## ВВЕДЕНИЕ

Данная курсовая работа выполнена в соответствии с заданием на курсовое проектирование. Темой работы является «Система управления «Центр детского творчества».

В современном мире стремительными темпами развиваются информационные технологии и те сферы человеческой деятельности, которые с ними связаны. С каждым годом всё больший и больший объём информации обрабатывается при помощи современных и продолжающих совершенствоваться персональных компьютерах.

Любая организация нуждается в своевременном доступе к информации. Ценность информации в современном мире очень высока. Роль распорядителей информации в современном мире чаще всего выполняют базы данных. Базы данных обеспечивают надежное хранение информации, структурированном виде и своевременный доступ к ней. Практически любая современная организация нуждается в базе данных, удовлетворяющей те или иные потребности по хранению, управлению и администрированию данных.

Главная цель, которая стоит перед написанием курсового проекта – это приобретение практических навыков проектирования базы данных «Центр детского творчества» с использованием программ СУБД. В ходе выполнения данного курсового проекта необходимо будет разработать информационную базу данных и программное обеспечение для работы совместно с базой данных для центра детского творчества, которые помогут пользователю легко найти нужную информацию о сотруднике, детях, группах и событиях в любом момент времени.

## ПОСТАНОВКА ЗАДАЧИ

## Описание предметной области

Предметом курсового проекта является проектирование информационной базы данных и создание программного обеспечения для работы совместно с базой данных для центра детского творчества.

Программный продукт «Система управления «Центр детского творчества» поможет пользователю легко найти нужную информацию о сотруднике, детях, группах и событиях в любом момент времени, автоматизировать все рутинные процессы по учету и управлению данными.

Сфера применения программы «Система управления «Центр детского творчества» это администрирование детских развивающих центров, детских творческих клубов, центров раннего развития и предприятий подобной направленности.

С помощью программы «Центр детского творчества» можно будет выполнять следующие функции:

* Вести клиентскую базу детей с учетом их сопровождающих (родители, родственники)
* Вести клиентскую базу педагогов
* Проводить мероприятия
* В удобном виде регистрировать посещения детей на занятия
* Планировать расписание занятий на любой период
* Составлять график работы педагогов
* И много другое…

На сегодняшний день существует большое количество аналогов. Для сравнения функционального и визуального обеспечения программы «Центр детского творчества», были выбраны программы «Хеликс: Детский Центр» и «CRM-система для автоматизации детских центров Отмечалка».

«Хеликс: Детский Центр» - компьютерная программа, разработанная специально для детских развивающих центров, которая поможет автоматизировать все рутинные процессы по учету и управлению бизнесом.

"Хеликс: Детский Центр" разработан на платформе "1С:Предприятие 8.3". Из плюсов программы можно выделить то что она максимально простая и удобная в работе, а также в ней реализованы множество функциональных возможностей. Из минусов можно отметить, что «Хеликс: Детский Центр» является платной программой, для её установки требуется выезд специалиста, и гарантия на обслуживание составляет всего 3 месяца.

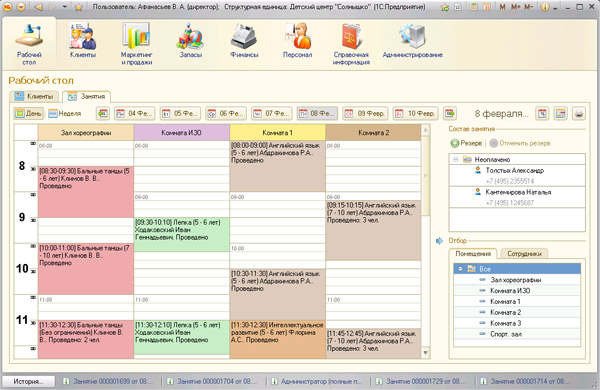


Рисунок 2.1.1 – Хеликс: Детский Центр»

«CRM-система для автоматизации детских центров Отмечалка» - это система автоматизации, предоставляющая полный набор инструментов, которые помогут раскрыть потенциал детского центра, повысить его прибыльность и конкурентоспособность.

Из плюсов это программы можно отметить то что в ней есть бесплатный 14 дневный период, постоянное обновление программы и высокая надёжность хранения данных.

В минусы этой программы также входят можно записать необходимость платить за её использование, а также необходимость подключения к сети интернет.

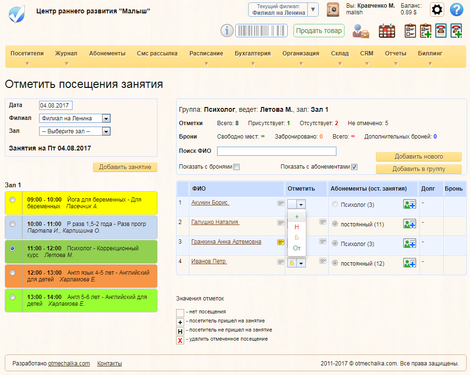


Рисунок 2.1.2 – «CRM-система Отмечалка»

Сравнительный анализ показал, что «Центр детского творчества», разрабатываемый в рамках курсового проектирования, по основным характеристикам не уступает существующим аналогам.

Входной информацией в создаваемой программе будут являться запросы пользователей.

Выходной информацией является подробная информация о детях, сотрудниках, группах и событиях представленная в виде таблиц.

Для достижения цели курсового проекта необходимо решить следующие задачи:

* анализ предметной области;
* построение концептуальной модели базы данных;
* организация базы данных;
* разработка прикладной программы;
* наполнение и сопровождение базы данных;

## Цели и задачи проектирования

## Проектирование

## Проектирование схемы данных

Проектирование базы данных осуществляется в три этапа:

1. Концептуальное (инфологическое) проектирование;
2. Логическое проектирование;
3. Физическое проектирование;

Цель инфологического этапа проектирования состоит в получении семантических (концептуальных) моделей, отражающих предметную область и информационные потребности пользователей. В качестве инструмента для построения семантических моделей данных на этапе инфологического проектирования является неформальная модель "Сущность-Связь" (ER-модель - Entity-Relationship). Моделирование предметной области базируется на использовании графических диаграмм, включающих небольшое число разнородных компонентов.

Основными понятиями ER-модели являются сущность, связь и атрибут.

Сущность (объект) - это реальный или представляемый объект предметной области, информация о котором должна сохраняться и быть доступна. Различают такие понятия, как тип сущности и экземпляр сущности. Понятие тип сущности относится к набору однородных предметов, событий, личностей, выступающих как единое целое. Экземпляр сущности относится к конкретной вещи в наборе. В диаграммах ER-модели сущность представляется в виде прямоугольника (в нотации Баркера), содержащего имя сущности.

Выделим базовые сущности данного курсового проекта:

* **Дети**. Атрибуты: информация о детях, родители.
* **События**. Атрибуты – описание, дата проведения, заметки, цена.
* **Группы**. Атрибуты – сотрудник, класс, расписание, размер группы, цена.
* **Сотрудники**. Атрибуты – ФИО, дата рождения, квалификация.

**Родители** и **Информацию о детях** будем рассматривать как связь с детьми. Атрибуты родителей: ФИО мамы, ФИО папы, дата рождения мамы, дата рождения папы. Атрибуты информации о детях: ФИО, дата рождения, адрес.

**Класс** будем рассматривать как связь с группой. Атрибуты класса – название, описание, заметки, категория. А **Категории** рассматриваем как связь с классом. Атрибуты категории – название, описание.

**Лист** в данной работе рассматривается как связь между такими сущностями как дети, события и группы.

Атрибут - поименованная характеристика сущности, определяющая его свойства и принимающая значения из некоторого множества значений. Каждый атрибут обеспечивается именем, уникальным в пределах сущности.

Связь (Relationship) - это поименованная графически изображаемая ассоциация, устанавливаемая между сущностями и представляющая собой абстракцию набора отношений, которые систематически возникают между различными видами предметов в реальном мире. Большинство связей относятся к категории бинарных и имеют место между двумя сущностями.

Среди бинарных связей существуют три фундаментальных вида связи: один-к-одному (1:1), один-ко-многим (1:M), многие-ко-многим (M:M). Связь один-к-одному (1:1) существует, когда один экземпляр одной сущности связан с единственным экземпляром другой сущности. Связь один-ко-многим (1:M) имеет место, когда один экземпляр одной сущности связан с одним или более экземпляром другой сущности и каждый экземпляр второй сущности связан только с одним экземпляром первой сущности. Связь многие-ко-многим (М:N) существует, когда один экземпляр одной сущности связан с одним или более экземпляром другой сущности и каждый экземпляр второй сущности связан с одним или более экземпляром первой сущности.

В данной курсовой работе предоставлены все три вида связей.

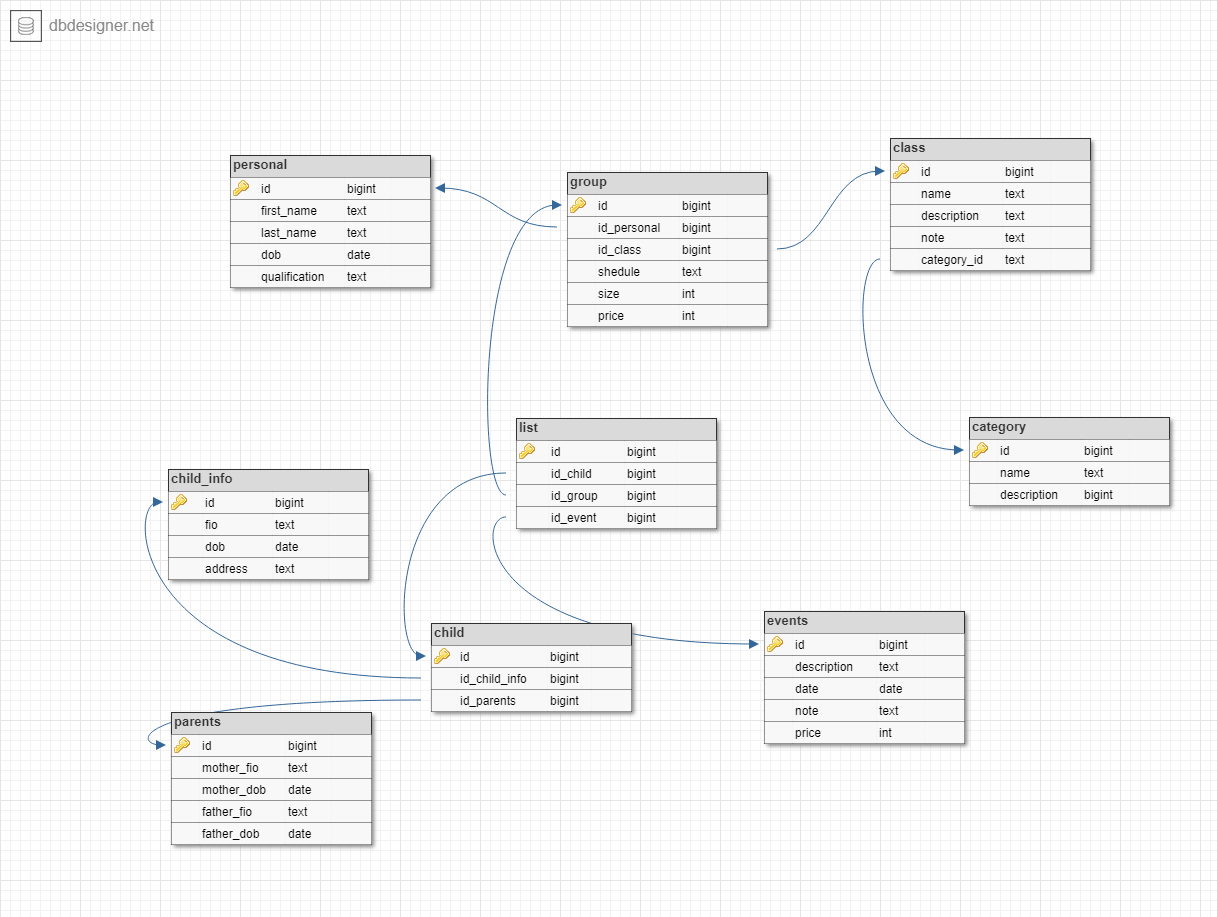


Рисунок 2.2.1 – ER диаграмма сущностей

Функциональные возможности:

* ведение БД (запись, чтение, модификация, удаление);
* обеспечение логической непротиворечивости БД;
* реализация наиболее часто встречающихся запросов в готовом виде;
* предоставление возможности сформировать произвольный запрос на языке манипулирования данными.

**З**апросы:

* получение списка всех групп, включая информацию о детях и классах;
* получение списка всех событий, включая информацию о детях;
* получение списка персонала;
* получение списка детей;
* получение списка полной информации о детях;
* получение списка родителей;
* получение списка классов;
* получение списка категорий;
* удаление данных из выше перечисленных списков;
* редактирование данных из выше перечисленных списков;
* добавление данных в выше перечисленные списки;

Логическое проектирование это проектированием логической структуры БД, что означает определение всех информационных единиц и связей между ними, задание их имен и типов, а также некоторых количественных характеристик (например, длины поля).

При проектировании логической структуры БД, осуществляется преобразование исходной инфологической модели в модель данных, поддерживаемую конкретной СУБД, и проверка адекватности, полученной логической модели отображаемой предметной области.

В данном курсовом проекте БД содержит 11 таблиц.

Таблица 1. Реляционная таблица «Categoty»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибуты | Тип | Ключ | Описание |
| id | bigint | primary | ID категории |
| name | text |  | Название |
| description | text |  | Описание |

Таблица 2. Реляционная таблица «Child»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибуты | Тип | Ключ | Описание |
| id | bigint | primary | ID |
| id\_child\_info | bigint |  | id информации о ребёнке |
| id\_parents | bigint |  | id родителей |

Таблица 3. Реляционная таблица «Child\_info»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибуты | Тип | Ключ | Описание |
| id | bigint | primary | ID |
| fio | text |  | фио |
| dob | date |  | дата рождения |
| address | text |  | адрес |

  Реляционная таблица «Class»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибуты | Тип | Ключ | Описание |
| id | bigint | primary | ID |
| name | text |  | названиие |
| description | text |  | описание |
| note | text |  | заметки |
| category\_id | text |  | номер категории |

Таблица 5.Реляционная таблица «Events»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибуты | Тип | Ключ | Описание |
| id | bigint | primary | ID |
| description | text |  | описание |
| date | date |  | дата проведения |
| note | text |  | заметки |
| price | int |  | стоимость |

Таблица 6. Реляционная таблица «Group»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибуты | Тип | Ключ | Описание |
| id | bigint | primary | ID |
| id\_personal | bigint |  | учитель |
| id\_class | bigint |  | тип группы |
| shedule | text |  | распиание |
| size | int |  | размер группы |
| price | int |  | стоимость |

Таблица 7. Реляционная таблица «List»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибуты | Тип | Ключ | Описание |
| id | bigint | primary | ID |
| id\_child | bigint |  | id ребёнка |
| id\_group | bigint |  | id группы |
| id\_event | bigint |  | id события |

Таблица 8. Реляционная таблица «Parents»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибуты | Тип | Ключ | Описание |
| id | bigint | primary | ID |
| mother\_fio | text |  | фио матери |
| mother\_dob | date |  | дата рождения матери |
| father\_fio | text |  | фио отца |
| father\_dob | date |  | дата рождения отца |

Таблица 9. Реляционная таблица «Personal»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибуты | Тип | Ключ | Описание |
| id | bigint | primary | ID |
| fio | text |  | фио |
| dob | date |  | дата рождения |
| qualification | text |  | квалификация |

Таблица 10. Реляционная таблица «Removed\_events»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибуты | Тип | Ключ | Описание |
| id | bigint | primary | ID |
| description | text |  | описание |
| date | date |  | дата проведения |
| note | text |  | заметки |
| price | int |  | стоимость |

Таблица 11. Реляционная таблица «Removed\_group»

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибуты | Тип | Ключ | Описание |
| id | bigint | primary | ID |
| id\_personal | bigint |  | учитель |
| id\_class | bigint |  | тип группы |
| shedule | text |  | распиание |
| size | int |  | размер группы |
| price | int |  | стоимость |

Таблицы соответствуют трём нормальным формам. Для связи между разными таблицами используется один и тот же ключ в разных таблицах и специально связующие таблицы. Все таблицы идентифицируются числом.

## Обоснование выбора технологий

C# — объектно-ориентированный язык программирования. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Основные достоинства С#:

* C# создавался параллельно с каркасом Framework .Net и в полной мере учитывает все его возможности — как FCL, так и CLR;
* C# является полностью объектно-ориентированным языком, где даже типы, встроенные в язык, представлены классами;
* C# является мощным объектным языком с возможностями наследования и универсализации;
* C# является наследником языков C/C++, сохраняя лучшие черты этих популярных языков программирования. Общий с этими языками синтаксис, знакомые операторы языка облегчают переход программистов от С++ к C#;
* сохранив основные черты своего великого родителя, язык стал проще и надежнее. Простота и надежность, главным образом, связаны с тем, что на C# хотя и допускаются, но не поощряются такие опасные свойства С++ как указатели, адресация, разыменование, адресная арифметика;
* благодаря каркасу Framework .Net, ставшему надстройкой над операционной системой, программисты C# получают те же преимущества работы с виртуальной машиной, что и программисты Java. Эффективность кода даже повышается, поскольку исполнительная среда CLR представляет собой компилятор промежуточного языка, в то время как виртуальная Java-машина является интерпретатором байт-кода;
* мощная библиотека каркаса поддерживает удобство построения различных типов приложений на C#, позволяя легко строить Web-службы, другие виды компонентов, достаточно просто сохранять и получать информацию из базы данных и других хранилищ данных;
* реализация, сочетающая построение надежного и эффективного кода, является немаловажным фактором, способствующим успеху C#

## Описание среды реализации

СУБД MySql — предоставляет мощные средства для доступа, настройки, администрирования, разработки всех компонентов базы данных и управления ими. MySql — это реляционная система управления базами данных. То есть данные в ее базах хранятся в виде логически связанных между собой таблиц, доступ к которым осуществляется с помощью языка запросов SQL. MySql — свободно распространяемая система. Кроме того, это достаточно быстрая, надежная и, главное, простая в использовании СУБД.

Работать с MySql можно не только в текстовом режиме, но и в графическом. Существует очень популярный визуальный интерфейс для работы с этой СУБД — PhpMyAdmin. Этот интерфейс позволяет значительно упростить работу с базами данных в MySql.

PhpMyAdmin позволяет пользоваться всеми достоинствами браузера, включая прокрутку изображения, если оно не умещается на экран. Многие из базовых SQL-функций работы с данными в PhpMyAdmin сведены к интуитивно понятным интерфейсам и действиям, напоминающим переход по ссылкам в Internet.

Microsoft Visual Studio — линейка продуктов компании Майкрософт, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и как отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. VisualStudio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (например, Subversion и VisualSourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода).

Главным преимуществом Visual Studio 2017 является производительность. Обеспечивает возможность создания разнообразных приложений на основе одного набора навыков.

## Программная реализация

## Физическая структура базы данных

## Структура приложения

## Тестирование

## Описание применение

## ЗАКЛЮЧЕНИЕ

Результатом выполнения курсовой работы стало разработанное приложение баз данных, позволяющее автоматизировать рутинные процессы по учету и управлению данными. в детских центрах. Разработанное приложение отвечает всем требованиям предметной области, таблицы созданной базы данных отвечают требованиям нормализации, что позволяет обеспечить целостность и непротиворечивость информации. В данной работе разработана база данных и приложение для работы совместно с базой данных.

В процессе выполнения курсовой работы были закреплены знания, полученные при изучении дисциплины. Были изучены такие пункты:

* анализ предметной области;
* построение концептуальной модели базы данных;
* организация базы данных;
* разработка прикладной программы;
* наполнение и сопровождение базы данных;

В процессе организации БД проведен до необходимого уровня абстракций анализ предметной области, построена реляционная модель БД, произведена нормализация реляционной БД. Реализация проекта была выполнена на современных программных платформах. В качестве технологии доступа к данным была использована объектно-реляционная модель, которая позволяет просто и лаконично осуществлять запросы к базе данных.

Даная курсовая работа была реализована на языке программирования C# с использованием среды разработки Microsoft Visual Studio.

Для проверки корректности работы программы были изучены методы тестирования и проведены тесты, по результатам которых были исправлены ошибки.

Цель и соответствующие задачи, поставленные перед выполнение курсового проекта выполнены в полном объёме.

## Список использованных источников

1. В.А.Гвоздева, И.Ю.Лаврентьева. Основы построения АИС, Москва, ИД «Форум» - ИНФРА-М, 2009.

2. А.В.Рудаков, Технология разработки программных продуктов, Москва, Издательский центр «Академия», 2008

3. Л.Г.Гагарина, Д.В.Киселев, Е.Л.Федотова, Разработка и эксплуатация АИС. Москва, ИД «Форум» - ИНФРА-М, 2009.

4. Г.Ю.Максимович, А.Г.Романенко, О.Ф.Самойлюк. Информационные системы. Москва 2007, Федеральное агентство по образованию

5. Фаронов В.В. Программирование баз данных в Delphi 7: Учебный курс. – СПб.: Питер, 2004 - 464 с.

6. Кренке Д. Теория и практика построения баз данных: [пер.с англ] / Д. Кренке. - 9 - е изд. - СПб.: Питер, 2005. - 858 с.