Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Волгоградский государственный технический университет»

Себряковский филиал

Специальность 09.02.04

«Информационные системы (по отраслям)»

**КУРСОВОЙ ПРОЕКТ**

**по дисциплине**

**«Основы проектирования баз данных»**

**Пояснительная записка на тему:**

**«Разработка базы данных для системы учёта посещаемости»**

**Выполнил**

студент гр <Группа> <ФИО>

\_\_\_\_\_\_\_\_\_\_\_ (подпись)

**Проверил**:

преподаватель

<ФИО>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (подпись)

Дата\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*<Город>*

*2021*

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 4](#_Toc1)

[I. ТЕОРИТИЧЕСКАЯ ЧАСТЬ 6](#_Toc2)

[1.1 Основные понятия 6](#_Toc3)

[1.2 Этапы проектирования БД 8](#_Toc4)

[1.2.1 Концептуальное проектирование 8](#_Toc5)

[1.2.2 Логическое проектирование 10](#_Toc6)

[1.2.3 Физическое проектирование 11](#_Toc7)

[1.3 СУБД 12](#_Toc8)

[1.3.1 Классификация СУБД 13](#_Toc9)

[1.4 Структура реляционной базы данных 14](#_Toc10)

[1.5 Типы данных 15](#_Toc11)

[1.6 Предметная область 16](#_Toc12)

[II. ПРАКТИЧЕСКАЯ ЧАСТЬ 18](#_Toc13)

[2.1 Предметная область 18](#_Toc14)

[2.2 Концептуальное проектирование 19](#_Toc15)

[2.3 Логическое проектирование 20](#_Toc16)

[2.4 Физическое проектирование 22](#_Toc17)

[2.5 Интерфейс 25](#_Toc18)

[2.5.1 Структура веб-сайта 25](#_Toc19)

[2.5.2 Главная 26](#_Toc20)

[2.5.3 Дневник 28](#_Toc21)

[2.5.4 Студент 29](#_Toc22)

[2.5.5 Преподаватель 30](#_Toc23)

[2.5.6 Проверка визита 31](#_Toc24)

[2.5.7 Запись визита 31](#_Toc25)

[2.5.8 Автоматическое обновление данных в базе 32](#_Toc26)

[2.5.9 Скрипты JavaScript 33](#_Toc27)

[ЗАКЛЮЧЕНИЕ 34](#_Toc28)

[СПИСОК ЛИТЕРАТУРЫ 35](#_Toc29)

# **ВВЕДЕНИЕ**

В современном обществе преобладает производство информационных продуктов, а реальные продукты становятся более информационными. В результате возникли противоречия между ограниченными способностями людей по восприятию и анализу информации и имеющимися массивами хранящейся и обрабатываемой информации. Появилось много избыточной информации, в которой иногда сложно ориентироваться и выбрать подходящие сведения.

Ключевые идеи нынешней информационной технологии основываются на идее, согласно которой информация должна быть организована в базы данных и приемлемо отображать изменения современного мира, удовлетворяя информационные потребности людей.

Система должна давать возможность хранения, масштабирования и учёта информации. Нужно гарантировать просмотр этих данных, их изменение и дополнение для поддержки актуальности.

Система должна обеспечивать возможность нахождения и просмотра отдельных частей собранный данных.

Цель данной курсовой работы не просто разработка базы данных, а создание полноценной системы учёта посещаемости в учебном заведении, с возможностью ввода данных полуавтоматическим способом, а также альтернативным (ручным). Система предполагает постоянное функционирование и обновление данных в фоновом режиме, а также предоставляет накопленную информацию в удобном для анализа виде (интерактивные графики и таблицы).

Задачи:

* разработка моделей базы данных и моделей поведения базы данных,
* описание связей сущностей базы данных,
* непосредственная разработка базы данных,
* создание связей,
* создание интерфейса базы данных, способного обеспечить работу с системой неподготовленного пользователя.

# I. ТЕОРИТИЧЕСКАЯ ЧАСТЬ

## 1.1 Основные понятия

База данных (сокр. БД) - совокупность данных, хранимых в соответствии с определённой структурой и архитектурой. Информация базы данных хранится в таблицах. Таблицы с данными состоят из набора однообразных записей, расположенных друг за другом, образуя тем самым строки таблицы, которые можно добавлять, удалять или изменять.

Данные - информация, которую можно получить, обработать, сохранить, передать и которая может быть подтверждена или опровергнута.

Архитектура БД - совокупность её функциональных компонентов, средств обеспечения их взаимодействия друг с другом, с пользователем и с персоналом.

СУБД - комплекс программных и языковых средств, позволяющий манипулировать как самой базой данных, так и данными в ней, например создание базы данных, создание таблиц, добавление, удаление и редактирование записей таблицы и многое другое.

Сущность - объект, данные о котором хранятся в отношении (таблице, в которой содержатся данные).

Таблицы – главные объекты базы данных, состоящие из строк и столбцов и содержащие все данные, хранимые в БД.

Домен - выражение, определяющее значение, разрешённое для данного атрибута.

Запросы — это объект базы данных, который служит для манипулирования данными в таблице, например добавление, изменение, удаление, выборка.

Представления - виртуальные таблицы, содержимое которых определяется запросом.

Поля – это основные элементы структуры таблицы. Они имеют свойства. От свойств зависят типы данных, которые можно хранить в поле, а также операции над данными, содержащимися в этом поле.

Хранимая запись - набор связанных полей.

Первичный ключ – это поле, обеспечивающее уникальность данных в текущей таблице. Т.е. данные в поле первичного ключа в каждой записи таблицы уникальны.

Внешний ключ – это поле, которое подставляет данные из индексного поля другой таблицы.

Индекс – структура, созданная для оптимизации поиска данных в таблицах. Он определяется для индексированных столбцов. Он состоит из отсортированных значений индексного столбца или столбцов, содержащих ссылки на соответствующую запись изначальной таблицы или представления. Повышение скорости поиска достигается благодаря сортировке данных.

Предметная область - совокупность предметов, свойств и отношений, представляющих часть реального мира, рассматриваемого в данном контексте.

Атрибут – свойство сущности в предметной области (заголовок столбца таблицы).

Модель данных - это определение объектов, операторов и иных элементов, образующих абстрактную машину доступа к информации, с необходимой пользователю.

ER-модель (Entity-Relationship model) - средство описания предметной области на этапе концептуального проектирования. С её помощью можно отметить главные сущности и показать связи, которые присутствуют между данными сущностями. В момент проектирования баз данных, схемы, созданные на основе ER-модели, преобразуются в определённую схему базы данных на основе заранее определённой модели данных (реляционной, объектной, сетевой или др.). Пример ER-модели представлен на рисунке 1.1)

Рисунок 1.1 - Пример ER-модели



## 1.2 Этапы проектирования БД

Разработка базы данных подразумевает:

• концептуальное проектирование

• логическое проектирование

• физическое проектирование

Что подразумевает создание концептуальной, логической и физической моделей данных соответственно.

### 1.2.1 Концептуальное проектирование

Концептуальная модель БД - это некоторая визуальная диаграмма, созданная из общественно признанных обозначений и полностью описывающая связи объектов между собой и свойств этих объектов. Концептуальная модель строиться для последующего создания БД и преобразования ее, в определённую модель данных. Эта модель визуально показывает связи между объектами данных характеристиками этих объектов.

Данная модель данных проектируется на основе сведений, записанных в спецификациях.

Концептуальное проектирование БД полностью независимо от физических подробностей устройства реализуемой базы данных таких как: СУБД, наборы программ, которые будут работать с данными базы или каким-либо образом их обрабатывать, от языков программирования, типа и архитектуры платформы, на которой предполагается работа системы с БД, и прочего).

Концептуальная модель включает:

* типы сущностей
* типы связей
* атрибуты и домены атрибутов
* первичные и альтернативные ключи
* ограничения целостности

Концептуальное моделирование поддерживается документацией, включая ER-диаграммы и словарь данных.

Пример концептуальной модели представлен на рисунке 1.2.

Рисунок 1.2 - Концептуальная модель



### 1.2.2 Логическое проектирование

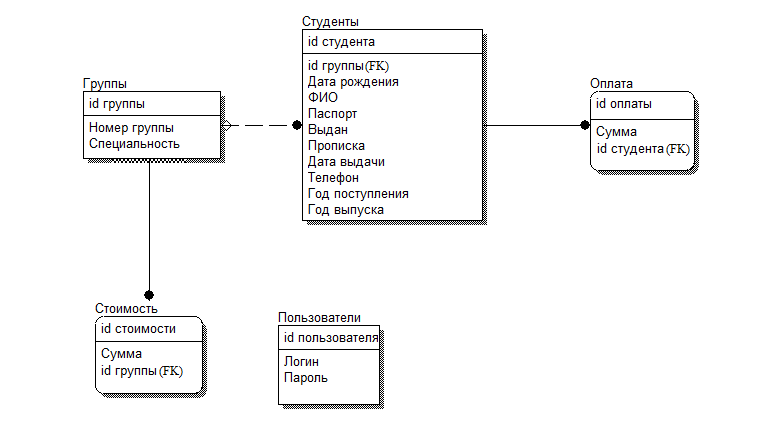
Логическое проектирование БД состоит в проектировании модели сведений, которые будут использоваться в основе заранее определённой модели данных, но данная модель не затрагивает многие аспекты физической реализации (такие как тип СУБД). Концептуальная модель данных, полученная на предшествующем шаге, усложняется и переходит в логическую модель данных. Данная модель должна строиться на основе выбранной модели организации данных в системе управления базами данных, на которой она будет реализована.

На этом шаге должна быть определена конкретная СУБД, которая будет являться целевой. В то же время на этом шаге не учитываются прочие характеристики этой СУБД, это значит, что все особенности реализации структур физического хранения данных в выбранной СУБД и особенности определения индексов.

При физическом проектировании логическая модель является источником сведений для разработчика физической базы данных и даёт ему возможность для поиска компромиссов, необходимых для достижения поставленных задач. Также данная модель помогает на этапе эксплуатации базы данных.

Пример логической модели представлен на рисунке 1.3.

Рисунок 1.3 - Логическая модель



### 1.2.3 Физическое проектирование

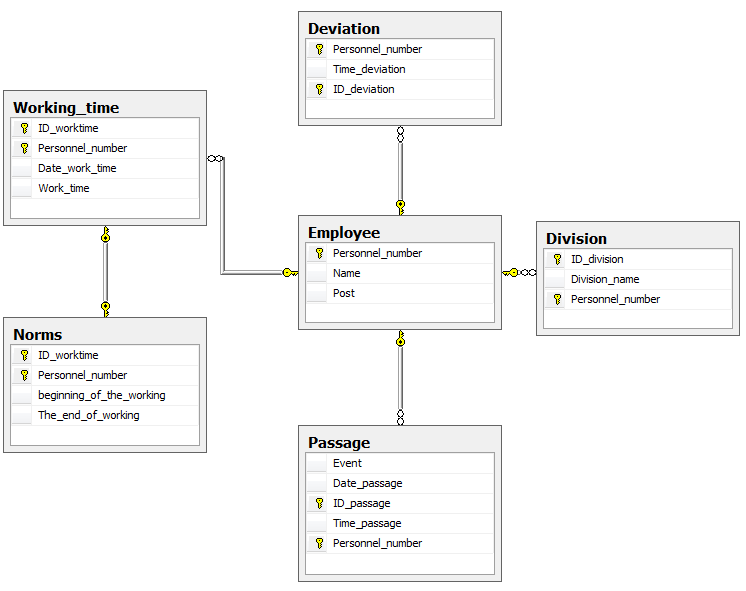
Физическое проектирование БД заключается в создании схемы, которая описывает физическую реализацию БД на внешних устройствах. При физическом проектировании определяются основные отношения и связи, построенные на основе индексов, необходимые для легкого и производительного доступа к данным и поиска данных в таблицах.

Физическое проектирование - это последний шаг проектирования БД, на нём проектировщик решает вопросы реализации создаваемой БД. Так как мы имеем логическую модель, следовательно у нас определена логическая структура БД, которая описывает отношения в принятой предметной области. Начиная физическое проектирование, нужно определить СУБД, которая будет использоваться в качестве основной. Исходя из этого физическое проектирование прямо зависит от выбранной СУБД. Логическое и физическое проектирование тесно связаны и могут как дополнять, так и изменять структуру друг друга.

Основная цель физического проектирования БД - это создание описания того, как следует реализовывать физическую и логическую структуры проекта, в котором используется данная база данных.

Пример физической модели представлен на рисунке 1.4.

Рисунок 1.4 - Физическая модель



## 1.3 СУБД

Система управления базами данных (англ. Database Management System) — специализированное программное обеспечение, предназначенное для манипулирования как самой базой данных и её структурой, так и данными в таблицах.

Состав СУБД:

* Ядро;
* Процессор языка базы данных;
* Подсистема поддержки времени исполнения;
* Сервисные программы;

### 1.3.1 Классификация СУБД

Иерархическая - СУБД, основанная на модели данных, в которой база данных представляется как иерархическая или же древовидная структура, которая состоит из объектов (данных) разных уровней (рис. 1.5).

Сетевая - СУБД, являющаяся дополнением иерархического подхода.

Реляционная СУБД основана на модели данных, при котрой данные хранятся в строках таблицы, занимая поля определённого типа (рис. 1.6).

Объектно-ориентированная - СУБД, которая представляет данные в виде абстрактных объектов, которые наделены свойствами и которые взаимодействуют с другими объектами базы данных посредством внутренних методов.

Объектно-реляционная - реляционная СУБД (РСУБД), которая использует часть технологий, объектно-ориентированных СУБД и использует объектно-ориентированный подход, то есть содержит объекты, а также классы и наследование.

Рисунок 1.5 - Иерархическая СУБД

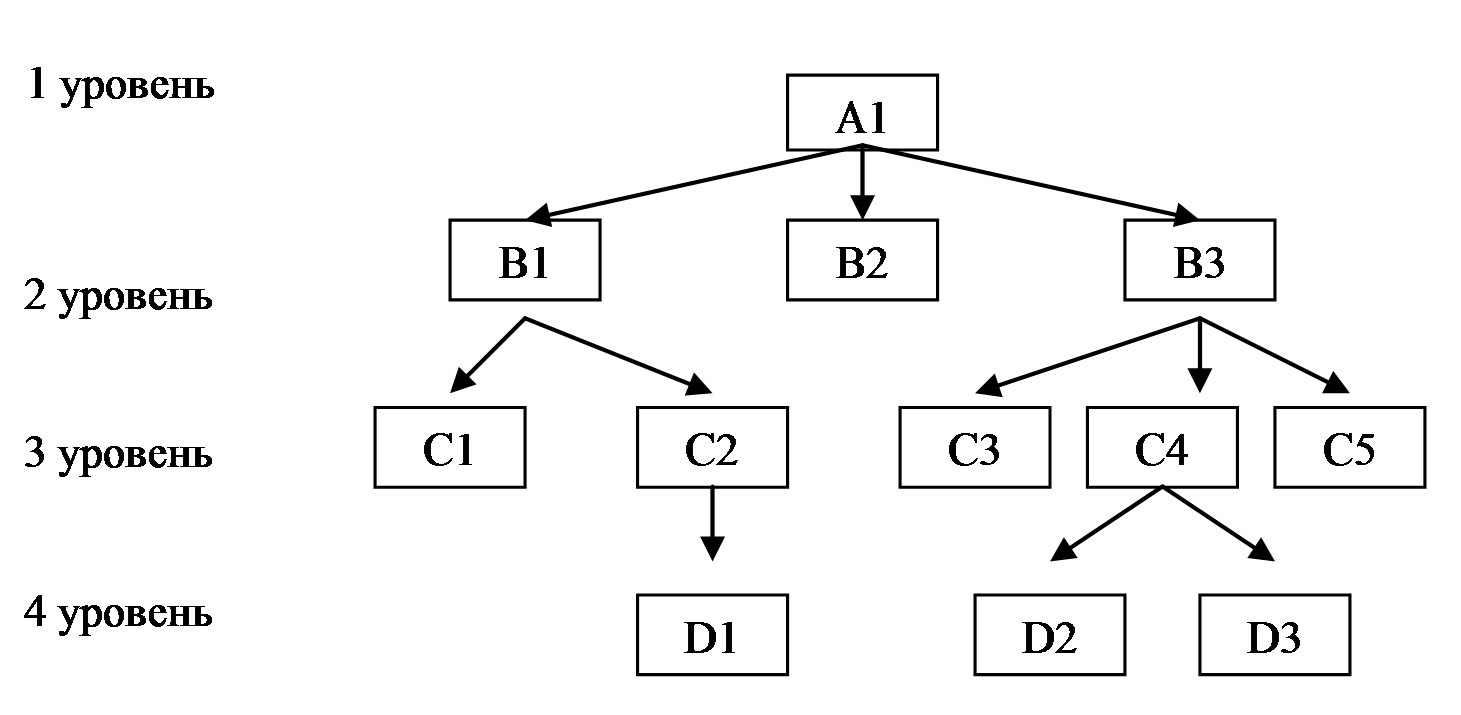
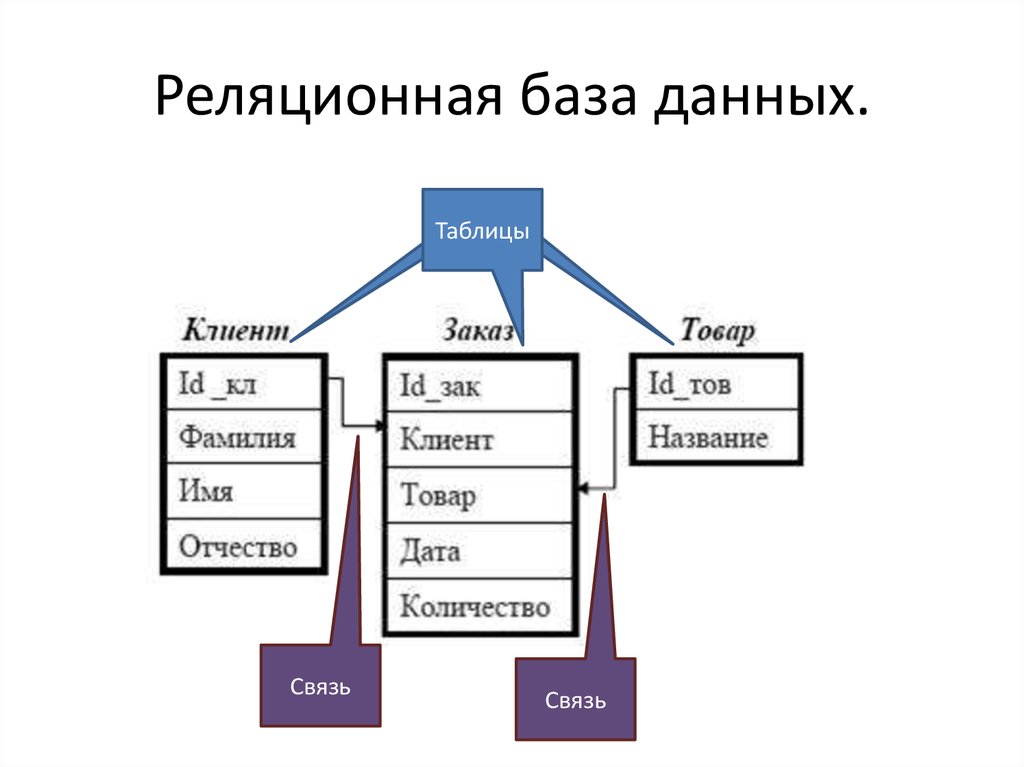


Рисунок 1.6 - Реляционная СУБД



## 1.4 Структура реляционной базы данных

Основной объект любой БД - таблица.

Большинство БД имеют табличную структуру. В данной структуре адрес ячейки определяется пересечением строк и столбцов. В БД столбцы называют полями, а строки, состоящие из полей – записями. Поля задают структуру, а записи - это данные, находящиеся в ней.

Поля базы данных не только создают структуру базы и таблицы – они еще и задают групповые свойства данных, записываемых в ячейки, принадлежащие каждому из полей.

Тип поля – определяет, какие данные могут находиться в поле данного типа.

Размер поля – задаёт максимальное количество символов (данных), которое может вместить данное поле.

Значение по умолчанию – то значение, которое вставляется в ячейку, если не задано альтернативное значение.

Поскольку в разных полях могут содержаться данные разного типа, то и свойства у полей могут различаться в зависимости от типа данных.

## 1.5 Типы данных

В языке запросов SQL существует шесть типов данных, которые определяют стандарт.

Символьные данные - тип, который предназначен для хранения символов. Символы - это буквы алфавита, цифры, пробел, знаки препинания, математические знаки и всё остальное, что мы можем напечатать на клавиатуре. Но не только.

Битовый тип данных хранит значение равное 0 или 1.

Тип точных числовых данных хранит числа, имеющие точное представление.

Тип округленных чисел хранит данные, которые невозможно точно представить в компьютере (действительные числа).

Тип данных "дата/время" хранит момент времени с определённой установленной точностью.

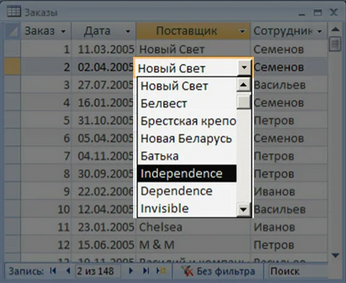
Базы данных Microsoft Access имеют некоторые дополнительные типы данных:

Счетчик – тип данных для создания внешних ключей и определения уникальных идентификаторов численного типа. Имеет автоинкремент (каждая новая запись имеет число в данном поле на 1 больше чем предыдущая).

Гиперссылка – специализированное поле для записи URL адресов. При щелчке на ссылке автоматически запускается браузер и открывается содержащаяся в поле URL-ссылка.

Мастер подстановок – это объект, который автоматически подставляет данные из другого поля другой таблицы и выводит его в виде списка (рис. 1.7).

Рисунок 1.7 - Список мастера подстановок



## 1.6 Предметная область

Предметная область базы данных - часть реального мира, которая рассматривается в границах выбранного контекста.

Объектное ядро предметной области - это группа объектов рассматриваемого мира, относящиеся к данной предметной области.

Сущность предметной области - совокупность однотипных объектов.

Примеры сущностей (с точки зрения ИС): ученик, группа учеников, класс, время занятий, числа. Быть объектом - это означает быть дискретным и различимым. Невозможно классифицировать как объект предметной области мир, время, смысл.

Ситуации - это взаимосвязи, отражающие отношения объектов друг с другом.

Модель предметной области – это проекция наших знаний о данной предметной области. Они могут быть в виде неформальных знаний в голове эксперта, а также они могут быть выражены формально с помощью каких-либо средств ( текстовые описания, наборы должностных инструкций и т.п.) Текстовый способ малоэффективен. Более информативны и полезны специализированные графические нотации.

# II. ПРАКТИЧЕСКАЯ ЧАСТЬ

## 2.1 Предметная область

Во многих учебных заведениях еще принято вести учёт посещаемости обучающихся на бумажных носителях. Данный способ имеет как плюсы так и минусы, но в целом является устаревшим и неудобным, поскольку получение статистических данных занимает достаточно большое время. А в случаях, когда число обучающихся слишком велико эта задача становиться практически невыполнимой.

На замену «бумажному» учёту приходит автоматический учёт. Он способен без лишних временных и физиологических затрат отметить посещаемость студента в данный день и с такой же лёгкостью предоставить статистические данные как по всему учебному заведению, так и по конкретной группе обучающихся.

Предметная область данного курсового проекта - это учёт посещаемости.

Ядро этой предметной области составляют обучающиеся, преподаватели и прочие получатели статистической информации, а также технический персонал, способный вводить данные или редактировать их.

Сущностями данной предметной области будут являться: группы, учащиеся, преподаватели, визиты, месяцы обучения, показатели групповой посещаемости за один день.

Ситуациями являются: отношения студентов с содержащими их группами, отношения визитов и студентов, их производящих, а также процента групповой посещаемости с конкретной группой.

## 2.2 Концептуальное проектирование

При проектировании сложных систем на основе баз данных очень важным элементом является проработка концепции и построение всевозможных текстовых и графических схем. Это необходимо для полного понимания того, над чем предстоит работать в ходе создания базы данных и какие данные будут храниться в базе, а также их структуру и взаимодействия между собой и внешними программами.

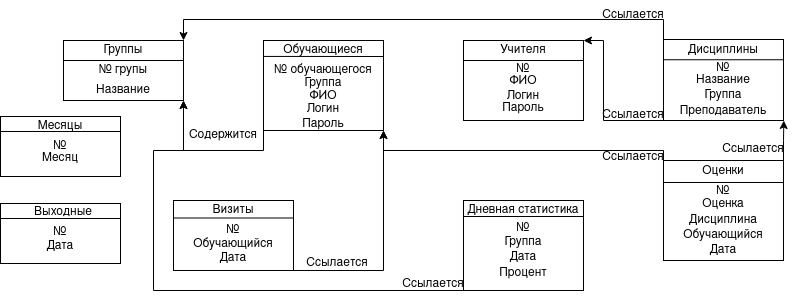
Один из первых этапов в проектировании базы данных является создание концептуальной модели, которая должна отображать общую структуру базы данных, то есть показывать основные сущности (таблицы) и взаимосвязи этих сущностей.

Сущностями для проектной базы данных являются: группы, обучающиеся, учителя, месяцы, дневная групповая статистика, выходные, визиты, дисциплины и оценки.

Взаимосвязи сущностей: идентификатор группы содержится в каждой записи студентов (каждый студент имеет свою группу), в каждой записи дневной статистики группы (каждая запись ссылается на конкретную группу), а также в каждой записи дисциплины (каждая дисциплина относится к конкретной группе), каждый визит содержи идентификатор студента, который произвёл этот визит, каждая запись дисциплины помимо «ссылки» на группу имеет «ссылку» на преподавателя, который этим предметом заведует, а также связь имеют сущности оценка, дисциплина и ученик (оценка отсылается на конкретную дисциплину и конкретного ученика).

Более подробно это можно увидеть на концептуальной модели данных (рис. 2.1).

Рисунок 2.1- Концептуальная модель



## 2.3 Логическое проектирование

При логическом проектировании разрабатывается модель базы данных, основываясь на конкретной системе управления базами данных. Данная модель отражает названия таблиц и столбцов в том виде, в котором они будут находиться в базе данных, также для каждого столбца указывается тип, данных, которые будут в нём храниться и некоторые свойства, например автоинкремент и primary key.

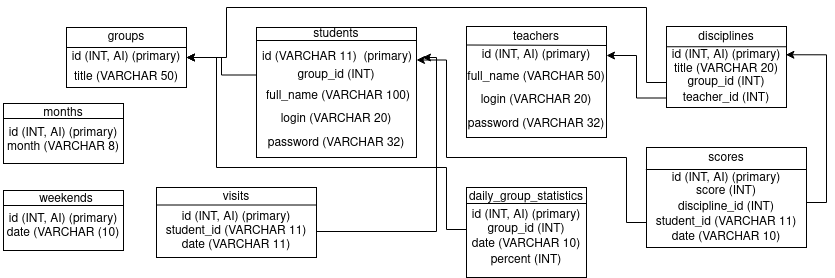
Перед началом проектирования логической модели данных, необходимо определить какая система управления базами данных будет использоваться для создания проекта. При написании этого курсового проекта была выбрана связка MySQL, PhpMyAdmin (для более удобного манипулирования СУБД), php, html, css, JavaScript (в качестве интерфейса для системы). Предполагается, что система будет функционировать на выделенном сервере (локальном или удалённом) и иметь возможность выполнять операции в фоновом режиме, иначе система может работать некорректно.

Логическая модель данных содержит следующие элементы:

* groups (содержит поля: id, которое является ключевым, title),
* students (содержит поля: id (является уникальным и в отличии от всех остальных полей идентификации имеет тип VARCAR, это связано с тем, что идентификатор студента - это идентификатор RFID-метки, которая ему выдаётся), group\_id (имеет тип INT), full\_name, login (хранит логин ученика), password (хранит хэш-сумму от пароля ученика (функция md5))),
* teachers (содержит поля: id, full\_name, login, password),
* visits (содержит поля: id, student\_id (имеет тип VARCHAR, т.к. ссылается на id студента), date (имеет тип VARCHAR длинной 10 символов, т.к. хранит строку, возвращаемую php-функцией date("d/m/Y") (например, «11/11/2021»))),
* daily\_group\_statistics (содержит поля: id, group\_id, date, percent (имеет тип INT и содержит процент посещаемости указанной группы в указанные день)),
* month (содержит поля: id, month ( хранит строку вида: «/11/2021»)),
* weekends (содержит поля: id, date),
* disciplines (содержит поля: id, title (содержит название предмета), group\_id (идентификатор группы, которой принадлежит этот предмет), teacher\_id (преподаватель, заведующий этим предметом)),
* scores (содержит поля: id, score (оценка 2-5), discipline\_id, student\_id, date).

Графический вид логической модели базы данных представлен на рисунке 2.2

Рисунок 2.2 - Логическая модель



## 2.4 Физическое проектирование

Физическое проектирование заключается в непосредственном создании базы данных на ранее выбранной СУБД. Как ранее уже было сказано, в данном проекте используется MySQL+phpMyAdmin.

После создания базы данных необходимо создать таблицу (рис. 2.3, 2.4). Таким образом были созданы все таблицы, описанные в логической модели (рис. 2.5).

Рисунок 2.3 - Создание таблицы

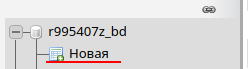


Рисунок 2.4 - Создание таблицы

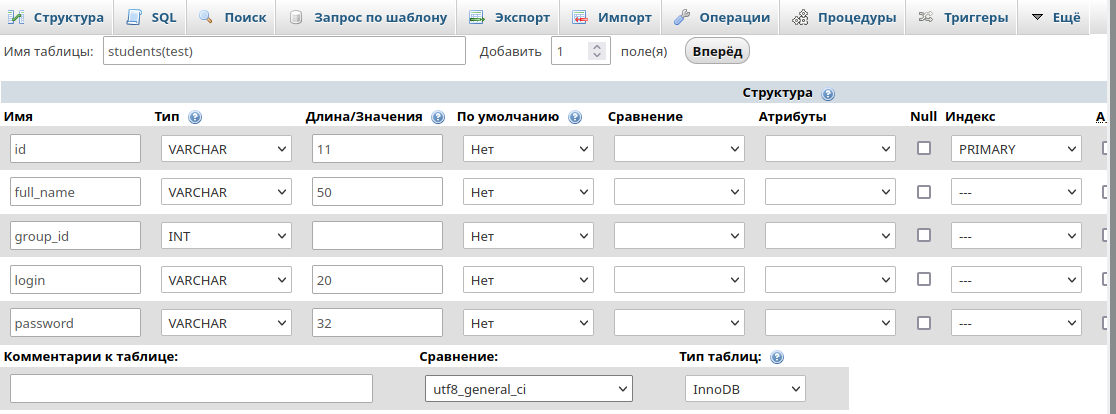
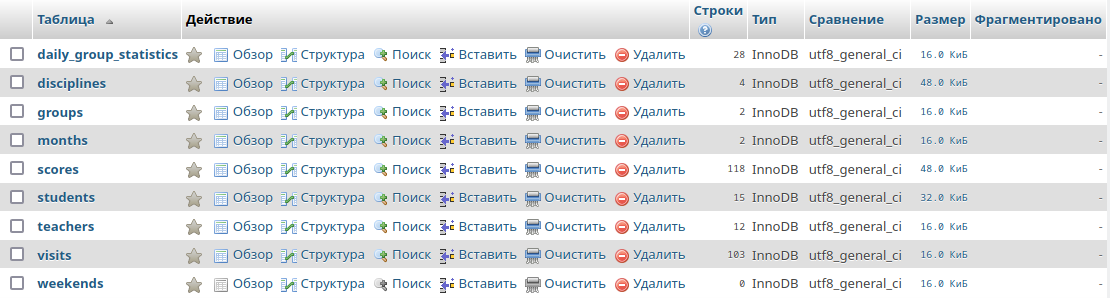


Рисунок 2.5 - Таблицы базы данных



Создавать связи между таблицами можно в разделе «Структура» → «Связи». В данном разделе (рис. 2.6) необходимо указать поле, в которое будет производиться подстановка, а также поле из которого будет производиться подстановка. При вставке (рис. 2.7) в поле, куда производиться подстановка появляется выпадающий список, со значениями из подставляемой таблицы.

Рисунок 2.6 - Добавление связей

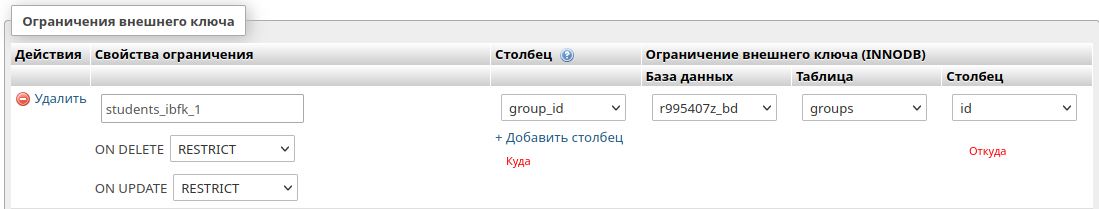
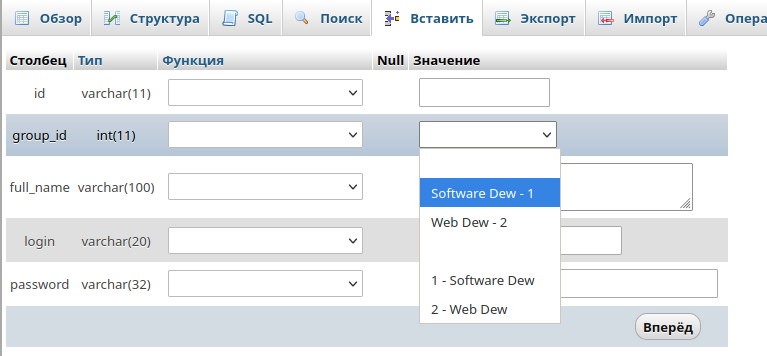
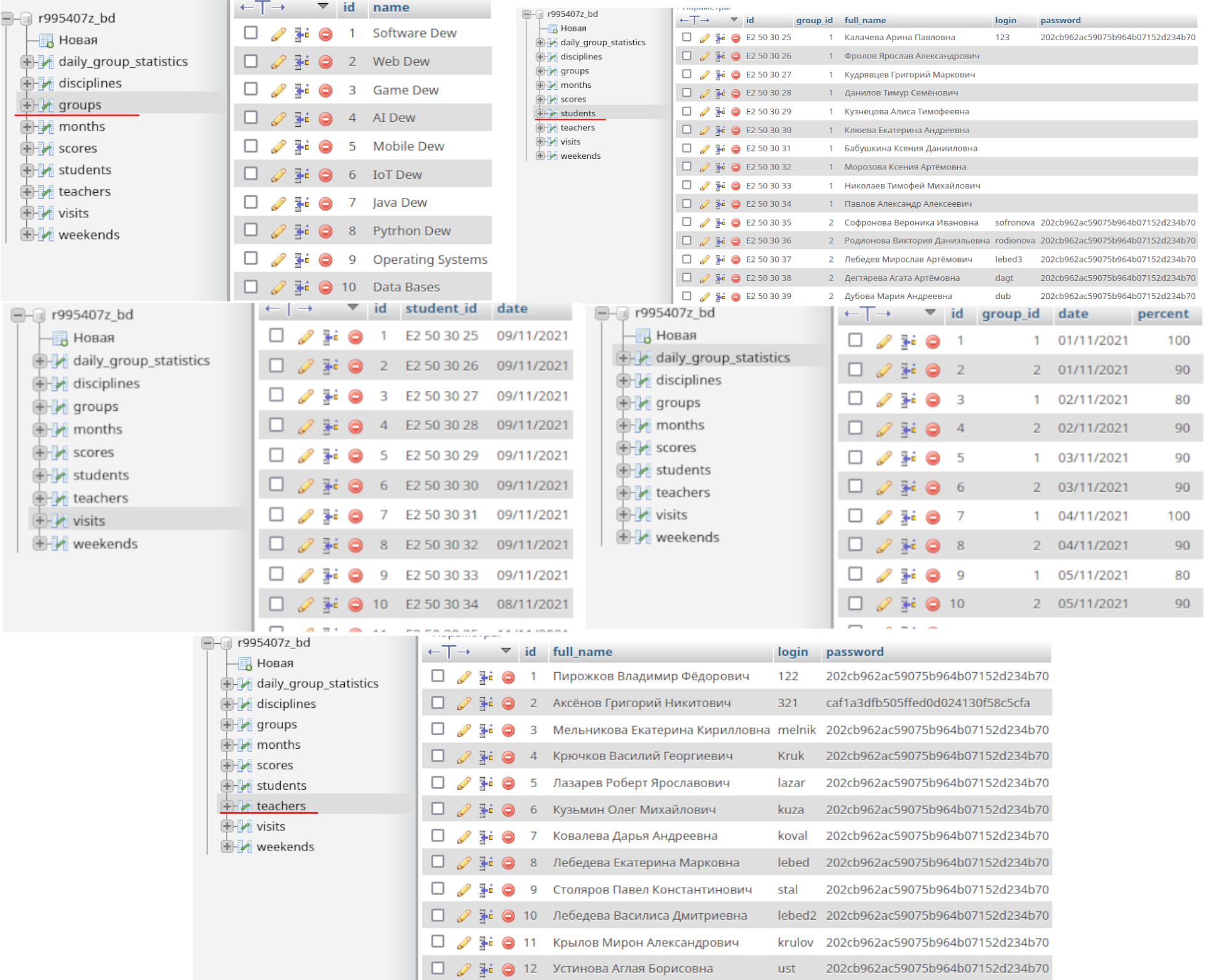


Рисунок 2.7 - Вставка записей



Таблицы были заполнены данными (рис. 2.8).

Рисунок 2.8 - Заполненные таблицы



## 2.5 Интерфейс

### 2.5.1 Структура веб-сайта

Структура веб сайта относительно корня:

* css/
  + style.css
  + main.css
* diary/
  + student/
    - index.php
  + teacher/
    - discipline/
      * index.php
    - index.php
  + index.php
  + login.php
* forgettable/
  + getstudentsfromgroup.php
  + index.php
* img/
* js/
  + scripts.js
* checkreq.php
* footer.php
* header.php
* index.php
* updatedataindb.php
* vars.php

Css содержит файлы каскадных таблиц стилей, описывающих визуальную составляющую сайта. Js содержит файл скриптов, благодаря которому на сайте работают интерактивные элементы. Img содержит картинки, которые используются в интерфейсе.

Diary - это директория электронного дневника, переходя в которую пользователя переносит либо на страницу студента (student), либо на страницу преподавателя (teacher). Forgettable содержит интерфейс для добавления студента, который забыл или потерял RFID-метку. index.php - это главная страница сайта.

Файл vars.php содержит переменные, которые используются на большинстве страниц сайта, например, переменная подключения к базе данных, текущий день и выбранные месяц (фильтр). Header.php содержит верхнюю часть сайта, которая не меняется от страницы к странице, а также подключения файлов стилей. Footer.php содержит подключение js-скриптов и закрывающие теги.

### 2.5.2 Главная

Главная страница (рис. 2.9) состоит из статичной шапки (не меняется для любой страницы сайта) и основного блока с контентом, который содержит интерактивный график посещаемости всего учебного заведения и селектор с выбором месяца, данные которого будут отображаться на графике.

График посещаемости строиться благодаря js-библиотеке charts.js. Для этого на странице расположен блок «<canvas id="grafica"></canvas>» в который помещается график, а также блок со скриптом (рис. 2.10). В этом скрипте создаётся два массива: с датами и процентами посещаемости.

Рисунок 2.9 - Главная страница

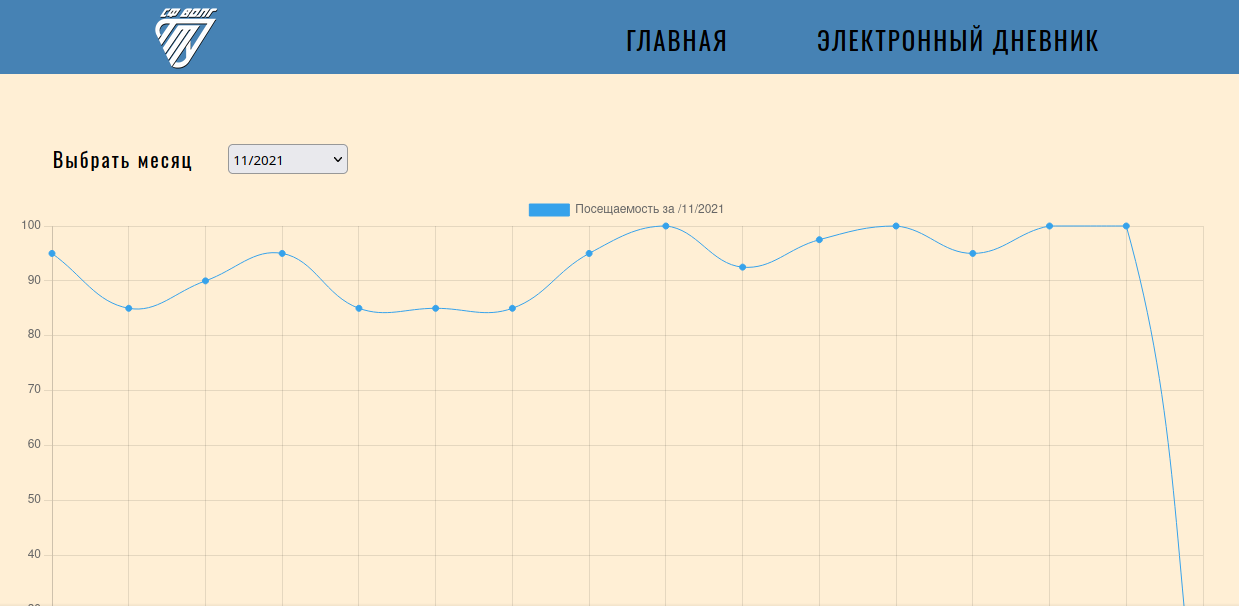


Рисунок 2.10 - Скрипт генерации графика



### 2.5.3 Дневник

Переходя по ссылке на страницу дневника пользователь попадает в index.php, где с помощью этого кода (рис. 2.11) проверяется авторизован ли пользователь.

Если у пользователя открыта сессия под названием autorization, то значит пользователь авторизован и его следует перебросить в соответствии с типом его ученой записи на страницу студента или страницу преподавателя.

Если у пользователя не открыта эта сессия, значит он не авторизован и его следует перебросить на страницу авторизации (login.php).

На странице авторизации повторно проверяется наличие сессии. При ее отсутствии на страницу выводиться форма авторизации (рис. 2.12). После ввода данных они попадают в этот же скрипт, где пароля хэшируется (md5($password)) и сначала данные ищутся в таблице со студентами, а затем в таблице с преподавателями (рис. 2.13).

Рисунок 2.11- Проверка авторизации

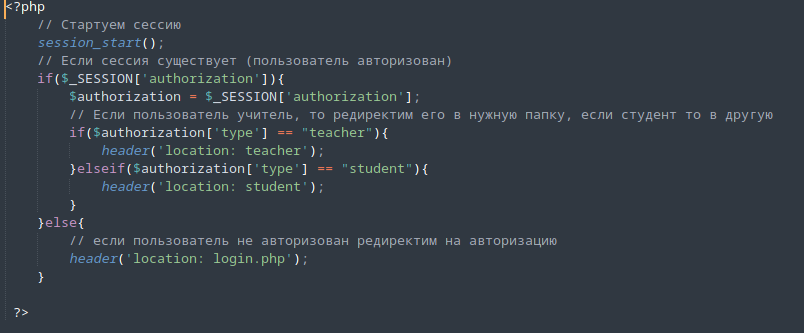


Рисунок 2.12 - Форма авторизации

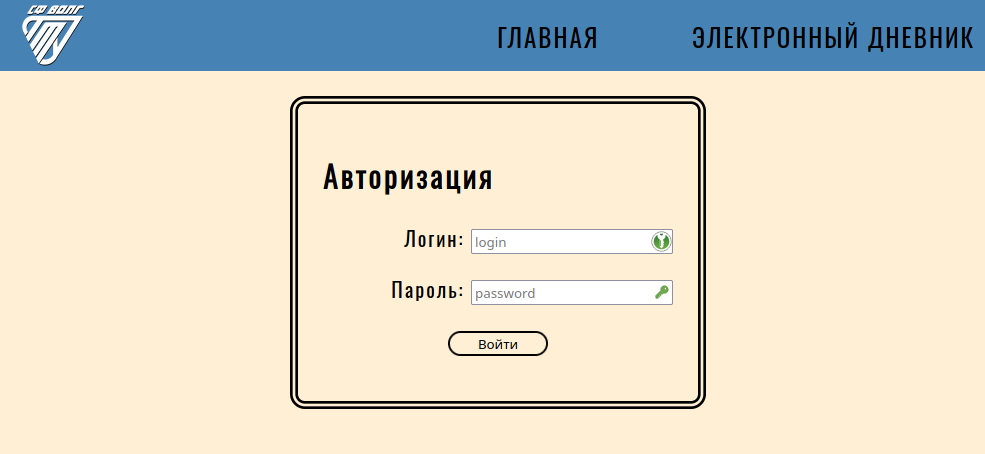
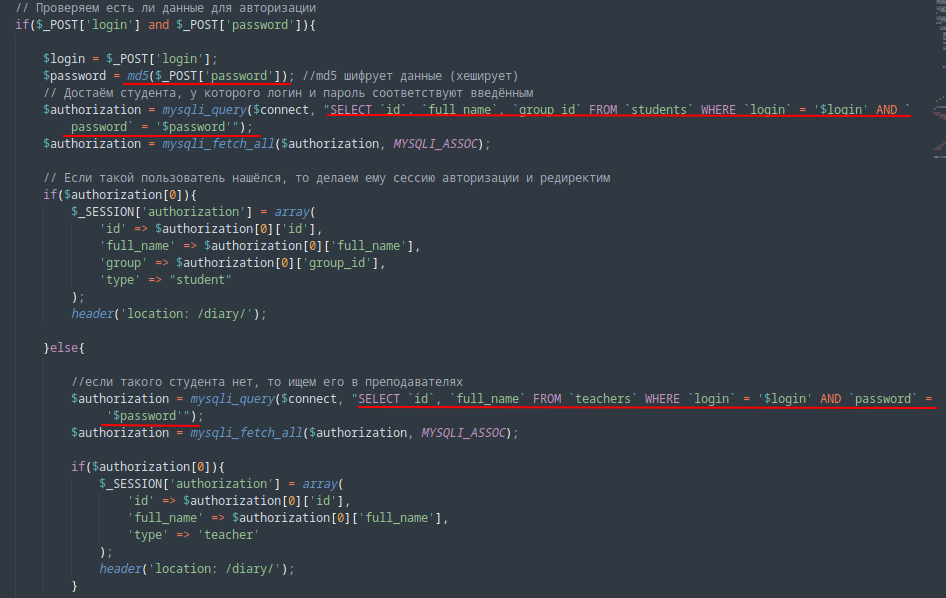


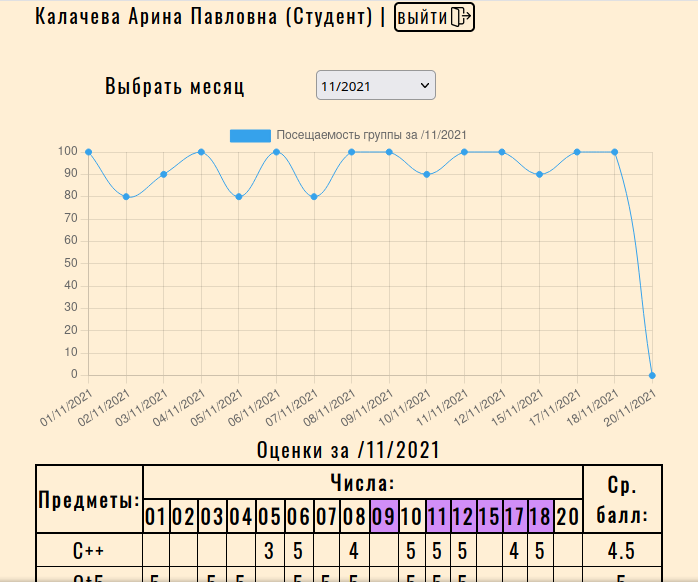
Рисунок 2.13 - Аворизация



### 2.5.4 Студент

Страница студента (рис. 2.14) состоит из графика посещаемости группы, в которой он состоит и таблицы с оценками, в корой ячейки с датами имеют разные цвета (если пользователь в этот день присутствовал (отмечен в визитах), то ячейка имеет фиолетовый цвет).

Рисунок 2.14 - Страница студента

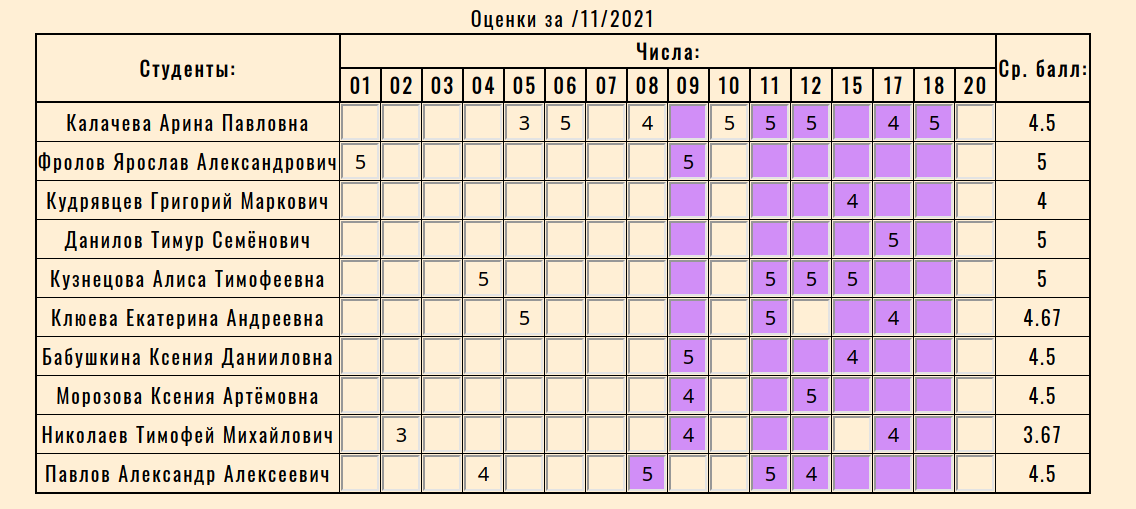


### 2.5.5 Преподаватель

На странице преподавателя выводятся предметы, которыми заведует этот преподаватель.

На странице предмета (рис. 2.15) расположен график посещаемости группы и таблица управления оценками, в которой ячейки окрашиваются в разные цвета, в зависимости от того, был ли этот ученик в этот день.

Рисунок 2.15 - Страница предмета



### 2.5.6 Проверка визита

Запись визитов производиться специальным устройством, которое считывает UID метки, который является id студента, имеющего эту метку. Далее это устройство посылает этот UID файлу checkreq.php, который ищет студента с таким id и если находит, то записывает его в таблицу визитов, отравляя устройству ответ «Ok!», тем самым давая понять, что это этот человек «свой» и его можно пропустить (открыть турникет) (это необязательная, но возможная функция).

Если checkreq.php не нашёл данного студента с таблице, то оно сообщает об этом в виде текста «error».

### 2.5.7 Запись визита

Если учащийся забыл свою метку-пропуск, то есть возможность ввода данных в ручную. Это может делать сторож вахтёр. Для этого существует страница forgettable (рис. 2.16), здесь можно выбрать группу и с помощью AJAX-запроса студенты из этой группы автоматически подгрузятся в элемент <select>. И с помощью той же технологии AJAX идентификатор студента отправляется checkreq.php, аналогично, как это делает специальное устройство.

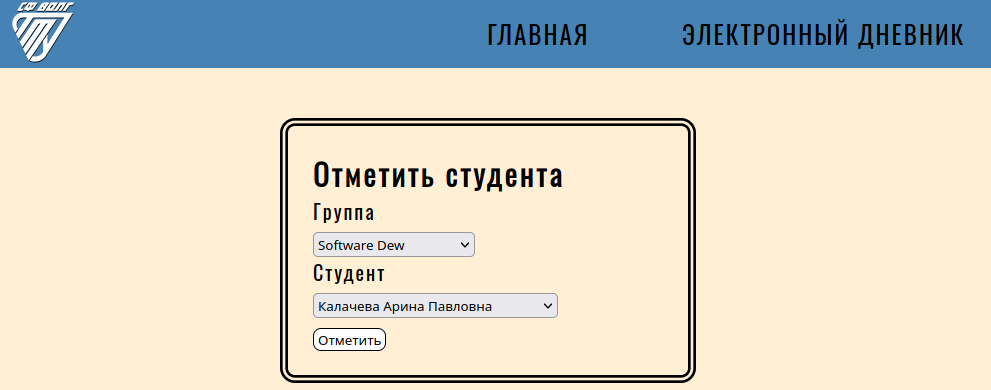


Рисунок 2.16 - Отметить студента

### 2.5.8 Автоматическое обновление данных в базе

В базе данных имеется таблица daily\_group\_statistics, на основе этой таблицы строиться вся логика работы с датами. Если в этой таблице не будет отмечена посещаемость хотя бы одной группы в определённый день, то этот день будет считаться выходным и будет исключён из общей выборки для графиков, а как следствие для таблиц с оценками, т.к. они строятся на датах, полученных при генерации графика. Поэтому очень важно ежедневно добавлять в таблицу ежедневной посещаемости групп новые данные. Этим занимается скрипт updatedataindb.php, который выполняется ежедневно в 23.00 (или другое удобное время) через задачи cron.

Данный скрипт проверяет есть ли в текущий день посещаемость, если она есть, то он проверяет посещаемость для каждой группы и заносит данные в таблицу.

### 2.5.9 Скрипты JavaScript

JavaScript необходим для придания интерактивности странице. В данном проекте js использовался для:

* Смены месяца выборки. При изменении значения элемента с выбором месяца к текущей странице добавляется get-аргумент mont и страница перезагружается.
* Подгрузки студентов для отметки посещаемости. Это реализовано с помощью AJAX.
* Отправки данных о визите студента. Это реализовано с помощью AJAX.
* Манипуляции оценками (добавление, изменение, удаление). Также реализовано через AJAX.
* Добавления границы для активной строки студента

# ЗАКЛЮЧЕНИЕ

Продуктом этого курсового проекта стала система учёта посещаемости, которая упрощает ведение этого учёта, а также получения статистических данных.

В ходе проекта были выполнены задачи:

* разработка моделей базы данных и моделей поведения базы данных,
* описание связей сущностей базы данных,
* непосредственная разработка базы данных,
* создание связей,
* создание интерфейса базы данных, способного обеспечить работу с системой неподготовленного пользователя.

Следующим шагом в развитии проекта станет разработка устройства считывания UID RFID-метки и отправки его на сервер, а также программы для упрощения получения UID, для создания новой записи пользователя. Программа должна будет считать данные из serial и записать их в буфер обмена.

# СПИСОК ЛИТЕРАТУРЫ

1. База данных как модель предметной области [электронный ресурс]. – Режим доступа: https://иванов-ам.рф/informatika\_11\_fgos/informatika\_materialy\_zanytii\_11\_19\_fgos\_02.html – Загл. с экрана (21.11.2021)
2. Базы данных и СУБД [электронный ресурс]. – Режим доступа: https://timeweb.com/ru/community/articles/bazy-dannyh-i-subd-1 – Загл. с экрана (21.11.2021)
3. Концептуальная модель базы данных — диаграмма связи между объектами [электронный ресурс]. – Режим доступа: https://webonto.ru/kontseptualnaya-model-bazyi-dannyih/ – Загл. с экрана (21.11.2021)
4. Основные понятия баз данных [электронный ресурс]. – Режим доступа: https://intuit.ru/studies/courses/3439/681/lecture/14017 – Загл. с экрана (21.11.2021)
5. Проектирование баз данных. Реляционнаямодель данных [электронный ресурс]. – Режим доступа: https://edu.tltsu.ru/sites/sites\_content/site216/html/media67140/lec3-4\_is1.pdf – Загл. с экрана (21.11.2021)
6. Система управления базами данных [электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0\_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F\_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8\_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85 – Загл. с экрана (21.11.2021)