Разработка школьного сайта для учащихся

Разработал:

Студент гр. ПР-41 \_\_\_\_\_\_\_\_\_\_\_\_\_\_/Д.А. Пинигина

Руководитель: \_\_\_\_\_\_\_\_\_/А.А. Кузьмина

2024

**ЗАДАНИЕ**

**на проектирование программного продукта**

**Курс IV**

**Студенту:**

**Фамилия** Пинигина

**Имя** Дарья

**Отчество** АндреевнаГруппа ПР-41

Тема программы: Разработка школьного сайта для учащихся

**Техническое задание должно содержать следующие разделы:**

- Титульный лист

- Лист задания на курсовой проект

- Содержание

Введение

1. Описание предметной области. Постановка задачи
2. Выбор средств/методологии проектирования. Выбор СУБД
3. Построение инфологической (концептуальной) модели предметной области
4. Проектирование логической структуры БД
5. Формулировка ограничений целостности

5.1 Выявление полного перечня ограниченной целостности, присущего данной предметной области

5.2 Определение перечня ограничений целостности, которые будут контролироваться в данном проекте

5.3 Выбор способа реализации контроля целостности для каждого из ограничений

1. Проектирование физической структуры базы данных
2. Организация ввода данных в БД
3. Организация корректировки БД
4. Реализация запросов пользователя, получение отчетов
5. Разработка пользовательского интерфейса
6. Реализация проекта в среде конкретной СУБД

Условные сокращения

Заключение

Список литературы

Приложения

Приложение А Основные таблицы БД

Приложение Б Схема межтабличных связей

Приложение В Интерфейсные формы диалога пользователей с БД

Приложение Г Образцы отчетов пользователей

Приложение Д Образцы запросов пользователей к БД

**Содержание**

Введение

1. Описание предметной области. Постановка задачи
2. Выбор средств/методологии проектирования. Выбор СУБД
3. Построение инфологической (концептуальной) модели предметной области
4. Проектирование логической структуры БД
5. Формулировка ограничений целостности

5.1 Выявление полного перечня ограниченной целостности, присущего данной предметной области

5.2 Определение перечня ограничений целостности, которые будут контролироваться в данном проекте

5.3 Выбор способа реализации контроля целостности для каждого из ограничений

1. Проектирование физической структуры базы данных
2. Организация ввода данных в БД
3. Организация корректировки БД
4. Реализация запросов пользователей, получение отчетов
5. Разработка пользовательского интерфейса
6. Реализация проекта в среде конкретной СУБД

Условные сокращения

Заключение

Список литературы

Приложения

Приложение А Основные таблицы БД

Приложение Б Схема межтабличных связей

Приложение В Интерфейсные формы диалога пользователей с БД

Приложение Г Образцы отчётов пользователей

Приложение Д Образцы запросов пользователей к БД

# ВВЕДЕНИЕ

Постоянные расспросы домашних работ, некорректные работы с общедоступными сайтами для выставления домашних работ, плохая успеваемость в школах. Другие ученики могут досаждать с просьбами о том, а какое домашнее задание задали по математике. Это может конкретно раздражать некоторых или создавать некомфортные условия для работы с учениками.

Исходя из этого, данная проблема точно нуждается в решении. Для этого потребуется связаться с учебной частью школы, которая понимает, как устроена школа и что хотят от этого люди в ней. Для продвижения и помощи ученикам и учителям следует также связаться с теми, кто знает и имеет все данные о школе. Информационная система, база данных которой содержит сведения о школе, учениках, предметах, мероприятиях, могла бы оказать помощь в решении вышеуказанных проблем.

Цель проекта – разработать школьный сайт для учащихся.

# 1 Описание предметной области. Постановка задач

Школа — это учебное заведение, предназначенное для организации учебного пространства и учебной среды для обучения школьников под руководством учителей. Это место, где дети получают образование, а также место, где они проводят значительную часть своего дня, участвуя в различных мероприятиях. Школы могут управляться как государственными, так и частными организациями и включать в себя начальную школу для детей младшего возраста и среднюю школу для подростков, закончивших начальное образование. Школы также могут быть посвящены одной конкретной области, например, экономическая школа или школа танцев. Школьная среда включает в себя школьные помещения и различные правила, которые делают ее идеальным местом для обучения.

Учителя — это профессионалы, которые создают увлекательные уроки, поддерживают процесс обучения и способствуют академической успеваемости в классе. Они необходимы для поддержки учащихся на всех этапах жизни - от детей младшего возраста и студентов колледжей до высших учебных заведений. Они являются экспертами в специальных предметах, которые преподают, или работают по всем направлениям, обучая целому ряду навыков и тем. Преподаватели отвечают за разработку и реализацию планов занятий, оценку успехов учащихся и создание позитивной учебной среды.

Ученик — это человек, официально занимающийся обучением, особенно зачисленный в школу. Это люди, которые исследуют и изучают что-то новое. Учеников можно охарактеризовать по-разному, например, как целеустремленных, обучающихся, аналитиков, инициативных, сосредоточенных, зрелых, уравновешенных, понимающих, бойцов, энергичных, скромных и амбициозных. Они приходят в школу готовыми к восприятию новых знаний и критически мыслящими.

Школьное расписание — это справочный документ, созданный профессионалами, который наглядно показывает, как школьные ресурсы, такие как учителя и классы, сочетаются с расписанием учащихся и школьным графиком, а также с днями недели. Это план или схема в школе, показывающая ежедневное распределение времени между несколькими предметами, видами деятельности и классами. Это зеркало, в котором отражается вся образовательная программа, реализуемая в школе. Расписание — это организационный инструмент, используемый для четкого представления о будущих событиях. Оно представляет собой структуру, которая позволяет людям видеть, как выстраиваются конкретные мероприятия.

Директор школы — это руководитель образовательного процесса и главный управляющий школы. Они отвечают за руководство, управление и надзор за повседневной деятельностью школы.

Директора обеспечивают дальновидное руководство школой, задавая тон и направление для достижения академических и организационных целей. Они создают безопасную учебную среду, ставят задачи перед учениками и учителями и контролируют процесс достижения этих целей.

Директора контролируют работу учителей и педагогического персонала и следят за успеваемостью учащихся. Они оценивают преподавателей и других сотрудников, чтобы убедиться в том, что стандартная учебная программа соблюдается, и выявляют области, в которых необходимо улучшить работу.

Директора служат публичным лицом школы, поэтому они должны быть уверены в себе и обладать отличными навыками письменного и устного общения. Они представляют школу на конференциях и других мероприятиях, общаются с коллегами-администраторами и преподавателями, чтобы быть в курсе последних достижений в области образования.

Директора также справляются с чрезвычайными ситуациями и школьными кризисами. Они являются решительными, аналитическими лидерами, способными налаживать отношения с различными группами.

В заключение следует отметить, что школы, директора, учителя, ученики и расписания — все это неотъемлемые части системы образования. Каждый из них играет свою уникальную роль в создании среды, способствующей обучению и развитию.

Таким образом можно выделить следующие объекты предметной области:

* Пользователь - список пользователей, которые могут войти в аккаунт и иметь доступ к сайту. Каждый пользователь имеет свой логин, пароль и роль, в зависимости от возможности;
* Роль - список ролей, которые необходимы для работы пользователям. Каждая роль имеет своё название;
* Учитель - список учителей, которые участвуют в жизнедеятельности школы. Каждый учитель имеет своё имя, фамилию, отчество, данные от пользователя, пол;
* Ученик - список учеников, которые обучаются и проживают жизнь в школе. Каждый ученик имеет своё имя, фамилию, отчество, класс, данные от пользователя, пол;
* Директор - список директоров, которые управляют самой школой. Каждый директор имеет своё имя, фамилию, отчество, статус, данные от пользователя, пол;
* Пол - список полов, которые указывают на различия человека. Каждый пол имеет своё название;
* День недели - список дней недели, которые повторяются по кругу. Каждый день недели имеет своё название и номер недели;
* Расписание - список уроков, предметов и классов в определенный урок и смену. Каждое расписание имеет день недели, номер урока, смена, учителя, предмет, класс, кабинет, домашнее задание;
* Мероприятия - список мероприятий, в которых участвуют учителя. Каждое мероприятие имеет название, учителя, который там участвует, смена, номер урока, дата;
* Предмет - список предметов, которые изучают в школе. Каждый предмет имеет своё название, и количество часов за весь учебный год;
* Родители - список родителей, которые являются законными представителями ребенка. Каждый родитель имеет своё имя, фамилию, отчество, и свой статус (мать, отец);
* Класс - список классов, которые обучаются в школе. Каждый класс имеет своё название и классного руководителя;
* Статус родителя - список статусов родителей для определения того, кто каким родителем является. Каждый статус имеет своё название;
* Статус директора - список статуса директоров, для определения своих обязанностей. Каждый статус директора имеет своё название.

Исходя из всего вышеперечисленного можно составить следующие требования:

* Возможность входа для пользователей;
* Регистрация пользователей, если пользователь имеет роль администратора;
* Добавление учеников если пользователь имеет роль администратора;
* Добавление учителей если пользователь имеет роль администратора;
* Добавление директоров если пользователь имеет роль администратора;
* Вывод расписания на день для учеников;
* Вывод расписания всех уроков для учителей на весь день;
* Вывод полного расписания школы на весь день для директоров;
* Вывод информации о родителях для учеников;
* Вывод информации о мероприятиях для учителей;
* Добавление мероприятий для директоров;
* Добавление домашнего задания для учителей;
* Вывод информации об учениках для директора;
* Возможность выхода из аккаунта;
* Просмотр расписания звонков;
* Экстренный переход на главную страницу для пользователей.

# 2 Выбор средств/методологии проектирования. Выбор СУБД.

Проектирование базы данных – процесс, который необходим для создания схем баз данных и определение нужных ограничений целостности.

Основные задачи проектирования включают в себя:

* Хранение информации в базе данных;
* Получение данных по запросам;
* Сокращение дублирования данных;
* Предотвращение потерь данных.

Поэтому, для создания необходимой базы данных нужно выделить методы и средства в плане проектирования.

Среди различных методов проектирования баз данных можно выделить метод «сущность – связь» (ER-модель), который описывает связь данных и общую их структуру. Данный метод использует сущности, которые являются теми или иными объектами, и отношения между ними, которые указывают на то, как они связаны.

Средства проектирования баз данных моделируют данные и генерируют схемы данных. Для этого используются CASE-средства. CASE-средства – это методы и технологии, которые проектируют необходимые схемы данных для создания по ним уже полноценных информационных систем.

Среди всех средств был выбран diagrams.net. Diagrams.net – это бесплатное приложение, которое используется для создания схем, диаграмм, моделей. Разработчиком данного продукта является JGraph. Интерфейс данного приложения максимально понятен, имеет функции перетаскивания, редактирования, создания различных схем и диаграмм.

Среди плюсов diagrams.net можно выделить следующие:

* Применение репозитория;
* Моделирование схем и диаграмм;
* Наличие API;
* Импорт и экспорт данных.

Если сравнивать с diagrams.net с другими средствами, то можно перечислить ряд недостатков. Visio – продукт от Майкрософт, но даже у него много минусов. Он дорого стоит и очень сложный для новичков, что может сразу оттолкнуть будущих пользователей. Erwin является таким же средством проектирования, но он имеет тоже много пробелов. В ней нет кнопок отмены или повтора, после чего проект нужно создавать заново, нет перевода на русский язык, отсутствует совместимость с базами данных, которые созданы в других программах.

Исходя из всего вышеперечисленного, выбор падает на diagrams.net. Важно также выбрать СУБД, в котором будет происходить само проектирование баз данных.

СУБД – это программа, которая используется для создания, управления, изменения данных. Она имеет интерфейс, с помощью которого пользователь может иметь доступ к различным данным. СУБД необходимо обеспечивать эффективность её использования и безопасность.

Среди возможного множества СУБД была выбрана Microsoft SQL Server. Данный продукт довольно простой для пользователей, обеспечивает скорость и стабильность. Данная СУБД поддерживает запросы и другие языки баз данных. Единственным её недостатком можно считать – высокая цена для юридических лиц, которую очень сложно оплатить для таких пользователей.

Для выбора именно Microsoft SQL Server происходило сравнение других СУБД. Например, PostgreSQL – это бесплатная СУБД, которая также справляется с задачами. К сожалению, для её использования нужно изучение, но документации очень туманны, скорость при выполнении запросов очень сильно падает. MariaDB – это тоже бесплатная СУБД, которая на данный момент быстро развивается, но она очень нестабильная и за её поддержку необходимо платить за поддержку.

Поэтому, среди всех систем управлением баз данных была выбрана Microsoft SQL Server. Она пригодится для работы в будущем на следующих этапах.

# 3 Построение инфологической (концептуальной) модели предметной области

В результате анализа требований пользователя и структур данных, описывающих деятельность школы, следует определить, как сущности следующие объекты:

* Пол (Код пола – потенциальный ключ, Название пола – потенциальный ключ);
* День недели (Код дня недели – потенциальный ключ, Название дня недели – потенциальный ключ);
* Статус (Код статуса – потенциальный ключ, Название статуса – потенциальный ключ);
* Роль (Код роли – потенциальный ключ, Название роли – потенциальный ключ);
* Статус родителей (Код статуса родителей – потенциальный ключ, Название статуса родителей – потенциальный ключ);
* Класс (Код класса – потенциальный ключ, Название класса – потенциальный ключ, Классный руководитель);
* Предмет (Код предмета – потенциальный ключ, Название предмета – потенциальный ключ, Норматив);
* Пользователь (Код пользователя – потенциальный ключ, Логин– потенциальный ключ, Пароль, Роль);
* Родитель (Код родителя – потенциальный ключ, Имя – потенциальный ключ, Фамилия – потенциальный ключ, Отчество – потенциальный ключ, Статус);
* Мероприятие (Код мероприятия – потенциальный ключ, Название мероприятия – потенциальный ключ, Номер урока, Смена, Учитель, Дата);
* Расписание (Код расписания – потенциальный ключ, Номер урока, Смена, Учитель, Предмет, Класс, Кабинет, Домашнее задание);
* Учитель (Код учителя – потенциальный ключ, Имя – потенциальный ключ, Фамилия – потенциальный ключ, Отчество – потенциальный ключ, Пользователь, Дата рождения, Пол);
* Ученик (Код ученика – потенциальный ключ, Имя – потенциальный ключ, Фамилия – потенциальный ключ, Отчество – потенциальный ключ, Пользователь, Дата рождения, Пол, Класс);
* Директор (Код директора – потенциальный ключ, Имя – потенциальный ключ, Фамилия – потенциальный ключ, Отчество – потенциальный ключ, Пользователь, Дата рождения, Статус).

Сущности могут взаимодействовать друг с другом, поэтому их связи будут следующими:

* Имеет (Родитель статус) - показывает, какой статус имеет каждый родитель в базе данных. Бинарная связь - один ко многим. Так как один статус, может быть, у нескольких родителей, а у родителя может быть только один статус. Класс обоих сущностей обязателен, так как статус по закону обязателен для каждого родителя, а родитель в свою очередь обязан быть со статусом. (см. Рисунок 3.1);

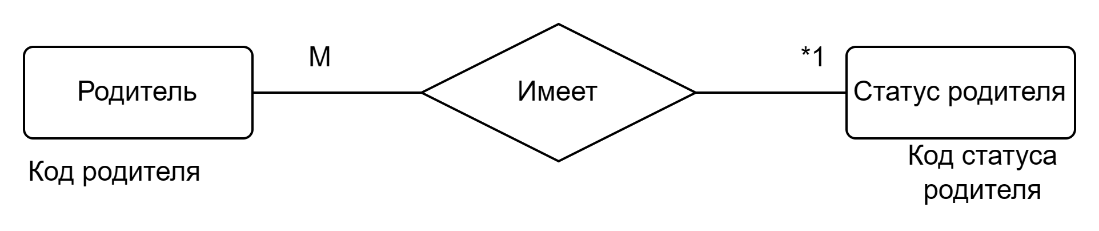


Рисунок 3.1 — Связь между сущностями «Родитель» и «Статус родителя»

* Участвует (Учитель в мероприятии) - показывает, какой учитель участвует в определенном мероприятии. Бинарная связь - один ко многим. Так как один учитель, может быть, в нескольких мероприятиях, а у мероприятия может быть только один учитель. Класс сущностей необязателен, как так не каждый учитель может принимать участие. (см. Рисунок 3.2);

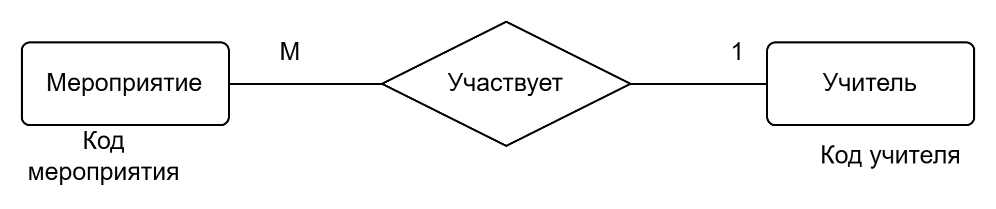


Рисунок 3.2 — Связь между сущностями «Учитель» и «Мероприятие»

* Руководит (Учитель классом) - показывает, какой преподаватель руководит каким классом. Бинарная связь - один к одному. Так как у каждого класса только один руководитель и у каждого руководителя только один класс, которым он руководит. “Класс” обязателен, так как он обязан иметь класс, а “Учитель” необязателен, так как не у всех преподавателей есть свой класс (см. Рисунок 3.3);



Рисунок 3.3 — Связь между сущностями «Учитель» и «Класс»

* Является (Ученик определенного пола) - показывает, какой пол присущ каждому ученику. Бинарная связь - один ко многим. Так как один пол, может быть, у нескольких учеников, а у ученика только один пол. Класс обоих сущностей обязателен, так как пол обязателен для указания в документах, а ученик обязан иметь пол. (см. Рисунок 3.4);



Рисунок 3.4 — Связь между сущностями «Ученик» и «Пол»

* Является (Учитель определенного пола) - показывает, какой пол присущ каждому учителю. Бинарная связь - один ко многим. Так как один пол, может быть, у нескольких учителей, а у учителя только один пол. Класс обоих сущностей обязателен, так как пол обязателен для указания в документах, а учитель обязан иметь пол. (см. Рисунок 3.5);



Рисунок 3.5 — Связь между сущностями «Учитель» и «Пол»

* Является (Директор определенного пола) - показывает, какой пол присущ каждому директору. Бинарная связь - один ко многим. Так как один пол, может быть, у нескольких директоров, а у директора только один пол. Класс обоих сущностей обязателен, так как пол обязателен для указания в документах, а директор обязан иметь пол (см. Рисунок 3.6);



Рисунок 3.6 — Связь между сущностями «Директор» и «Пол»

* Является (Пользователь учеником) - показывает, какой пользователь является определенным учеником. Бинарная связь - один к одному. Так как у каждого пользователя только один ученик, закрепленный к нему, а у ученика только один пользователь, через который он может войти. Класс обоих сущностей обязателен, так как каждому ученику необходим свой пользователь, а каждому пользователю необходим свой ученик. (см. Рисунок 3.7);



Рисунок 3.7 — Связь между сущностями «Ученик» и «Пользователь»

* Является (Пользователь учителем) - показывает, какой пользователь является определенным учителем. Бинарная связь - один к одному. Так как у каждого пользователя только один учитель, закрепленный к нему, а у учителя только один пользователь, через который он может войти. Класс обоих сущностей обязателен, так как каждому учитель необходим свой пользователь, а каждому пользователю необходим свой учитель (см. Рисунок 3.8);



Рисунок 3.8 — Связь между сущностями «Учитель» и «Пользователь»

* Является (Пользователь директором) - показывает, какой пользователь является определенным директором. Бинарная связь - один к одному. Так как у каждого пользователя только один директор, закрепленный к нему, а у директора только один пользователь, через который он может войти. Класс обоих сущностей обязателен, так как каждому директору необходим свой пользователь, а каждому пользователю необходим свой директор (см. Рисунок 3.9);



Рисунок 3.9 — Связь между сущностями «Директор» и «Пользователь»

* Имеет (Ученик класс) - показывает, какой класс имеет ученик. Бинарная связь - один ко многим. Так как один класс, может быть, у нескольких учеников, а у ученика только один класс. Класс обоих сущностей обязателен, так как класс обязан иметь учеников, а ученик обязан быть в классе (см. Рисунок 3.10);

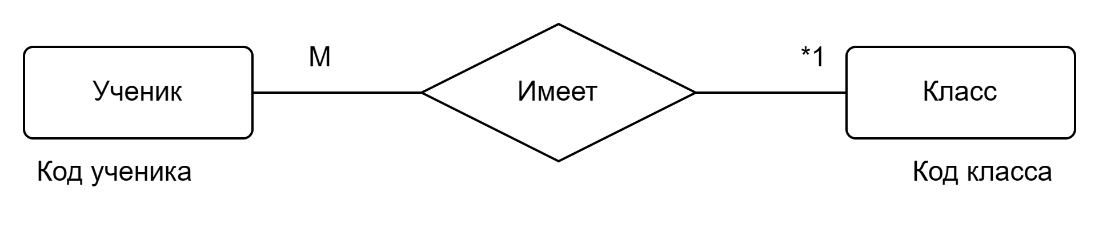


Рисунок 3.10 — Связь между сущностями «Ученик» и «Класс»

* Является (Директор определенного статуса) - показывает, какой статус присущ каждому директору. Бинарная связь - один ко многим. Так как один статус, может быть, у нескольких директоров, а у директора только один статус. Класс обоих сущностей обязателен, так как статус обязателен для указания в договорах, а директор обязан иметь свой статус (Рисунок 3.11);

Изображение выглядит как текст, Шрифт, снимок экрана, логотип

Автоматически созданное описание

Рисунок 3.11 – Связь между сущностями «Директор» и «Статус»

* Дана (Роль пользователю) - показывает, какая роль дана каждому пользователю. Бинарная связь - один ко многим. Так как одна роль, может быть, у нескольких пользователей, а у пользователя только одна роль. Класс обоих сущностей обязателен, так как роль обязательна для работы с программой, а пользователь обязан иметь свою роль (Рисунок 3.12);

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описание

Рисунок 3.12 – Связь между сущностями «Пользователь» и «Роль»

* Указывает (Расписание на день недели) - показывает, какое расписание будет для каждого дня недели. Бинарная связь - один ко многим. Так как один день недели, может быть, у нескольких расписаний, а у у расписания только один день. Класс обоих сущностей обязателен, так как расписание устроено для каждого дня (Рисунок 3.13);

Изображение выглядит как текст, Шрифт, логотип, снимок экрана

Автоматически созданное описание

Рисунок 3.13 – Связь между сущностями «Расписание» и «День недели»

* Обучается (Класс в расписании) - показывает, какой класс учится в расписании. Бинарная связь - один ко многим. Так как один класс, может быть, у нескольких расписаний, а у расписания только класс. Класс обоих сущностей обязателен, так как класс обязателен для указания в расписании, а расписание обязано иметь класс в своей информации (Рисунок 3.14);

Изображение выглядит как текст, Шрифт, снимок экрана, логотип

Автоматически созданное описание

Рисунок 3.14 – Связь между сущностями «Расписание» и «Класс»

* Являются родственниками (Ученик родителю) - показывает, какие ученики являются друг другу детьми родителям. Бинарная связь - многие ко многим. Так как один и тот же ученик может иметь несколько родителей, а родитель может иметь сразу несколько учеников. Класс обоих сущностей обязателен, так как каждый ученик обязан иметь родителей, а родитель обязан иметь своих детей (Рисунок 3.15);

Изображение выглядит как текст, Шрифт, белый, снимок экрана

Автоматически созданное описание

Рисунок 3.15 - Связь между сущностями «Родитель» и «Ученик»

* Проводит (Учитель предмет) - показывает, какие предметы проводятся учителями. Бинарная связь - многие ко многим. Так как один и тот же предмет может вести несколько учителей, а у того же учителя могут быть сразу несколько предметов, которые они проводят. Класс обоих сущностей обязателен, так как каждый предмет обязан проводиться учителем, а учитель обязан проводить свои предметы. Проводится расписанием - показывает, какой предмет и учитель будут проводиться в расписании. Бинарная связь - один ко многим. Так как один учитель с предметом могут проводиться несколько раз за расписание, а у расписания только одно такое указание. Класс обоих сущностей обязателен, так как каждый предмет должен проводиться учителем в расписании, а каждое расписание должно указывать наличие возможности проведения этого урока определенным учителем (Рисунок 3.16).

Изображение выглядит как текст, снимок экрана, Шрифт, Самоклеющийся листок

Автоматически созданное описание

Рисунок 3.16 - Связь между сущностями «Расписание», «Учитель» и «Предмет»

Исходя из вышесказанного можно построить следующую ER-диаграмму (см. Рисунок 3.17):

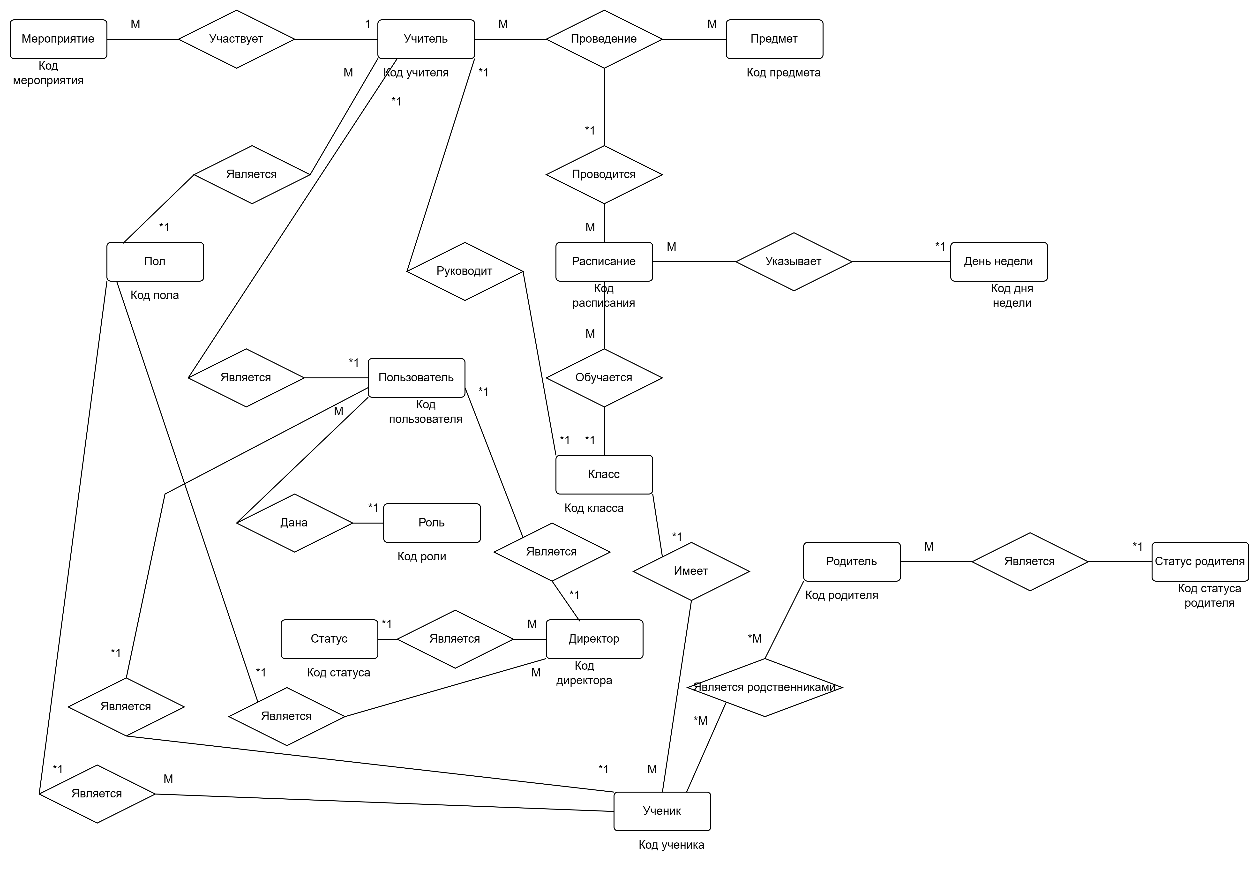


Рисунок 3.17 — Общая ER-диаграмма

# 4 Проектирование логической структуры БД

Логическая структура баз данных необходима для определения того, какая будет структура её таблиц, которые между собой связаны. Для её проектирования необходимо составить таблицы, которые включают в себя атрибуты и тип данных этого самого атрибута.

Можно выделить следующие таблицы:

Логическая модель сущности «Пол» (см. Рисунок 4.1) состоит из следующих атрибутов:

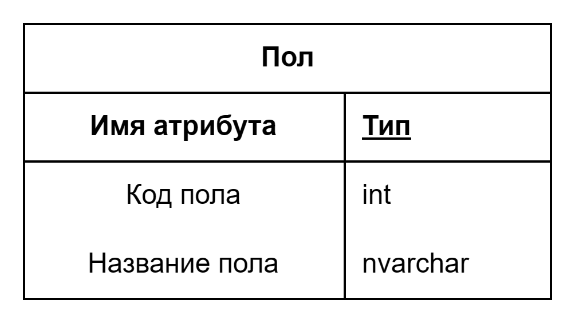


Рисунок 4.1 — Атрибуты сущности «Пол»

Логическая модель сущности «День недели» (см. Рисунок 4.2) состоит из следующих атрибутов:



Рисунок 4.2 — Атрибуты сущности «День недели»

Логическая модель сущности «Статус» (см. Рисунок 4.3) состоит из следующих атрибутов:

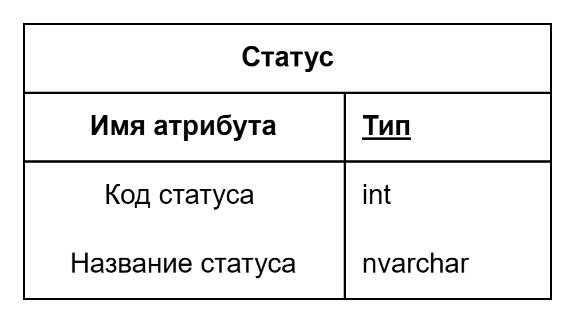


Рисунок 4.3 — Атрибуты сущности «Статус»

Логическая модель сущности «Роль» (см. Рисунок 4.4) состоит из следующих атрибутов:

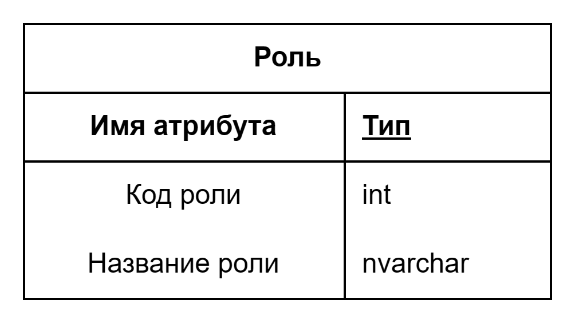


Рисунок 4.4 — Атрибуты сущности «Роль»

Логическая модель сущности «Статус родителей» (см. Рисунок 4.5) состоит из следующих атрибутов:



Рисунок 4.5 — Атрибуты сущности «Статус родителей»

Логическая модель сущности «Класс» (см. Рисунок 4.6) состоит из следующих атрибутов:



Рисунок 4.6 — Атрибуты сущности «Класс»

Логическая модель сущности «Предмет» (см. Рисунок 4.7) состоит из следующих атрибутов:



Рисунок 4.7 — Атрибуты сущности «Предмет»

Логическая модель сущности «Пользователь» (см. Рисунок 4.8) состоит из следующих атрибутов:



Рисунок 4.8 — Атрибуты сущности «Пользователь»

Логическая модель сущности «Родитель» (см. Рисунок 4.9) состоит из следующих атрибутов:

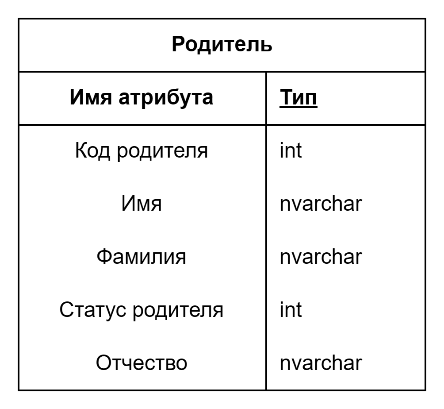


Рисунок 4.9 — Атрибуты сущности «Родитель»

Логическая модель слабой сущности «Учитель\_Предмет» (см. Рисунок 4.10) состоит из следующих атрибутов:

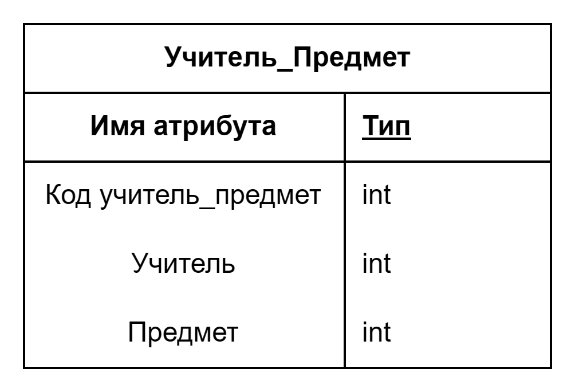


Рисунок 4.10 — Атрибуты связи «Учитель\_Предмет»

Логическая модель слабой сущности «Ученик\_Родитель» (см. Рисунок 4.11) состоит из следующих атрибутов:

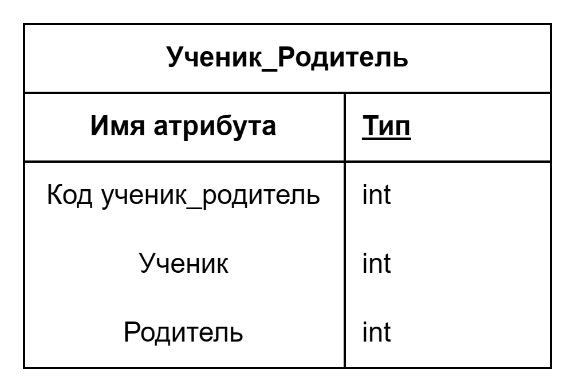


Рисунок 4.11 — Атрибуты связи «Ученик\_Родитель»

Логическая модель сущности «Мероприятие» (см. Рисунок 4.12) состоит из следующих атрибутов:

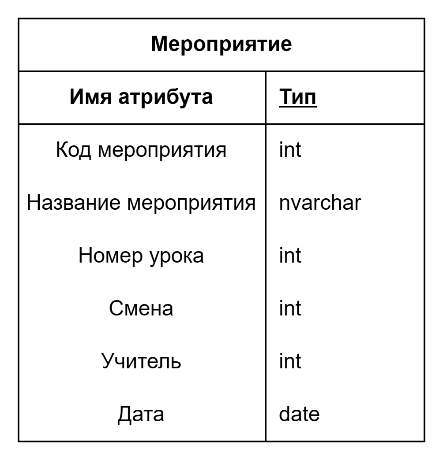


Рисунок 4.12 — Атрибуты сущности «Мероприятие»

Логическая модель сущности «Расписание» (см. Рисунок 4.13) состоит из следующих атрибутов:

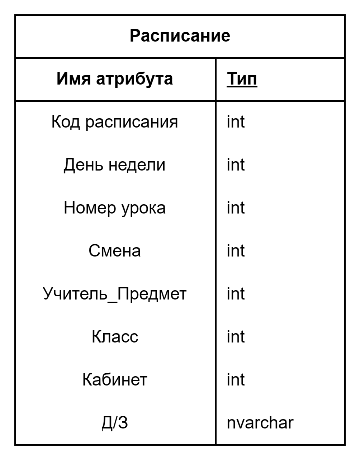


Рисунок 4.13 — Атрибуты сущности «Расписание»

Логическая модель сущности «Учитель» (см. Рисунок 4.14) состоит из следующих атрибутов:

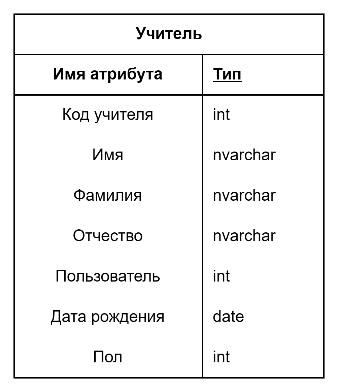


Рисунок 4.14 — Атрибуты сущности «Учитель»

Логическая модель сущности «Ученик» (см. Рисунок 4.15) состоит из следующих атрибутов:

Изображение выглядит как снимок экрана, черный, текст

Автоматически созданное описание

Рисунок 4.15 – Атрибуты сущности «Ученик»

Логическая модель сущности «Директор» (см. Рисунок 4.16) состоит из следующих атрибутов:

Изображение выглядит как снимок экрана, текст, черный, Шрифт

Автоматически созданное описание

Рисунок 4.16 - Атрибуты сущности «Директор»

Теперь можно переходить к формулировкам ограниченной целостности и составлению физической структуры базы данных.

# 5 Формулировка ограниченной целостности

**5.1 Выявление полного перечня ограниченной целостности, присущего данной предметной области**

При изучении предметной области были выявлены различные ограничения в ней для полноценной работы системы.

У каждого лица в базе данных должны быть свои личные данные, которые не повторяются.

Ученик имеет доступ только для просмотра информации о расписании или о родителях.

Только учитель с правами администратора может регистрировать новых пользователей.

При добавлении мероприятий не должно быть так, что учитель должен находиться сразу на двух мероприятиях одновременно в одно время и дату.

Нельзя указывать в датах мероприятие, которое меньше сегодняшней. Только больше сегодня или сегодняшняя дата.

Учитель не может два класса одновременно в один и тот же урок в то же время.

**5.2 Определение перечня ограничений целостности, которые будут контролироваться в данном проекте**

При проектировании базы данных будут использоваться определенные ограничения для сохранения целостности данных:

* У каждого лица в базе данных должны быть свои личные данные, которые не повторяются;
* Ученик имеет доступ только для просмотра информации о расписании или о родителях;
* Только учитель с правами администратора может регистрировать новых пользователей;
* При добавлении мероприятий не должно быть так, что учитель должен находиться сразу на двух мероприятиях одновременно в одно время и дату;
* Нельзя указывать в датах мероприятие, которое меньше сегодняшней. Только больше сегодня или сегодняшняя дата;
* Учитель не может два класса одновременно в один и тот же урок в то же время.

**5.3 Выбор способа реализации контроля целостности для каждого из ограничений**

Для того чтобы уследить за всеми ограничениями, нужно выполнить следующее:

* Использовать запросы для проверки тех или иных сущностей на существование. Если они есть, то происходит предупреждения, что нужно менять данные на другие;
* Составлять те списки сущностей с помощью запросов, которые не были нигде использованы;
* Проверять вошедшего пользователя, является ли он администратором;
* С помощью логики узнавать, кем является вышедший пользователь (ученик, учитель, студент);
* Использовать специальные элементы для выбора того или иного числа в месяце и году для выбора конкретной даты.

# 6 Проектирование физической структуры базы данных

Перед проектированием физической структуры базы данных необходимо разобраться с методами доступа к файлам. Можно выделить следующие:

* Последовательный (Выполняется последовательный просмотр всех записей таблицы и поиск нужных из них. Этот метод доступа является крайне неэффективным и приводит к значительным временным затратам на поиск, прямо пропорциональным размеру таблицы. Поэтому его рекомендуется использовать только для относительно небольших таблиц);
* Прямой (Нужная запись выбирается в таблице на основании ключа или индекса. При этом просмотр других записей не выполняется. При поиске записи выполняется не последовательный просмотр всей таблицы, а непосредственный доступ к записи на основании ссылки);
* Индексно-последовательный (Метод доступа включает в себя элементы последовательного и прямого методов доступа и используется при поиске группы записей. Этот метод реализуется только при наличии индекса, построенного по полям, значения которых должны быть найдены);
* Навигационный (Обработка каждой отдельной записи таблицы. Этот способ обычно используется в локальных БД или в удаленных БД небольшого размера. Если необходимо обработать несколько записей, то все они обрабатываются поочередно);
* Реляционный (Обработка сразу группы записей, при этом, если необходимо обработать одну запись, то обрабатывается группа, состоящая из одной записи).

Будет использоваться последовательный метод доступа к файлам, так как он предполагает хранение физических записей в логической последовательности в небольших объемах данных.

Для составления физической структуры базы данных использовались вышеперечисленные модели, структуры и ограничения, из которых по итогу и состоит данная структура.

В результате проектирования физической структуры базы данных получилась следующее (см. Рисунок 6.1):

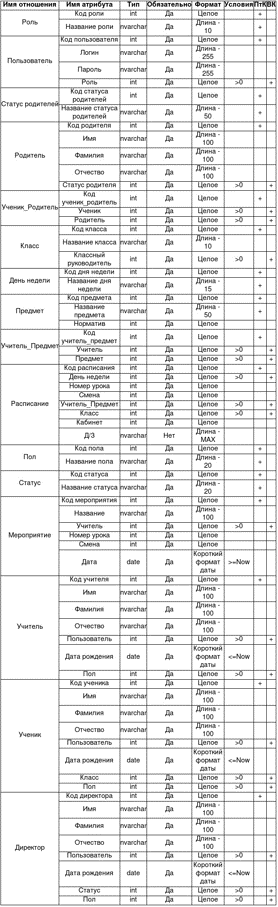


Рисунок 6.1 — Физическая структура БД

# 7 Организация ввода данных в БД

При работе с базой данных были разработаны специальные представления для добавления данных, которые добавляются непосредственно в саму БД.

На Рисунке 7.1 экранная форма «Регистрация пользователя» выглядит следующим образом:



Рисунок 7.1 – Экранная форма «Регистрация пользователя»

На Рисунке 7.2 экранная форма уже с введенными данными.

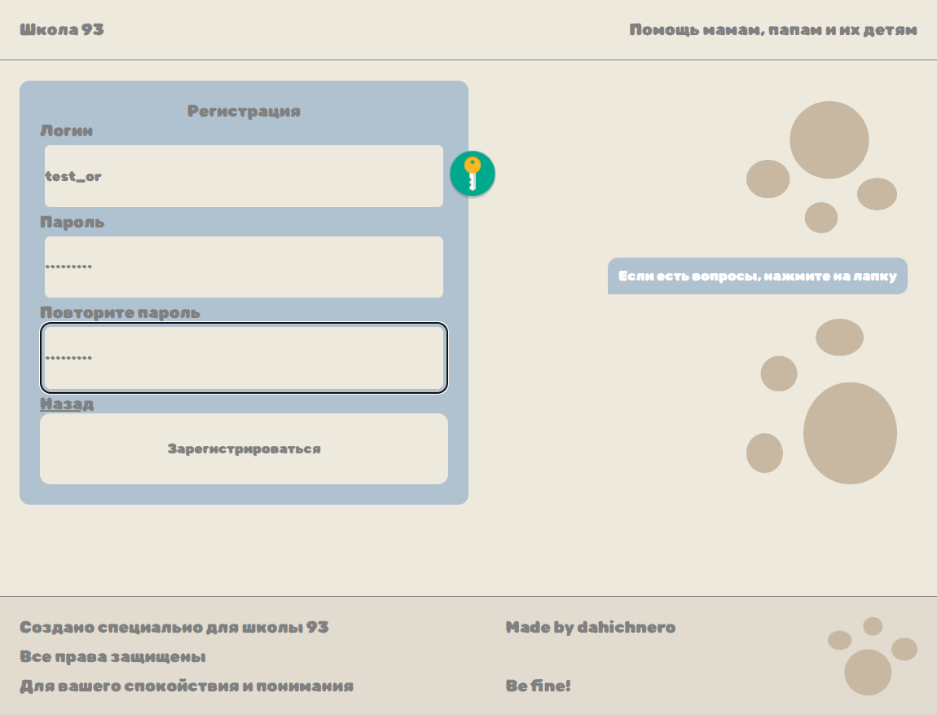


Рисунок 7.2 – Экранная форма «Регистрация пользователя» с введенными данными

На рисунке 7.3 показано, что добавленный пользователь появится в БД:



Рисунок 7.3 – Добавленный пользователь в БД

Другие представления диалога приведены в Приложении В.

# 8 Организация корректировки БД

Для корректировки данных были созданы формы, которые изменяют данные касательно выбранного учителя.

На Рисунке 8.1 выбранного учителя поменяем Имя (Ранее Милана):



Рисунок 8.1 – Изменение имени у выбранного учителя

На Рисунке 8.2 видно, что имя у учителя поменялось.

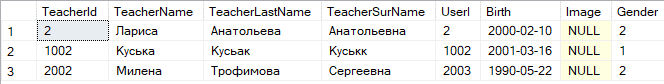


Рисунок 8.2 – Изменение имени учителя в БД

Другие представления для реализации корректировки данных приведены в Приложении В.

# 9 Реализация запросов, получение отчётов

Исходя из вышеперечисленных требований в разделе «Описание предметной области» необходима их реализация в качестве запросов и получения отчетов их выполнения.

На Рисунке 9.1 происходит запрос о выводе информации о родителях вошедшего ученика (ссылка «Родители»):



Рисунок 9.1 – Запрос о выводе информации о родителях вошедшего ученика

На рисунке 9.2 предоставлен отчет о выполненном запросе.

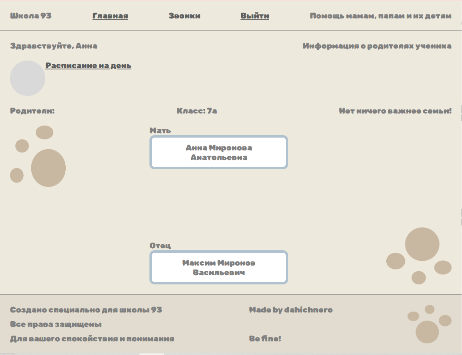


Рисунок 9.2 - Отчет по запросу о выводе информации о родителях вошедшего ученика

Другая часть запросов приведена в Приложении Д, а составленные отчёты по ним приведены в Приложении Г.

# 10 Разработка интерфейса

В разработанном интерфейсе был реализован весь функционал для работы. При входе в программу появляется страница входа, где администратор может ввести свои данные для входа в систему, или же ученик, учитель или директор без прав админа может войти (см. Рисунок 10.1).



Рисунок 10.1 -Вход в программу

При входе как клиент, происходит переход на страницу, где содержится весь доступный функционал для работы с сайтом. Для данного пользователя доступ к редактированию разрешен. Данные функции может осуществлять только администратор (см. Рисунок 10.2).



Рисунок 10.2 - Главное окно при входе как учитель

Другие элементы разработанного интерфейса приведены в Приложении В.

# 11 Реализация проекта в среде конкретной СУБД.

На Рисунке 11.1 при проектировании проекта в SQL Server Management Studio была составлена следующая диаграмма:

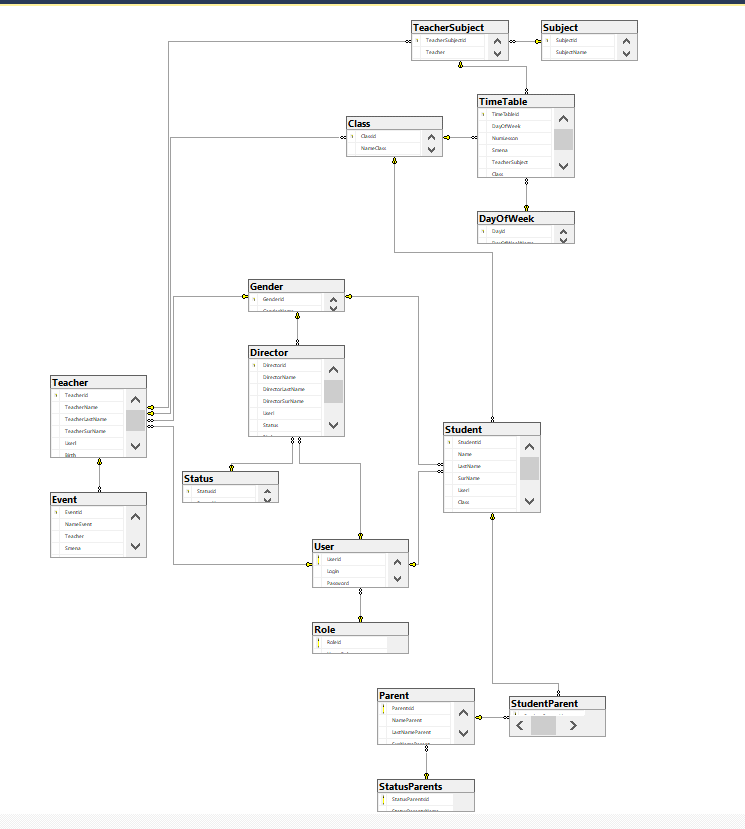


Рисунок 11.1 - Диаграмма базы данных

При проверке работоспособности БД составлялись определенные запросы. На рисунке 11.2 изображена структура запроса по добавлению ученика:

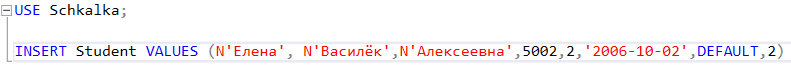


Рисунок 11.2 - Структура запроса для нового ученика

На Рисунке 11.3 изображен результат запроса.



Рисунок 11.3 - Результат запроса по добавлению ученика

На Рисунке 11.4 изображен запрос по добавлению учителя:



Рисунок 11.4 - Запрос по добавлению учителя

На Рисунке 11.5 изображен результат запроса.

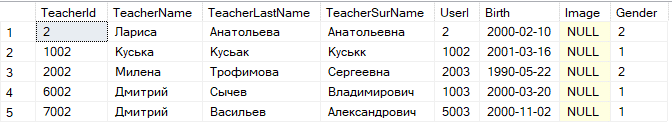


Рисунок 11.5 - Результат запроса по добавлению учителя

На Рисунке 11.6 изображен запрос по добавлению директора:



Рисунок 11.6 - Запрос по добавлению директора

На Рисунке 11.7 изображен результат запроса.

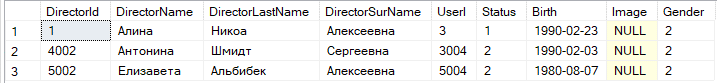


Рисунок 11.7 - Результат запроса по добавлению директора

На Рисунке 11.8 изображен запрос по добавлению мероприятия:

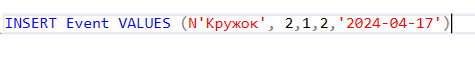


Рисунок 11.8 - Запрос по добавлению мероприятия

На Рисунке 11.9 изображен результат запроса.

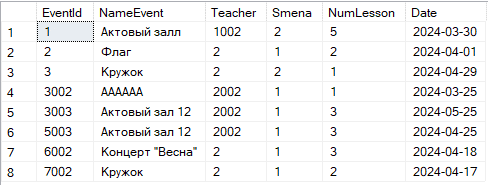


Рисунок 11.9 - Результат запроса по добавлению мероприятия

На Рисунке 11.10 изображен запрос по добавлению домашнего задания:



Рисунок 11.10 - Запрос по добавлению домашнего задания

На Рисунке 11.11 изображен результат запроса.

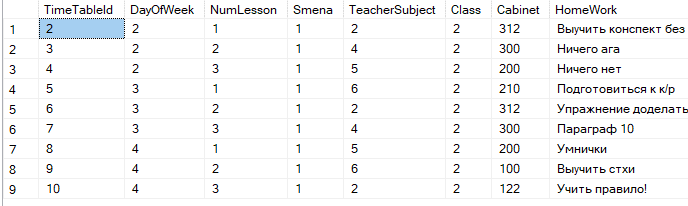


Рисунок 11.11 - Результат запроса по добавлению домашнего задания

# Условные сокращения

SQL (Structured Query Language) – это язык запросов, который применяют, чтобы работать с базами данных, структурированных особым образом.

БД (База данных) – это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе.

СУБД (Система управления базами данных) — это программное обеспечение, предназначенное для создания, управления, обновления и анализа баз данных.

API (Application Programming Interface) – это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными.

SSMS (SQL Server Management Studio) - это интегрированная среда для управления любой инфраструктурой SQL, от SQL Server до баз данных SQL Azure.

ER-модель (Entity-Relationship model) - модель данных, позволяющая описывать концептуальные схемы предметной области.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения проекта была создана БД, которая предназначена для работы со школой. Она имеет ряд удобств:

* Экономит количество затрачиваемого времени на изменение и добавление данных;
* Имеет возможность в помощи нахождения той или иной домашней работы;
* Может помочь учителям спокойно посещать мероприятия без их отсутствия;
* Проста при использовании и понимании.

Составленные формы для добавления и редактирования данных спокойно оповещают о некорректно введенных данных и не дают сохранить изменения, если хоть одно из поле ввода введено неверно. Разработанный ранее интерфейс позволяет спокойно работать с данными базы данных без использования самой СУБД. Он разработан без лишних и ярких деталей, максимально дружелюбен с пользователями.

Использование школьной базы данных поможет ученикам, учителям и самим директорам в работе с внутренней системой школы. Ученики в свою очередь будут видеть, какое домашнее задание задали, а учителя видеть, к каким предметам на завтра им готовиться и какой класс им нужно будет вести.

На будущее этой БД можно спрогнозировать следующее:

* Возможность составлять расписание без прямого использования базы данных;
* Возможность добавления фотографии в профиль пользователя.

# Список литературы

* Школа как образовательная система: современный взгляд [Электронный ресурс]. – Режим доступа: https://science-education.ru/ru/article/view?id=13167&ysclid=luwhrs1yf0878763420 (дата обращения: 12.03.2024);
* Управление современной школой: какой должна быть команда [Электронный ресурс]. – Режим доступа: https://rosuchebnik.ru/material/upravlenie-shkoloy/ (дата обращения: 12.03.2024);
* Школа - Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%A8%D0%BA%D0%BE%D0%BB%D0%B0 (дата обращения: 12.03.2024);
* Базы данных и начало работы с SQL / Хабр [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/companies/first/articles/755832/> (дата обращения: 12.03.2024);
* Что такое база данных | Oracle [Электронный ресурс]. – Режим доступа: <https://www.oracle.com/cis/database/what-is-database/> (дата обращения: 12.03.2024);

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Основные таблицы БД

Таблица «Пользователь»

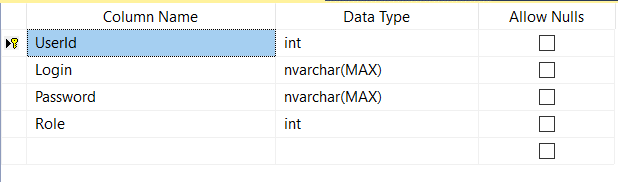


Таблица «Класс»

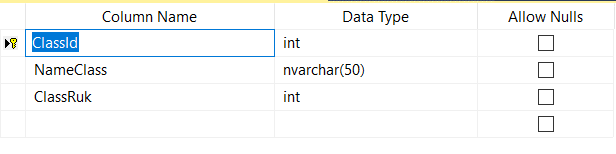


Таблица «День недели»

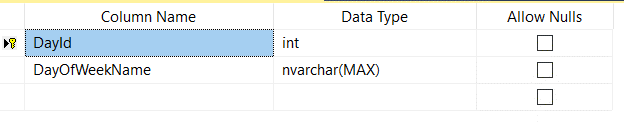


Таблица «Директор»

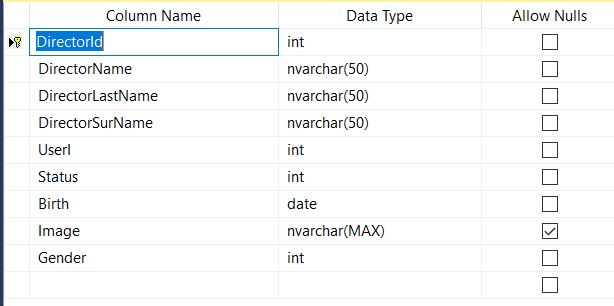


Таблица «Мероприятие»

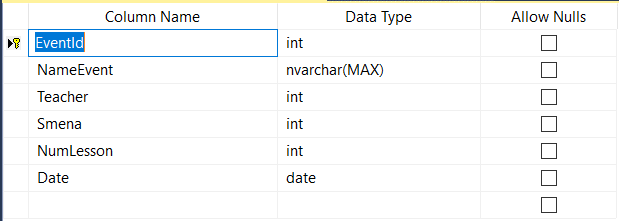


Таблица «Пол»

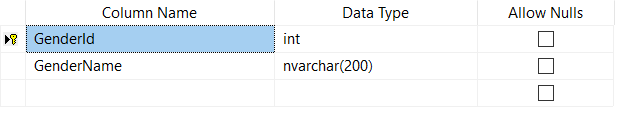


Таблица «Родитель»

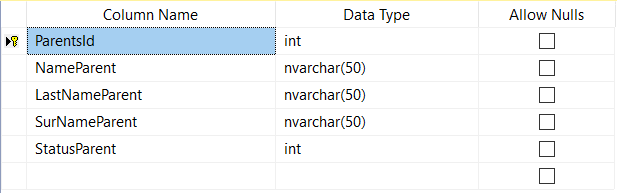


Таблица «Роль»

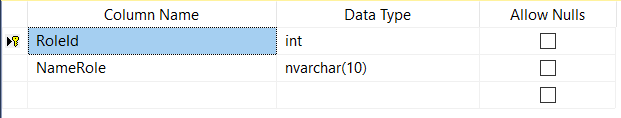


Таблица «Статус»

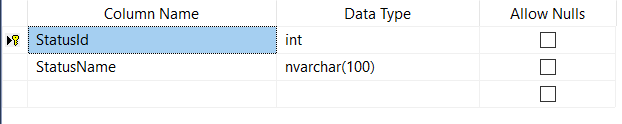


Таблица «Статус родителя»

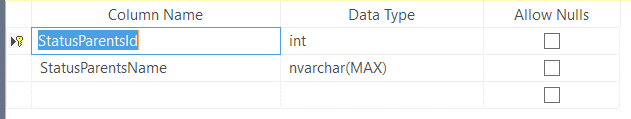


Таблица «Ученик»

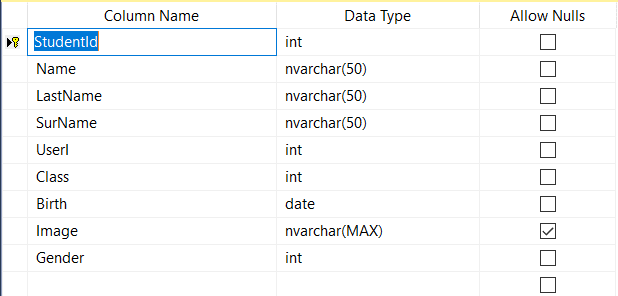


Таблица «Ученик\_Родитель»

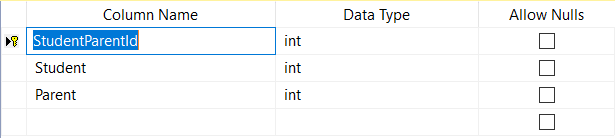


Таблица «Предмет»

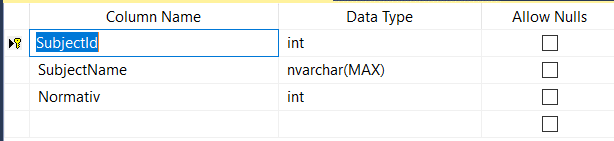


Таблица «Учитель»

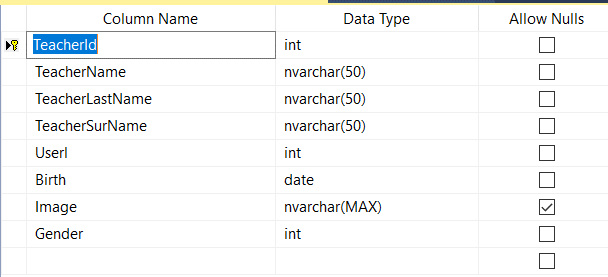


Таблица «Учитель\_Предмет»

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Таблица «Расписание»

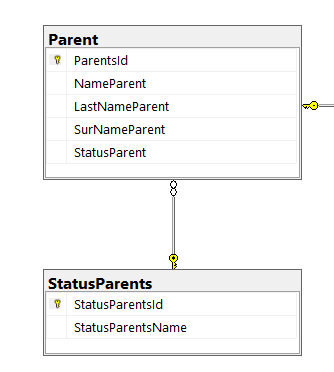
Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

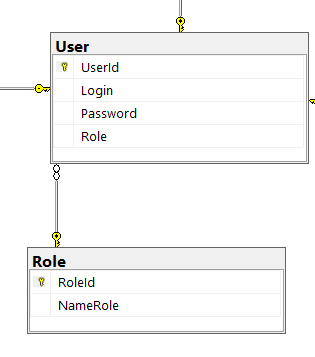
ПРИЛОЖЕНИЕ Б

Схема межтабличных связей

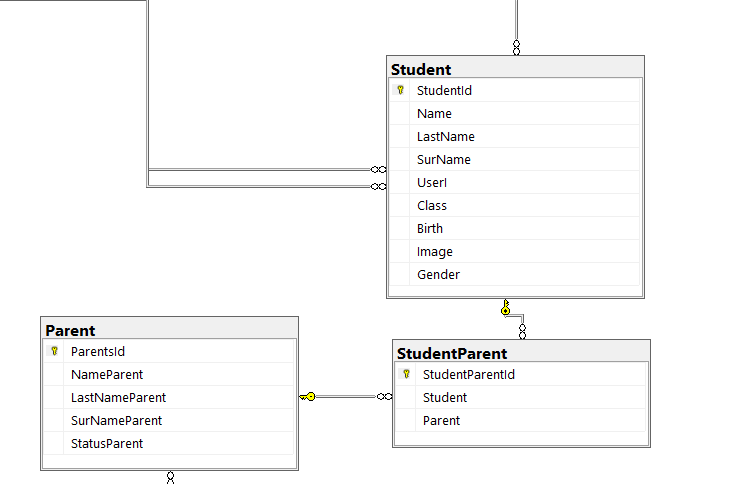
Связь между таблицами «Родитель» и «Статус родителя» (один ко многим)



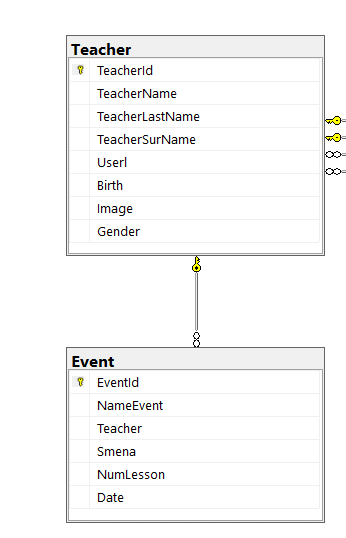
Связь между таблицами «Роль» и «Пользователь» (один ко многим)



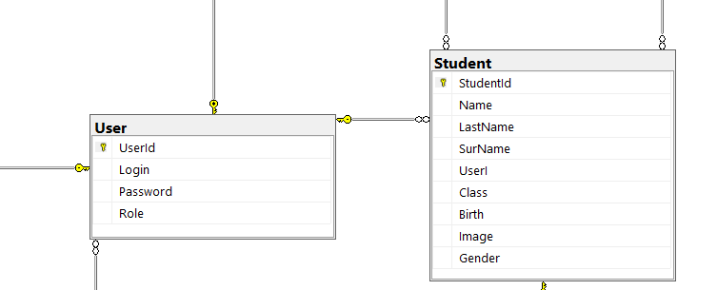
Связь между таблицами «Родитель» и «Ученик» (многие ко многим)



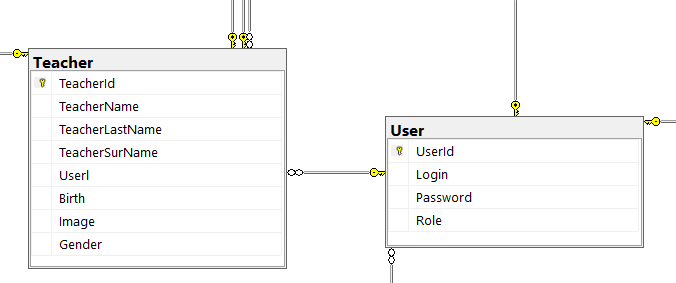
Связь между таблицами «Учитель» и «Мероприятие» (один ко многим)



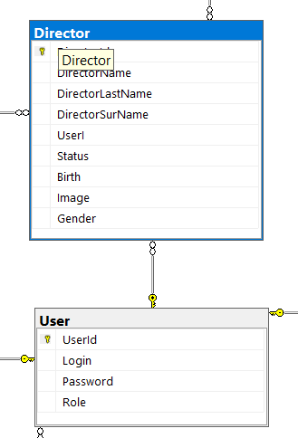
Связь между таблицами «Пользователь» и «Ученик» (один ко одному)



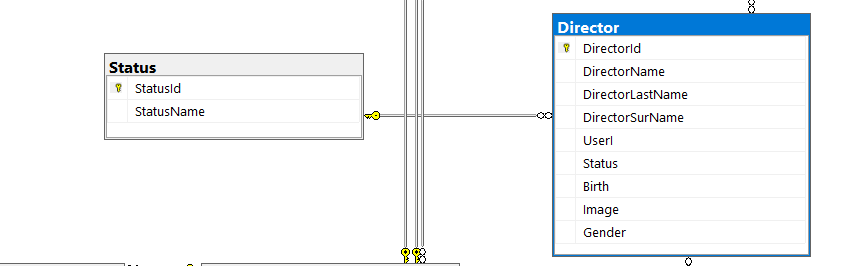
Связь между таблицами «Пользователь» и «Учитель» (один ко одному)



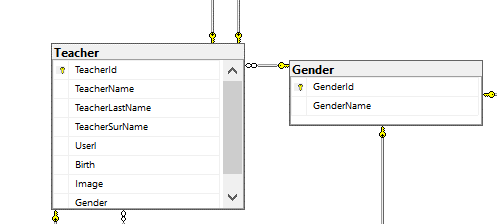
Связь между таблицами «Пользователь» и «Директор» (один ко одному)



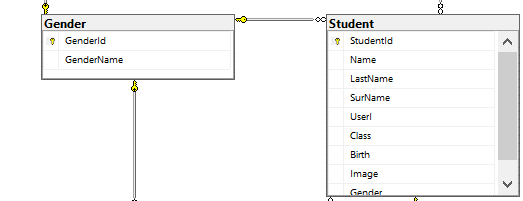
Связь между таблицами «Статус» и «Директор» (один ко многим)



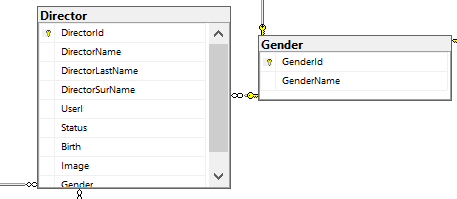
Связь между таблицами «Учитель» и «Пол» (один ко многим)



Связь между таблицами «Ученик» и «Пол» (один ко многим)



Связь между таблицами «Директор» и «Пол» (один ко многим)



Связь между таблицами «Учитель» и «Класс» (один ко многим)

Изображение выглядит как текст, снимок экрана, линия, диаграмма

Автоматически созданное описание

Связь между таблицами «Класс» и «Расписание» (один ко многим) и «День недели» и «Расписание» (один ко многим)

Изображение выглядит как текст, снимок экрана, число, диаграмма

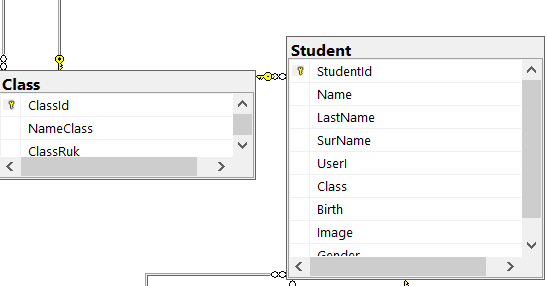
Автоматически созданное описание

Связь между таблицами «Учитель» и «Предмет» (многие ко многим) и «Расписание» (один ко многим)

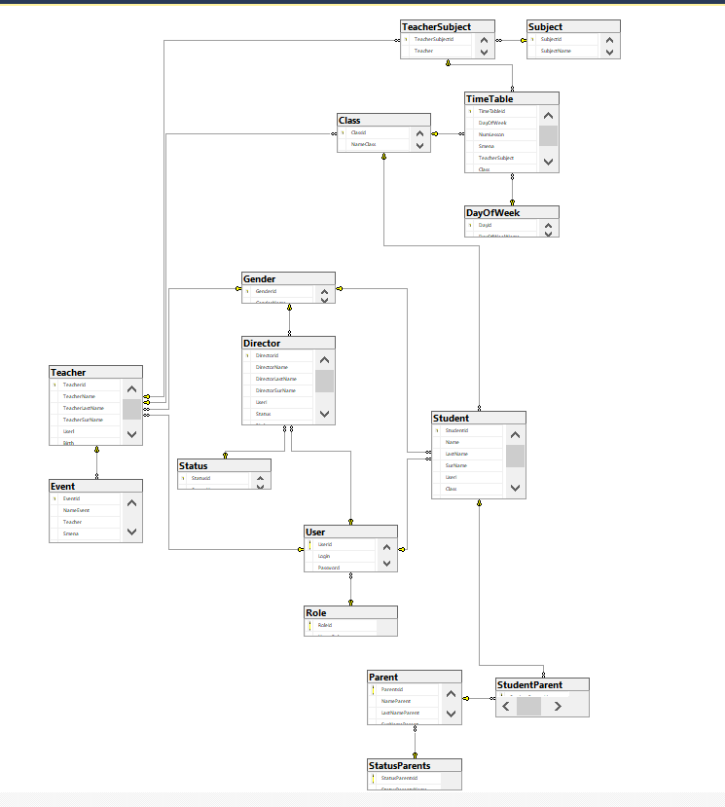
Изображение выглядит как текст, снимок экрана, число, диаграмма

Автоматически созданное описание

Связь между сущностями «Класс» и «Ученик» (один ко многим)



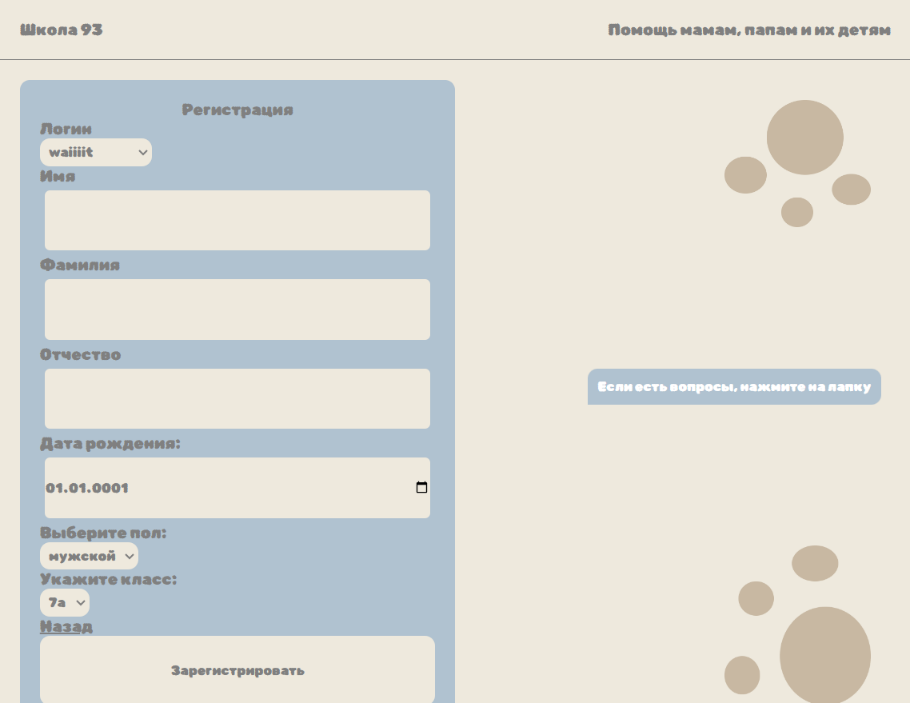
Общая связь между всеми таблицами



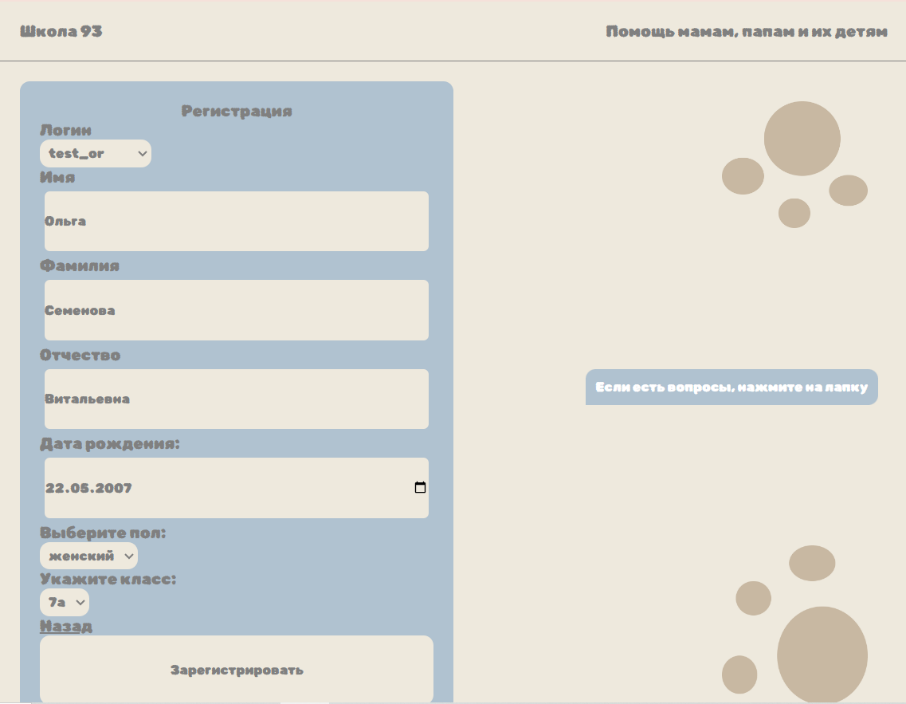
ПРИЛОЖЕНИЕ В

Интерфейсные формы диалога пользователей с БД

Экранная форма «Добавление ученика»



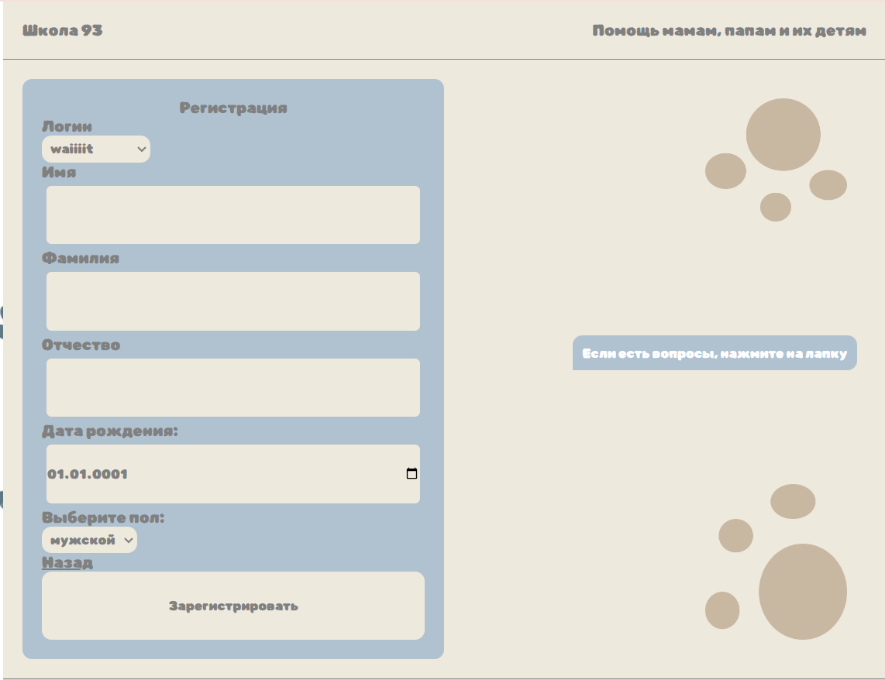
Экранная форма уже с введенными данными



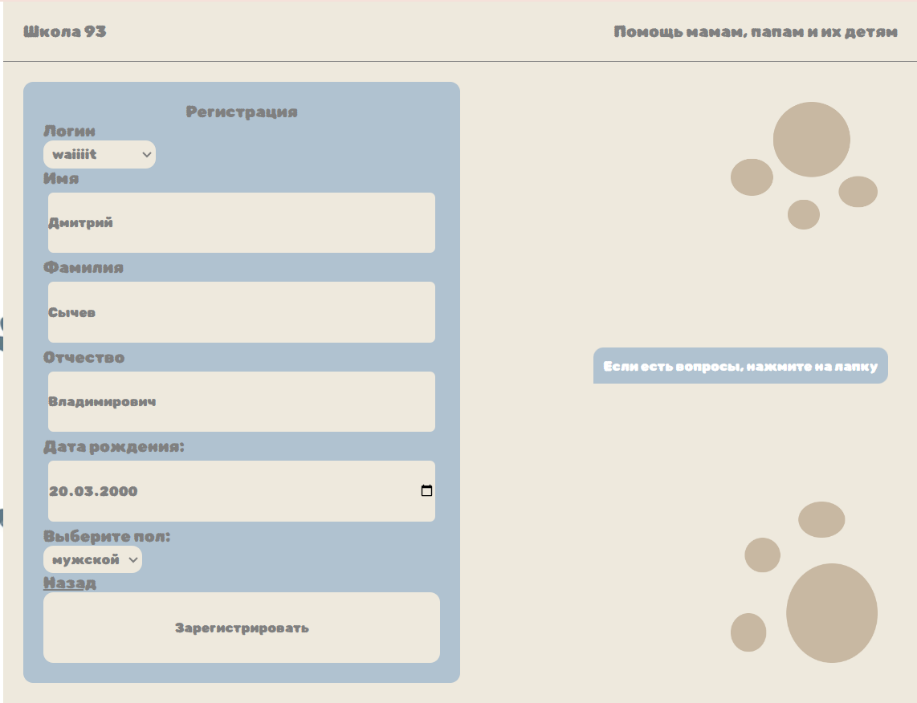
Добавленный ученик появится в БД



Экранная форма «Добавление учителя»



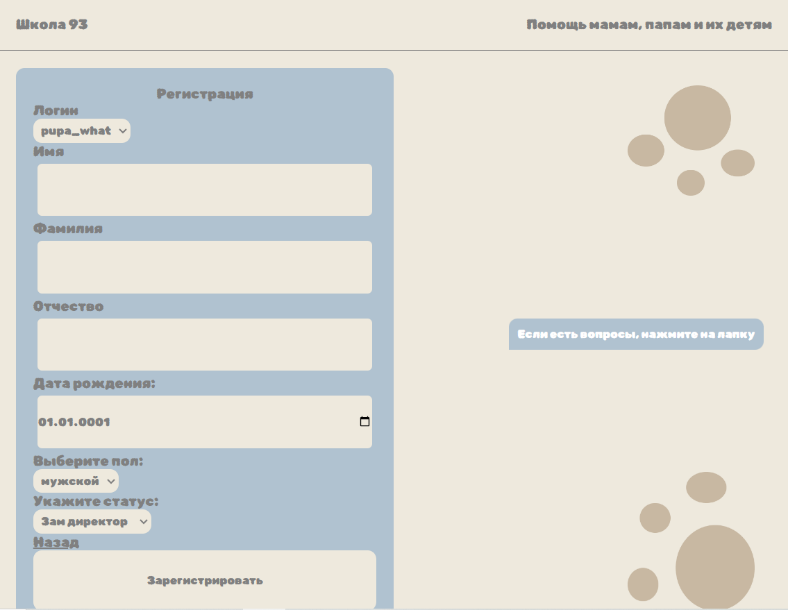
Экранная форма с введенными данными



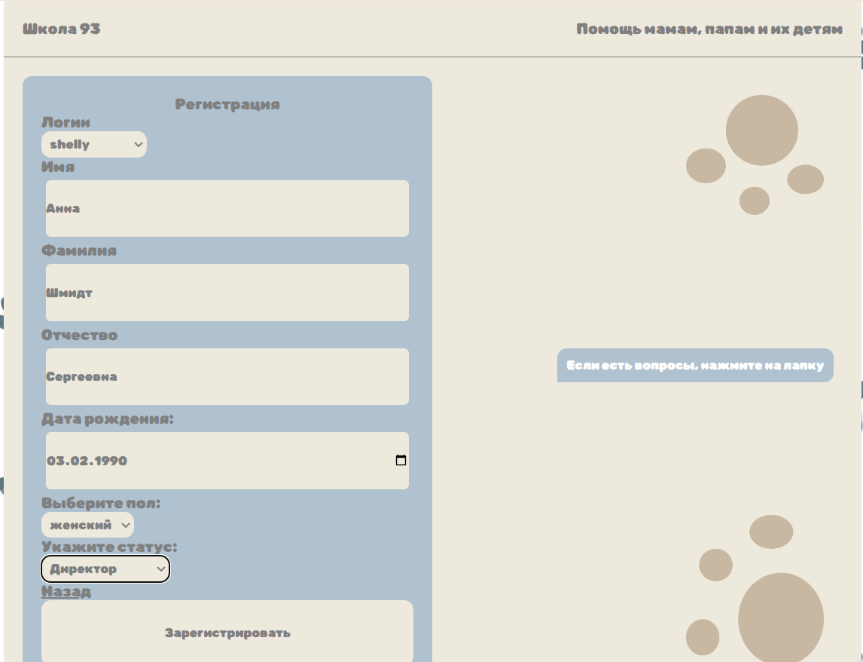
Добавленный учитель появится в БД



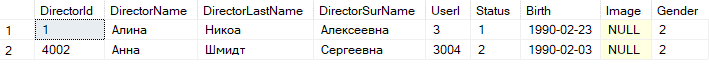
Экранная форма «Добавление других»



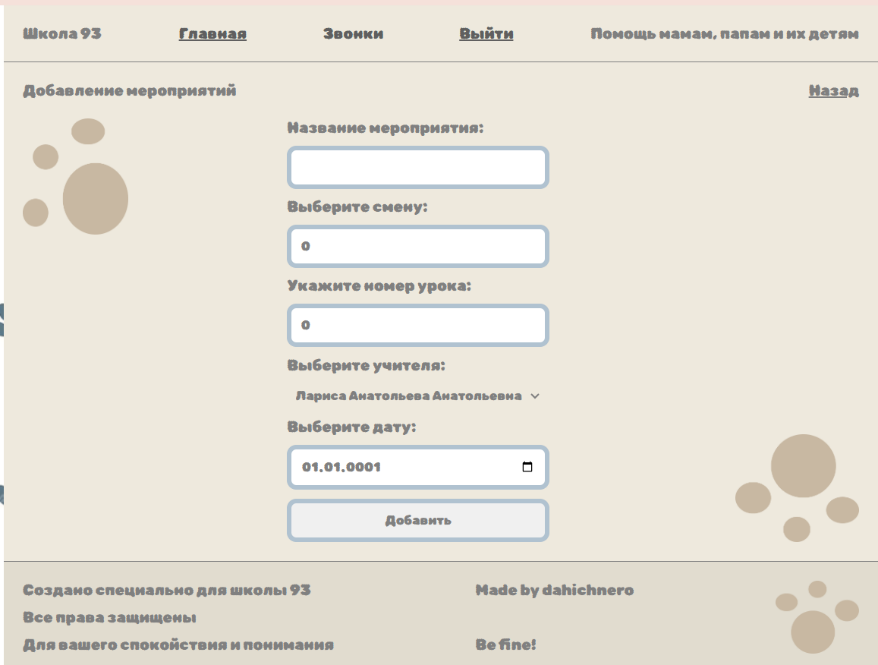
Экранная форма уже с введенными данными



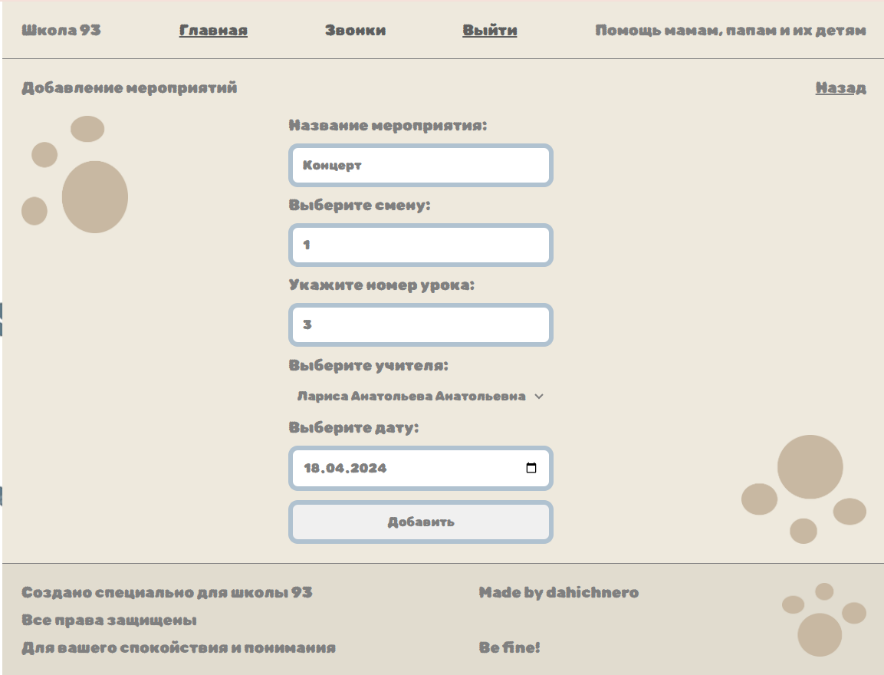
Добавленный директор появится в БД



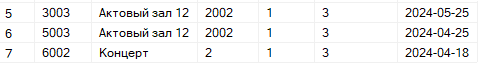
Экранная форма «Добавление мероприятия»



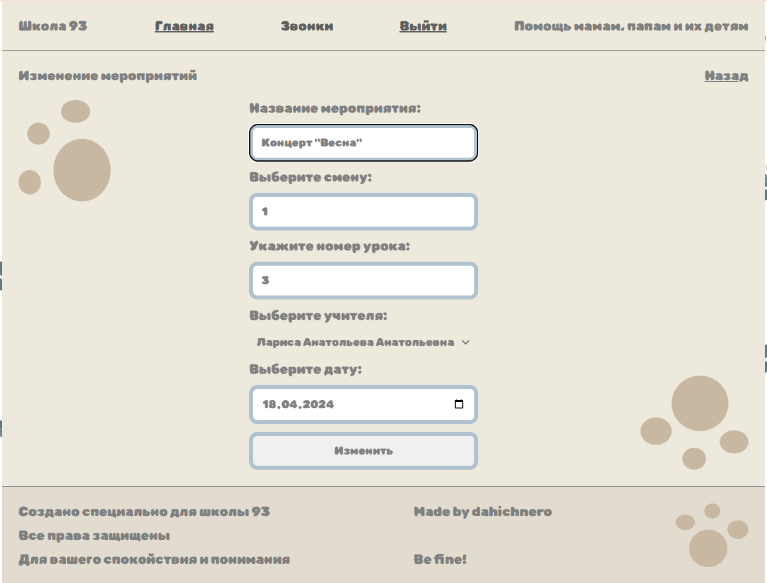
Экранная форма уже с введенными данными



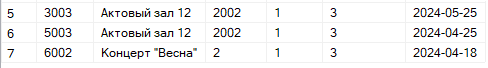
Добавленное мероприятие появится в БД



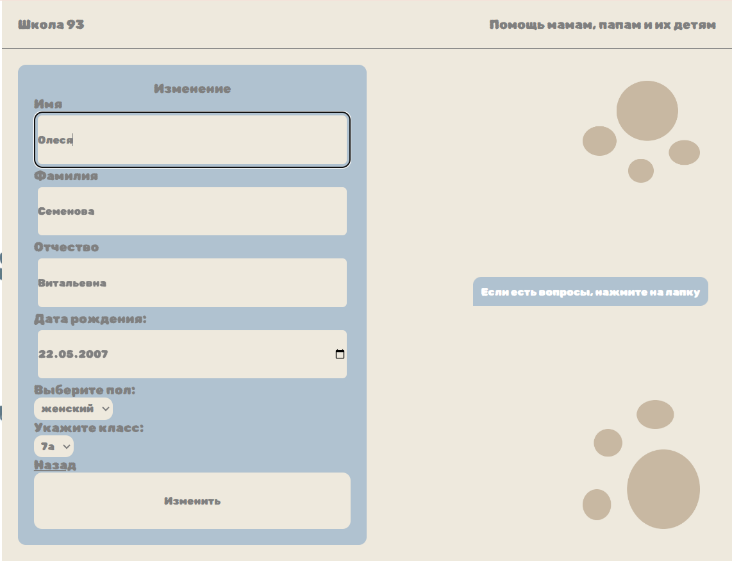
Изменение названия мероприятия в Мероприятие:



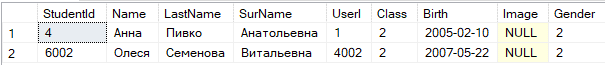
Название мероприятия изменится



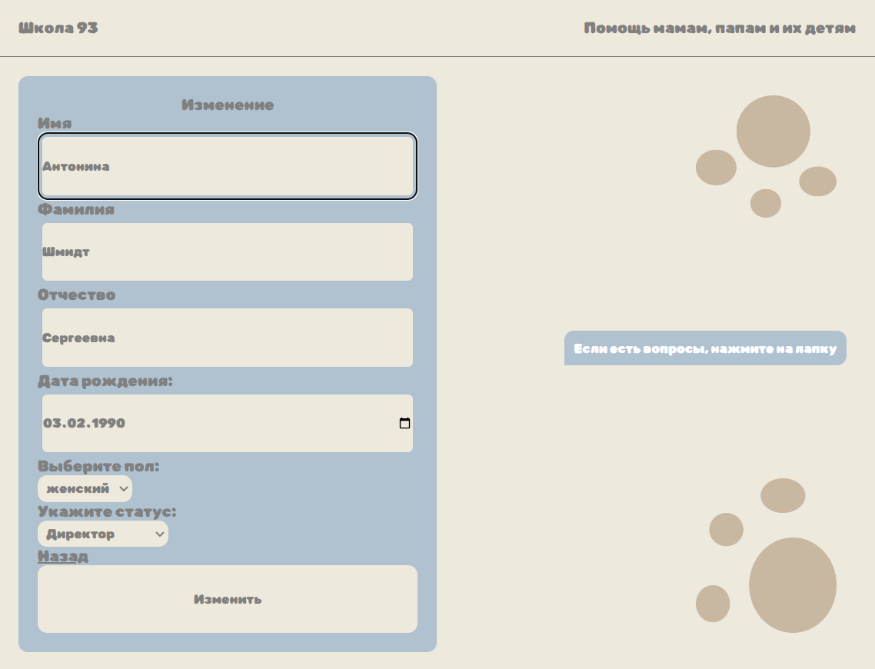
Изменение имени в Ученике:



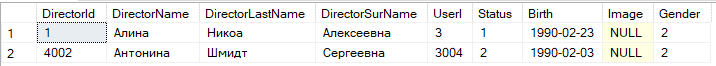
Имя изменится.



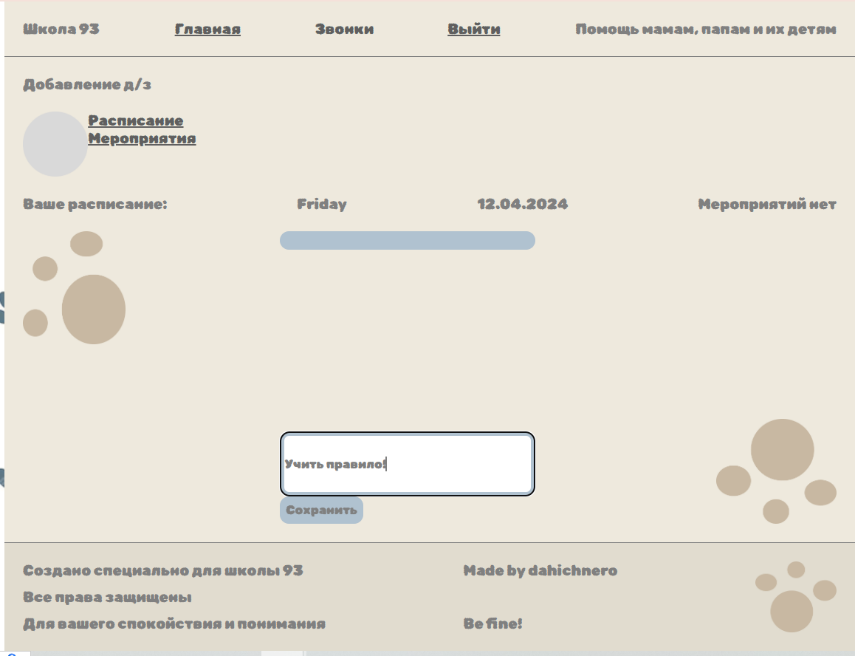
Изменение имени у директора:



Имя изменится.



Изменение д/з в расписании:



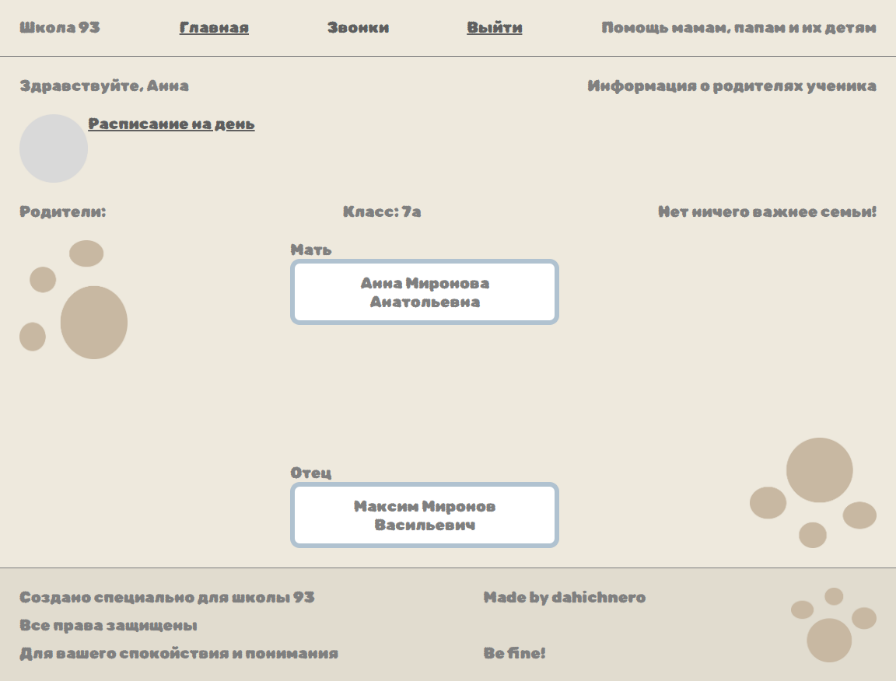
Домашнее задание изменится.



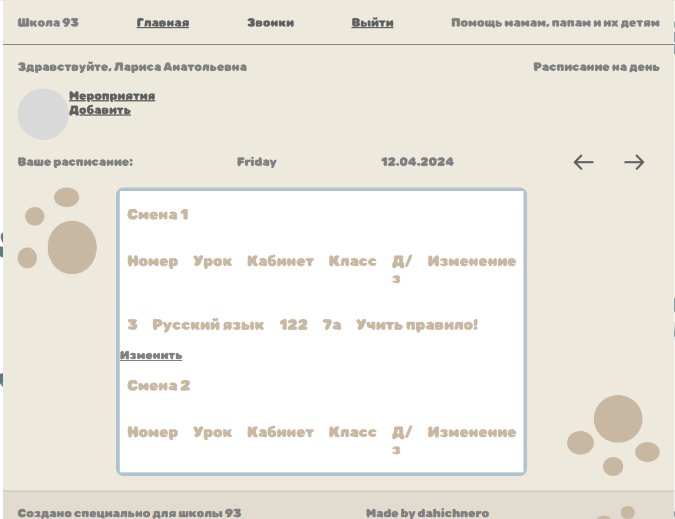
Окно с расписанием ученика



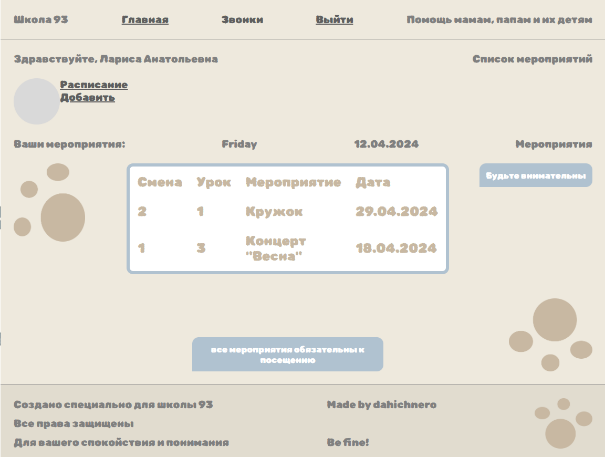
Окно с информацией о родителях ученика



Окно с информацией о расписании учителя



Информация о мероприятиях учителя



Окно для выбора добавления или изменения

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Окна для списка учеников

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Окна для списка учителей

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Окна для списков других людей

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Окно с расписанием директора на день

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Окно для просмотра учеников

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Окно для просмотра мероприятий

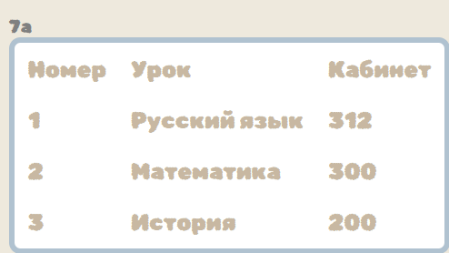
Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

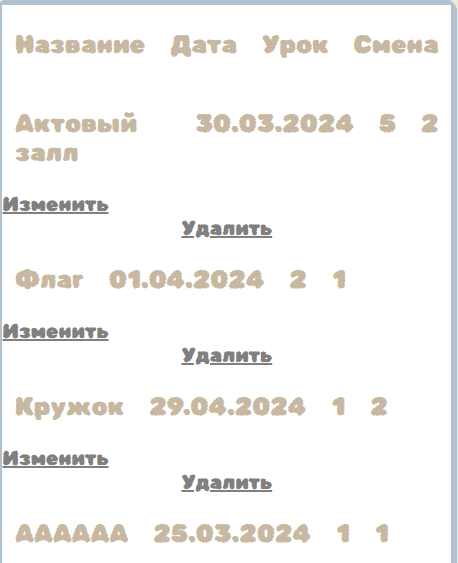
ПРИЛОЖЕНИЕ Г

Образцы отчетов пользователей

Отчет по запросу о выводе расписания всех классов



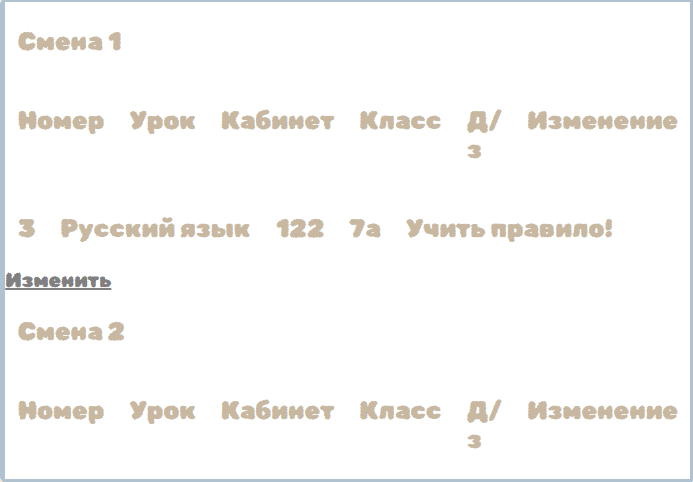
Отчет по запросу о выводе всех мероприятий



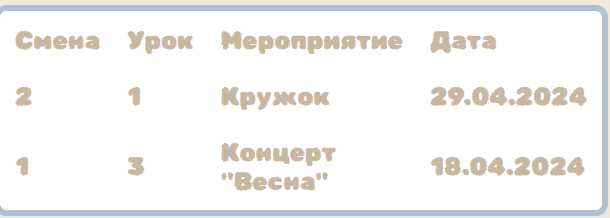
Отчет по запросу о выводе всех учеников



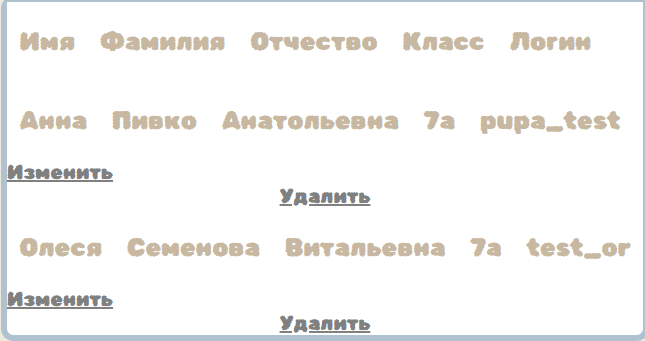
Отчет по запросу о выводе расписания на день



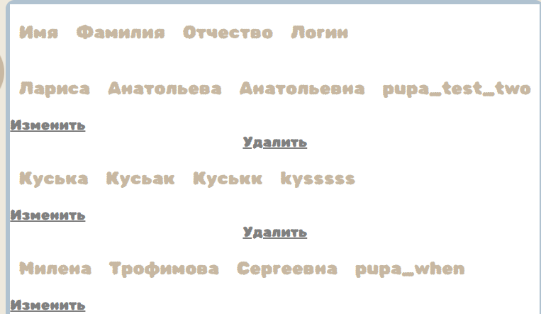
Отчет по запросу о выводе мероприятий для учителя



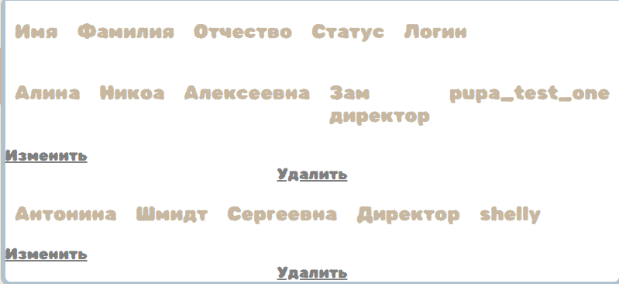
Отчет по запросу о выводе списка учеников



Отчет по запросу о выводе списка учителей



Отчет по запросу о выводе списка директоров и других



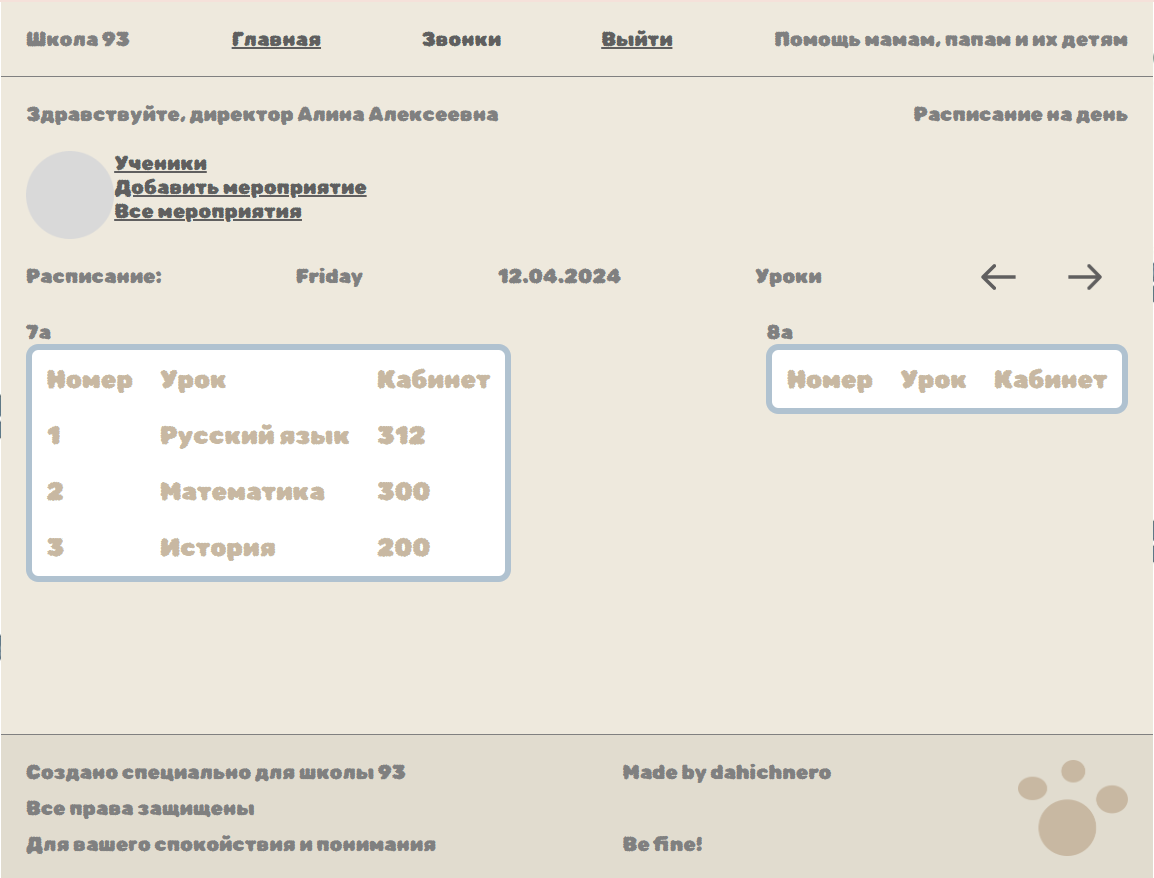
Отчет по запросу о выводе расписания для учеников



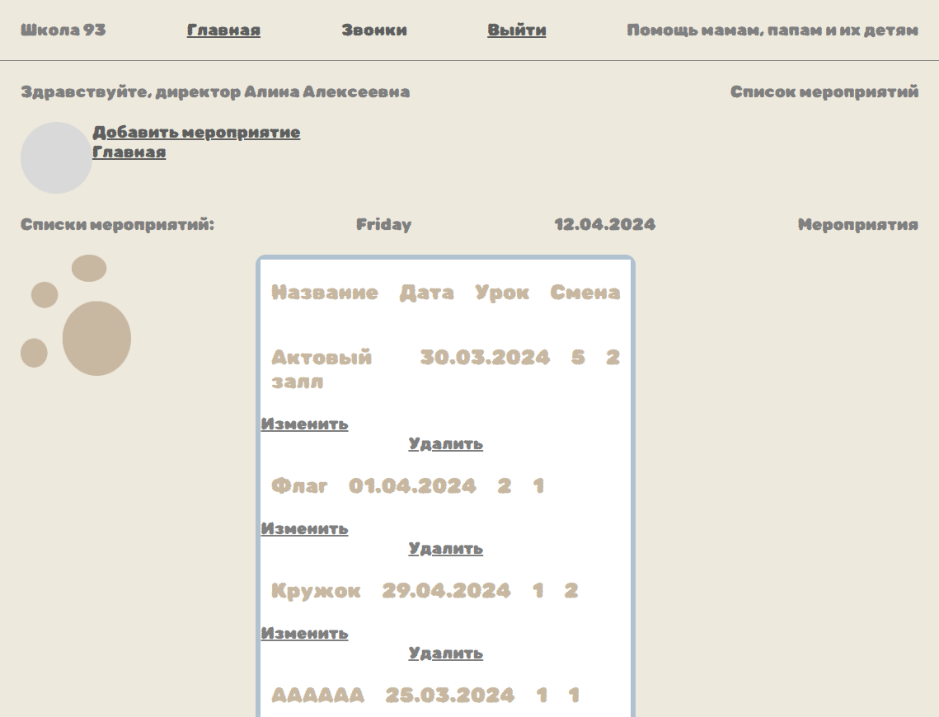
ПРИЛОЖЕНИЕ Д

Образцы запросов пользователей к БД

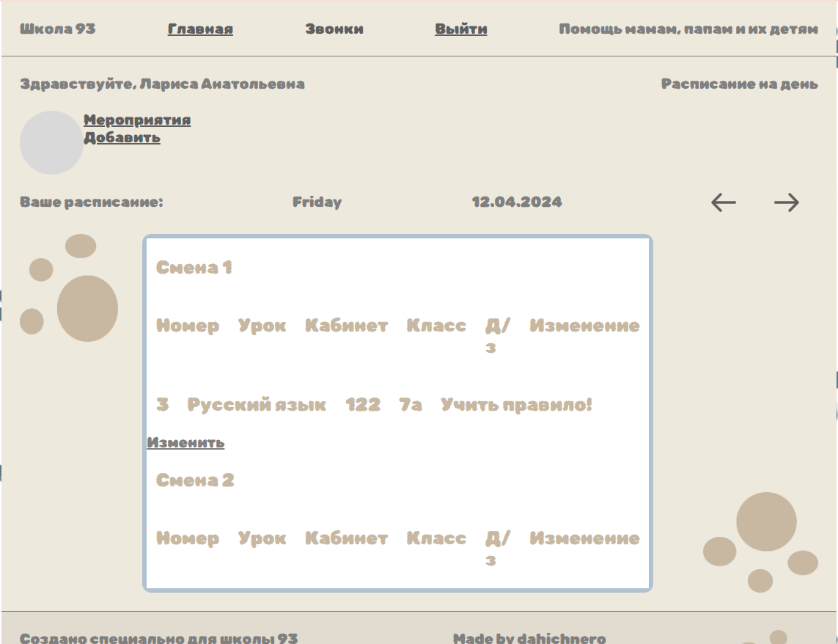
Запрос о выводе расписания всех классов



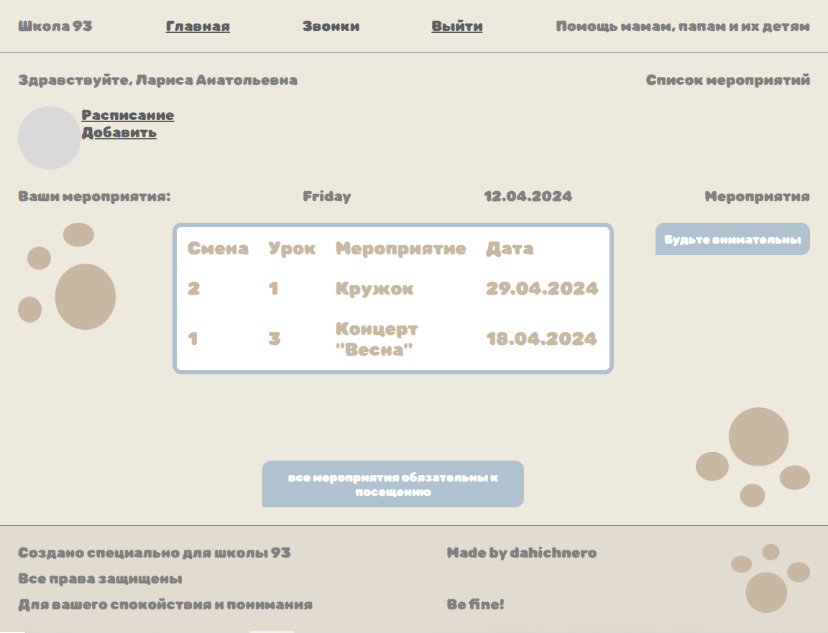
Запроса о выводе списка мероприятий



Запрос о выводе расписания на день



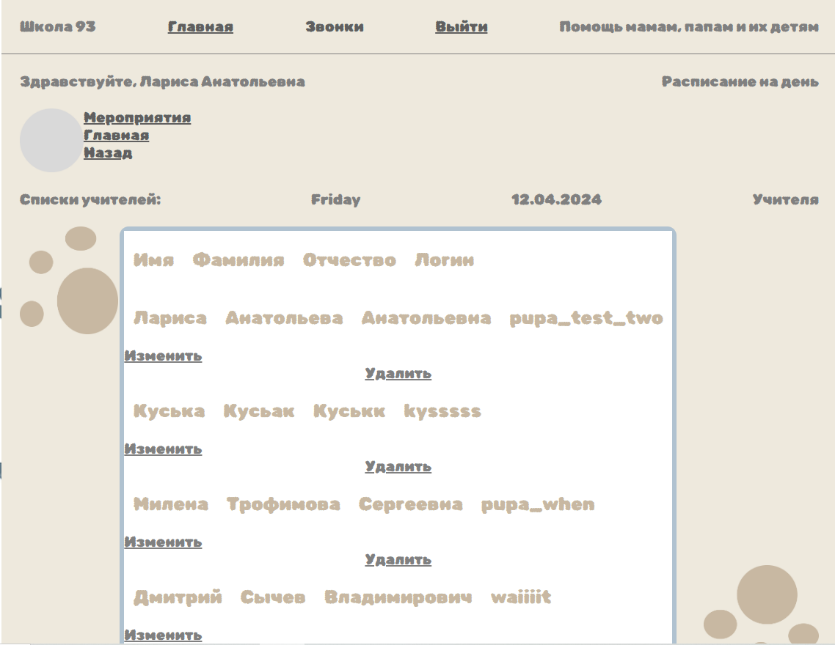
Запрос о выводе мероприятиях учителя



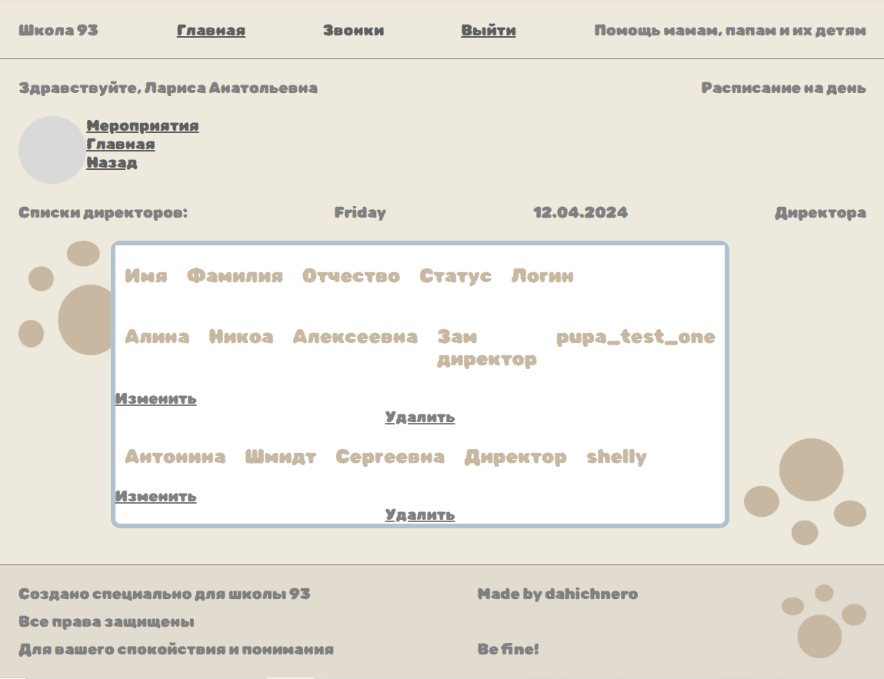
Запрос о выводе учеников



Запрос о выводе учителей



Запрос о выводе директоров и других



Запрос о выводе расписания для ученика

