**Государственное Бюджетное Учреждение Калининградской Области Профессиональная Образовательная Организация   
«Колледж Информационных Технологий и Строительства»**

**Разработка автоматизированного рабочего места для спорткомплекса**

**Проект выполнен студентами гр. ИСа 22-1**

**Баербах Рудольфом**

**Алексанян Артемием**

**Гладких Спартаком**

**Руководитель**

**Большакова-Стрекалова Анна Викторовна**

**Калининград 2023**

Содержание

**Введение**

**1. Описание предметной области автоматизации расписания для спорткомплекса**

**1.1 Документы предметной области**

**1.2 Описание предметной области**

**2. Постановка задачи**

**2.1 Организационно-экономическая сущность задачи**

**2.2 Входная информация**

**2.3 Выходная информация**

**3. Информационное обеспечение задачи**

**3.1 Информационный анализ предметной области и выделение информационных объектов задачи**

**3.2 Определение логической структуры реляционной базы данных (ERD-модель)**

**4. Алгоритмы решения задачи**

**4.1 Обобщенный алгоритм решения задачи и его декомпозиция на модули**

**4.2 Классификация и реализация используемых запросов**

**5. Технология решения задачи**

**5.1 Описание дерева диалога (форм ввода-вывода)**

**5.2 Технология ввода и накопления входной информации, обеспечивающей решение задачи**

**5.3 Технология осуществления запросов**

**5.4 Технология получения отчетов**

**6. Тестирование программы**

**6.1 Тестовые наборы**

**Заключение**

**Сценарий**

1. **Установка сроков**
2. **Осматриваем объект.**
3. **Анализ предстоящей работы.**
4. **Рассказываем что нужно приобрести.**
5. **Рассказываем процесс работы .**
6. **Процесс установки.**
7. **Цена на приборы ПК и т.д.**
8. **Обучение персонала по пользованию Программного Обеспечения.**
9. **Рассказываем цену.**
10. **Оценка результатов**

**1. Описание предметной области автоматизации расписания для спорткомплекса**

**1.1 Документы предметной области**

В своей деятельности спорткомплекс опирается на следующие документы и данные: таблицы расписания, услуги, данные о помещениях, тренерах, посетителях и группах на которые подразделяются посетители. Таблица расписания выводит индивидуальную информацию для каждого элемента подразделения. Для помещений расписание содержит информацию о том, в какой день недели и в какое время будет проходить занятие, а также, какой тренер, и какая группа будут заниматься в помещении. Для групп расписание содержит информацию о дне недели, времени, виде занятий, номере помещения и тренере. Аналогично выводится расписание, для тренеров включающее в себя номер помещения и название группы.

Каталог помещений содержит информацию о названии помещения, виде занятия и месторасположении данного помещения. Данные о тренерах содержат информацию, о Ф.И.О. тренера, адресе проживания, телефоне и возрасте тренера. Данные о группах содержат информацию, о названии группы, специализации и посетителях которые относятся к определенной группе.

**1.2 Описание предметной области**

Главным звеном в данной информационной системе является расписание, на основе которого формируется система учета свободного времени и учет занятости для помещений и тренеров. Так же важной частью системы, является возможность добавления новой информации для подразделений, на основе которой строиться само расписание.

Функционирование автоматизированной системы спорткомплекса можно разделить на три части. Первая часть это возможность добавления новой и редактирования старой информации о тренерах, помещениях, группах и посетителях. Вторая часть это формирование и вывод расписания. Третья часть строиться на основе первой и второй, в ней выводится информация о свободном времени тренеров и помещений, отчеты по загрузкам тренеров и помещений, количестве человек в группе, виде занятий. Система предназначена для добавления и работы только со штатными тренерами. Нет возможности формирования расписания для тренеров с почасовым режимом работы.

**2. Постановка задачи**

**Ответственный – менеджер проекта Артемий Алексанян, сроки – 2 недели**

**2.1 Организационно-экономическая сущность задачи**

Администрация спорткомплекса заказала разработку информационной системы для организации своей работы. Система предназначена для обработки данных о времени проведения занятий, о дне недели, кол-во человек в группе, вид занятий, учет помещений, фамилии тренеров.

Система должна выдавать отчеты по запросу менеджера спорткомплекса: расписание, учет свободного времени, отчеты по загрузкам тренера и помещений.

Менеджер рассматривает списки всех имеющихся групп, тренеров, и помещений. Имеются списки, отображающие свободное время для помещений и тренеров. Эта информация необходима для формирования расписания.

Расписание для помещений содержит информацию о дне недели, времени начала и окончания занятия, виде занятий, группе, у которой будет проводиться занятие и тренере, который будет проводить занятие. Аналогично формируется расписание для групп и тренеров. Расписание для групп содержит информацию о номере помещения и Ф.И.О. тренера. В расписание тренеров входит информация о номере помещения и названии группы. Менеджер имеет возможность добавлять тренера, группу, помещение, а также регистрировать посетителей и определять к какой группе они относятся.

**2.2 Входная информация**

Описание входной информации:

* Данные о тренере (номер тренера, Ф.И.О., адрес, телефон, возраст);
* Данные о помещении (номер помещения, название помещения, вид проводимых занятий, этаж);
* Данные о группе (номер группы, название группы, специализация);
* Данные о посетителях (номер посетителя, Ф.И.О., название группы);
* Расписание (номер времени, номер тренера, номер помещения, номер группы).

**2.3 Выходная информация**

Описание выходной информации:

* Таблица расписания для помещений (день недели, время, тренер, вид спортивного занятия, группа);
* Таблица расписания для групп (день недели, время, название помещения, тренер);
* Таблица расписания тренеров (день недели, время, номер помещения, группа);
* Таблица учета свободного времени для помещений и тренеров (день недели, время);
* Отчет по помещениям (название помещения, количество пар за неделю, вид занятия);
* Отчет по группам (название группы, количество человек в группе);
* Отчет по тренерам (Ф.И.О., количество пар за неделю, время работы за неделю);
* Информационная таблица обо всех помещениях (название помещения, вид занятия, этаж);
* Информационная таблица обо всех группа (название группы, специализация);
* Информационная таблица обо всех тренерах (Ф.И.О., адрес, телефон, возраст);
* Информационная таблица обо всех посетителях (Ф.И.О., группа).

**3. Информационное обеспечение задачи**

**Ответственный – ведущий специалист отдела информационного обеспечения Гладких Спартак Никитич, сроки – 2 месяца**

**3.1 Информационный анализ предметной области и выделение информационных объектов задачи**

При анализе предметной области были выделены следующие наиболее важные объекты и их свойства.

Помещение (код помещения, название, вид занятий, этаж). Объект содержит информацию о помещениях, в которых проводятся занятия.

Тренер (код тренера, Ф.И.О., адрес, телефон, возраст). Объект содержит информацию о тренерах, работающих в спорткомплексе.

Группа (код группы, название, специализация). Объект содержит информацию обо всех группах, которые занимаются в спорткомплексе.

Посетитель (код посетителя, код группы, Ф.И.О.). Объект содержит информацию обо всех посетителях, которые подразделены на группы.

День недели (код дня, день недели). Объект хранит все дни недели, в которые работает спорткомплекс.

Время (код времени, время). Объект хранит информацию о времени начала и окончания занятий.

**3.2 Определение логической структуры реляционной базы данных (ERD-модель)**

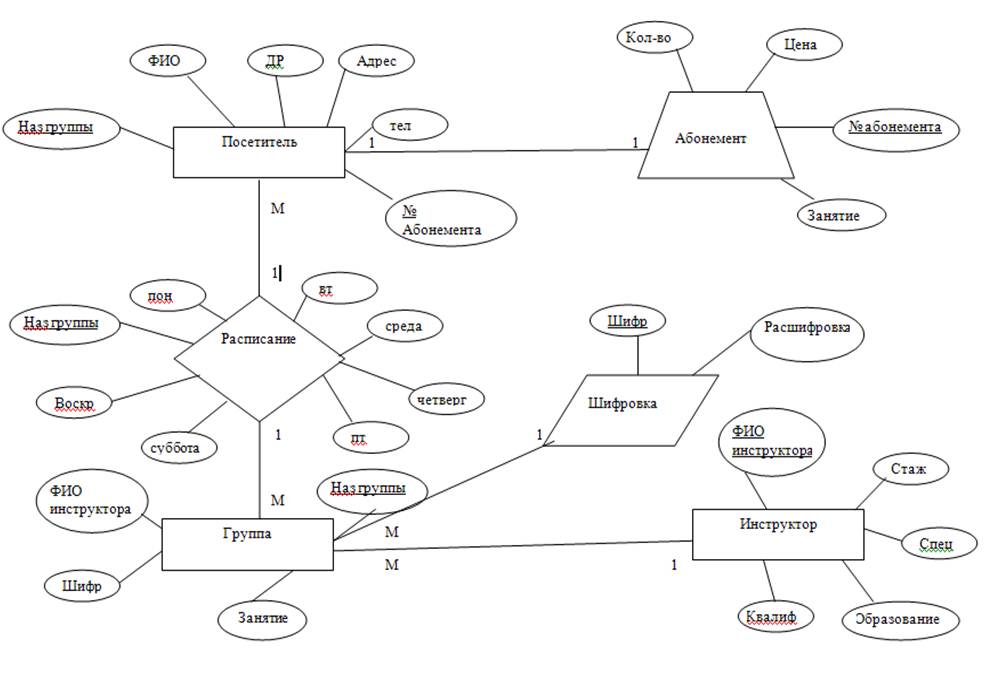
Для системы спорткомплекса были выделены следующие сущности: «Помещение», «Тренер», «Группа», «Посетитель», «Расписание», «День и время», «День проведения занятий» и «Время проведения занятий». Рассмотрим наглядно взаимодействие этих сущностей. Для этого построим ERD-модель данных (Рисунок 1).

Рисунок 1 – ERD-модель.

Связи между сущностями:

* Сущность «группа» связана (М:М) с «посетителями», один посетитель может находиться в нескольких группах.
* Сущность «группа» связана (1:М) с «расписанием» значит для определенной группы может быть сформировано расписание с разными тренерами, помещениями и в разные дни недели и время.
* Сущность «тренер» связана (1:М) с «расписанием» значит для определенного тренера может быть сформировано расписание с разными группами, помещениями, днями недели и временем.
* Сущность «помещение» связана (1:М) с «расписанием» значит, в определенном помещении могут заниматься множество групп с различными тренерами, и в разные дни недели и время.
* Сущности «день проведения занятий» и «время проведения занятий» связаны (1:M) c сущностью «день и время», каждый рабочий день недели содержит все промежутки времени начала и окончания занятий.
* Сущность «день и время» связано (1:М) с «расписанием», расписание содержит разные дни недели и разное время занятий проводимых в спорткомплексе для всех подразделений.
* Сущность «группа» связана (1:М) с сущностью «вид спортивного занятия», группа занимается определенным видом спорта.

Анализируя каждую сущность и связи между ними, создадим необходимые таблицы базы данных.

**4. Алгоритмы решения задачи**

**4.1 Обобщенный алгоритм решения задачи и его декомпозиция на модули**

**Connect.php** – модуль соединения с БД. Find.php – модуль осуществляющий поиск данных. Index.php – модуль включающий в себя основной интерфейс и вызов остальных модулей и функций.

**Function.php** – модуль содержащий следующие функции:

* Данные о тренере (номер тренера, Ф.И.О., адрес, телефон, возраст);
* Function rasp\_trener – выводит расписание для тренера;
* Function rasp\_group – выводит расписание для группы;
* Function add\_raspisanie – добавляет расписание для любого из подразделений, в зависимости от параметров;
* Function edit – редактирует расписание для любого из подразделений;
* Function sw\_time – выводит таблицу учета свободного времени;
* Function spisok – формирует поле со списком для каждого подразделения.

**View\_info.php** – модуль формирующий отчеты и формы для необходимого подразделения.

**4.2 Классификация и реализация используемых запросов**

При написании курсовой работы были использованы следующие классы SQL запросов:

* запросы на создание таблиц БД;
* запросы на выборку данных со всевозможными условиями;
* запросы на редактирования записей в таблицах;
* запросы на удаление записей из таблиц;
* запросы на добавление записей в таблицы.

Все запросы, написанные для реализации данной информационной системы, не имеют общего места расположения, а используются повсеместно в разных частях исходного кода (во всех модулях) по мере необходимости. Запросы на создание таблиц БД. SQL запрос на создание таблицы time (Время):

* CREATE TABLE `time` (`id\_time` INT(11) NOT NULL AUTO\_INCREMENT,`time` VARCHAR(20) NOT NULL, PRIMARY KEY (`id\_time`));
* CREATE TABLE `weekday` (`id\_weekday` INT(11) NOT NULL AUTO\_INCREMENT,`weekday` VARCHAR(11) NOT NULL, PRIMARY KEY (`id\_weekday`));
* SQL запрос на создание таблицы time\_off\_weekday (Время дня недели). CREATE TABLE `time\_off\_weekday` (`id\_timeweekday` INT(11) NOT NULL AUTO\_INCREMENT,`id\_time` INT(11) NOT NULL,`id\_weekday` INT(11) NOT NULL, PRIMARY KEY (`id\_timeweekday`));
* SQL запрос на создание таблицы auditorium (Помещения). CREATE TABLE `auditorium` (`id\_auditorium` INT(11) NOT NULL AUTO\_INCREMENT,`number\_auditorium` VARCHAR(10) NOT NULL,`type\_occupation` VARCHAR(15) NOT NULL,`floor` VARCHAR(5) NOT NULL, PRIMARY KEY (`id\_auditorium`));
* SQL запрос на создание таблицы group\_off\_student (Группы). CREATE TABLE `group\_off\_student` (`id\_group` INT(11) NOT NULL AUTO\_INCREMENT,`name` VARCHAR(10) NOT NULL,`decryption` VARCHAR(30) NOT NULL, PRIMARY KEY (`id\_group`));
* SQL запрос на создание таблицы peoples (Посетители). CREATE TABLE ` peoples ` (`id\_peoples ` INT(11) NOT NULL AUTO\_INCREMENT,`id\_group` INT(11) NOT NULL,`name` VARCHAR(15) NOT NULL,`surname` VARCHAR(15) NOT NULL,`patronymic\_name` VARCHAR(15) NOT NULL, PRIMARY KEY (`id\_student`));
* SQL запрос на создание таблицы teacher (Тренеры). CREATE TABLE `teacher` (`id\_teacher` INT(11) NOT NULL AUTO\_INCREMENT,`name` VARCHAR(15) NOT NULL,`surname` VARCHAR(15) NOT NULL,`patronymic\_name` VARCHAR(15) NOT NULL,`vid\_zanyatiy` VARCHAR(15) NOT NULL,`Address` VARCHAR(30) NOT NULL,`Telephone` VARCHAR(15) NOT NULL,`Age` VARCHAR(10) NOT NULL, PRIMARY KEY (`id\_teacher`));
* SQL запрос на создание таблицы raspisanie (Расписание). CREATE TABLE `raspisanie` (`id\_raspisanie` INT(11) NOT NULL AUTO\_INCREMENT,`id\_timeweekday` INT(11) NOT NULL,`id\_auditorium` INT(11) NOT NULL,`id\_teacher` INT(11) NOT NULL,`id\_group` INT(11) NOT NULL, PRIMARY KEY (`id\_raspisanie`));

**5. Технология решения задачи**

**5.1 Описание дерева диалога (форм ввода-вывода)**

Как правило, для удобства пользователей создаются экранные формы, которые позволяют рационально расположить данные на экране, использовать разнообразные элементы оформления, обеспечивать возможности специфических проверок при вводе данных.

Экранные формы уже давно применяются в практике создания баз данных. Роль форм сильно возросла и изменилась в последнее время. Сейчас экранные формы используются не только для ввода данных в базу, корректировки данных, а также позаписного ввода данных, но и для реализации просмотров базы данных по определенным условиям, и даже для создания заставок и меню.

Если первоначально формы определялись только как способ отображения одной записи в БД, то сейчас в формах допускается наличие табличной (многострочной) части; более того, допускается наличие множества динамических частей, возможно, из разных баз данных.

Экранные формы можно классифицировать по ряду признаков:

* по характеру связи с таблицами;
* по распределению данных по экранам (страницам);
* по способу реализации;
* по форме представления информации;
* по выполняемым функциям различают формы ввода, вывода, управляющие, смешанные.

Назначение каждого вида этих форм ясно из их названия. Следует обратить внимание на то, что даже в случае, если формы ввода и вывода полностью совпадают по своему внешнему виду, иногда целесообразно их выполнить как самостоятельные формы в целях обеспечения безопасности данных.

При проектировании данной базы данных также использовалось большое количество форм ввода-вывода. Например, формами ввода и просмотра информации являются такие формы как «просмотр информации о подразделениях», позволяющая вносить и выводить данные, «просмотр расписания» и «просмотр свободного времени», дающие возможность вносить и просматривать данные.

**5.2 Технология ввода и накопления входной информации, обеспечивающей решение задачи**

Организуя ввод данных, в базу данных, необходимо помнить, что человек, использующий какую либо форму для ввода информации в базу может допустить всевозможные ошибки.

При организации ведения баз данных нужно стремиться к реализации принципа однократного ввода информации. Необходимо стараться до минимума сократить количество ручных операций.

Надо обеспечивать контроль правильности введенных данных. Необходимо осуществлять фильтрацию данных, проверку на запрещенные символы, заполнение всех полей формы, длину строк и др.

Введенные данные проходят тщательную обработку (фильтрацию), для предотвращения записи не корректно сформированных данных или ошибочной информации. После фильтрации данные записываются в базу данных, и хранятся там столько сколько потребуется**.**

**5.3 Технология осуществления запросов**

Если пользователю необходимо получить какую-либо информацию, касающуюся данной предметной области, то он может воспользоваться запросами, существующими в базе данных. Все SQL запросы прописаны в исходном коде программы. Обрабатывая, какую либо форму, выполняется тот запрос, который принадлежит данной форме.

Если это форма добавления нового тренера, то обрабатывается запрос добавления в базу данных записи с информацией обо всех необходимых атрибутах таблицы. Если хотя бы один из атрибутов не будет передан, то есть, какое либо из полей формы не будет заполнено или будет не соответствовать требованиям, запрос не будет выполнен, и появится сообщение об ошибке. В противном случае информация со всех полей формы передастся в поле таблицы. Таким образом, выполняются все остальные запросы, принадлежащие различным формам, таким как форма редактирования, удаления и вывода информации. Если пользователь попытается осуществить свой запрос, ему необходимо будет использовать исходный код системы и в нем сформировать все необходимы формы для данного запроса. Примеры некоторых запросов используемых в системе приведены ниже.

Запрос на выборку посетителей из определенной группы.

SELECT name, surname, patronymic\_name FROM student WHERE id\_group = ‘ВМ-06–01’;

Запрос, подсчитывающий количество человек в группе.

SELECT COUNT (id\_group) FROM student WHERE id\_group = ‘ВМ-06–01’;

Запрос, выводящий таблицу учета свободного времени у тренеров.

SELECT id\_weekday, id\_time FROM time\_off\_weekday WHERE id\_timeweekday NOT IN (SELECT id\_timeweekday FROM raspisanie WHERE id\_teacher = 1;

**5.4 Технология получения отчетов**

Есть несколько способов получения отчетов. На панели главного меню расположены две формы, одна из них служит для получения отчетов методом выбора нужного элемента из поля со списком. Таких поля три, каждое поле соответствует определенному подразделению, такому как помещения, группы и тренеры. При выборе элемента из списка выводится отчет (о расписании, учете свободного времени и информация о выбранном элементе) в виде таблицы, но кроме отчета предоставлены еще несколько форм для работы с отчетов, благодаря этим формам можно производить изменения данных прямо в отчете и смотреть, как формируется отчет на основе этих изменений. Это удобно и позволяет нам не тратить время на переходы от формы вывода информации (отчета) к форме редактирования, все происходит в одном месте. Кроме всего этого существует форма позволяющая вывести всю информацию об определенном подразделении и др.

**6. Тестирование программы**

**Ответственный – ведущий тестировщик Баербах Рудольф Алексеевич, сроки – 2 месяца**

**6.1 Тестовые наборы**

Перед тем как поместить данные из заполненной формы в БД, необходимо провести обработку данных (фильтрацию). В данной системе предусмотрена проверка вводимых данных. Запись в базу будет производиться в том случае, когда будет выполнен ряд условий, таких как:

* проверка на заполнение всех полей формы;
* проверка да длину строки;
* проверка на соответствие типов;
* проверка на запрещенные символы.

Если введенные данные соответствуют всем требованиях, перед отправкой их обрабатывает функция htmlspecialchars, которая служит для защиты БД MySQL.

Если данные не проходят всех уровней фильтрации, выводятся различные ошибки, в зависимости от определенного условия. Например, не выполнили заполнение полей формы, выводится ошибка

В таблицы 2, 3 и 4 не вносятся новые записи, они служат для вывода и обработки существующих в них записей. Тестовую информацию в них вносить не обязательно. Таблица 2 – День проведения занятия

**Заключение**

Исходя из практически возникшей необходимости, в данной выпускной квалификационной работе была спроектирована и реализована информационная система "Автоматизированное рабочее место администратора спортивного комплекса", которая представляет собой приложение, автоматизирующее работу администратора спортивного комплекса:

1. ведение клиентской базы;
2. создание различных видов абонементов;
3. удобное отображение действующих абонементов;
4. удобное отображение информации по тренерам;
5. обеспечение ввода, удаления, хранения и редактирования информации, которая содержится в таблицах данных.

Приложение "Автоматизированное рабочее место администратора спортивного комплекса" реализовано в современной и перспективной сисеме управления базами данных Microsoft Office Access.

В системе разработан удобный интерфейс, не требующий дополнительного обучения для работы с ней. Программное приложение предназначено для использования непосредственно в спортивном комплексе.

Таким образом, созданное приложение позволяет наиболее достоверно, быстро и безошибочно собирать и производить различные операции с данными. А значит, позволит быстрее и качественнее выполнять администратору спортивного комплекса свою работу, не отвлекаясь на перепроверку данных.