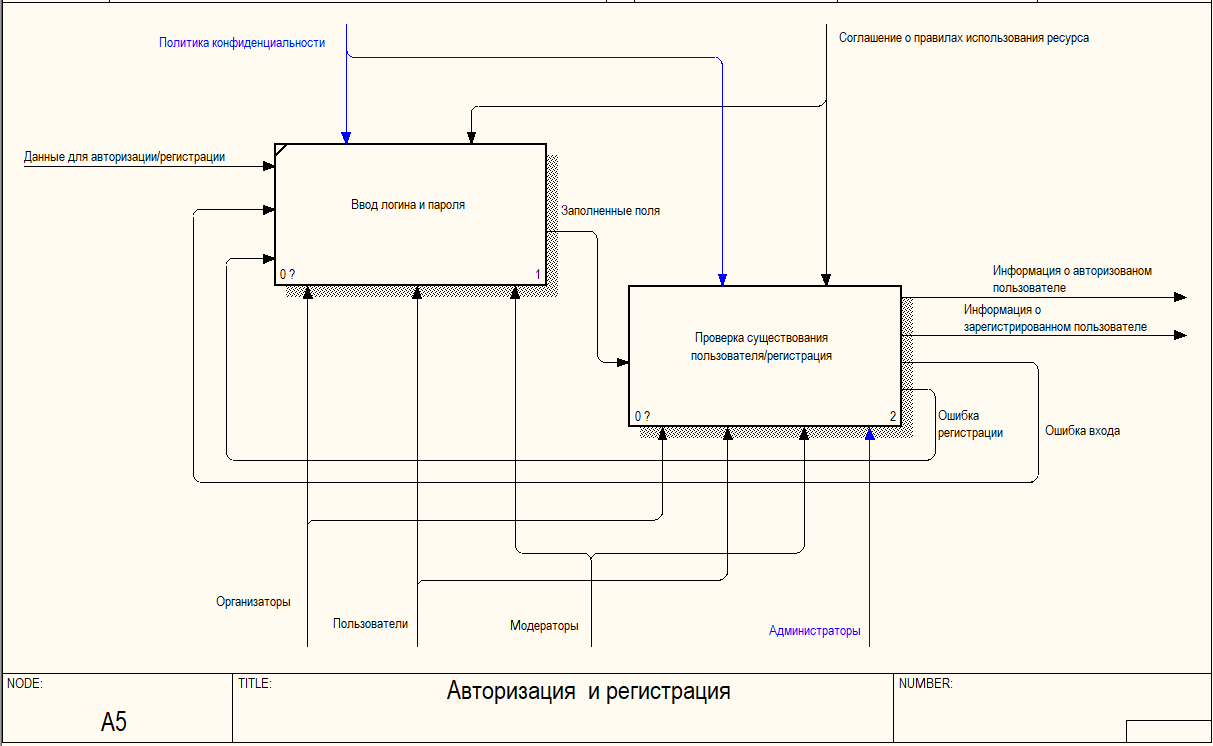


Рисунок 7 – Декомпозиция процесса «Авторизация и регистрация»

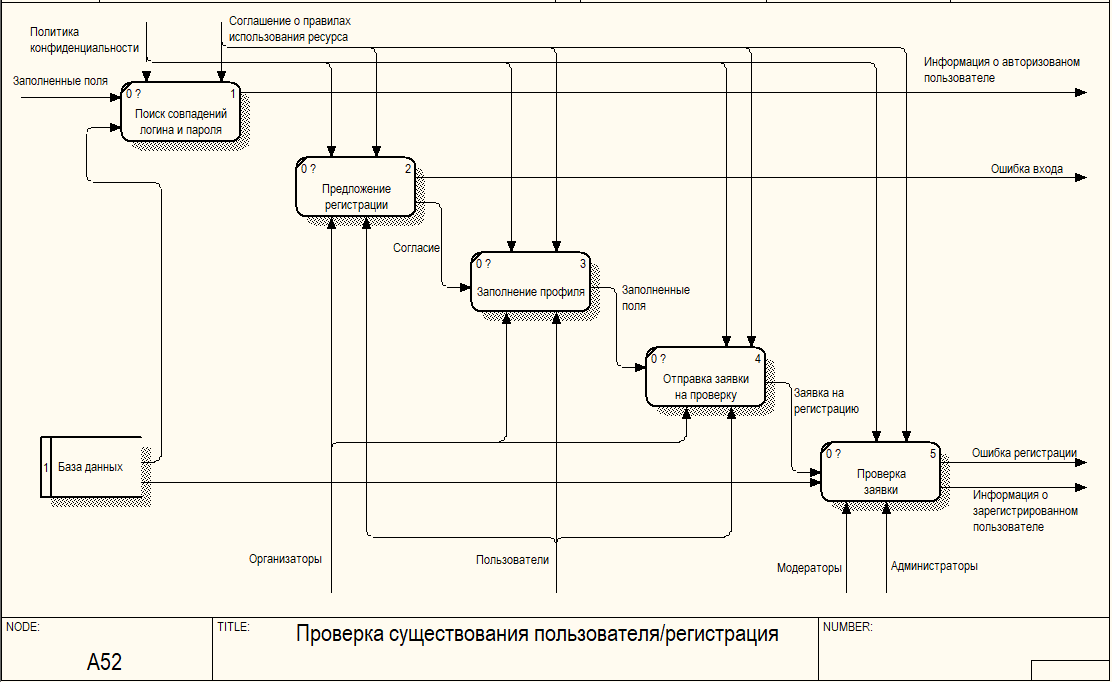


Рисунок 8 – Декомпозиция процесса «Проверка существования пользователя/регистрация»

2.1.2 Диаграмма дерева узлов

Диаграмма дерева узлов проектируемой базы данных представлена на рисунке 12.

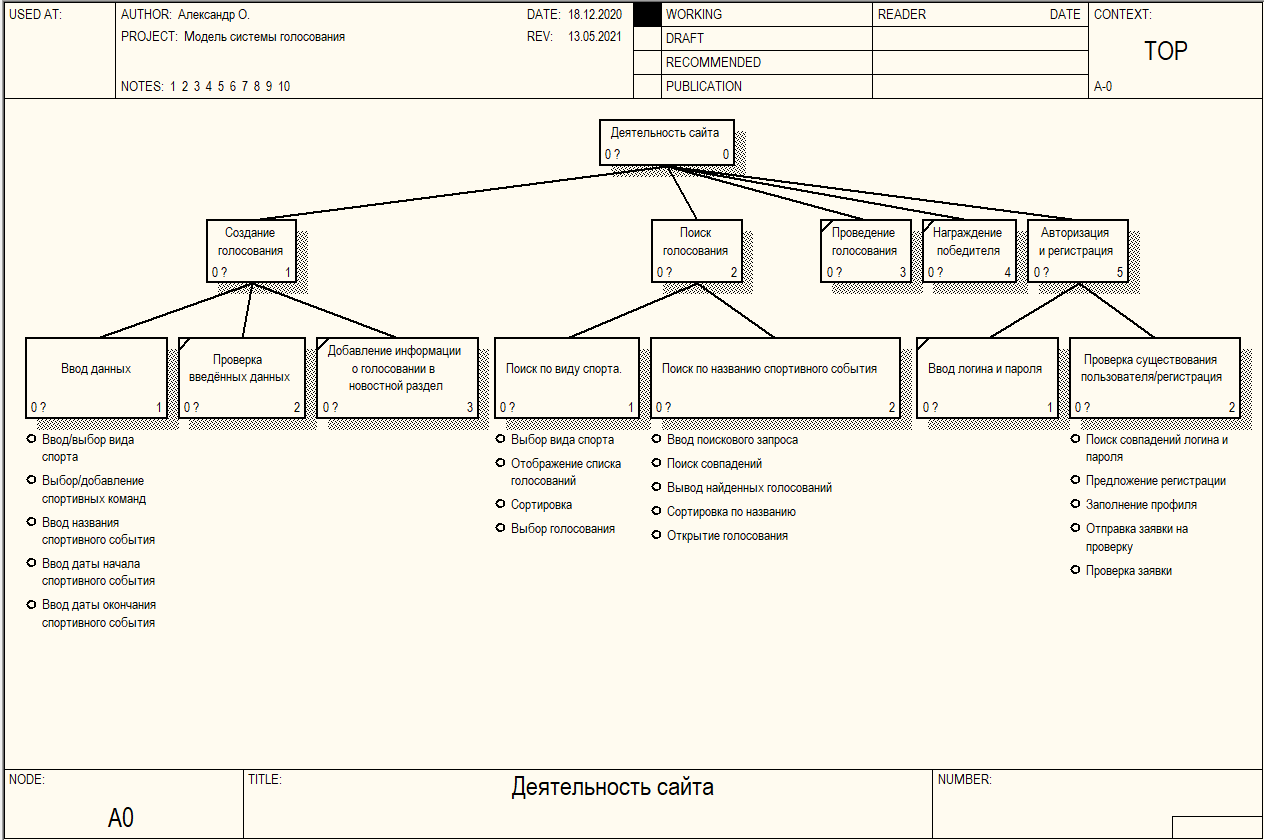


Рисунок 12 – Диаграмма дерева узлов

2.2 Информационная модель

2.2.1 Идентификация сущностей и связей. ER-диаграмма логического уровня

Erwin имеет два уровня представления модели – логический и физический. Логический уровень – это абстрактный взгляд на данные. Объекты модели, представляемые на нём, называются сущностями и атрибутами. Логическая модель данных является универсальной, т.к. не зависит от конкретной СУБД.

Для отображения информационной модели рассматриваемого процесса на логической модели используются следующие сущности:

1. Спортивные события – Команды

Связь многие ко многим, так как в одном спортивном событии участвует несколько команд, но одна команда может участвовать в разных спортивных событиях. На физическом уровне преобразуется в две связи один ко многим со связующей таблицей Команды в спортивном событии.

1. Спортивные события – Виды спорта

Связь один ко многим, идентифицирующая, так как вид спорта не имеют смысла без спортивных событий. В спортивном событии может быть только один вид спорта, но один вид спорта может быть у разных спортивных событий.

1. Спортивные события – Голосования

Связь один ко одному, неидентифицирующая, так как спортивные события могут существовать без голосований. По спортивному событию может быть только одно голосование, голосование может проводиться только по одному спортивному событию.

1. Голосования – Пользователи

Связь один ко многим, неидентифицирующая, так как пользователи могут существовать без голосований. Пользователь может создать несколько голосований, но голосование может создать только один пользователь.

1. Пользователи – Роли

Связь один ко многим, идентифицирующая, так как роли не могут существовать без пользователей. У пользователя может быть только одна роль, но одна и та же роль может быть у нескольких пользователей.

1. Пользователи – Персональные данные

Связь один к одному, идентифицирующая, так как персональные данные не могут существовать без пользователей. Персональные данные могут относиться только к одному пользователю, а у каждого пользователя может быть только одна запись персональных данных.

1. Голоса пользователей – Пользователи

Связь один ко многим, идентифицирующая, так как голоса пользователей не могут существовать без пользователей. Один пользователь может голосовать несколько раз, однако голос может принадлежать только одному пользователю.

1. Голоса пользователей – Голосования

Связь один ко многим, идентифицирующая, так как голоса пользователей не могут существовать без голосований. Один голос может относиться только к одному голосованию, но голосование может содержать несколько голосов.

1. Новостные записи – Голосования

Связь один ко многим, неидентифицирующая, так как новостные записи могут существовать без голосований. Новостная запись может быть только по одному голосованию, но по одному голосованию может быть несколько новостных записей.

1. Новостные записи – Пользователи

Связь один ко многим, неидентифицирующая, так как новостные записи могут существовать без пользователей. Новостную запись может создать только один пользователь, но один пользователь может создать несколько новостных записей.



Рисунок 13 – ER-диаграмма логического уровня

2.2.2 ER-диаграмма физического уровня. Ограничения ссылочной целостности. Переопределение триггеров. Индексирование отношений.

Все связи в модели имеют одинаковую ссылочную целостность. Везде поддерживается каскадное удаление и обновление, т.к. при отсутствии или изменении некой информации в родительской сущности, она теряет свой смысл при использовании её в дочерних сущностях.



Рисунок 14 – ER-диаграмма физического уровня

Типы данных:

1. Рассмотрим таблицу «Команды». Если взять поле «ИмяКоманды», то оно будет представлено строковым типом данных.
2. Рассмотрим таблицу «Спортивные события». Если взять поле «ДатаСпортивногоСобытия», то она представлена типом Datetime.
3. Рассмотрим таблицу «Голосования». Если взять поле «КодПользователя», то оно представлено целочисленным типом данных (integer).
4. Рассмотрим таблицу «Спортивные события». Если взять поле «РезультатСпортивногоСобытия», то оно представлено строковым типом данных.

В базе данных триггеры определены для сущностей «Пользователи», «Спортивные события», «Голосования». Данные триггеры на такие операции как INSERT, DELETE и UPDATE.

2.2.3 Определение представлений, хранимых процедур серверной компоненты

Для данной базы данных были созданы следующие представления:

1. UsersAndPersonalData – выбирает все поля из таблицы Users, добавляя при помощи JOIN все соответствующие поля из таблицы PersonalData
2. SportsEventsWithSports – выбирает все поля из таблицы SportsEvents, добавляя названия видов спорта из таблицы Sports;
3. UsersCountByRoles – полностью выводится таблица Roles с дополнительным полем UsersCount (количество пользователей каждой роли);
4. UsersWithOrganizerRole – выводятся все пользователи из таблицы Users с ролью Organizer;
5. UsersWithAdminRole – выводятся все пользователи из таблицы Users с ролью Admin;
6. UsersWithModeratorRole – выводятся все пользователи из таблицы Users с ролью Moderator;
7. UsersWithUserRole – выводятся все пользователи из таблицы Users с ролью User;
8. SportsEventsCountBySports – выводится таблица Sports с дополнительным полем SportsEventsCount (количество спортивных событий по данному виду спорта).
9. PassedVotings – выводится таблица Votings, где все голосования уже закончились;
10. CurrentVotings – выводит актуальные голосования из таблицы Votings;
11. PassedVotingsWithoutResult – выводит все прошедшие голосования из таблицы Votings, у которых ещё не внесён результат;
12. VotesCountByVotings – выводит все голосования из таблицы Votings с дополнительным полем VotesCount (количество голосов пользователей по данному голосованию);
13. SportsEventsCountByPlace – выводит из таблицы SportsEvents поле Place и дополнительное поле SportsEventsCount (количество спортивных событий в данной локации);
14. SportsEventsInBelarus – выводит из таблицы SportsEvents все записи с локацией в Беларуси;
15. SportsEventsFootball – выводит из таблицы SportsEvents все записи с видом спорта футбол.

2.3 Верификация спроектированной логической модели

| **Arrow Name** | **Entity Name** | **Attribute Name** |
| --- | --- | --- |
| Администраторы | Персональные данные |  |
|  |  | КодПерсональныхДанных |
|  |  | НомерТелефонаПользователя |
|  |  | ПочтаПользователя |
|  |  | ФИОПользователя |
|  | Пользователи |  |
|  |  | ИмяПользователя |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | ПарольПользователя |
|  | Роли |  |
|  |  | ИмяРоли |
|  |  | КодРоли |
| Выбранный вид спорта | Виды спорта | ИмяВидаСпорта |
|  |  | КодВидаСпорта |
| Данные для авторизации/регистрации | Пользователи | ИмяПользователя |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | ПарольПользователя |
| Данные о голосовании | Голосования | КодВидаСпорта |
|  |  | КодГолосования |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | КодСпортивногоСобытия |
| Данные о спортивном событии | Спортивные события | ДатаСпортивногоСобытия |
|  |  | ИмяСпортивногоСобытия |
|  |  | КодВидаСпорта |
|  |  | КодСпортивногоСобытия |
|  |  | МестоСпортивногоСобытия |
|  |  | РезультатСпортивногоСобытия |
| Данные, не прошедшие модерацию | Голосования | КодВидаСпорта |
|  |  | КодГолосования |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | КодСпортивногоСобытия |
| Информация о авторизованом пользователе | Персональные данные | КодПерсональныхДанных |
|  |  | НомерТелефонаПользователя |
|  |  | ПочтаПользователя |
|  |  | ФИОПользователя |
|  | Пользователи | ИмяПользователя |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | ПарольПользователя |
|  | Роли | ИмяРоли |
|  |  | КодРоли |
| Информация о голосовании в новостном разделе | Голосования | КодВидаСпорта |
|  |  | КодГолосования |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | КодСпортивногоСобытия |
|  | Новостные записи | ИмяНовостнойЗаписи |
|  |  | КодГолосования |
|  |  | КодНовостнойЗаписи |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | ТекстНовостнойЗаписи |
| Информация о зарегистрированном пользователе | Персональные данные | КодПерсональныхДанных |
|  |  | НомерТелефонаПользователя |
|  |  | ПочтаПользователя |
|  |  | ФИОПользователя |
|  | Пользователи | ИмяПользователя |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | ПарольПользователя |
|  | Роли | ИмяРоли |
|  |  | КодРоли |
| Информация о награждённом пользователе | Персональные данные | КодПерсональныхДанных |
|  |  | НомерТелефонаПользователя |
|  |  | ПочтаПользователя |
|  |  | ФИОПользователя |
|  | Пользователи | ИмяПользователя |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | ПарольПользователя |
|  | Роли | ИмяРоли |
|  |  | КодРоли |
| Модераторы | Персональные данные |  |
|  |  | КодПерсональныхДанных |
|  |  | НомерТелефонаПользователя |
|  |  | ПочтаПользователя |
|  |  | ФИОПользователя |
|  | Пользователи |  |
|  |  | ИмяПользователя |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | ПарольПользователя |
|  | Роли |  |
|  |  | ИмяРоли |
|  |  | КодРоли |
| Название голосования | Голосования | КодВидаСпорта |
|  |  | КодГолосования |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | КодСпортивногоСобытия |
| Найденные голосования | Голосования | КодВидаСпорта |
|  |  | КодГолосования |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | КодСпортивногоСобытия |
| Организаторы | Персональные данные |  |
|  |  | КодПерсональныхДанных |
|  |  | НомерТелефонаПользователя |
|  |  | ПочтаПользователя |
|  |  | ФИОПользователя |
|  | Пользователи |  |
|  |  | ИмяПользователя |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | ПарольПользователя |
|  | Роли |  |
|  |  | ИмяРоли |
|  |  | КодРоли |
| Отсортированнные голосования | Голосования | КодВидаСпорта |
|  |  | КодГолосования |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | КодСпортивногоСобытия |
| Отсортированные голосования | Голосования | КодВидаСпорта |
|  |  | КодГолосования |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | КодСпортивногоСобытия |
| Пользователи | Персональные данные |  |
|  |  | КодПерсональныхДанных |
|  |  | НомерТелефонаПользователя |
|  |  | ПочтаПользователя |
|  |  | ФИОПользователя |
|  | Пользователи |  |
|  |  | ИмяПользователя |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | ПарольПользователя |
|  | Роли |  |
|  |  | ИмяРоли |
|  |  | КодРоли |
| Результаты голосования | Голоса пользователей | ВыбранныйРезультатСобытия |
|  |  | КодГолосаПользователя |
|  |  | КодГолосования |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  | Голосования | КодВидаСпорта |
|  |  | КодГолосования |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | КодСпортивногоСобытия |
| Результаты поиска голосования | Голосования | КодВидаСпорта |
|  |  | КодГолосования |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | КодСпортивногоСобытия |
| Созданное голосование | Голосования |  |
|  |  | КодВидаСпорта |
|  |  | КодГолосования |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | КодСпортивногоСобытия |
| Список голосований | Голосования | КодВидаСпорта |
|  |  | КодГолосования |
|  |  | КодПерсональныхДанных |
|  |  | КодПользователя |
|  |  | КодРоли |
|  |  | КодСпортивногоСобытия |

3 Реализация системы

3.1 SQL-определения регламентированных запросов

UsersAndPersonalData выбирает информацию пользователях и персональных данных

CREATE VIEW [UsersAndPersonalData]

AS

SELECT U.Id,

U.Login,

U.Password,

Roles.Name AS RoleName,

PD.Name,

PD.PhoneNumber,

PD.Email

FROM Users AS U

LEFT JOIN PersonalData AS PD ON PD.UserId = U.Id

LEFT JOIN Roles ON Roles.Id = U.RoleId

GO

SportsEventsWithSports выбирает информацию спортивных событиях и видах спорта

CREATE VIEW [SportsEventsWithSports]

AS

SELECT SE.Id,

SE.Name,

SE.Place,

SE.Date,

S.Name AS SportName,

SE.EventResult

FROM SportsEvents AS SE

LEFT JOIN Sports AS S ON S.Id = SE.SportId

GO

UsersCountByRoles выводит количество пользователей по ролям

CREATE VIEW [UsersCountByRoles]

AS

SELECT Р.Id,

Р.Name,

COUNT(U.Id) AS UsersCount

FROM Roles AS Р

LEFT JOIN Users AS U ON U.RoleId = Р.Id

GROUP BY Р.Id, Р.Name

GO

UsersWithOrganizerRole выбирает пользователей с ролью Organizer

CREATE VIEW [UsersWithOrganizerRole]

AS

SELECT U.Id,

U.Login,

U.Password

FROM Users AS U

WHERE U.RoleId = (SELECT Id FROM Roles WHERE Name = 'Organizer')

GO

UsersWithAdminRole выбирает пользователей с ролью Admin

CREATE VIEW [UsersWithAdminRole]

AS

SELECT U.Id,

U.Login,

U.Password

FROM Users AS U

WHERE U.RoleId = (SELECT Id FROM Roles WHERE Name = 'Admin')

GO

UsersWithModeratorRole выбирает пользователей с ролью Moderator

CREATE VIEW [UsersWithModeratorRole]

AS

SELECT U.Id,

U.Login,

U.Password

FROM Users AS U

WHERE U.RoleId = (SELECT Id FROM Roles WHERE Name = 'Moderator')

GO

UsersWithUserRole выбирает пользователей с ролью User

CREATE VIEW [UsersWithUserRole]

AS

SELECT U.Id,

U.Login,

U.Password

FROM Users AS U

WHERE U.RoleId = (SELECT Id FROM Roles WHERE Name = 'User')

GO

SportsEventsCountBySports выводит количество спортивных событий по видам спорта

CREATE VIEW [SportsEventsCountBySports]

AS

SELECT S.Id,

S.Name,

COUNT(SE.Id) AS SportsEventsCount

FROM Sports AS S

LEFT JOIN SportsEvents AS SE ON SE.SportId = S.Id

GROUP BY S.Id, S.Name

GO

PassedVotings выводит информацию о завершённых голосованиях

CREATE VIEW [PassedVotings]

AS

SELECT V.Id,

SE.Name,

SE.Place,

SE.Date,

S.Name AS SportName,

SE.EventResult

FROM Votings AS V

LEFT JOIN SportsEvents AS SE ON SE.Id = V.SportEventId

LEFT JOIN Sports AS S ON S.Id = SE.SportId

WHERE SE.Date < GETDATE()

GO

CurrentVotings выводит информацию о текущих голосованиях

CREATE VIEW [CurrentVotings]

AS

SELECT V.Id,

SE.Name,

SE.Place,

SE.Date,

S.Name AS SportName,

SE.EventResult

FROM Votings AS V

LEFT JOIN SportsEvents AS SE ON SE.Id = V.SportEventId

LEFT JOIN Sports AS S ON S.Id = SE.SportId

WHERE SE.Date >= GETDATE()

GO

PassedVotingsWithoutResult выводит завершённые голосования без результата

CREATE VIEW [PassedVotingsWithoutResult]

AS

SELECT V.Id,

SE.Name,

SE.Place,

SE.Date,

S.Name AS SportName,

SE.EventResult

FROM Votings AS V

LEFT JOIN SportsEvents AS SE ON SE.Id = V.SportEventId

LEFT JOIN Sports AS S ON S.Id = SE.SportId

WHERE SE.Date < GETDATE() AND (SE.EventResult = NULL OR SE.EventResult = '')

GO

VotesCountByVotings выводит количество голосов по голосованиям

CREATE VIEW [VotesCountByVotings]

AS

SELECT V.Id,

SE.Name,

COUNT(UV.VotingId) AS VotesCount

FROM Votings AS V

LEFT JOIN SportsEvents AS SE ON SE.Id = V.SportEventId

LEFT JOIN UsersVotes AS UV ON UV.VotingId = V.Id

GROUP BY V.Id, SE.Name

GO

SportsEventsCountByPlace выводит количество спортивных событий по локациям

CREATE VIEW [SportsEventsCountByPlace]

AS

SELECT SE.Place,

COUNT(\*) AS SportsEventsCount

FROM SportsEvents AS SE

GROUP BY SE.Place

GO

SportsEventsInBelarus выводит спортивные события в Беларуси

CREATE VIEW [SportsEventsInBelarus]

AS

SELECT SE.Id,

SE.Name,

SE.Place,

SE.Date,

S.Name AS SportName,

SE.EventResult

FROM SportsEvents AS SE

LEFT JOIN Sports AS S ON S.Id = SE.SportId

WHERE SE.Place = 'Беларусь'

GO

SportsEventsFootball выводит спортивные события по футболу

CREATE VIEW [SportsEventsFootball]

AS

SELECT SE.Id,

SE.Name,

SE.Place,

SE.Date,

S.Name AS SportName,

SE.EventResult

FROM SportsEvents AS SE

LEFT JOIN Sports AS S ON S.Id = SE.SportId

WHERE SE.SportId = (SELECT Id FROM Sports WHERE Name = 'Футбол')

GO

3.2 SQL-определения триггеров

Users\_DELETE перемещающает удалённых пользователей в специальную таблицу

CREATE TRIGGER Users\_DELETE

ON Users

AFTER DELETE

AS

INSERT INTO DeletedUsers

(Login, Password, RoleId)

SELECT Login, Password, RoleId

FROM deleted

GO

PersonalData\_INSERT отменяет операцию добавления персональных данных, если пользователя не существует

CREATE TRIGGER PersonalData\_INSERT

ON PersonalData

AFTER INSERT

AS

IF NOT EXISTS(SELECT \* FROM Users, inserted WHERE Users.Id = inserted.Id)

BEGIN

PRINT 'Пользователя не существует.'

ROLLBACK TRANSACTION

END

GO

PersonalData\_DELETE перемещающает удалённые персональные данные в специальную таблицу

CREATE TRIGGER PersonalData\_DELETE

ON PersonalData

AFTER DELETE

AS

INSERT INTO DeletedPersonalData

(Name, PhoneNumber, Email, UserId)

SELECT Name, PhoneNumber, Email, UserId

FROM deleted

GO

Users\_INSERT отменяет операцию добавления пользователя, если логин уже занят

CREATE TRIGGER Users\_INSERT

ON Users

AFTER INSERT

AS

IF EXISTS(SELECT \* FROM Users, inserted WHERE Users.Login = inserted.Login)

BEGIN

PRINT 'Логин занят.'

ROLLBACK TRANSACTION

END

GO

Users\_UPDATE отменяет операцию обновления пользователя, если новый логин уже занят другим пользователем

CREATE TRIGGER Users\_UPDATE

ON Users

AFTER UPDATE

AS

IF EXISTS(SELECT \* FROM Users, inserted WHERE Users.Login = inserted.Login)

BEGIN

PRINT 'Логин занят.'

ROLLBACK TRANSACTION

END

GO

SportsEvents\_INSERT отменяет операцию добавления спортивного события без заполненного результат, если событие завершено

CREATE TRIGGER SportsEvents\_INSERT

ON SportsEvents

AFTER INSERT

AS

IF EXISTS(SELECT \* FROM inserted WHERE inserted.Date < GETDATE() AND inserted.Result = NULL)

BEGIN

PRINT 'Введите результат события.'

ROLLBACK TRANSACTION

END

GO

Votings\_INSERT добавляет в специальную таблицу дату вставки новой записи в таблицу Votings

CREATE TRIGGER Votings\_INSERT

ON Votings

AFTER INSERT

AS

INSERT INTO InsertedVotings

(InsertDate)

VALUES (GETDATE())

GO

Teams\_INSERT отменяет операцию добавления спортивной команды, если её вид спорта отсутствует в базе

CREATE TRIGGER Teams\_INSERT

ON Teams

AFTER INSERT

AS

IF NOT EXISTS(SELECT \* FROM Sports, inserted WHERE Sports.Id = inserted.SportId)

BEGIN

PRINT 'Виды спорта нет в базе. Сначала добавьте вид спорта'

ROLLBACK TRANSACTION

END

GO

Teams\_DELETE перемещает удалённые спортивные команды в специальную таблицу

CREATE TRIGGER Teams\_DELETE

ON Teams

AFTER DELETE

AS

INSERT INTO DeletedTeams

(Name, SportId)

SELECT Name, SportId

FROM deleted

GO

3.3 SQL-определения хранимых процедур

VotingsWithoutResult выводит все голосования без результатов

CREATE PROCEDURE [VotingsWithoutResult]

AS

SELECT V.Id,

SE.Name,

SE.Place,

SE.Date,

S.Name,

SE.EventResult

FROM Votings AS V

LEFT JOIN SportsEvents AS SE ON SE.Id = V.SportEventId

LEFT JOIN Sports AS S ON S.Id = SE.SportId

WHERE SE.EventResult = NULL OR SE.EventResult = ''

GO

AllVotings выводит информацию о всех голосованиях все голосования

CREATE PROCEDURE [AllVotings]

AS

SELECT V.Id,

SE.Name,

SE.Place,

SE.Date,

S.Name,

SE.EventResult

FROM Votings AS V

LEFT JOIN SportsEvents AS SE ON SE.Id = V.SportEventId

LEFT JOIN Sports AS S ON S.Id = SE.SportId

GO

UserVotings принимает как параметр id пользователя и выводит все голоса, сделанные им

CREATE PROCEDURE [UserVotings]

@userId INT

AS

SELECT V.Id,

SE.Name,

SE.Place,

SE.Date,

S.Name,

SE.EventResult

FROM Votings AS V

LEFT JOIN SportsEvents AS SE ON SE.Id = V.SportEventId

LEFT JOIN Sports AS S ON S.Id = SE.SportId

WHERE V.UserId = @userId

GO

AddSport принимает название вида спорта и добавляет новый вид спорта в таблицу Sports

CREATE PROCEDURE [AddSport]

@name NVARCHAR(50)

AS

INSERT INTO Sports (Name)

VALUES (@name)

GO

AddNewsEntry принимает в качестве параметров данные о новостной записи и добавляет новостную запись в таблицу NewsEntries

CREATE PROCEDURE [AddNewsEntry]

@name NVARCHAR(100),

@text NVARCHAR(750),

@voteId INT,

@userId INT

AS

INSERT INTO NewsEntries(Name, Text, VotingId, UserId)

VALUES (@name, @text, @voteId, @userId)

GO

AddSportEvent добавляет новое спортивное события с данными из передаваемых параметров

CREATE PROCEDURE [AddSportEvent]

@name NVARCHAR(100),

@date DATE,

@palce NVARCHAR(50),

@result NVARCHAR(50)

AS

INSERT INTO SportsEvents (Name, Date, Place, EventResult)

VALUES (@name, @date, @palce, @result)

GO

ChangeUserLogin принимает id пользователя и логин, заменяя старый логин у новый у пользователя с этим id

CREATE PROCEDURE [ChangeUserLogin]

@id INT, @login NVARCHAR(50)

AS

UPDATE Users

SET Login = @login

WHERE Id = @id

GO

AllUsers выводит данные всех пользователей

CREATE PROCEDURE [AllUsers]

AS

SELECT U.Id,

U.Login,

U.Password

FROM Users AS U

GO

ChangeSportEventResult устанавливает результат спортивного события

CREATE PROCEDURE [ChangeSportEventResult]

@id INT, @result NVARCHAR(50)

AS

UPDATE SportsEvents

SET EventResult = @result

WHERE Id = @id

GO

AdminsCount выводит количество пользователей с ролью Admin

CREATE PROCEDURE [AdminsCount]

AS

SELECT COUNT(\*)

FROM Users

WHERE RoleId = 1

GO

CurrentVotingsCount выводит количество актуальных голосований

CREATE PROCEDURE [CurrentVotingsCount]

AS

SELECT COUNT(\*)

FROM SportsEvents

WHERE SportsEvents.Date >= GETDATE()

GO

AllSports выводит данные всех видов спорта

CREATE PROCEDURE [AllSports]

AS

SELECT Id, Name

FROM Sports

GO

SportsCount выводит количество всех видов спорта

CREATE PROCEDURE [SportsCount]

AS

SELECT COUNT(\*)

FROM Sports

GO

UsersWithRole принимает название роли и выводит всех пользователей с данной ролью

CREATE PROCEDURE [UsersWithRole]

@roleName NVARCHAR(50)

AS

SELECT U.Id,

U.Login,

U.Password

FROM Users AS U

WHERE U.RoleId = (SELECT Id FROM Roles WHERE Name = @roleName)

GO

AllSportsEvents выводит данные всех спортивных событий

CREATE PROCEDURE [AllSportsEvents]

AS

SELECT SE.Id,

SE.Name,

SE.Place,

SE.Date,

S.Name,

SE.EventResult

FROM SportsEvents AS SE

LEFT JOIN Sports AS S ON S.Id = SE.SportId

GO

SportsEventsBySport принимает название вида спорта и выводит все спортивные события с этим видом спорта

CREATE PROCEDURE [SportsEventsBySport]

@sport NVARCHAR(50)

AS

SELECT SE.Id,

SE.Name,

SE.Place,

SE.Date,

S.Name,

SE.EventResult

FROM SportsEvents AS SE

LEFT JOIN Sports AS S ON S.Id = SE.SportId

WHERE SE.SportId = (SELECT Id FROM Sports WHERE Name = @sport)

GO

SportsEventsByPlace принимает название локации и выводит все спортивные события с этой локацией

CREATE PROCEDURE [SportsEventsByPlace]

@place NVARCHAR(50)

AS

SELECT SE.Id,

SE.Name,

SE.Place,

SE.Date,

S.Name,

SE.EventResult

FROM SportsEvents AS SE

LEFT JOIN Sports AS S ON S.Id = SE.SportId

WHERE SE.Place = @place

GO

DeleteVotingById удаляет голосование по id

CREATE PROCEDURE [DeleteVotingById]

@id INT

AS

DELETE FROM Votings

WHERE Id = @id

GO

AllNewsEntries выводит информацию о всех новостных записях

CREATE PROCEDURE [AllNewsEntries]

AS

SELECT Id, Name, Text, VotingId, UserId

FROM NewsEntries

GO

DeleteNewsEntryById удаляет новостные записи по id

CREATE PROCEDURE [DeleteNewsEntryById]

@id INT

AS

DELETE FROM NewsEntries

WHERE Id = @id

GO

3.4 SQL-определения курсоров

Cursor1 используется для подсчёта количества пользователей

DECLARE Cursor1 CURSOR

SCROLL KEYSET

TYPE\_WARNING

FOR

SELECT Id FROM Users

FOR READ ONLY

OPEN Cursor1

DECLARE @Counter INT

SET @Counter = @@CURSOR\_ROWS

SELECT @Counter

ClOSE Cursor1

DEALLOCATE Cursor1

GO

Cursor2 используется для подсчёта количества спортивных событий

DECLARE Cursor2 CURSOR

SCROLL KEYSET

TYPE\_WARNING

FOR

SELECT Id FROM SportsEvents

FOR READ ONLY

OPEN Cursor2

DECLARE @Counter INT

SET @Counter = @@CURSOR\_ROWS

SELECT @Counter

ClOSE Cursor2

DEALLOCATE Cursor2

GO

Cursor3 используется для вывода данных всех пользователей и подсчёта обсчего количества пользователей

DECLARE Cursor3 CURSOR

GLOBAL SCROLL KEYSET

TYPE\_WARNING

FOR

SELECT Login, Password, RoleId

FROM Users

FOR READ ONLY

DECLARE @login NVARCHAR(50), @password NVARCHAR(50), @roleId INT

OPEN Cursor3

DECLARE @Counter INT

SET @Counter = @@CURSOR\_ROWS

FETCH NEXT FROM Cursor3 INTO @login, @password, @roleId

WHILE @@FETCH\_STATUS = 0

BEGIN

PRINT @login + ' ' + @password + ' ' + CONVERT(NVARCHAR(50), @roleId)

FETCH NEXT FROM Cursor3 INTO @login, @password, @roleId

END

PRINT 'Всего записей: ' + CONVERT(NVARCHAR(5), @Counter)

ClOSE Cursor3

DEALLOCATE Cursor3

GO

Cursor4 используется для изменения пароля пользователя

DECLARE Cursor4 CURSOR

FOR

SELECT Password FROM Users

OPEN Cursor4

FETCH Cursor4

UPDATE Users SET Password = ''

WHERE CURRENT OF Cursor4

SELECT Password FROM Users WHERE Id = 1

ClOSE Cursor4

DEALLOCATE Cursor4

GO

Cursor5 используется для вывода данных о пользователях, у которых имя короче 25 символов, и их количества

DECLARE Cursor5 CURSOR

GLOBAL SCROLL KEYSET

TYPE\_WARNING

FOR

SELECT Name, PhoneNumber, Email

FROM PersonalData

FOR READ ONLY

DECLARE @name NVARCHAR(100), @phone NVARCHAR(50), @email NVARCHAR(50)

OPEN Cursor5

DECLARE @counter INT

SET @counter = 0

PRINT 'Пользователи с ФИО, короче 25 символов: '

FETCH NEXT FROM Cursor5 INTO @name, @phone, @email

WHILE @@FETCH\_STATUS = 0

BEGIN

IF LEN(@name) < 25

BEGIN

SET @counter = @counter + 1

PRINT @name + @phone + @email

END

FETCH NEXT FROM Cursor5 INTO @name, @phone, @email

END

PRINT 'Всего: ' + CONVERT(NVARCHAR(5), @counter)

CLOSE Cursor5

DEALLOCATE Cursor5

GO

3.5 Распределение прав доступа пользователей

В соответствии с указанными ранее категориями пользователей – пользователь, организатор, модератор и администратор, в базу данных добавлены аккаунты для них.

CREATE LOGIN Admin

WITH PASSWORD = 'Admin';

CREATE LOGIN Moderator

WITH PASSWORD = 'Moderator';

CREATE LOGIN Organizer

WITH PASSWORD = 'Organizer';

CREATE LOGIN User1

WITH PASSWORD = 'User1';

CREATE USER Admin FOR LOGIN Admin;

CREATE USER Moderator FOR LOGIN Moderator;

CREATE USER Organizer FOR LOGIN Organizer;

CREATE USER User1 FOR LOGIN User1;

Пользователю Admin будут предоставлены основные права на чтение, изменение, добавление и изменение данных во всех таблицах базы данных.

GRANT SELECT, INSERT, UPDATE, DELETE ON [Votings] TO Admin

GRANT SELECT, INSERT, UPDATE, DELETE ON [Users] TO Admin

GRANT SELECT, INSERT, UPDATE, DELETE ON [PersonalData] TO Admin

GRANT SELECT, INSERT, UPDATE, DELETE ON [NewsEntries] TO Admin

GRANT SELECT, INSERT, UPDATE, DELETE ON [Sports] TO Admin

GRANT SELECT, INSERT, UPDATE, DELETE ON [SportsEvents] TO Admin

GRANT SELECT, INSERT, UPDATE, DELETE ON [Teams] TO Admin

GRANT SELECT, INSERT, UPDATE, DELETE ON [TeamsInSportEvent] TO Admin

Пользователю Moderator предоставляются разрешения на чтение данных из всех таблиц.

GRANT SELECT ON [Votings] TO Moderator

GRANT SELECT ON [Users] TO Moderator

GRANT SELECT ON [PersonalData] TO Moderator

GRANT SELECT ON [NewsEntries] TO Moderator

GRANT SELECT ON [Sports] TO Moderator

GRANT SELECT ON [SportsEvents] TO Moderator

GRANT SELECT ON [Teams] TO Moderator

GRANT SELECT ON [TeamsInSportEvent] TO Moderator

Пользователю Organizer предоставляются права на чтение, добавление, изменение и удаление данных в таблицах с голосованиями, новостными записями, спортивными событиями, командами и командами в спортивных событиях, а также права на чтение и добавление данных в таблице с видами спорта. Таким образом организатор может просматривать и создавать, а также изменять и удалять свои голосования и всё, что с ними связано.

GRANT SELECT, INSERT, UPDATE, DELETE ON [Votings] TO Organizer

GRANT SELECT, INSERT, UPDATE, DELETE ON [NewsEntries] TO Organizer

GRANT SELECT, INSERT ON [Sports] TO Organizer

GRANT SELECT, INSERT, UPDATE, DELETE ON [SportsEvents] TO Organizer

GRANT SELECT, INSERT, UPDATE, DELETE ON [Teams] TO Organizer

GRANT SELECT, INSERT, UPDATE, DELETE ON [TeamsInSportEvent] TO Organizer

Пользователю User1 предоставлены права чтение данных из таблиц с голосованиями, новостными записями, видами спорта, спортивными событиями, командами и командами в спортивных событиях.

GRANT SELECT ON [Votings] TO User1

GRANT SELECT ON [NewsEntries] TO User1

GRANT SELECT ON [Sports] TO User1

GRANT SELECT ON [SportsEvents] TO User1

GRANT SELECT ON [Teams] TO User1

GRANT SELECT ON [TeamsInSportEvent] TO User1

4 Результаты тестирования информационной системы

При запуске приложения открывается главная страница – страница с текущими голосованиями.

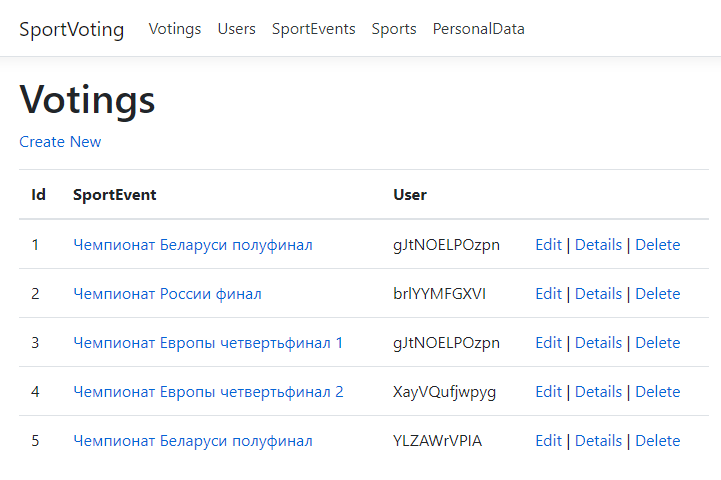


Рисунок 15 – Главная страница приложения

В верхней части главной страницы расположено навигационное меню, используя которое пользователь может перемещаться между страницами с голосованиями, пользователями, спортивными событиями, видами спорта.

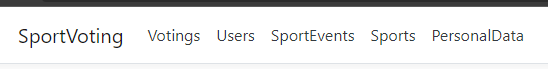


Рисунок 16 – Навигационное меню сайта

На странице с голосованиями расположен список голосований с основной информацией, а также на каждой строке списка имеются три кнопки действий: Edit (изменить), Details (детали), Delete (удалить).



Рисунок 17 – Кнопки действий

По нажатию на кнопку Details открывается страница с деталями голосования: название спортивного события, по которому проводится голосование, и логин пользователя, который создал это голосование.

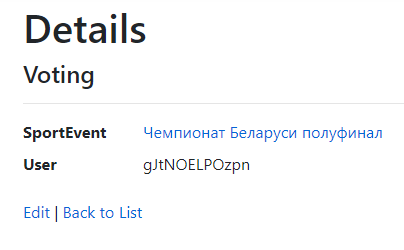


Рисунок 18 – Страница с деталями голосования

Внизу этой страницы есть две кнопки:

1. Edit – аналогична той, что находится на странице с голосованиями.
2. Back to List – кнопка возврата к списку голосований.

Если на странице с голосованиями нажать кнопку Delete, откроется страница подтверждения действия. Это позволяет избежать случайного удаления данных.

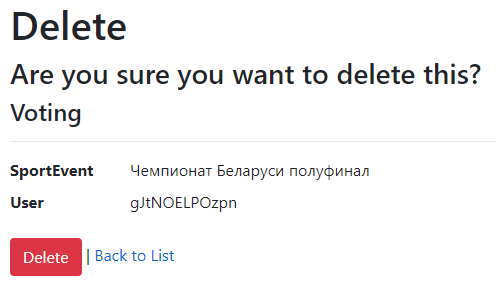


Рисунок 19 – Страница подтверждения удаления голосования

Для удаления нужно нажать красную кнопку Delete, что приведёт к удалению голосования и возврату к списку всех голосований. Если удалять голосование не требуется, можно нажать кнопку Back to List, которая снова откроет таблицу со списком голосований.

При нажатии на кнопку Edit на странице со списком голосований или на странице с деталями голосования откроется страница редактирования голосования.

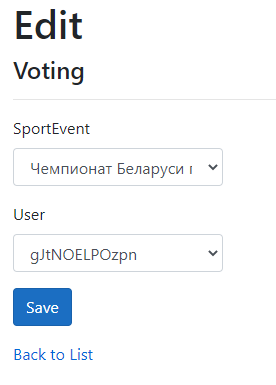


Рисунок 20 – Страница редактирования голосования

На этой странице можно изменить голосование. В зависимости от сущности на странице могут находится текстовые поля, выпадающие списки, а также датапикеры (поля для выбора даты с выпадающим календарём). Для сохранения изменений нужно нажать на синюю кнопку Save. Для отмены изменений можно нажать на кнопку Back to List.

На странице со списком голосований в самом верху находится кнопка добавления нового голосования Create New.

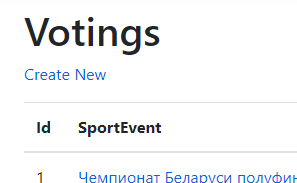


Рисунок 21 – кнопка добавления нового голосования

По нажатию на неё открывается страница добавления голосования. Она аналогично странице изменения, но текстовые поля в ней изначально будут пустыми, а селекторы (выпадающие списки) буду содержать значение по умолчанию.

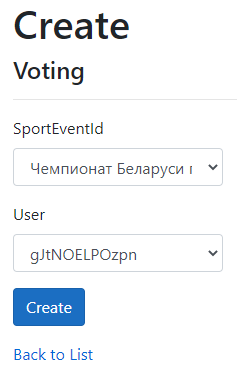


Рисунок 22 – Страница добавления голосования

По нажатию на синюю кнопку Create произойдёт добавление голосования и переход на страницу со списком голосований. Для отмены действия можно нажать кнопку Back to List.

На странице со списком голосований и с деталями голосования можно нажать на название спортивного события, по которому проводится голосование, что приведёт к открытию страницы с деталями этого спортивного события.

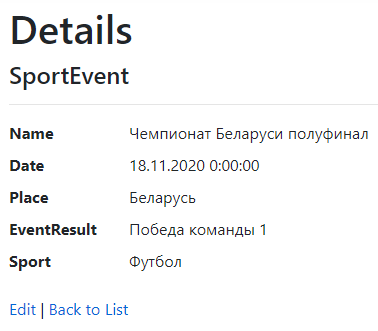


Рисунок 23 – Страница с деталями спортивного события

Остальные разделы сайта имеют ту же структуру и функционал.

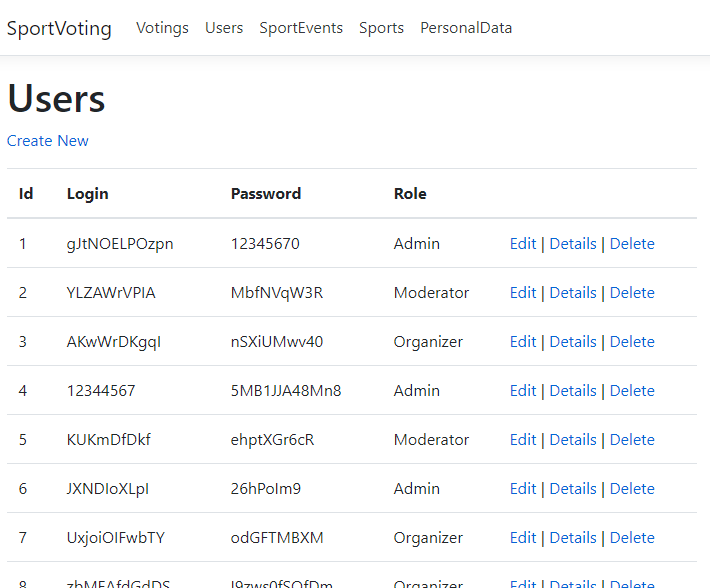


Рисунок 24 – Страница со списком пользователей

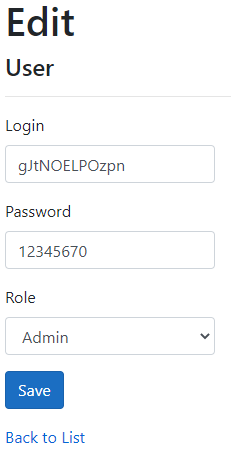


Рисунок 25 – Страница изменения пользователя

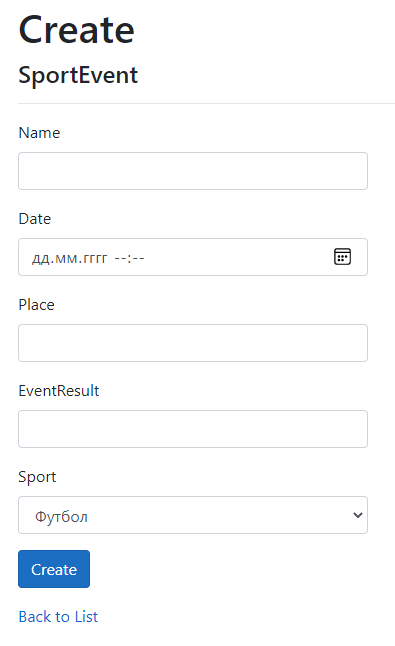


Рисунок 26 – Страница добавления спортивного события

Заключение

В ходе выполнения курсовой работы был разработан программный модуль, необходимый для организации и проведения онлайн-голосований за исходы спортивных событий. Написанное приложение агрегирует в себе большое количество спортивных событий по различным видам спорта расположена в сети интернет, что позволяет увеличить охват пользователей.

Список использованных источников

* 1. Маркин, А. В. Построение запросов и программирование на SQL. Учебное пособие / А.В. Маркин. - М.: Диалог-Мифи, 2014. - 384 c.
  2. [Бхамидипати](http://oz.by/books/more106510.html?id_search=127378#writer), К. SQL. Справочник программиста/ К.Бхамидипати–М.: Эком, 2003. – 304 с.
  3. Бьюли, А. Изучаем SQL / А.Бьюли, Э. Оппель. – М.: Символ, 2007. – 312 с.
  4. Жилинский, А. Самоучитель Microsoft SQL Server 2005 / А.Жилинский – СПб: BHV, 2004. – 224 с.
  5. Форта, Бен Освой самостоятельно SQL за 10 минут / Бен Форта. - М.: Вильямс, 2015. - 288 c.
  6. Павловская, Т. А. С#. Программирование на языке высокого уровня: учебник для вузов / Т. А. Павловская. – СПб. : Питер, 2007. – 432 с.: ил.
  7. Подбельский, В. В. Язык С#. Решение задач / В. В. Подбельский. – М.: Инфра-М, 2014. – 296 с.
  8. Троелсен, Э. Язык программирования С# 4.0 и платформа .NET 4.0 : пер. с англ. / Э. Троелсен. – М. : Вильямс, 2011. – 1392 с.
  9. Дейт, К. Дж. SQL и реляционная теория. Как грамотно писать код на SQL / К.Дж. Дейт. - М.: Символ-плюс, **2017**. - 480 c.
  10. Еремин И.И., Астафьев Н.Н. Введение в теорию линейного и выпуклого программирования -М.: Наука, 1976 г. – 239 с.
  11. Общий курс высшей математики для экономистов. Учебник / под ред В.И. Ермакова.- М.: ИНФА - М. - 656 с. - (серия «высшее образование»).
  12. Грабер, Мартин SQL для простых смертных / Мартин Грабер. - М.: ЛОРИ, 2014. - 378 c.