МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет	*Кафедра*
вычислительной техники	«Вычислительная техника»

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»

Профиль подготовки: «Программное обеспечение средств ВТ и автоматизированных систем»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

БАКАЛАВРА

на тему

Система учета пользователей ЛВС ПГУ. Разработка базы данных

Студент Майоров, 03.06.20 (подпись, дата) Майорова Алла Романовна (ФИО полностью)

Руководитель 03.06.20 (подпись, дата) К.В. Попов (фамилия, инициалы)

Нормоконтролёр (подпись, дата) А.В. Кучин (фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 03.06.2020 г. № 12)

Заведующий кафедрой (подпись) М.А. Митрохин (фамилия, инициалы)

Работа защищена с отметкой отлично (протокол заседания ГЭК от 11.06.2020 № 1)

Секретарь ГЭК (подпись) Е.И. Гурин (фамилия, инициалы)

Пенза 2020

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
«30» апреля 2020г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ БАКАЛАВРА

1. Студент Майорова Алла Романовна гр.16ВВ2 факультета вычислительной техники

2. Тема работы: «Система учета пользователей ЛВС ПГУ. Разработка базы данных»

Тема утверждена приказом ПГУ № 0335а/о от "30" апреля 2020г.

3. Руководитель бакалаврской работы к.т.н., доцент кафедры «ВТ» Попов К.В.

4. Задание на работу (назначение разработки, исходные данные и т.п)
Назначение разработки - приложение предназначено для учета пользователей ЛВС ПГУ. Разработать базу данных, связанную с веб-интерфейсом, которая позволит автоматизировать процесс сбора и хранение информации.

5. Перечень подлежащих разработке вопросов.
Требуется разработать базу данных для системы, осуществляющей учет пользователей ЛВС ПГУ. Которая будет хранить в себе данные о строениях, подразделениях, аудиториях и ip-адресах университета. И позволит администратору удалять, добавлять и редактировать необходимую информацию, а также разграничить права доступа для сотрудников университета, использующих систему.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области	10	30.04.2020	
2.	Выбор средств разработки	10	08.05.2020	
3.	Изучение основных характеристик СУБД	10	13.05.2020	
4.	Проектирование базы данных	20	18.05.2020	
5.	Разработка базы данных	30	20.05.2020	
6.	Заполнение тестовой базы данных и разработка SQL-запросов	10	26.05.2020	
7.	Оформление выпускной работы	10	28.05.2020	

Дата выдачи задания "30" апреля 2020 г.

Руководитель бакалаврской работы _____
(подпись, дата)

Задание к исполнению принял студент Иванов, 30.04.2020
(подпись, дата)

Работу к защите допустить
Декан факультета _____
(подпись, дата)

РЕФЕРАТ

Пояснительная записка содержит 77 листов, 30 рисунков, 17 таблиц, 17 источников, 3 приложения.

СУБД MYSQL, PHPMYADMIN, BOOTSTRAP, MAMP, SQL, БАЗА ДАННЫХ, ИНФОРМАЦИОННАЯ СИСТЕМА, SQL-ЗАПРОСЫ.

В рамках данной выпускной квалификационной работы была создана база данных - MYSQL, с использованием phpMyAdmin для ее проектирования.

Объект исследования - база данных компьютерного фонда ПГУ.

Предмет исследования - особенности проектирования базы данных на базе платформы MYSQL.

Цель выпускной квалификационной работы - разработка и внедрение полученной системы учета пользователей ЛВС ПГУ, и реализация этого проекта с использованием базы данных – MYSQL

Задача выпускной квалификационной работы есть разработка базы данных для учета компьютерного фонда университета.

В результате выпускной квалификационной работы было реализовано и протестировано приложение, удовлетворяющее поставленной задаче.

Графическая часть включает в себя: структуру взаимодействия компонентов веб-приложения; UML – диаграммы прав доступа и модель базы данных.

Разработка проводилась посредством использования компонентов локального веб-сервера MAMP. Использовались языки: программирования PHP и JavaScript, структурированных запросов SQL, гипертекстовой разметки HTML, каскадных таблиц стилей CSS. Для хранения данных выбрана СУБД MySQL, для ее разработки и администрирования применялся инструмент phpMyAdmin.

СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 7
ВВЕДЕНИЕ 8
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 10
1.1 Постановка задачи 10
1.2 Анализ существующих разработок в предметной области 11
1.2.1 10-Strike LANState 11
1.2.2 Spiceworks 12
1.2.3 Zabbix 12
1.2.4 Простая система учета оборудования и ip-адресов IPM v2.0.4b 14
1.2.5 IPplan 15
1.3 Архитектура СУБД 16
1.4 Ключевые компоненты СУБД 18
1.5 Основные характеристики СУБД 21
1.5.1 Многоплатформенная реализация 21
1.5.2 Административные утилиты 21
1.5.3 Возможность резервного копирования 21
1.5.4 Обслуживание репликаций 22
1.5.6 Поддержка доступа к данным с помощью сети 22
1.6 Преимущества современных СУБД 22
1.7 Актуальность выбранной темы 24
2 ТРЕБОВАНИЯ К РАЗРАБОТКЕ ПРИЛОЖЕНИЯ 26
2.2 Выбор и обоснование средств разработки 26
2.3 MAMP 26
2.4 PHPMyAdmin 28
2.5 СУБД MySQL 29
2.7 Язык PHP 31
3 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ 32
3.1 Алгоритм работы 32
3.2 Формирование ролей 32
3.3.1 Определение сущностей базы данных 34

3.3.2 Определение уникальных идентификаторов объектов 39
3.3.3 Создание связей между таблицами 40
3.3.4 Проектирование схемы базы данных 43
3.3.5 Нормализация БД 43
3.3.6 Создание таблиц с помощью языка SQL 44
3.4 Добавление данных 45
3.5 SQL – запросы 50
4 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ 52
4.1 Введение 52
4.2 Вредные факторы 52
4.3 Компьютерные синдромы 53
4.4 Влияние компьютера на женщин и детей 54
4.5 Вредные факторы согласно ГОСТ 55
4.6 Действия для снижения опасности 56
4.7 Соблюдение режима труда и отдыха 57
ЗАКЛЮЧЕНИЕ 60
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 61
ПРИЛОЖЕНИЯ 63
Приложение А SQL – код базы данных 63
Приложение Б Модель базы данных 70
Приложение В Презентация 71

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

СУБД - система управления базами данных;

БД - база данных;

PHP - (англ. Personal Home Page Tools) препроцессор гипертекста или инструмент для создания персональных веб-страниц;

SQL - (англ. structured query language) язык структурированных запросов;

ПЭВМ- персональная электронно-вычислительная машина;

RDP- (англ. Remote Desktop Protocol) -протокол удалённого рабочего стола;

IP- (англ. Internet Protocol) - межсетевой протокол;

PK-(англ. Primary Key)-первичный ключ;

FK-(англ. Foreign Key)-внешний ключ.

ВВЕДЕНИЕ

Развитие технологий в области информатизации привело к тому, что в современных организациях, как частных, так и государственных, работает большое количество различных информационных систем.

Образовательные учреждения также используют новейшие технологии для решения повседневных задач, такие как информационные системы университета, предназначенные для обмена информацией между студентами и преподавателями, а также работы с документами, автоматизированные системы управления, например «Абитуриент», работающие с поступающими и взаимодействующие с сайтом организации, ЭИОС-это электронная информационно-образовательная среда, представляющая собой совокупность образовательных ресурсов, а так же множество других систем.

Локальная вычислительная сеть позволяет одновременно использовать ресурсы и подключённые устройства компьютеров, подключенных к одной сети. Именно в совместном доступе заключается ее основная задача. Соответственно учет пользователей ЛВС имеет большое практическое значение.

Разработка носит прикладной характер, предложенная система учета пользователей ЛВС ПГУ помогает решить ряд задач при возникновении трудноразрешимых проблем с сетью или при утечке трафика, кроме этого разработанная и спроектированная база данных позволяет быстро определять, в каком корпусе и аудитории произошла проблема для более быстрого ее устранения.

Основными задачами, решаемыми автором в выпускной квалификационной работе, были:

- охарактеризовать предметную область решаемой задачи;
- провести анализ существующих разработок в предметной области;
- рассмотреть основные функции системы учета пользователей;
- разработать требований к базе данных;

- выбрать среду разработки;
- разработать базы данных для приложения;
- сформировать запросы к базе данных средствами MySQL.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Постановка задачи

Требуется разработать приложение, позволяющее выполнять учет компьютерного фонда университета.

Для реализации приложения должна использоваться структура на основе двух компонентов:

- клиентская часть;
- база данных.

Клиентская часть данного приложения представляет собой графический интерфейс, а также программу для обработки запросов к базе данных. Пользователи могут осуществлять доступ и взаимодействие при помощи браузера, а также различных элементов управления: окон, кнопок, полей ввода, выпадающих списков и другое.

База данных – это организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации преимущественно больших объемов. Она выдается по запросу веб-приложения. Структура взаимодействия компонентов для одного из вариантов использования представлена на рисунке 1.

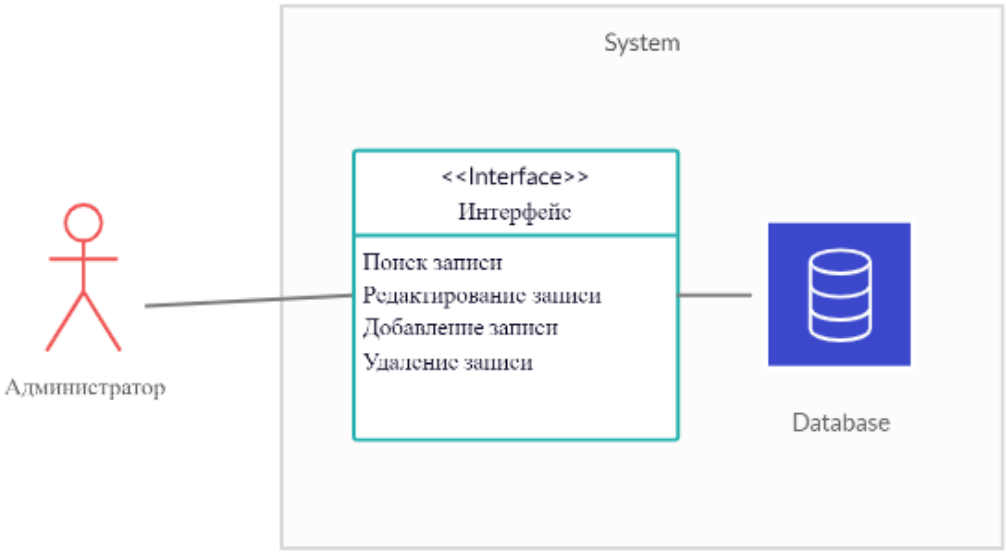


Рисунок 1-Структура взаимодействия компонентов

1.2 Анализ существующих разработок в предметной области

Программы для мониторинга сети являются главными помощниками каждого системного администратора. Они позволяют быть в курсе всех сетевых процессов и оперативно реагировать на аномальную деятельность в пределах локальной вычислительной сети. Таким образом, появляется возможность автоматизировать часть рутинной деятельности: прежде всего той, что связана с обеспечением сетевой безопасности. Также такие программы позволяют узнать месторасположения компьютера и устранить неполадку в кратчайшие сроки. А некоторые даже имеют возможность устранять небольшие проблемы самостоятельно.

1.2.1 10-Strike LANState

"10-Strike LANState" –это программа, предназначенная для администрирования и мониторинга сети.

У нее есть много возможностей, к ним можно отнести:

- Сканирование сети и построение графической карты

Программа осуществляет построение схемы сети. Если коммутаторы управляемые, то программа может считывать с них информацию о подключениях рисовать графическую карту.

- Администрирование хостов

По карте сети удобно администрировать устройства. Так как можно увидеть какие устройства подключены, а какие отключены. Также это позволяет просматривать некоторою информацию на удаленных компьютерах, например, реестр, Event Log, процессы, службы, а также настраивать контекстное меню для вызова частых операций, таких как RDP, ping и других.

- Осуществление мониторинга и получение оповещений об ошибках

Можно также получать различные уведомления, о событиях, происходящих на удаленных компьютерах. Например, можно увидеть: неполадки сети, нехватку места на дисках, остановку служб.

- Получение графиков и статистики

Программа накапливает статистику по результатам проверок мониторинга и можно с легкостью создать отчет и проанализировать его. Таким образом, обнаружив ухудшения параметров (например, рост трафика), можно заранее предотвратить аварию.

1.2.2 Spiceworks

Это удобная и мощная система, позволяющая собрать и структурировать информацию о сети. Используется различными предприятиями для мониторинга и инвентаризации сети.

Она позволяет получить готовую систему управления активами.

Ее достоинствами являются:

- Быстрое сканирование подсетей на наличие любого оборудования с IP-адресом, включая компьютеры Windows и Mac, маршрутизаторы, коммутаторы, принтеры и многое другое.
- Предоставление полученной информации в виде таблиц и ее фильтрация. Также есть возможность сравнения устройств.
- Имеет понятный веб-интерфейс
- Имеет многопользовательский доступ к данным
- Предоставление инструментов для мониторинга и поддержки пользователей

К недостаткам можно отнести только то, что ввиду использования веб-интерфейса есть некоторые ограничения на визуализацию процессов, но это компенсируется всеми достоинствами программы.

1.2.3 Zabbix

Zabbix –это универсальная система мониторинга. Основными функциями которой являются:

- Сбор данных;
- Автоматическое обнаружение проблем. Данная система анализирует собранные данные, выделяя причины проблем. Она имеет

насколько уровней критичности, а также позволяет получать оповещения о проблемах высокой важности по СМС и e-mail;

- Визуализация. Программа имеет несколько вариантов представления веб-интерфейса для обзора IT-инфраструктуры;
- Безопасность и аутентификация. Позволяет защищать данные на всех уровнях благодаря надежному шифрованию между компонентами и нескольким уровням аутентификации;
- Простота внедрения. Эта система позволяет создавать пользовательские шаблоны, что позволяет эффективно управлять тысячами устройств. Также возможно осуществлять экспорт и импорт узлов сети. Данные экспортируются в формате XML, который легко читать и модифицировать;
- Автоматизация. Осуществление мониторинга больших инфраструктур может стать ночным кошмаром для администратора без использования автоматизации. В Zabbix есть 3 подхода для автоматического обнаружения и управления элементами инфраструктуры сетевое обнаружение, низкоуровневое обнаружение и авторегистрация агентов, каждый подход покрывает свое поле деятельности;
- Выполнение команд удаленно через брандмауэр;
- Мониторинг состояния веб-сайта с помощью автоматического выполнения сценария;

Zabbix довольно мощная система, она имеет множество функций, которые упрощают наблюдение за сетью. Система написана на C и PHP. Веб-интерфейс Zabbix написан на PHP. Для хранения данных используются различные СУБД, такие как MySQL, PostgreSQL и множество других;

В настоящее время эта система используется Пензенским государственным университетом для мониторинга сети;

1.2.4 Простая система учета оборудования и ip-адресов IPM v2.0.4b

Система является свободной и предоставляется бесплатно под лицензией AGPLv3. Она представляет собой пакет модулей к популярной системе учета заявок OTRS.

Пакет помогает администраторам локальных сетей и IT-служб. Предназначен для ведения учета оборудования и IP-адресов в локальных сетях.

Позволяет по серийному или инвентарному номеру, MAC-адресу устройства и доступным полям фильтра найти IP адрес оборудования и наоборот. Эта функция нужна, если оборудование находится на складе или на удаленном объекте и нужно, например, по инвентарному или серийному номеру узнать его ip-адрес.

Визуальные индикаторы состояния ip-адреса опция упрощают обслуживание адресного пространства.

Встроенные функции экспорта/импорта позволяют данные из таблиц Excel в систему учета IPM.

Пакет IPM v2.0.4b включает:

- Систему учета оборудования и IP-адресов;

- MAC Address Vendor Lookup;

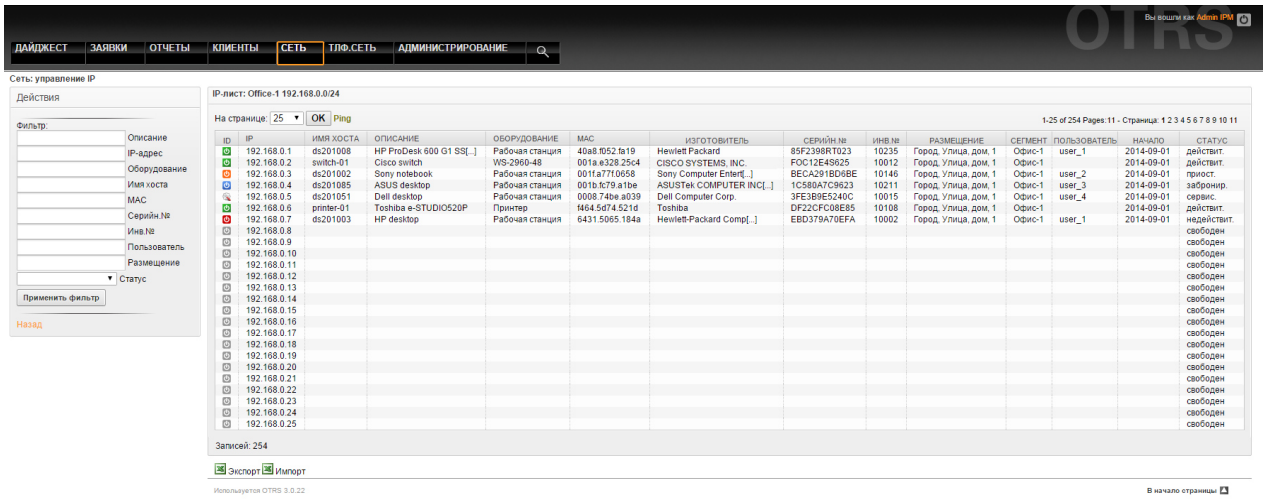


Рисунок 2-Пример системы

1.2.5 IPplan

IPplan – это доступное, веб-многоязычное программное обеспечение для управления IP-адресами и инструмент отслеживания, написанный на php4, упрощающий администрирование пространства IP-адресов.

IPplan выходит за рамки управления адресами TCP/IP, он также включает в себя возможность администрирования DNS, управление файлами конфигурации, управление расположением и хранение информации об оборудовании, которое настраивается дополнительно.

Эта система может обслуживать одну сеть или обслуживать несколько сетей и клиентов. Она делает управление IP-адресами простыми и легкими.

IPplan работает со многими базами данных и был протестирован с MySQL, Postgress и Oracle. Также известно, что MSSQL работает так же, как и большинство других баз данных, хотя они не были напрямую протестированы на IPplan. Система может работать на любой операционной системе, которая поддерживает язык PHP и одну из вышеуказанных баз данных.

Основными функциями и достоинствами IPplan являются:

- поддержка IPv6
- интернационализация
- импорт определений сети из таблиц маршрутизации
- импорт определений из TAB файлов с разделителями в XML - формат
- несколько администраторов с разными профилями доступа (для каждой группы, разрешающей доступ для каждого клиента, для каждой сети.)
- определение границы полномочий адресного пространства для каждой группы
- поиск свободное адресное пространство в диапазоне
- разделение и объединение сетей, чтобы сделать их все меньше и меньше (определенные ip остаются неизменными)
- отображение перекрывающегося адресного пространство между сетями

- возможности поиска
- журнал аудита
- статистика
- администрирование DNS (прямая и обратная зоны, импорт существующих зон с помощью передачи зон)
- система шаблонов для расширения IPplan, содержащая специфичную для сайта информацию, например, данные схемы, данные конфигурации хоста, и другое.
- управление файлами конфигурации устройства
- позволяет связывать адреса вместе через указатели
- имеет внешнюю таблицу стилей для изменения внешнего вида дисплея
- наличие триггеров - каждое пользовательское событие может вызывать пользовательскую функцию
- external poller - проверка подсети на наличие активных адресов для сбора статистики использования
- система запроса IP-адресов - позволяет пользователям запрашивать статические IP-адреса из базы данных.

Доступны два метода аутентификации - либо собственная схема внутренней аутентификации IPplan, либо в качестве альтернативы использование любого внешнего модуля аутентификации Apache. Такие как системы единого входа, например, SiteMinder или собственные схемы, основанные на LDAP, или любые другие системы, совместимые с Apache.

1.3 Архитектура СУБД

Системные модели определяют:

- Архитектуру информационной системы
- Функции и роли компонентов
- Правила взаимодействия компонентов
- Состав оборудования

- Программные средства для реализации и функционирования системы

К программным средствам относятся: программы общего назначения, системы управления базами данных, специальные пакеты программ, специальные программные компоненты.

Можно выделить три роли программных компонентов:

1. Клиент-реализует взаимодействие с человеком
2. Сервер данных-обеспечивает хранение необходимой информации и доступ к ней
3. Функциональный модуль-реализует основные функции системы

Можно выделить 4 группы функций приложений, использующих БД в своем составе:

- Функции ввода и отображения данных
- Прикладные функции
- Служебные функции
- Фундаментальные функции управления ресурсами и их хранение

Модель доступа к удаленным ресурсам представлена на рисунке 2.

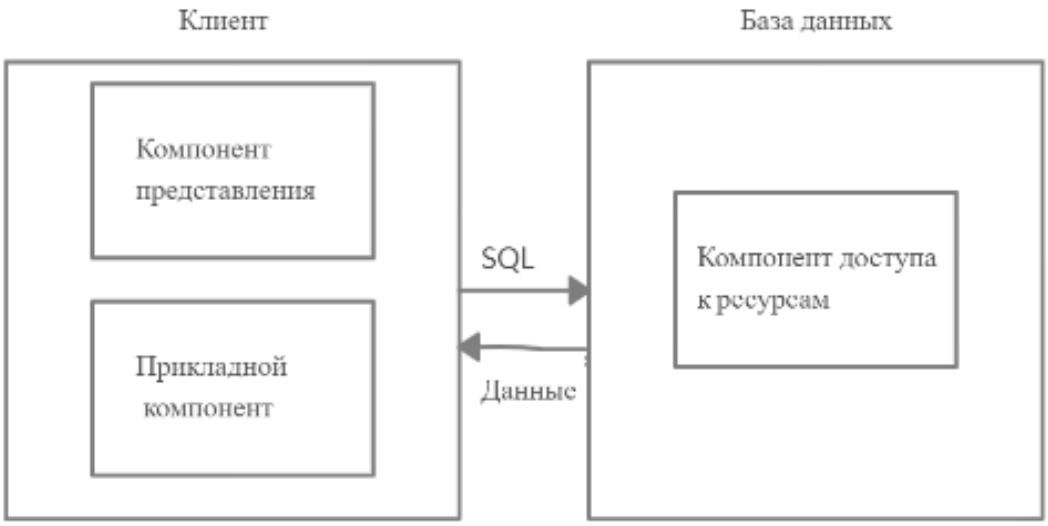


Рисунок 3-Модель доступа к удаленным данным

1.4 Ключевые компоненты СУБД

Система управления базами данных относится к технологическому решению, используемому для оптимизации и управления хранением и извлечением данных из баз.

СУБД предлагает системный подход к управлению базами данных через интерфейс для пользователей, а также к рабочим нагрузкам, обращающимся к базам данных через приложения. Обязанности по управлению СУБД включают информацию в базах данных, процессы, применяемые к базам данных (такие как доступ и изменение), и логическую структуру базы данных. СУБД также облегчает, помимо прочего, дополнительные административные операции, такие как управление изменениями, аварийное восстановление, соответствие требованиям и мониторинг производительности.

Для облегчения этих функций СУБД имеет следующие ключевые компоненты:

- Программное обеспечение

СУБД - это прежде всего программная система, которая содержит в себе консоль или графический интерфейс для взаимодействия с базами данных. Интерфейс также распространяется по физическим системам реального мира, которые вносят данные в серверные базы данных. Операционная система, сетевое программное обеспечение и аппаратная инфраструктура участвуют в создании, доступе, управлении и обработке баз данных.

- Данные

СУБД содержит данные, доступ к записям базы данных и метаданным в качестве ресурса для выполнения необходимых функций. Данные включают в себя файлы, разного типа, административную информацию, словари данных, используемые для представления потоков данных и связей с другими записями или объектами.

- Процедуры

Процедуры не являются частью программного обеспечения СУБД, но они могут рассматриваться как инструкции по использованию СУБД. Они существуют чтобы помочь пользователю и разработчику в проектировании, изменении, удалении данных, а также обработки самой базы.

- Языки баз данных

Языки являются компонентами, для доступа, изменения, хранения и извлечения элементов из баз данных.

Типы языков СУБД включают:

1. группу операторов определения данных (DDL)-это такие операторы, как CREATE, ALTER, DROP

2. язык для манипуляции данными (DML)-это самые распространенные операторы языка SQL: SELECT, INSERT, UPDATE, DELETE.

3. язык доступа данными (DCL). Определяется: GRANT, REVOKE, DENY

4. группу операндов для управления транзакциями (TCL). Определяет действия над транзакциями, например, COMMIT TRANSACTION

- Обработчик запросов.

В качестве основного компонента СУБД процессор запросов выступает в качестве посредника между пользователями и механизмом обработки данных СУБД для передачи запросов. Когда пользователи вводят инструкцию на языке SQL, команда выполняется от языка высокого уровня до языка низкого уровня, который базовый компьютер может понять и обработать для выполнения соответствующих функций СУБД. В дополнение к разбору и переводу инструкций обработчик запросов также оптимизирует запросы для обеспечения быстрой обработки и точных результатов.

- Менеджер времени выполнения.

Компонент централизованного управления СУБД, который обрабатывает функциональность, связанную с данными времени выполнения, которая обычно используется для доступа к базам данных на основе контекста.

Этот компонент:

1 проверяет авторизацию пользователя для запроса;

2 обрабатывает утвержденные запросы;

3 разрабатывает оптимальную стратегию выполнения запроса;

4 поддерживает параллелизм, позволяющий нескольким пользователям одновременно работать с одними и теми же базами данных;

5 обеспечивает целостность данных, записанных в базы данных.

- Менеджер баз данных.

Менеджер базы данных предназначен для выполнения функций СУБД, связанных с данными в базах, а также позволяет использовать набор команд для выполнения различных операций СУБД, включая создание, удаление, резервное копирование, восстановление, клонирование и другие задачи обслуживания базы данных. Менеджер базы данных также может быть использован для обновления базы данных с помощью исправлений от поставщиков.

- Движки баз данных

Это основной программный компонент в решении СУБД, который выполняет основные функции, связанные с хранением и извлечением данных. Существуют самые разные движки баз данных, например, MyISAM, InnoDB, Memory, Archive, CSV, DBD, Federated, Blackhole.

Ядро базы данных также доступно через API, которые позволяют пользователям или приложениям создавать, читать, записывать и удалять записи в базах данных.

- Создание отчетов

Генератор отчетов извлекает полезную информацию из файлов СУБД и отображает ее в структурированном формате. Эта информация используется для дальнейшего анализа и принятия взвешенных решений.

Наиболее интуитивно понятный интерфейс для этой цели имеет программный продукт Access, в большинстве случаев используется именно он.

1.5 Основные характеристики СУБД

1.5.1 Многоплатформенная реализация

Большинство серверов баз данных в настоящее время могут существовать в нескольких версиях для различных платформ, например, Windows NT Server, а также UNIX — Solaris, HP/UX и другие.

Также существуют однопользовательские серверы СУБД, они обычно используются в так называемых «переносных» системах. Например, для работы с базами данных отдельно от центрального офиса компании с последующей репликацией данных или же для отстранения разработчика от работающей в настоящее время версии для внесения изменений или переопределения.

В настоящее время многие производители серверных СУБД выпускают также версии для Linux.

Но некоторые компании так и не позволяют Linux, как и некоторым другим ОС использовать свои продукты. Таковыми являются Microsoft SQL Server.

1.5.2 Административные утилиты

Административные утилиты предпочитают опытные профессионалы, которые занимаются администрированием. Они используют их как замену консольному приложению.

Большинство современных СУБД поставляется уже со встроенными или комплектными утилитами.

Административные утилиты являются чуть ли не самым важным пунктом при выборе СУБД.

1.5.3 Возможность резервного копирования

Резервное копирование данных и журналов транзакций поддерживается всеми без исключения коммерческими СУБД. Разница заключается лишь в том, есть ли возможность производить резервное копирование в процессе работы пользователей. Если такая возможность присутствует, то какие пользовательские операции в это время нельзя выполнять. А также некоторые

СУБД могут уже содержать различные специальные утилиты для использования различных внешних устройств в своем комплекте постаки.

1.5.4 Обслуживание репликаций

Репликация подразумевает под собой копирование информации из одной базы данных в другие. Она в основном используется для разделения нагрузки между серверами в сети и для перемещения поднаборов данных на вспомогательные серверы, а также для синхронизации данных на нескольких серверах и многих других целей.

Репликации в настоящее время поддерживаются всеми современными серверными СУБД. Разница лишь в поддержке тех или иных конкретных сценариев репликаций. Исключением является только IB Database, в последней представленной версии она не поддерживает репликации. Если есть необходимость иметь более одной базы данных, то стоит обратить внимание на то, какие сценарии репликаций поддерживает выбранная СУБД.

1.5.6 Поддержка доступа к данным с помощью сети

Без поддержки публикации данных в сети или получения данных от удаленных Internet-клиентов сегодня не обходится ни одна СУБД. Производители СУБД в большинстве своем сейчас поддерживают Web-технологии. Эта поддержка осуществляется с помощью Web-серверов собственного производства, либо посредством создания расширений для существующих Web-серверов. А также путем включения в комплект поставки утилит, генерирующих Web-страницы согласно определенному расписанию.

1.6 Преимущества современных СУБД

СУБД была разработана для решения фундаментальных проблем, связанных с хранением, управлением, доступом, защитой и аудитом данных в традиционных файловых системах.

В устаревших базах данных были четко определены проблемы избыточности данных, изоляции, ограничения целостности и трудностей

управления доступом к данным Данные проблемы появились из-за разработки поверх баз.

Для решения данных проблем необходим уровень абстракции между пользователями или приложениями и базами данных на физическом и логическом уровне.

Внедрение программного обеспечения СУБД для управления базами данных дает следующие преимущества:

- Безопасность данных.

СУБД позволяет организациям применять политики, обеспечивающие соответствие и безопасность. Базы данных доступны для соответствующих пользователей в соответствии с политикой организации. Система СУБД также отвечает за поддержание оптимальной производительности операций запросов при обеспечении достоверности, безопасности и согласованности элементов данных, обновляемых в базе данных.

- Обмен данными

Быстрое и эффективное сотрудничество между пользователями.

- Доступ к данным и аудит.

Контролируемый доступ к базам данных. Регистрация связанных действий доступа позволяет организациям проводить аудит на предмет безопасности и соответствия требованиям.

- Интеграция данных.

Вместо того, чтобы работать с островными ресурсами баз данных, используется единый интерфейс для управления базами данных с логическими и физическими взаимосвязями.

- Абстракция и независимость.

Абстракция и независимость указывают на то, что физическая схема может изменяться организациями без внесения изменений в логические схемы, которая управляет связями в базах данных. В результате организации могут модернизировать хранилище и масштабировать инфраструктуру, не влияя на работу базы данных. Аналогичным образом, изменения в логической схеме

могут применяться без изменения приложений и служб, которые обращаются к базам данных.

- Единое управление и администрирование.

Единый интерфейс, который представляется в виде консоли для выполнения основных административных задач облегчает работу администраторам и пользователям. Также практически у всех БД существует возможность осуществить привязку к графическому интерфейсу, используя нужные утилиты.

Для бизнес-организаций, образовательных и медицинских учреждений управляемых данными, СУБД превращается в чрезвычайно сложные технологические решения, для которых требуются выделенные ресурсы и собственный опыт.

Размер, стоимость и производительность СУБД зависят от архитектуры системы и вариантов использования, и поэтому нужно оценивать СУБД соответствующим образом. Кроме того, сбой СУБД может привести к значительным убыткам для организаций, которые не в состоянии поддерживать оптимальную функциональность системы СУБД.

1.7 Актуальность выбранной темы

Исходя из анализа уже имеющихся решений, мы видим, что сейчас существует не так много приложений, предназначенных для учета пользователей и компьютеров в организации, а также для обзора их месторасположения. Рассмотренные выше приложения являются коммерческим программным обеспечением.

Существующие системы при внедрении в образовательное учреждение требуют серьезной доработки, так как они имеют достаточно широкий функционал, в связи с чем многие модули приложения использоваться не будут. Лицензия на некоторые из вышеперечисленных приложений является дорогостоящей, и переплата государственного учреждения за неиспользуемые функции является необоснованной.

Таким образом, выбрано решение упростить разработку приложения учета пользователей ЛВС. Оно будет дешевле, а также будет иметь лишь те задачи, которые необходимы в работе.

2 ТРЕБОВАНИЯ К РАЗРАБОТКЕ ПРИЛОЖЕНИЯ

Приложение для учета пользователей ЛВС ПГУ представляет собой приложение, предназначенное для использования внутри университета его сотрудниками. Оно использует базу данных с доступом через веб-интерфейс.

Пользователями данного приложения являются:

- администраторы
- операторы;

2.2 Выбор и обоснование средств разработки

В ходе разработки были проанализированы различные средства и технологии. Программные средства, использованные в данной работе:

- локальный сервер MAMP для запуска веб-интерфейса;
- СУБД MySQL;
- Веб-интерфейс PhpMyAdmin для администрирования СУБД MySql.

Использованный язык программирования:

SQL для работы с базой данных, PHP(PDO) для создания и добавления данных;

С помощью перечисленных функциональных средств можно создавать базы данных. Выбор средств разработки исходил из:

- функциональных возможностей;
- высокой производительности;
- бесплатного использования.

2.3 MAMP

MAMP - это бесплатная локальная серверная среда, которая может быть установлена под macOS и Windows очень просто. MAMP предоставляет все инструменты, необходимые для запуска WordPress на настольном ПК, например, для тестирования или разработки. Можно также легко протестировать свои проекты на мобильных устройствах.

Среди других популярных локальных веб-серверов можно выделить: AppServ, Open Server, Denwer. Их главный недостаток – отсутствие удобного преднастроенного программного комплекса.

AppServ это отличный сервер от тайского производителя, концепцией которого является легкая установка и настройка всех дистрибутивов за 1 минуту. AppServ очень прост в установке, он работает стабильно. Apache и MySQL устанавливаются как сервисы. Но так как нет никакой управляющей программы, то своей сути AppServ является графическим установщиком Apache, PHP, MySQL. Здесь компоненты просто устанавливаются в качестве сервисов и управляются самой Windows.

Open Server – это портативная серверная платформа и программная среда, созданная специально для веб-разработчиков с учётом их рекомендаций и пожеланий. Она имеет полный набор серверного программного обеспечения, удобный и многофункциональный интерфейс, обладает мощными возможностями по администрированию и настройке компонентов. Платформа широко используется с целью разработки, отладки и тестирования веб-проектов, а также для предоставления веб-сервисов в локальных сетях.

Denwer – один из самых популярных веб-серверов. Его комплектация: Apache, PHP и СУБД MySQL настраивается самостоятельно пользователем. Можно после установки этого веб-сервера изменить конфигурации и версии сервисов, которые входят в комплект.

MAMP включает различные версии PHP и систем управления базами данных, а так же есть возможность быстрого переключения между ними. Включенные СУБД модули: MySQL, PostgreSQL, MariaDB, MongoDB. В состав программного комплекса входят и различные веб-интерфейсы администрирования: PHPMyAdmin, Adminer, PHPRedisAdmin, PHPPgAdmin, PHPMemcachedAdmin и другие компоненты.

Все необходимые компоненты можно выбрать самостоятельно на усмотрение пользователя. Преимуществом использования MAMP является:

- дружественный интерфейс

- многофункциональность;
- быстрый переход между базами и веб-интерфейсом;
- мощные возможности проектирования;
- гибкие настройки;
- поддержка;
- постоянное развитие;
- выпуск новых версий программы.

2.4 PHPMyAdmin

PHPMyAdmin – это приложение, представляющее собой веб-интерфейс для администрирования СУБД MySQL. Оно позволяет с использованием веб-браузера осуществлять администрирование базы данных, выполнять SQL-запросы, просматривать содержимое таблиц и баз данных. Это приложение имеет интуитивно понятный и удобный интерфейс, поддерживает различные функции MySQL, а также является одним из самых популярных веб-интерфейсов. PHPMyAdmin обеспечивает полноценную, в том числе удаленную, работу с базой данных MySQL посредством браузера. PHPMyAdmin служит для разработки структуры БД и отладки запросов в веб-приложении.

Самые популярные действия с базами данных – это их импорт и экспорт, а также выполнение различных SQL – запросов; создание, удаление и редактирование записей – все это позволяет осуществлять работу с данным веб-интерфейсом.

Таким образом, с помощью PHPMyAdmin возможно:

- Создание и корректирование БД, таблиц и записей;
- создание пользователей;
- исполнение SQL – команд;
- поиск по БД.

Исходя из вышесказанного был выбран для разработки данный инструмент, так как является самым удобным для создания таблиц и написания запросов.

2.5 СУБД MySQL

SQL – язык структурированных запросов. Он применяется для создания, модификации и управления информацией в базе данных, которая в свою очередь управляется соответствующей СУБД.

MySQL – это одна из самых популярных систем управления базами данных.

Она чрезвычайно проста в использовании и является одним из главных инструментов, используемых при разработке веб-приложений. Обычно используется в сочетании с PHP. Существует множество инструментов, использующих преимущества MySQL, особенно платформы электронной коммерции с открытым исходным кодом, такие как OsCommerce, Magento, Joomla и многие другие.

Некоторые из преимуществ работы с MySQL:

1. MySQL безопасен: он использует надежные уровни защиты данных для защиты конфиденциальных данных от внешнего доступа. Индивидуальные права могут быть установлены для обеспечения нескольких уровней пользователя, что позволяет нескольким людям получать доступ к хранимым данным. Все пароли, используемые для доступа к базе данных MySQL, зашифрованы.

2. MySQL является недорогим:

поскольку программное обеспечение базы данных MySQL можно бесплатно загрузить с официального портала, и оно поставляется предварительно установленным на большинстве хост-серверов, его использование и обслуживание не требует больших затрат.

3. MySQL управляет памятью:

программное обеспечение базы данных отлично управляет памятью и фактически предотвращает утечки памяти, которые могут возникнуть на сервере, где он не развернут.

4. MySQL активно поддерживается:

можно запускать программное обеспечение на различных платформах и операционных системах, включая Windows, Linux, Novell NetWare, UNIX и другие.

5. MySQL быстр:

по сравнению с другими программами баз данных, такими как Sybase и Oracle, MySQL предлагает меньше возможностей, но это хорошо. Это означает, что MySQL намного быстрее с точки зрения производительности, и пользователи по-прежнему имеют доступ ко всем необходимым функциям, которые они получают через другие платформы баз данных.

6. Хранилище данных MySQL является масштабируемым:

по умолчанию MySQL может обрабатывать ограничение общего размера файла около 4 ГБ, однако его можно увеличить в соответствии с потребностями - до 8 ТБ данных.

7. MySQL поддерживает Novell Cluster Services:

Novell Cluster Services от NetWare предлагает расширение базы данных на стороне сервера для защиты ваших данных в случае аппаратного сбоя. Если контент MySQL хранится в кластере Novell, в случае сбоя одного сервера программное обеспечение перейдет на альтернативный сервер - это обеспечит бесперебойную работу приложений, без каких-либо сбоев. Клиенты никогда даже не заметят, что сервер вышел из строя.

Руководствуясь всеми перечисленными достоинствами был сделан вывод, что СУБД MySQL отлично подходит для работы с объемами данных использованных в данном приложении.

2.7 Язык PHP

Язык PHP –это один из самых часто используемых языков для разработки веб-интерфейса. Он является удобным, функциональным и конечно же бесплатным. PHP имеет множество различных версий, которые постоянно обновляются. Этот язык имеет понятный синтаксис и является своеобразной надстройкой над системным языком.

Для написания программы на данном языке можно использовать обычный текстовый редактор. Но для разработки данного приложения использовался VS Code, который является удобным инструментом для написания программ.

Весь код, используемый в приложении в момент загрузки страницы, выгружается в память сервера, обрабатывается. Если код написан без ошибок и не произошло никаких системных аварий, то он выполняется. Результат выполнения выводится пользователю.

Он хорошо работает с базой данных MySQL.Исходя из выше сказанного PHP был выбран для разработки приложения, т.к. полностью подходит для реализации поставленной задачи.

3 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ

3.1 Алгоритм работы

Приложение состоит из двух функциональных частей:

- клиентской части;
- базы данных.

Авторизованный пользователь осуществляет какое-либо действие, в соответствии с этим выполняется требуемая задача. Если возникает ошибка, об этом сообщается пользователю. В случае успеха он обращается к СУБД и выполняет SQL – запрос. При возникновении ошибки пользователю так же выводится об этом сообщение. В случае успешного прохождения запроса выполняется запрашиваемый SQL – запрос, и пользователь видит результат запроса.

3.2 Формирование ролей

Структура проекта включает в себя следующие роли:

- Администратор;
- Оператор.

Рассмотрим функциональные возможности каждого пользователя.

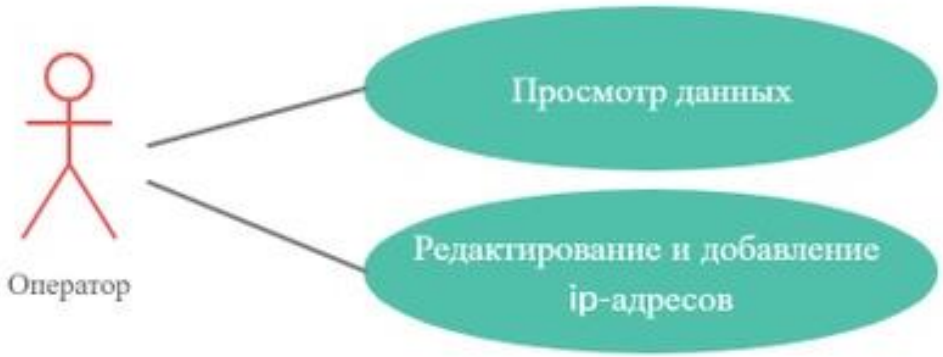


Рисунок 4- UML-диаграмма прав оператора

Рисунок 5- UML-диаграмма прав администратора

Таким образом администратор имеет расширенные права и может совершать любые манипуляции с данными. Оператор же имеет более узкий круг прав, и ему доступно только добавление, редактирование и удаление ip адресов аудиторий, а также возможности создания резервной копии и восстановления.

3.3. Проектирование базы данных

Качество проектирования базы данных влияет на удобство работы с ней. С хорошо спроектированной базой данных легче работать, легче писать к ней запросы.

Для качественного проектирования базы данных нужно:

1 Определить сущности и их атрибуты, которые будут храниться в базе данных, и сформировать по ним таблицы;

2 Разделить сложных атрибутов на более простые;

3 Определить уникальные идентификаторы (первичные ключи) объектов, которые хранятся в строках таблицы;

4 Определить отношения между таблицами с помощью внешних ключей;

5 Спроектировать схему базы данных;

6 Провести нормализацию базы данных;

7 Заполнить таблиц базы данных нужными данными.

3.3.1 Определение сущностей базы данных

На первом этапе произведем сущностей.

Сущность –это объект хранящийся в базе данных и соответствующий, как правилу объекту реального мира. У каждой сущности нужно определить набор атрибутов. Атрибут представляет свойство, которое описывает некоторую характеристику объекта.

Каждый столбец должен хранить один атрибут сущности. А каждая строка представляет отдельный объект или экземпляр сущности.

При проектировании базы данных на начальном этапе мы можем использовать два различных подхода:

- восходящий;
- нисходящий.

При восходящем подходе рассматривается выделение необходимых атрибутов, которые нужно хранить в БД.

Затем выделенные атрибуты группируются в сущности, для которых впоследствии создается таблицы. Такой подход больше подходит для проектирования небольших баз данных с небольшим количеством атрибутов.

Нисходящий подход подразумевает выявление сущностей. Затем происходит анализ сущностей, выявляются связи между ними, а потом и атрибуты сущностей.

Для нашей БД будет использоваться комбинированный метод. То есть описания разных частей системы будут использоваться разные подходы. А затем их результаты объединятся.

В базе данных есть три типа таблиц: вспомогательные таблицы для обеспечения разграничения прав пользователей, основные таблицы, являющиеся частью информационной системы и дополнительные таблицы.

Выделим сущности основных таблиц базы данных:

- Подразделение;
- Корпус;
- Комната;
- Адреса.

Основные имеющиеся таблицы:

«Department» – (Подразделения) содержит информацию о подразделениях ПГУ;

«Build» - (Корпуса) содержит информацию корпуса;

«Rooms» - (Аудитории) содержит информацию о компьютерных аудиториях;

«Address» - (Адреса) содержит информацию об адресе.

Опишем атрибуты сущностей основных таблиц базы данных, приведенных в таблице 1.

Таблица 1 – Сущности основных таблиц БД

Корпуса	Подразделения	Аудитории	ip-адреса
Номер корпуса	Название подразделения	Номер	Адрес
	Руководитель подразделения		
	Номер телефона		

В таблице корпуса содержатся поле Номер корпуса, в котором записан номер корпуса в формате: Корпус № X (часть у), если буква в названии корпуса отсутствует, то в таблице она также не указывается. Таблица подразделения содержит поле название подразделения, где указывается название подразделения, поле руководитель подразделения, в который заносится ФИО руководителя, и поле телефон подразделения, где хранится телефон подразделения. В таблице аудитории поле номер содержит номер аудитории в формате: 7-316а или 7б-205, а также поле в котором указывается назначение аудитории, оно является необязательным для заполнения. В таблице ip-адреса содержится сам адрес и назначение этого адреса. Поле назначение адреса является необязательным к заполнению.

Выделим сущности вспомогательных таблиц базы данных.

Имеющиеся вспомогательные таблицы, представлены в таблице 2 к ним относятся:

«Users» - (Пользователи)таблица со списком пользователей;

«Status» - (Статус)таблица со статусом пользователей.

Символом «*» в дальнейшем будет обозначается необязательное для заполнения поле.

Таблица 2 – Сущности вспомогательных таблиц БД

Пользователи	Статус
Логин	Код статуса
Пароль	Название
Имя	
Email*	
Статус	
active	

Таблица Пользователи

Поле логин включает в себя логин пользователя, пароль пользователя представляет зашифрованную комбинацию знаков для входа в учетную запись. Атрибут Имя используется для приветствия пользователя. Поле Email является не обязательным. Поле статус указывает на статус пользователя, в зависимости от кода статуса, присвоенного пользователю он может быть либо администратором, либо оператором. Active- признак активности учетной записи, может использоваться, например, при увольнении сотрудника для отключения, но не удаления его учетной записи.

Таблица статус

Поле код статуса включает в себя код статуса, код 1 означает оператора, код 10 администратора, кода от 2 до 9 пропущены для возможности добавления других типов пользователей. В поле название вводится текст с названием роли пользователя, например, «Администратор».

Дополнительные таблицы

Дополнительная таблица используется для организации древовидной структуры подразделений университета, представленной в таблице 3. Использована дополнительная таблица, а не простейший parent_id. Это решение является более эффективным для осуществления поиска и частых выборок из таблицы.

Таблица 3 – Сущности дополнительных таблиц БД

Иерархия
Предок
Потомок

Опишем атрибуты основных сущностей таблиц базы данных. Описание представлено в таблицах 4-10.

Основные имеющиеся таблицы:

Таблица 4 – Атрибуты основной сущности «Подразделение»

Имя столбца	Описание	Тип данных	Область допустимых значений
id	id статуса	int(11)	0 – 9
dep_name	номер	varchar(240)	0 – 9 А-я
head	название	text	-
phone	телефон	varchar(20)	0 – 9
is_deleted	флаг удаления	tinyint(1)	0,1

Таблица 5 – Атрибуты основной сущности «Корпуса»

Имя столбца	Описание	Тип данных	Область допустимых значений
b_id	id корпуса	int(11)	0 – 9
build_name	номер	varchar(255)	0 – 9 а-я
is_deleted	флаг удаления	tinyint(1)	0,1

Таблица 6 – Атрибуты основной сущности «Аудитории»

Имя столбца	Описание	Тип данных	Область допустимых значений
r_id	id комнаты	int(11)	0 – 9
room_name	номер	text	0 – 9 А-я
owner	подразделение владелец	int(11)	0 – 9
building_id	корпус	int(11)	0 – 9
address_id	id адреса	int(11)	0 – 9

Таблица 7 – Атрибуты основной сущности «Адреса»

Имя столбца	Описание	Тип данных	Область допустимых значений
id	id адреса	int(11)	0 – 9
ip_adress	адрес	varchar(20)	0 – 9

Имеющиеся вспомогательные таблицы:

Таблица 8 – Атрибуты вспомогательной сущности «Статус»

Имя столбца	Описание	Тип данных	Область допустимых значений
status_id	id статуса	tinyint(3)	0 – 9
status_name	название	text	-

Таблица 9 – Атрибуты вспомогательной сущности «Пользователь»

Имя столбца	Описание	Тип данных	Область допустимых значений
user_id	id пользователя	int(11)	0 – 9
login	логин	text	-
password	пароль	varchar(15)	0 – 9 а-я
name	фио	int(10)	0 – 9
e-mail	e-mail	varchar(320)	0 – 9 а-я
status_id	статус	int(11)	0 – 9
active	флаг активность	tinyint(1)	0,1

Дополнительная таблица:

Таблица 10 – Атрибуты дополнительной таблицы «Иерархия»

Имя столбца	Описание	Тип данных	Область допустимых значений
id	id	int(11)	0 – 9
ancestor	предок	int(11)	0 – 9
desancestor	потомок	int(11)	0 – 9

3.3.2 Определение уникальных идентификаторов объектов

На следующем этапе этап определим ключи, представляющие способ идентификации строк в таблице. С помощью ключей мы также будем связывать строки между различными таблицами в отношения.

Первичный ключ (pk) непосредственно применяется для идентификации строк в таблице. Он должен соответствовать следующим требованиям:

- Первичный ключ должен быть уникальным все время
- Первичный ключ должен постоянно присутствовать в таблице и иметь значение
- Первичный ключ не должен часто менять свое значение
- Первичный ключ в основном представляет один столбец таблицы, но также может быть составным и состоять из нескольких столбцов.
- Если для таблицы можно выделить потенциальный ключ, то его можно использовать в качестве первичного ключа.

3.3.3 Создание связей между таблицами

Базы данных хранят таблицы, которые связанны между собой специальными связями. За это и отвечает следующий этап. Связь представляет ассоциацию между сущностями разных типов.

В связях есть 2 типа таблиц дочерняя и родительская. Дочерняя таблица зависит от родительской.

Чтобы организовать связи нужно использовать внешние ключи. Внешний ключ представляет один или несколько столбцов из одной таблицы, который одновременно является потенциальным ключом из другой таблицы. Внешний ключ необязательно должен соответствовать первичному ключу из главной таблицы. Но обычно внешний ключ из зависимой таблицы указывает на первичный ключ из главной таблицы.

Связи между таблицами бывают следующих типов:

- Один к одному
- Один к многим
- Многие ко многим

В базе данных представленной в данной выпускной квалификационной работе большинство таблиц связывает связь типа один ко многим.

При изменении первичных и внешних ключей соблюдался такой аспект как ссылочная целостность данных. Ее основная идея состоит в том, чтобы две таблице в базе данных, которые хранят одни и те же данные, поддерживали их согласованность. Целостность данных представляет правильно выстроенные отношения между таблицами с корректной установкой ссылок между ними. Целостность данных может нарушить, например, аномалия удаления. Она возникает при удалении строки из главной таблицы. В этом случае внешний ключ из зависимой таблицы продолжает ссылаться на удаленную строку из главной таблицы. Поэтому важно организовать корректное удаление записей из родительских и дочерних таблиц.

Для связей в таблицах используются индексы. В таблицах 11=17 представлена полная структура с указанием ключей(ПК) и индексов(ИН).

Таблица 11 – Полная структура таблицы «Корпуса»

Имя столбца	Описание	Тип данных
ПК b_id	id корпуса	int(11)
build_name	номер	varchar(255)
ИН is_deleted	флаг удаления	tinyint(1)

Таблица 12 – Полная структура таблицы «Аудитории»

Имя столбца	Описание	Тип данных
ПК id	id комнаты	int(11)
room_name	номер	text
ИН owner	подразделение владелец	int(11)
ИН build_id	корпус	int(11)
ИН address_id	id адреса	int(11)

Таблица 13 – Полная структура таблицы «Адреса»

Имя столбца	Описание	Тип данных
ПК id	id адреса	int(11)
ip_address	адрес	varchar(20)

Таблица 14 – Полная структура таблицы «Подразделения»

Имя столбца	Описание	Тип данных
ПК d_id	id статуса	int(11)
dep_name	номер	varchar(240)
head	руководитель	text
phone	телефон	varchar(20)
ИН is_deleted	флаг удаления	tinyint(1)

Таблица 15 – Полная структура таблицы «Дерево»

Имя столбца	Описание	Тип данных
id	id	int(11)
ancestor	предок	int(11)
desancestor	потомок	int(11)

Таблица 16 – Полная структура таблицы «Пользователи»

Имя столбца	Описание	Тип данных
ПК user_id	id пользователя	int(11)
login	логин	varchar(32)
password	пароль	varchar(32)
name	фио	int(10)
e-mail	e-mail	text
ИН status_ number	статус	int(2)
active	активность	tinyint(1)

Таблица 17 – Полная структура таблицы «Статус»

Имя столбца	Описание	Тип данных
ПК status_id	id статуса	tunyint(3)
status_name	название	text

3.3.4 Проектирование схемы базы данных

На основании спроектированных таблиц была разработана схема базы данных. Она представлена на рисунке 6.

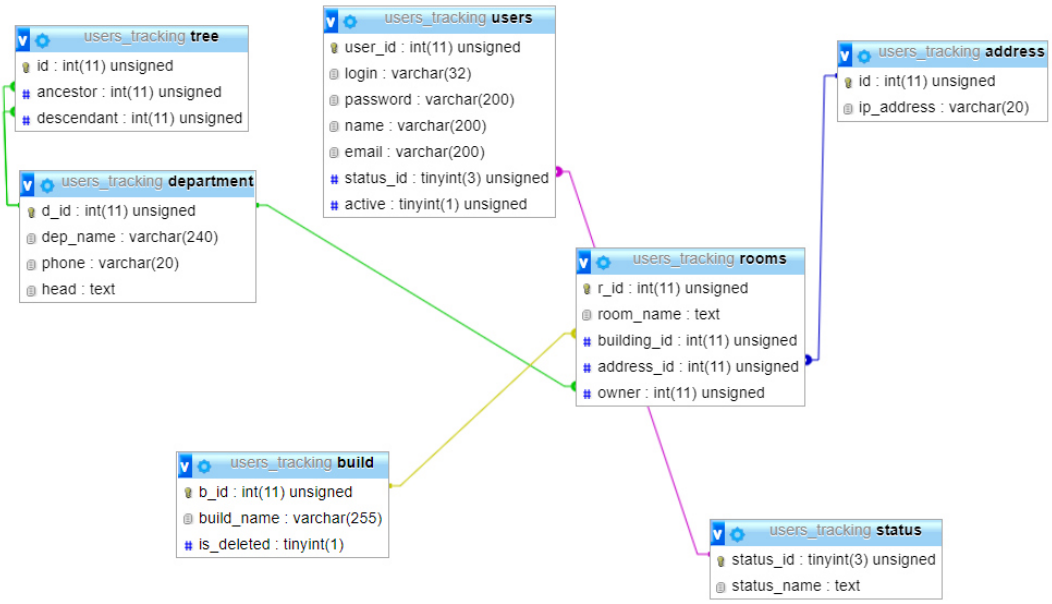


Рисунок 6-Схема физического проектирования базы данных

3.3.5 Нормализация БД

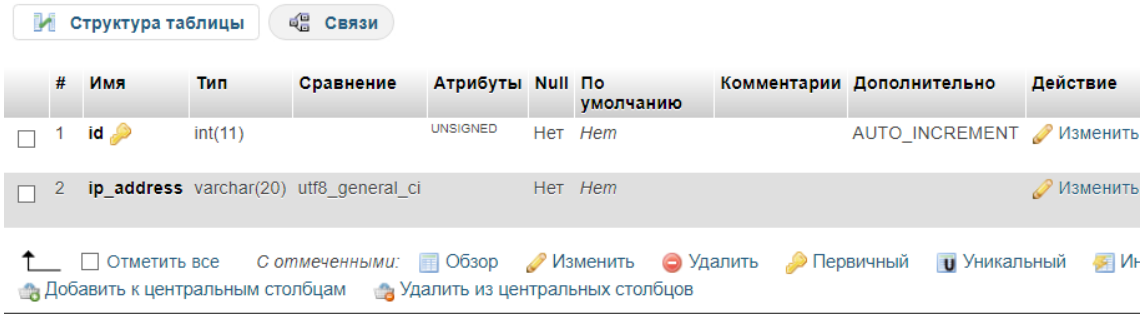
Была выполнено устранения избыточности данных.

В результате нормализации было получено:

- Ни в одной таблице не хранится информация о двух и более сущностях.
- Ни одна таблица не содержит повторяющихся столбцов
- Нормализация предполагает применение нормальных форм к структуре данных. Была применена третья нормальная форма, которая считается достаточной для решения большинства задач.

3.3.6 Создание таблиц с помощью языка SQL

На основе разработанных сущностей базы данных создаем с помощью языка SQL соответствующие таблицы: Users, status, units, floors, buildings, addresses, rooms. Для создания таблиц был использован оператор CREATE. Результаты представлены на рисунках 7-13


Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(11)		UNSIGNED	Нет	Нем		AUTO_INCREMENT	Изменить
2	ip_address	varchar(20)	utf8_general_ci		Нет	Нем			Изменить

Отметить все С отмеченными: Обзор Изменить Удалить Первичный Уникальный Ин
Добавить к центральным столбцам Удалить из центральных столбцов

Рисунок 7 – Таблица «Адреса»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	b_id	int(11)		UNSIGNED	Нет	Нем		AUTO_INCREMENT	Изменить
2	build_name	varchar(255)	utf8_general_ci		Да	NULL			Изменить
3	is_deleted	tinyint(1)			Нет	0			Изменить

Отметить все С отмеченными: Обзор Изменить Удалить Первичный Уникальный Индекс
Добавить к центральным столбцам Удалить из центральных столбцов

Рисунок 8– Таблица «Корпуса»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(11)			Нет	Нем		AUTO_INCREMENT	Изменить
2	ancestor	int(11)			Нет	Нем			Изменить
3	descendant	int(11)			Нет	Нем			Изменить

Отметить все С отмеченными: Обзор Изменить Удалить Первичный Уникальный И

Рисунок 9– Таблица «Иерархия»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	r_id	int(11)		UNSIGNED	Нет	Нем		AUTO_INCREMENT	Изменить
2	room_name	text	utf8_general_ci		Нет	Нем			Изменить
3	building_id	int(11)		UNSIGNED	Нет	Нем			Изменить
4	address_id	int(11)		UNSIGNED	Да	NULL			Изменить
5	owner	int(11)		UNSIGNED	Да	NULL			Изменить

Отметить все С отмеченными: Обзор Изменить Удалить Первичный Уникальный Ин
Добавить к центральным столбцам Удалить из центральных столбцов

Рисунок 10 – Таблица «Аудитории»

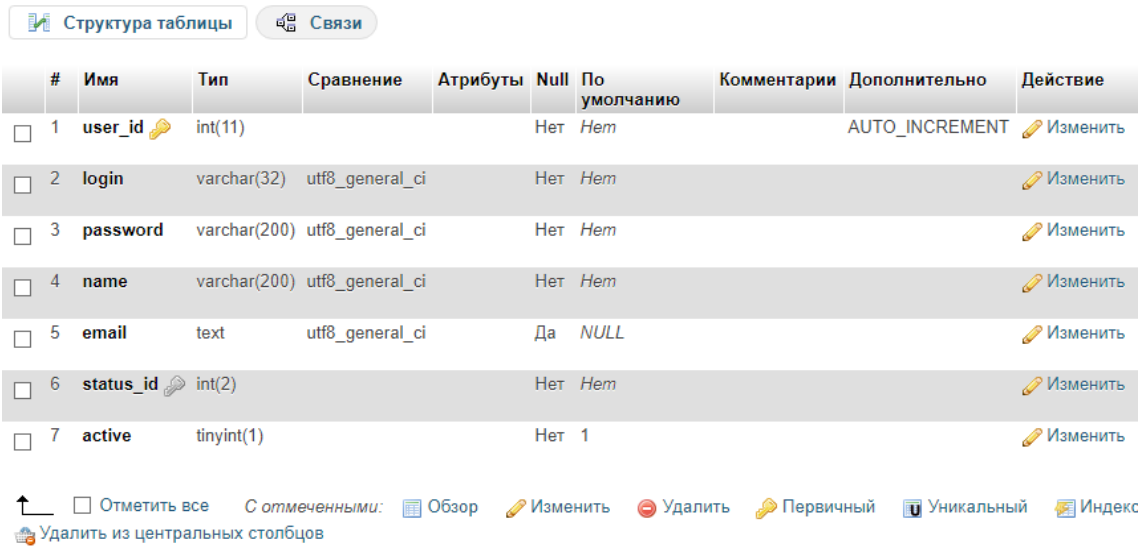


Рисунок 11 – Таблица «Пользователи»

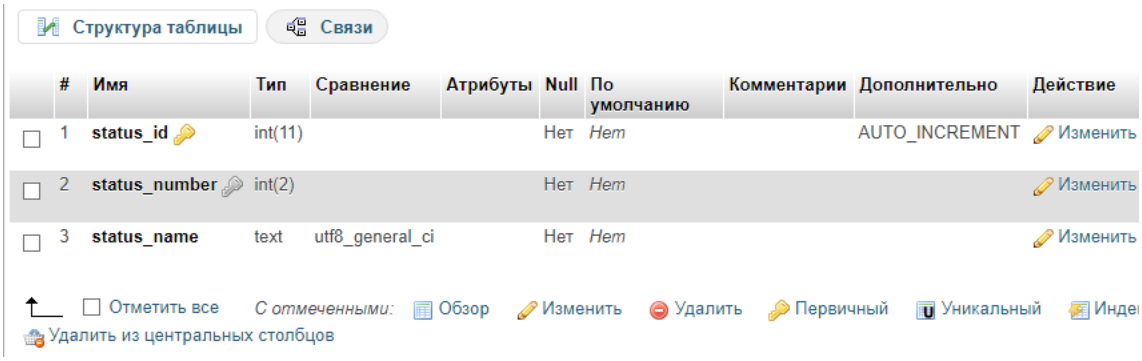


Рисунок 12 – Таблица «Статус»

Рисунок 13 – Таблица «Подразделения»

3.4 Добавление данных

Для добавления данных в таблицы был разработан web-интерфейс. Так как добавление данных может осуществлять только администратор, то разработана авторизация. Для которой используются таблицы Users и Status.

Таким образом, любой неавторизованный пользователь должен пройти процедуру авторизации и если он является администратором, ему будет предоставлено право добавления в таблицы.

Осуществить добавление в конкретную таблицу можно по средствам форм, разработанных на PHP. На рисунке представлен интерфейс добавления пользователя.

После нажатия на кнопку «добавить пользователя» появится интерфейс для добавления представленный на рисунках 14,15.

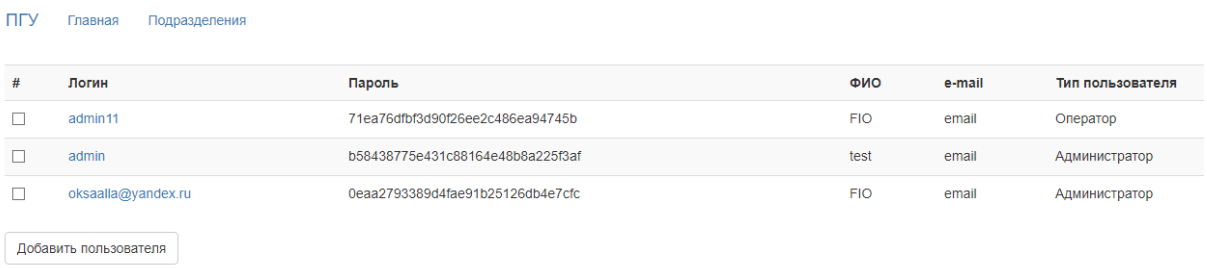


Рисунок 14 – Интерфейс таблицы пользователей

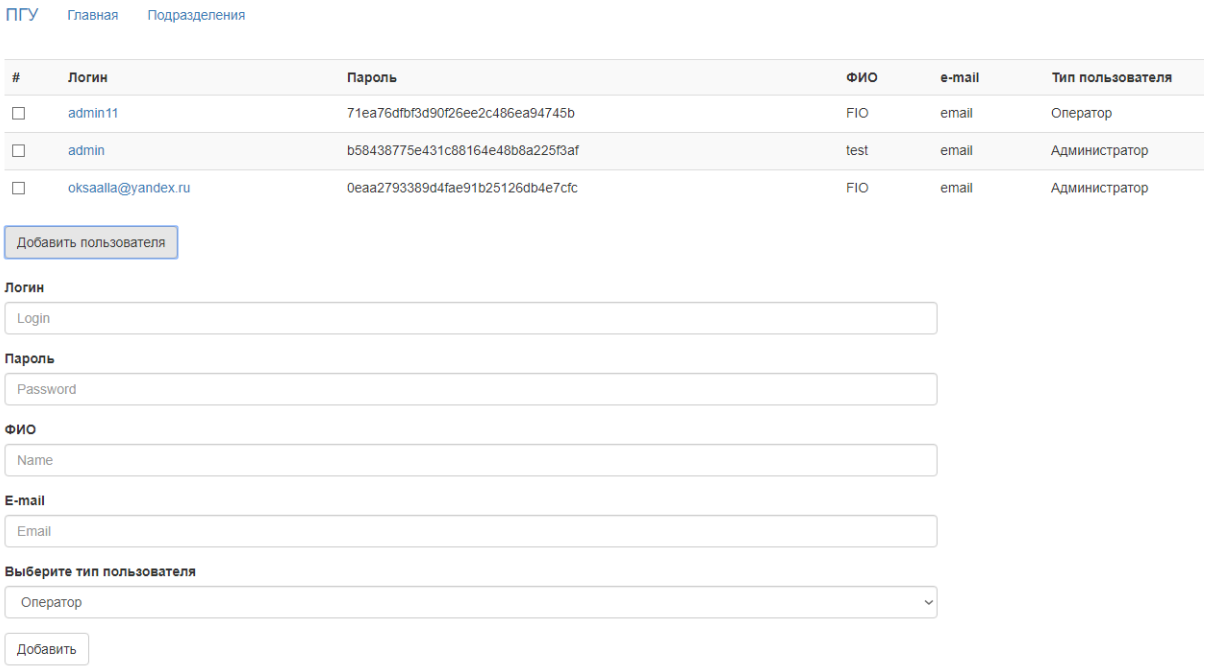


Рисунок 15 – Интерфейс добавления пользователя

Таким же способом можно осуществить и добавление подразделения. Результат на рисунках 16,17.

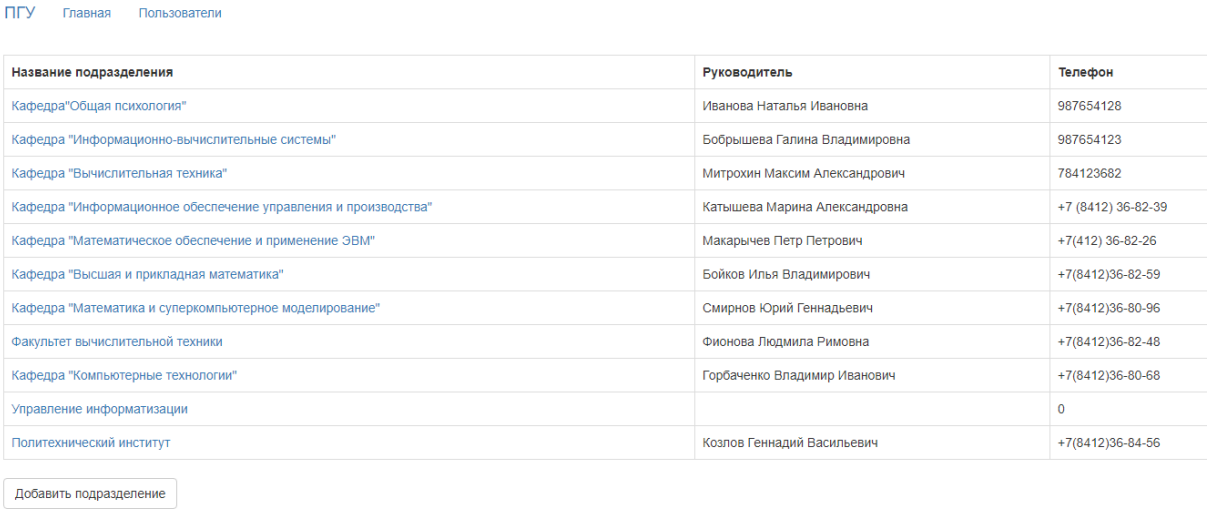


Рисунок 16 – Интерфейс таблицы подразделений

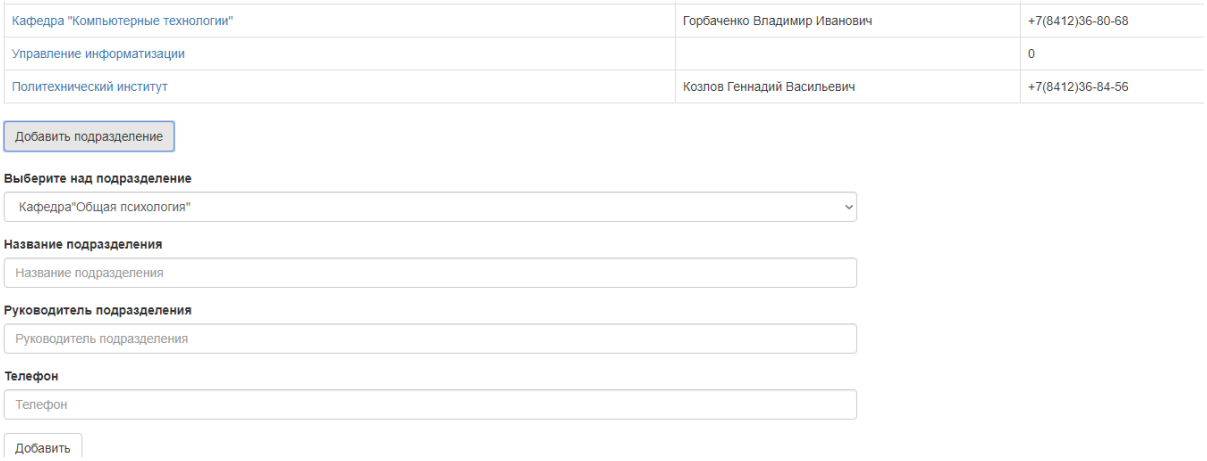


Рисунок 17 – Интерфейс добавления подразделений

Была создана таблица интерфейс для выгрузки данных. Она представлена на рисунке 18. А также форма для добавления данных, представленная на рисунке 19.

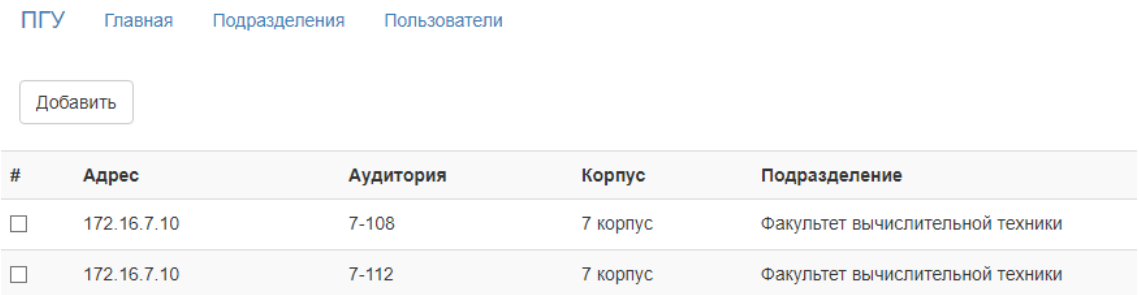


Рисунок 18 – Интерфейс для выгрузки данных

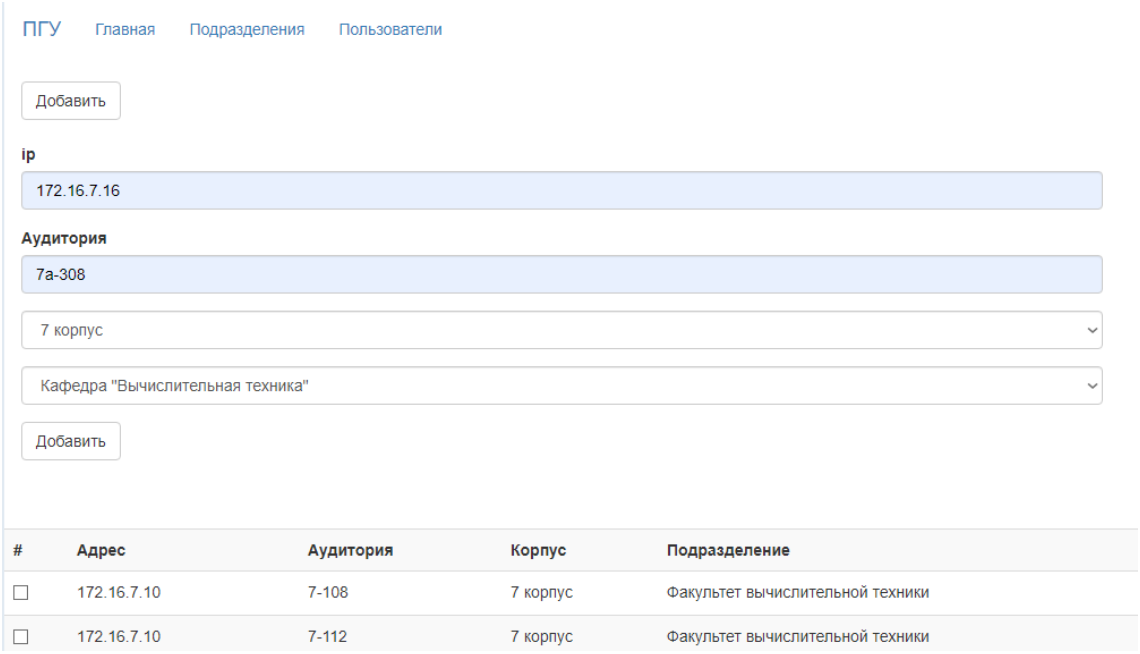


Рисунок 19 – Интерфейс добавления в таблицу

Также добавления данных можно осуществить с помощью команды INSERT. На рисунке 20 представлено тестовое заполнение таблиц.

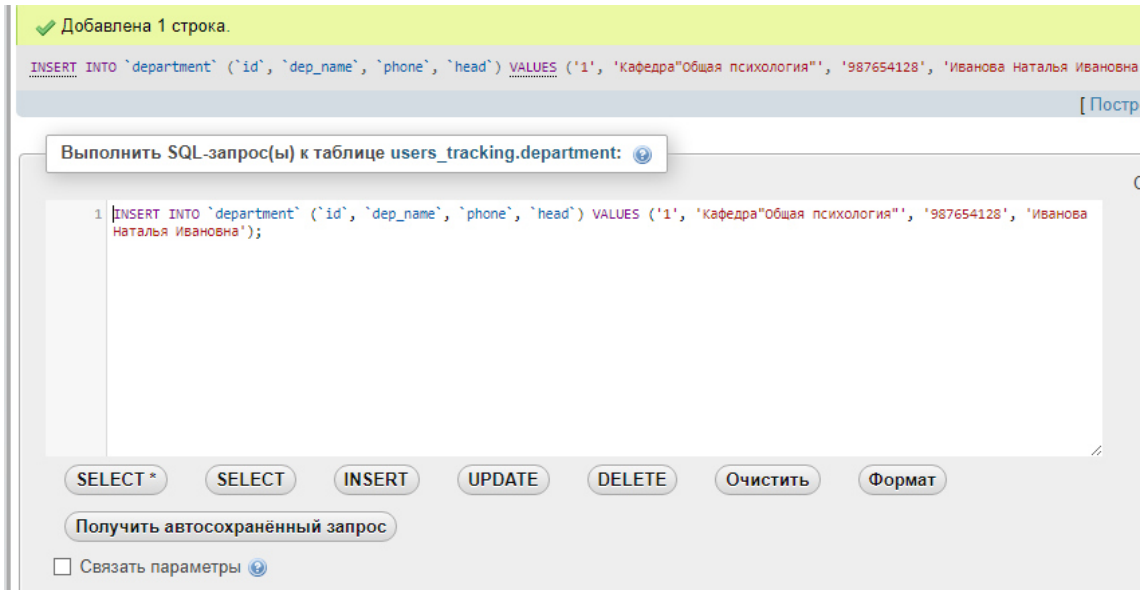


Рисунок 20 – Интерфейс добавления подразделений

Выведем все данные тестовых таблиц, они представлены на рисунках 21-27. Для скрытия паролей использована функция MD5 в базе данных, а также хэш-функция в интерфейсе.

+ Параметры

	d_id	dep_name	phone	head	is_deleted
Изменить Копировать Удалить	1	Кафедра"Общая психология"	987654128	Иванова Наталья Ивановна	0
Изменить Копировать Удалить	544152	Кафедра "Информационно-вычислительные системы"	987654123	Бобрышева Галина Владимировна	0
Изменить Копировать Удалить	544170	Кафедра "Вычислительная техника"	784123682	Митрохин Максим Александрович	0
Изменить Копировать Удалить	544206	Кафедра "Информационное обеспечение управления и п...	+7 (8412) 36-82-39	Катышева Марина Александровна	0
Изменить Копировать Удалить	544242	Кафедра "Математическое обеспечение и применение Э...	+7(412) 36-82-26	Макарычев Петр Петрович	0
Изменить Копировать Удалить	544368	Кафедра "Высшая и прикладная математика"	+7(8412)36-82-59	Бойков Илья Владимирович	0
Изменить Копировать Удалить	544842	Кафедра "Математика и суперкомпьютерное моделирова...	+7(8412)36-80-96	Смирнов Юрий Геннадьевич	0
Изменить Копировать Удалить	31429808	Факультет вычислительной техники	+7(8412)36-82-48	Фионова Людмила Римовна	0
Изменить Копировать Удалить	123595545	Кафедра "Компьютерные технологии"	+7(8412)36-80-68	Горбаченко Владимир Иванович	0
Изменить Копировать Удалить	161624869	Управление информатизации	0		0
Изменить Копировать Удалить	352685446	Политехнический институт	+7(8412)36-84-56	Козлов Геннадий Васильевич	0

Рисунок 20 – Таблица «Подразделения»

+ Параметры

	id	room_name	building_id	address_id	owner ▲ 1
Изменить Копировать Удалить	6856	7-316	7	1	544170
Изменить Копировать Удалить	6857	7-317	7	1	544170
Изменить Копировать Удалить	6858	7-319	7	1	544170
Изменить Копировать Удалить	6860	7-322	7	1	544170
Изменить Копировать Удалить	6955	8-703в	8	1	544368
Изменить Копировать Удалить	6956	8-703г	8	1	544368
Изменить Копировать Удалить	6957	8-704	8	1	544368
Изменить Копировать Удалить	6912	8-705	8	1	31429808

Рисунок 21 – Таблица «Комнаты»

+ Параметры

	status_id	status_name
Изменить Копировать Удалить	1	Оператор
Изменить Копировать Удалить	2	Администратор

Рисунок 22 – Таблица «Статус»

+ Параметры

	id	ip_address ▼ 1
Изменить Копировать Удалить	4	172.16.7.13
Изменить Копировать Удалить	3	172.16.7.12
Изменить Копировать Удалить	2	172.16.7.11
Изменить Копировать Удалить	1	172.16.7.10

Рисунок 23 – Таблица «Адреса»

Параметры

	id	ancestor	descendant
Изменить Копировать Удалить	1	31429808	544152
Изменить Копировать Удалить	3	31429808	544170
Изменить Копировать Удалить	4	31429808	544206
Изменить Копировать Удалить	5	31429808	544242
Изменить Копировать Удалить	6	31429808	544368
Изменить Копировать Удалить	7	31429808	544842
Изменить Копировать Удалить	8	31429808	123595545
Изменить Копировать Удалить	9	31429808	544842

Рисунок 24 – Таблица «Иерархия»

Изменить Копировать Удалить	7	7 корпус	
Изменить Копировать Удалить	8	8 корпус	
Изменить Копировать Удалить	9	9 корпус	

Рисунок 25 – Таблица «Корпуса»

	user_id	login	password	name	email	status_id	active
Изменить Копировать Удалить	44	admin	b58438775e431c88164e48b8a225f3af	test	email	2	1
Изменить Копировать Удалить	45	admin11	71ea76dfbf3d90f26ee2c486ea94745b	FIO	email	1	1
Изменить Копировать Удалить	46	oksaalla@yandex.ru	0eaa2793389d4fae91b25126db4e7cfc	FIO	email	2	1

Рисунок 26 – Таблица «Пользователи»

3.5 SQL – запросы

Для добавления и изменения данных можно использовать SQL-запросы не включенные в интерфейс. Это продемонстрированно на рисунках 27-29.

Затронута 1 строка.

```
UPDATE `units` SET `unit_name` = 'Педагогический институт', `unit_head` = 'Иванова Наталья Валентиновна' WHERE `units`.`unit_id` = 11;
```

[Построчное редактирование] [Изменить] [Создать PHP-код]

Отображение строк 0 - 1 (2 всего, Запрос занял 0.0008 сек.)

```
SELECT * FROM `units`
```

Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]

Рисунок 27 – SQL-заброс на изменение данных

Отображение строк 0 - 8 (9 всего, Запрос занял 0.0013 сек.)

```
SELECT * FROM `department` JOIN `tree` ON (`department`.`id` = `tree`.`descendant`) WHERE `tree`.ancestor = 31429808
```

Профилирование [Построчное редактирование] [Изменить] [Ана

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице | Сортировать по индексу:

+ Параметры

id	id_parent	name	phone	head	id	ancestor	descendant
544152	31429808	Кафедра "Информационно-вычислительные системы"	0		1	31429808	544152
544170	31429808	Кафедра "Вычислительная техника"	0		3	31429808	544170
544206	31429808	Кафедра "Информационное обеспечение управления и п...	0		4	31429808	544206
544242	31429808	Кафедра "Математическое обеспечение и применение Э...	0		5	31429808	544242
544368	31429808	Кафедра "Высшая и прикладная математика"	0		6	31429808	544368
544842	31429808	Кафедра "Математика и суперкомпьютерное моделирова...	0		7	31429808	544842
123595545	31429808	Кафедра "Компьютерные технологии"	0		8	31429808	123595545
544842	31429808	Кафедра "Математика и суперкомпьютерное моделирова...	0		9	31429808	544842
123595545	31429808	Кафедра "Компьютерные технологии"	0		10	31429808	123595545

Рисунок 28 – Отображение всех подразделений, принадлежащих “ФВТ»

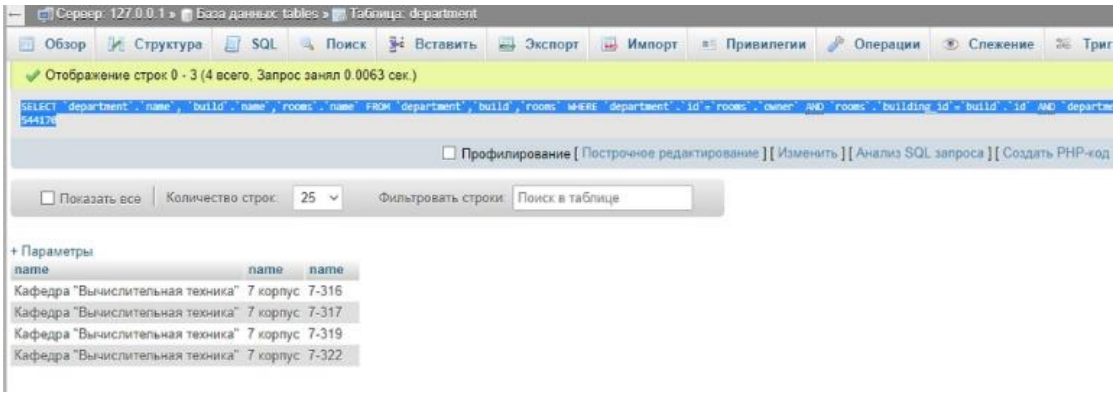


Рисунок 29 – Отображение аудиторий, принадлежащих кафедре «ВТ» в

SQL-запросы к БД через интерфейс представлены на рисунке 29.

```
C: > WebServers > home > test > www > api.php
15  function getAllUsers($db) {…
32  }
33  function getAllAddress($db) {…
46  }
47  function getAllBbuildings($db) {…
59  }
60  function getAllRooms($db) {
61
62      $sql =  "SELECT * FROM rooms
63      RIGHT JOIN address ON rooms.address_id=address.id
64      RIGHT JOIN build ON rooms.building_id=build.id
65      INNER JOIN department ON rooms.owner=department.id
66       " ;
67
68      $result = array();
69
70      $stmt = $db->prepare($sql);
71      $stmt->execute();
72
73      while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
74          $result[$row['id']] = $row;
75          }
76      return $result;
77  }
78  function getAllDepartment($db) {…
90  }
91
92  function addUser($db, $login, $password, $name, $email, $status_id){
93      $password=password_hash( $password,PASSWORD_DEFAULT);
94      $sql = "INSERT INTO users(login, password, name, email, status_id)
95              VALUES(:login, :password, :name, :email, :status_id )";
96      $stmt = $db->prepare($sql);
97      $stmt->bindValue(':login', $login, PDO::PARAM_STR);
98      $stmt->bindValue(':password', $password, PDO::PARAM_STR);
99      $stmt->bindValue(':name', $name, PDO::PARAM_STR);
100     $stmt->bindValue(':email', $email, PDO::PARAM_STR);
101     $stmt->bindValue(':status_id', $status_id, PDO::PARAM_INT);
102     $stmt->execute();
```

Рисунок 30 – SQL-запросы интерфейса

4 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

4.1 Введение

Современному человеку трудно даже представить себе жизнь без персонального компьютера. Мы привыкли к использованию данной техники при выполнении служебных обязанностей и в часы досуга. На работе и дома. Мало кто не сумел оценить по достоинству целый комплекс возможностей, которые открывают перед нами электронные устройства. Однако, как и в любой другой сфере деятельности, в работе с компьютером есть не только множество плюсов, но и свои минусы. Такая деятельность сопряжена с влиянием на оператора ряда негативных и даже опасных факторов.

Большинство таких факторов вредного влияния компьютера на здоровье человека невозможно определить с помощью наших органов чувств.

4.2 Вредные факторы

Вредные факторы делятся на 3 группы:

- Физически вредные и опасные факторы
- Химически вредные и опасные факторы
- Психофизические вредные и опасные факторы

Факторы, оказывающие физическое влияние.

За этим названием скрываются такие факторы, как: повышенный уровень электромагнитного, рентгеновского, ультрафиолетового и инфракрасного излучения; повышенный уровень статического электричества и запыленности воздуха рабочей зоны, также повышенное содержание положительных аэронов и пониженное содержание отрицательных аэройонов в воздухе рабочей зоны. Повышенный уровень блескости, неравномерность распределения яркости и повышенная яркость светового изображения представляет опасность для глаз. Повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека.

Факторы, оказывающие химическое влияние.

К химическим факторам относятся следующие: повышенное содержание двуокиси углерода, озона, аммиака, фенола и формальдегида в воздухе рабочей зоны.

Факторы, оказывающие психофизическое влияние.

К ним относятся такие показатели как: напряжение зрения и внимания, а также интеллектуальные, эмоциональные и длительные статические нагрузки. Монотонность труда и большой объем информации, обрабатываемый в единицу времени оказывает не лучшее воздействие на оператора.

Типичными ощущения, которые испытывают операторы ПЭВМ к концу рабочего дня являются: переутомление глаз, головная боль, тянущие боли в мышцах шеи, рук и спины, снижение концентрации внимания.

4.3 Компьютерные синдромы

Уже в первые годы компьютеризации было отмечено специфическое зрительное утомление у пользователей дисплеев. Всемирная организация здравоохранения ввела такое понятие как «компьютерный зрительный синдром». Его симптомами являются жжение в глазах, покраснение век и конъюктивиты, чувство инородного тела или песка под веками, боли в области глазниц и лба, затуманивание зрения, аккомодация.

Среди «компьютерных синдромов выделяют» также синдром компьютерного стресса.

Его симптомами являются: головная боль, раздражительность, быстрая утомляемость, вялость, признаки депрессии. Проявления этого заболевания разнообразны и многочисленны. Они сгруппированы по принципу воздействия на ту или иную часть организма.

Многие из тех, кто постоянно работает с компьютером, начинают жаловаться на боли в спине и шее. Такие недомогания называют синдромом длительных статических нагрузок. Главной причиной появления этого синдрома является длительное пребывание в сидячем положении, что перегружает мышцы спины и ног.

Самый страшный из всех симптомов-это затрудненное дыхание. Вынесенные вперед локти не дают свободно двигаться грудной клетке, и это приводит к астме, развитию приступов кашля и другое. Человек не может физически проводить долгие часы в фиксированном положении. Длительные периоды неподвижности снижают приток крови к мышцам, в следствие чего происходит накопление продуктов метаболизма, которые раздражают нервы задействованных мышц.

Остеохондроз. При длительном сидении с опущенными плечами возникает стойкое изменение костно-мышечной системы. Возникают искривление позвоночника, тендовагинит или воспаление сухожилий кисти, запястья или плеча, тендосиновит — воспаление синовиальной оболочки сухожильного основания кисти и запястья;

Существует еще и синдром запястного канала, который, он представляет собой травму запястья. Это поражение серединного нерва в запястном канале. Его симптомами является боль, снижение чувствительности, некоторая слабость и неловкость при движениях кистью, особенно при необходимости захватывающего движения большим пальцем.
В неврологии относится он относится к туннельным синдромам. Характерны ночные болевые приступы, вынуждающие пациентов пробуждаться. Синдром может носить двусторонний характер, но чаще и сильнее поражается доминирующая кисть.

4.4 Влияние компьютера на женщин и детей

Особый случай - это влияние компьютера на женщин в период беременности. Многочисленная статистика подтверждает, что работа за ПК нарушает нормальное течение беременности и повышает вероятность выкидыша и часто является причиной появления на свет детей с врожденными пороками, из которых наиболее существенными бывают дефекты развития головного мозга. Из-за того, что в процессе развития плода женщина набирает вес, увеличивается нагрузка на части тела, которые представляют опорно-

двигательную систему. Они подвергаются дополнительному напряжению. Поэтому необходимо, чтобы беременные или имеющие намерение забеременеть женщины минимизировали использование компьютера, как на работе, так и дома.

Неблагоприятное влияние оказывает работа с ПК на детей и подростков в силу анатомо-физиологических особенностей растущего организма и реакций его на внешние раздражители. Общее и зрительное утомление зачастую происходят на фоне повышения тревоги, раздражительности, подавленности, замкнутости или нарушения сна. Влияние многих факторов, особенно на школьников и студентов, изучено далеко не полностью.

4.5 Вредные факторы согласно ГОСТ

Есть специальный документ, регламентирующий работу с ПК-ГОСТ12.0.003-74 “ССБТ. Опасные и вредные производственные факторы. Классификация”, который определяет какие именно факторы воздействуют на человека при работе за компьютером. К ним относится, например, повышенная температура поверхностей ПК, повышенная или пониженная влажность воздуха, повышенный уровень электромагнитных излучений и другие.

Наиболее опасным является повышенный уровень электромагнитных излучений. Самыми чувствительными к ним являются центральная и сердечно-сосудистая системы. Так же влиянию подвергается и эндокринная система.

Сначала начинает повышаться возбудимость нервной системы, которая проявляется в раздражительности, нарушении сна, эмоциональной неустойчивости. Затем развиваются астенические состояния. Могут возникнуть гипотония, боли в сердце, головная боль, брадикардия. Указанные симптомы могут быть выражены в разной степени.

Повышение температуры тела из-за теплового воздействия ЭВМ проявляется в нагреве отдельных клеток и тканей. Интенсивность нагрева зависит от количества поглощенной энергии и скорости оттока тепла от облучаемых участков тела. Отток тепла затруднен в органах и тканях с плохим

кровоснабжением. Такой тканью является хрусталик глаза, вследствие чего возможно возникновение катаракты. Также воздействию подвергаются: печень, поджелудочная железа, желудок. Еще в следствие нагревания могут обострится хронические заболевания человека, который подвергается постоянному воздействию.

4.6 Действия для снижения опасности

В целях снижения пагубного влияния, опасные факторы, воздействия которых подвергается человек, работающий за ЭВМ, подлежат нормированию.

Все ПК должны иметь гигиенический сертификат, включающий оценку визуальных параметров. Во множестве стран существуют документы, регламентирующие правила пользования дисплеями. Наиболее известны шведские документы MPR-II 1990: 8 (Шведский национальный комитет по защите от излучений) и более жесткий стандарт ТСО-99 (Шведская конференция профсоюзов). Эти нормы применяются во всех странах Скандинавии и рекомендованы к распространению в странах ЕС.

Работа за ПЭВМ должна производится на комфортабельном рабочем месте, которое имеет достаточную освещенность для выполнения данной деятельности. Уровень освещенности также контролируется мировыми стандартами. Обычное офисное освещение часто бывает очень ярким для работы с компьютером. Экран не должен бликовать.

Для достижения четкости изображения нужно регулярно протирать экран антистатическим раствором.

Самым удобным местом для расположения монитора является расстоянии вытянутой руки работающего за ПЭВМ человека. Верхний край экрана должен располагаться на уровне глаз или чуть ниже.

Необходим регулярный отдых от работы за компьютером, поскольку однообразная поза достаточно утомительна для глаз, шеи и спины.

Воздух в помещении нуждается в увлажнении не зависимо от того есть ли в ней люди, работающие за компьютером или нет. В отопительный сезон

увлажнение воздуха становится еще более важным, так как отопительные приборы иссушают воздух. Сухой воздух оказывает пагубное влияние на кожу и дыхательные пути.

Мебель, используемая при работе на компьютере, отвечает за удобство расположения рук, ног и позвоночника. В настоящее время мебельные фабрики производят огромное количество офисных, компьютерных и других видов стульев, которые позволяют чувствовать себя комфортно.

Верхний край экрана должен располагаться выше линии глаз, чтобы голову не приходилось запрокидывать. Если человек долгое время будет находится в таком положении, то более в шее и спине не избежать. Под столом должно быть достаточно пространства для того, чтобы можно было вытянуть ноги

Глубина стола должна быть такой, чтобы расстояние до экрана монитора было не менее 50 см. Ширина его зависит от количества периферийных устройств. Чем массивнее стол - тем лучше: устойчивость - враг вибрации, а вибрация - враг техники.

Современные жидкокристаллические дисплеи более безопасны для человека, чем мониторы на основе электронно-лучевой трубки, которые являются источником электростатического поля, электромагнитных излучений, излучения оптического диапазона и рентгеновского излучения. Дисплеи с ЭЛТ притягивают и собирают пыли, могут быть причиной появления дерматитов, обострения бронхиальной астмы, раздражения слизистых оболочек

4.7 Соблюдение режима труда и отдыха

Время труда и отдыха при работе за компьютером регламентируется статьями 107 ТК РФ и 109 ТК РФ. Первая статья признает необходимость перерывов в течении рабочего дня для людей, осуществляющих работу за ПЭВМ.

Согласно статье 109 ТК РФ некоторые виды труда подразумевают необходимость предоставления работникам возможности отдохнуть в течение рабочего дня, что вызвано спецификой технологии, производства и условий работы.

В двадцать седьмой статье Закона о санитарно-эпидемиологическом благополучии отмечается, что работа с машинами и различными устройствами не должна вредно влиять на здоровье.

В зависимости от вида работ и степени нагрузки Приложение 7 к СанПиН 2.2.2/2.4.1340-03 устанавливает, что время отдыха при рассматриваемом виде работ должно составлять в течение рабочего дня от 50 до 140 минут. И это с учетом того, что данные перерывы длительность работы увеличивать не должны.

За нарушение режима труда и отдыха работников на работодателя накладываются административные штрафы, предусмотренные статьей 6.3 КоАП РФ.

В Типовой инструкции ТОИ Р-45-084-01 (утверждена 2 февраля 2001 года) содержится более детальная регламентация рассматриваемого вопроса.

Согласно данной Инструкции длительность работы с компьютером без перерыва может быть не более двух часов.

Целью перерывов является уменьшение напряжения, усталости глаз и т.д.

Инструкция устанавливает зависимость времени перерывов от вида и времени осуществляемой работы путем деления на группы:

А – чтение информации с монитора по сделанному запросу;

Б – печатание на клавиатуре с целью ввода информации;

В — творческая работа.

Кроме этого предусмотрено деление на категории сложности работ:

для группы А (не свыше 60000 считываемых знаков за смену) перерыв составляет 15 минут, предоставляется два раза – через два часа после начала работы и перерыва на обед;

для группы Б (не свыше 40000 напечатанных знаков за смену) перерыв составляет 10 минут через каждый трудовой час;

для группы В (не свыше шести 6 часов за смену) перерыв составляет 15 минут через каждый трудовой час.

ЗАКЛЮЧЕНИЕ

Результатом данной выпускной квалификационной работы является база данных и интерфейс добавления для системы учета пользователей ЛВС ПГУ. Была разработана база данных для ПГУ, разработаны запросы к базе данных, разработан интерфейс, оформлена документация на приложение. Проведенное тестирование показало, что часть приложения, созданная в ходе выпускной квалификационной работы, работает правильно.

В дальнейшем приложение может быть улучшено за счет расширения базы данных, усложнения структуры и добавления новых модулей. Предусмотрено дополнение вспомогательных таблиц новыми ролями.

В ходе выполнения выпускной квалификационной работы был значительно повышен уровень практических навыков по созданию веб-приложений, освоен процесс разработки баз данных для приложения. Были рассмотрены такие средства, как: MAMP, СУБД MySQL, PhpMyAdmin, Bootstrap. Получены навыки работы с множеством языков программирования: PHP, HTML, CSS, JavaScript и SQL – запросов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. К. Янк. PHP и MySQL. От новичка к профессионалу, 5-е изд. Москва, 2013. 585 с.

2. Л. Томсон, Л. Веллинг. Разработка web-приложений на PHP и MySQL, 5-е изд. Москва, Санкт-Петербург, Киев, 2017. 652 с.

3. Р. Никсон. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5, 4-е изд. 2016. 766 с.

4. А. Бьюли. Изучаем SQL. Санкт-Петербург – Москва, 2007. 309 с.

5. Н. Прохоренок, В. Дронов. HTML, JavaScript, PHP и MySQL. Джентельменский набор web-мастера, 5-е изд. Санкт – Петербург, 2019. 912 с.

6. Д. Дакетт. HTML и CSS. Разработка и дизайн веб-сайтов. Санкт – Петербург, 2013. 480 с.

7. К. Шмитт. CSS. Рецепты программирования, 3-е изд. Санкт – Петербург, 2011. 672 стр.

8. Д. Макфарланд. Новая большая книга CSS. Санкт – Петербург, 2018. 720 с.

9. Н. Закас. JavaScript для профессиональных веб-разработчиков. Санкт – Петербург, 2015. 960 с.

10. Д. Флэнаган. Javascript. Подробное руководство, 6-е изд. СанктПетербург – Москва, 2012. 1080 с. 11. Д. Флэнаган. JavaScript. Карманный справочник, 3-е изд. Москва, Санкт-Петербург, Киев, 2013. 314 с.

12. Э. Фримен, Э. Робсон. Изучаем программирование на JavaScript. Санкт – Петербург, 2015. 640 с.

14. MySQL [Электронный ресурс]. – Электрон.дан. URL: http://www.mysql.com/

15. Введение в базы данных. Часть 3. СУБД [электронный ресурс]

https://compress.ru/article.aspx?id=10551&part=pa1ext11 [Дата обращения 18.05.2020 г.]

16. Организация труда на предприятии [Электронный ресурс] // Организация труда на предприятии Режим доступа URL http://www.up-pro.ru/encyclopedia/organizaciya-truda.html [Дата обращения 10.05.20 20 г.].

17. Опасные и вредные факторы при работе с ПЭВМ [электронный ресурс] // Опасные и вредные факторы при работе с ПЭВМ Режим доступа URL http://rgrtu-640.ru/bezopasnost-zhiznedeyatelnosti/45.html [Дата обращения 18.05.2020 г.]

ПРИЛОЖЕНИЯ

Приложение А SQL – код базы данных

```
-- phpMyAdmin SQL Dump
-- version 4.6.4
-- https://www.phpmyadmin.net/
--
-- Хост: 127.0.0.1
-- Время создания: Май 26 2020 г., 09:22
-- Версия сервера: 5.7.15
-- Версия PHP: 7.0.10

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

-- База данных: `users_tracking`

-- Структура таблицы `address`

CREATE TABLE `address` (
  `id` int(11) UNSIGNED NOT NULL,
  `ip_address` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Дамп данных таблицы `address`

INSERT INTO `address` (`id`, `ip_address`) VALUES
(1, '255.255.255.255'),
(2, '255.255.255.254'),
(3, '255.255.255.253'),
(4, '192.168.255.222');

-- --------------------------------------------------------

-- Структура таблицы `build`

CREATE TABLE `build` (
  `b_id` int(11) UNSIGNED NOT NULL,
  `build_name` varchar(255) DEFAULT NULL,
  `is_deleted` tinyint(1) NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Таблица
помещений и корпусов';

-- Дамп данных таблицы `build`
```

```
INSERT INTO `build` (`b_id`, `build_name`, `is_deleted`) VALUES
(1, '1 корпус', 0),
(2, '2 корпус', 0),
(3, '3 корпус', 0),
(4, '4 корпус', 0),
(5, '5 корпус', 0),
(6, '6 корпус', 0),
(7, '7 корпус', 0),
(8, '8 корпус', 0),
(9, '9 корпус', 0),
(10, '10 корпус', 0),
(11, '11 корпус', 0),
(12, '12 корпус', 0),
(13, '13 корпус', 0),
-- --------------------------------------------------------

-- Структура таблицы `department`

CREATE TABLE `department` (
  `d_id` int(11) UNSIGNED NOT NULL,
  `dep_name` varchar(240) DEFAULT NULL,
  `phone` varchar(20) NOT NULL,
  `head` text NOT NULL,
  `is_deleted` tinyint(1) NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Дамп данных таблицы `department`

INSERT INTO `department` (`d_id`, `dep_name`, `phone`, `head`, `is_deleted`)
VALUES
(1, 'Кафедра"Общая психология"', '987654128', 'Иванова Наталья Ивановна', 0),
(544152, 'Кафедра "Информационно-вычислительные системы"', '987654123',
'Бобрышева Галина Владимировна', 0),
(544170, 'Кафедра "Вычислительная техника"', '784123682', 'Митрохин Максим
Александрович', 0),
(544206, 'Кафедра "Информационное обеспечение управления и производства"',
'+7 (8412) 36-82-39', 'Катышева Марина Александровна', 0),
(544242, 'Кафедра "Математическое обеспечение и применение ЭВМ"', '+7(412)
36-82-26    ', 'Макарычев Петр Петрович', 0),
(544368, 'Кафедра "Высшая и прикладная математика"', '+7(8412)36-82-59',
'Бойков Илья Владимирович', 0),
(544842, 'Кафедра "Математика и суперкомпьютерное моделирование"',
'+7(8412)36-80-96', 'Смирнов Юрий Геннадьевич', 0),
(31429808, 'Факультет вычислительной техники', '+7(8412)36-82-48', 'Фионова
Людмила Римовна', 0),
```

```
(123595545, 'Кафедра "Компьютерные технологии"', '+7(8412)36-80-68',
'Горбаченко Владимир Иванович', 0),
(161624869, 'Управление информатизации', '0', '', 0),
(352685446, 'Политехнический институт', '+7(8412)36-84-56', 'Козлов Геннадий
Васильевич', 0);

-- --------------------------------------------------------

-- Структура таблицы `rooms`

CREATE TABLE `rooms` (
  `r_id` int(11) UNSIGNED NOT NULL,
  `room_name` text NOT NULL,
  `building_id` int(11) UNSIGNED NOT NULL,
  `address_id` int(11) UNSIGNED DEFAULT NULL,
  `owner` int(11) UNSIGNED DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Дамп данных таблицы `rooms`

INSERT INTO `rooms` (`r_id`, `room_name`, `building_id`, `address_id`, `owner`)
VALUES
 (6848, '7-203', 7, 1, 31429808),
(6849, '7-309', 7, 1, 31429808),
(6850, '7-324', 7, 1, 31429808),
(6851, '7-107', 7, 1, 31429808),
(6852, '7-303в', 7, 1, 31429808),
(6853, '7-305', 7, 1, 31429808),
(6854, '7-305а', 7, 1, 31429808),
(6855, '7-306', 7, 1, 31429808),
(6856, '7-316', 7, 1, 544170),
(6857, '7-317', 7, 1, 544170),
(6858, '7-319', 7, 1, 544170),
(6859, '7-320', 7, 1, 31429808),
(6860, '7-322', 7, 1, 544170),
(6861, '7-402', 7, 1, 31429808),
(6862, '7-405', 7, 1, 31429808),
(6863, '7-406', 7, 1, 31429808),
(6864, '7-406а', 7, 1, 31429808),
(6865, '7-409', 7, 1, 31429808),
(6866, '7-413', 7, 1, 31429808),
 (6894, '7-223', 8, 1, 31429808),
(6895, '7-224', 7, 1, 31429808),
(6896, '7-225', 7, 1, 31429808),
(6897, '7-225а', 7, 1, 31429808),
```

```
(6912, '8-705', 8, 3, 31429808),
(6937, '8-706', 8, 1, 31429808),
(6953, '8-703а', 8, 1, 31429808),
(6954, '8-703б', 8, 1, 31429808),
(6955, '8-703в', 8, 1, 544368),
(6956, '8-703г', 8, 1, 544368),
(6957, '8-704', 8, 1, 544368),
(6978, '9-503', 9, 1, 31429808);

-- --------------------------------------------------------
-- Структура таблицы `status`

CREATE TABLE `status` (
  `status_id` tinyint(3) UNSIGNED NOT NULL,
  `status_name` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Дамп данных таблицы `status`

INSERT INTO `status` (`status_id`, `status_name`) VALUES
(1, 'Оператор'),
(2, 'Администратор');

-- --------------------------------------------------------

-- Структура таблицы `tree`

CREATE TABLE `tree` (
  `id` int(11) UNSIGNED NOT NULL,
  `ancestor` int(11) UNSIGNED NOT NULL,
  `descendant` int(11) UNSIGNED NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Дамп данных таблицы `tree`

INSERT INTO `tree` (`id`, `ancestor`, `descendant`) VALUES
(1, 31429808, 544152),
(3, 31429808, 544170),
(4, 31429808, 544206),
(5, 31429808, 544242),
(6, 31429808, 544368),
(7, 31429808, 544842),
(8, 31429808, 123595545),
(9, 31429808, 544842),
(10, 31429808, 123595545),
```

```
(11, 352685446, 31429808),
(12, 352685446, 352685446);

-- --------------------------------------------------------

-- Структура таблицы `users`

CREATE TABLE `users` (
  `user_id` int(11) UNSIGNED NOT NULL,
  `login` varchar(32) NOT NULL,
  `password` varchar(200) NOT NULL,
  `name` varchar(200) NOT NULL,
  `email` varchar(200) DEFAULT NULL,
  `status_id` tinyint(3) UNSIGNED NOT NULL,
  `active` tinyint(1) UNSIGNED NOT NULL DEFAULT '1'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Дамп данных таблицы `users`

INSERT INTO `users` (`user_id`, `login`, `password`, `name`, `email`, `status_id`,
`active`) VALUES
(44, 'admin', 'b58438775e431c88164e48b8a225f3af', 'test', 'email', 2, 1),
(45, 'admin11', '71ea76dfbf3d90f26ee2c486ea94745b', 'FIO', 'email', 1, 1),
(46, 'oksaalla@yandex.ru', '0eaa2793389d4fae91b25126db4e7cfc', 'FIO', 'email', 2,
1);

-- Индексы сохранённых таблиц

-- Индексы таблицы `address`
--
ALTER TABLE `address`
  ADD PRIMARY KEY (`id`);

-- Индексы таблицы `build`
--
ALTER TABLE `build`
  ADD PRIMARY KEY (`b_id`),
  ADD KEY `is_deleted` (`is_deleted`);

-- Индексы таблицы `department`
--
ALTER TABLE `department`
  ADD PRIMARY KEY (`d_id`),
  ADD KEY `is_deleted` (`is_deleted`);
```

```
-- Индексы таблицы `rooms`
--
ALTER TABLE `rooms`
  ADD PRIMARY KEY (`r_id`),
  ADD KEY `building_id` (`building_id`),
  ADD KEY `address_id` (`address_id`),
  ADD KEY `floor_id` (`owner`);

-- Индексы таблицы `status`
--
ALTER TABLE `status`
  ADD PRIMARY KEY (`status_id`);

-- Индексы таблицы `tree`
--
ALTER TABLE `tree`
  ADD PRIMARY KEY (`id`),
  ADD KEY `ancestor` (`ancestor`),
  ADD KEY `descendant` (`descendant`);

-- Индексы таблицы `users`
--
ALTER TABLE `users`
  ADD PRIMARY KEY (`user_id`),
  ADD KEY `status_number` (`status_id`);

-- AUTO_INCREMENT для сохранённых таблиц

-- AUTO_INCREMENT для таблицы `address`
ALTER TABLE `address`
  MODIFY `id` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=5;
-- AUTO_INCREMENT для таблицы `build`
ALTER TABLE `build`
  MODIFY `b_id` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=105;
-- AUTO_INCREMENT для таблицы `department`
ALTER TABLE `department`
  MODIFY `d_id` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=352685447;
-- AUTO_INCREMENT для таблицы `rooms`
ALTER TABLE `rooms`
  MODIFY `r_id` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=6979;
-- AUTO_INCREMENT для таблицы `status`
```

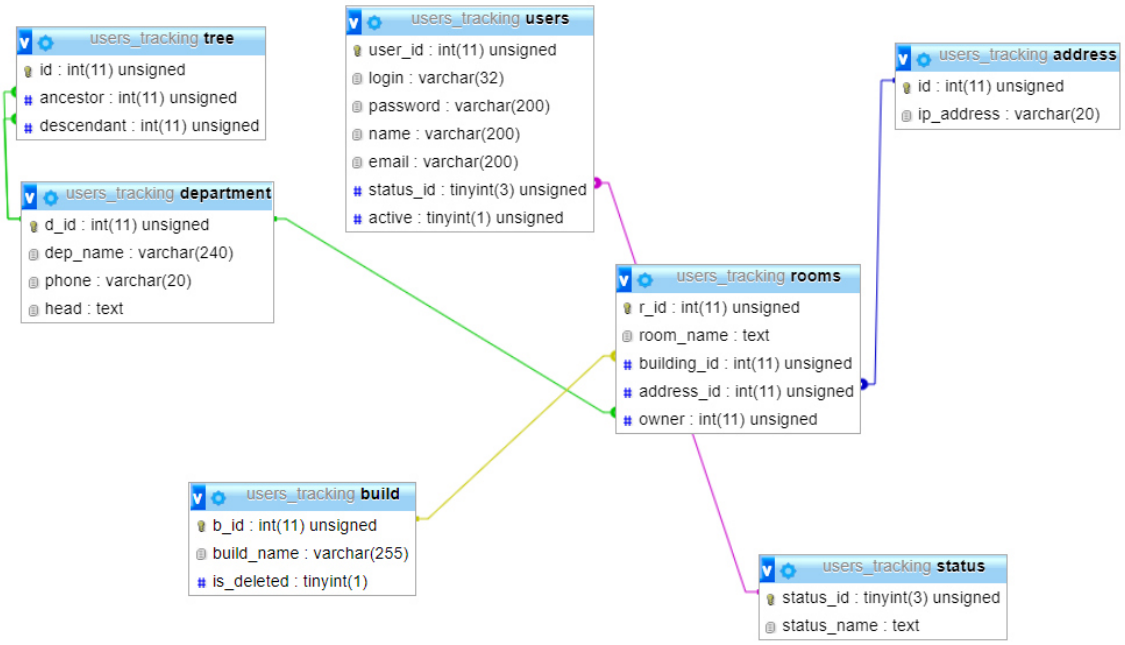
```
ALTER TABLE `status`
  MODIFY `status_id` tinyint(3) UNSIGNED NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=3;
-- AUTO_INCREMENT для таблицы `tree`
ALTER TABLE `tree`
  MODIFY `id` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=13;
-- AUTO_INCREMENT для таблицы `users`
ALTER TABLE `users`
  MODIFY `user_id` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=47;
-- Ограничения внешнего ключа сохраненных таблиц
-- Ограничения внешнего ключа таблицы `rooms`
ALTER TABLE `rooms`
  ADD CONSTRAINT `rooms_ibfk_1` FOREIGN KEY (`owner`) REFERENCES
`department` (`d_id`) ON UPDATE CASCADE,
  ADD CONSTRAINT `rooms_ibfk_2` FOREIGN KEY (`address_id`)
REFERENCES `address` (`id`) ON UPDATE CASCADE,
  ADD CONSTRAINT `rooms_ibfk_3` FOREIGN KEY (`building_id`)
REFERENCES `build` (`b_id`) ON UPDATE CASCADE;

-- Ограничения внешнего ключа таблицы `tree`
ALTER TABLE `tree`
  ADD CONSTRAINT `tree_ibfk_1` FOREIGN KEY (`ancestor`) REFERENCES
`department` (`d_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `tree_ibfk_2` FOREIGN KEY (`descendant`) REFERENCES
`department` (`d_id`) ON DELETE CASCADE ON UPDATE CASCADE;

-- Ограничения внешнего ключа таблицы `users`
ALTER TABLE `users`
  ADD CONSTRAINT `users_ibfk_1` FOREIGN KEY (`status_id`) REFERENCES
`status` (`status_id`) ON UPDATE CASCADE;
```

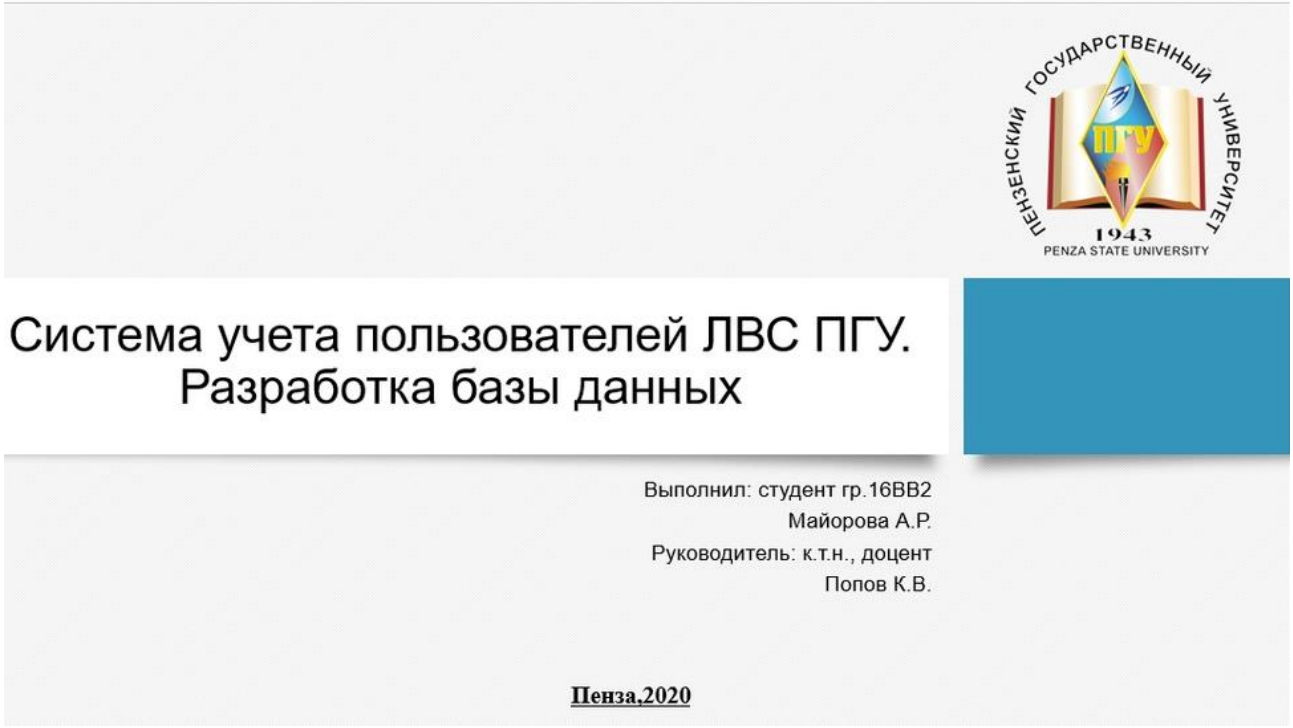
Приложение Б

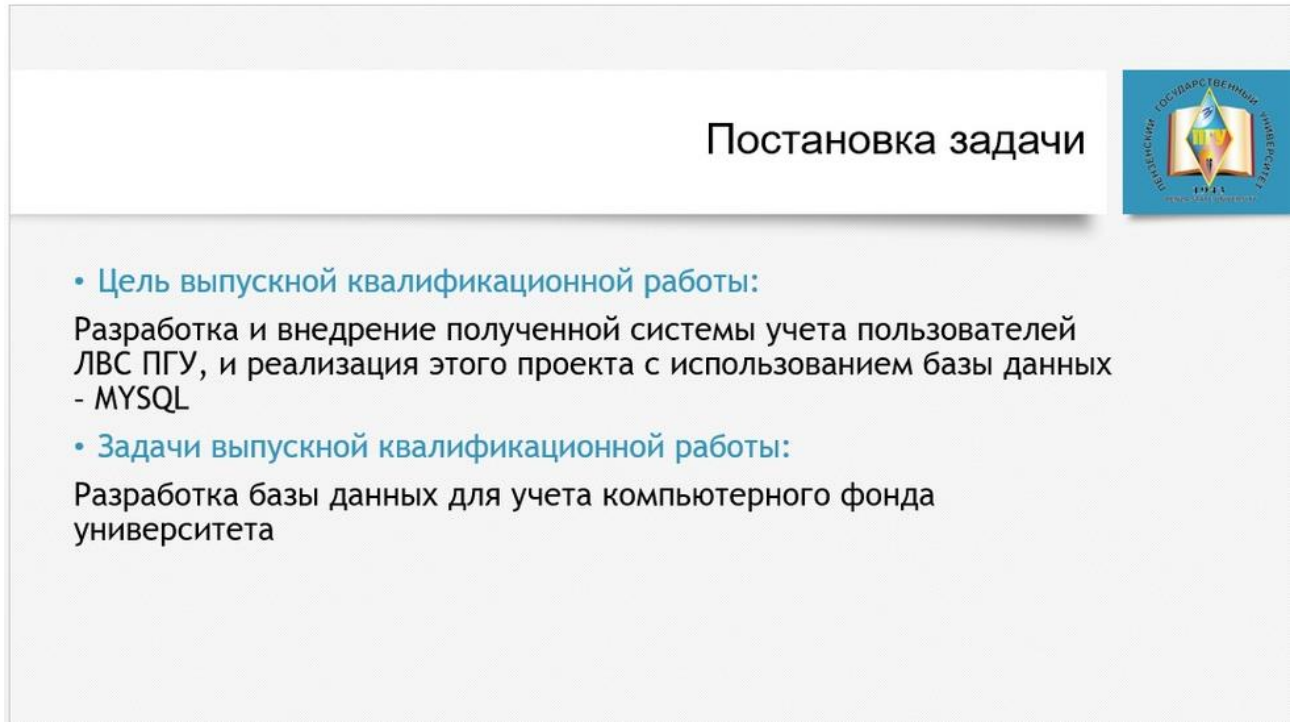
Модель базы данных



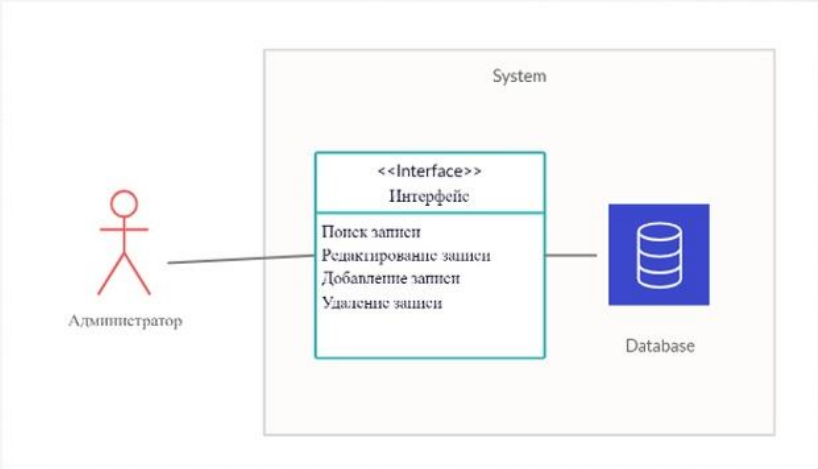
Приложение В

Презентация





Структура взаимодействия компонентов



Выбор средств и технологий разработки

Средства разработки

- Язык программирования PHP 7
- СУБД MYSQL
- Bootstrap

Программы

- Локальный сервер MAMP
- PhpMyAdmin

Формирование ролей

Определение сущностей основных таблиц

Корпуса	Подразделения	Аудитории	ip-адреса
Номер корпуса	Название подразделения	Номер	Адрес
	Руководитель подразделения		
	Номер телефона		

Определение сущностей вспомогательных и дополнительных таблиц

Вспомогательные таблицы

Пользователи	Статус
Логин	Код статуса
Пароль	Название
Имя	
Email*	
Статус	
active	

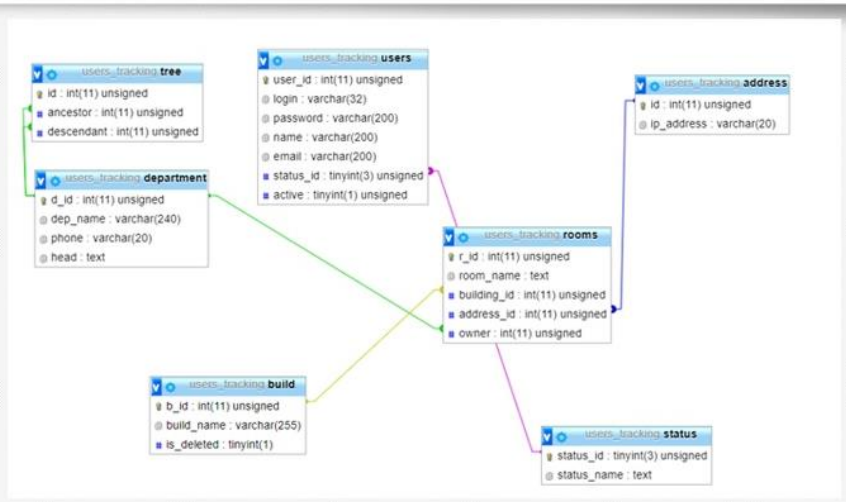
Дополнительные таблицы

Иерархия
Предок
Потомок

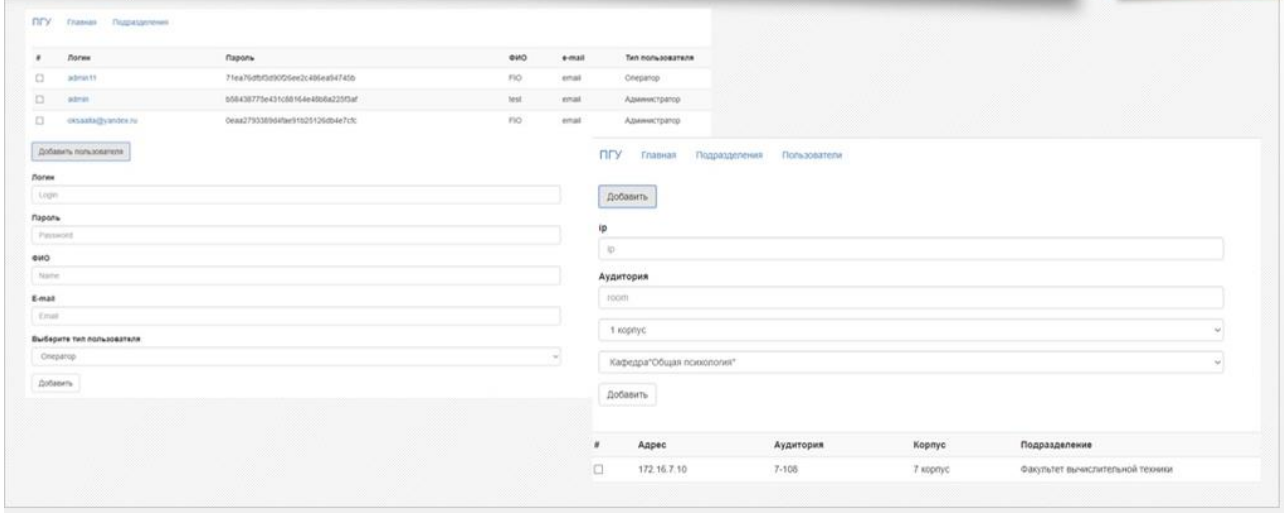
Создание первичных ключей и индексов

Корпуса	Подразделения	Аудитории	ip-адреса
Id	id	id	id
build_name	dep_name	room_name	ip_address
is_deleted	head	owner	
	phone	build_id	
	is_deleted	address_id	

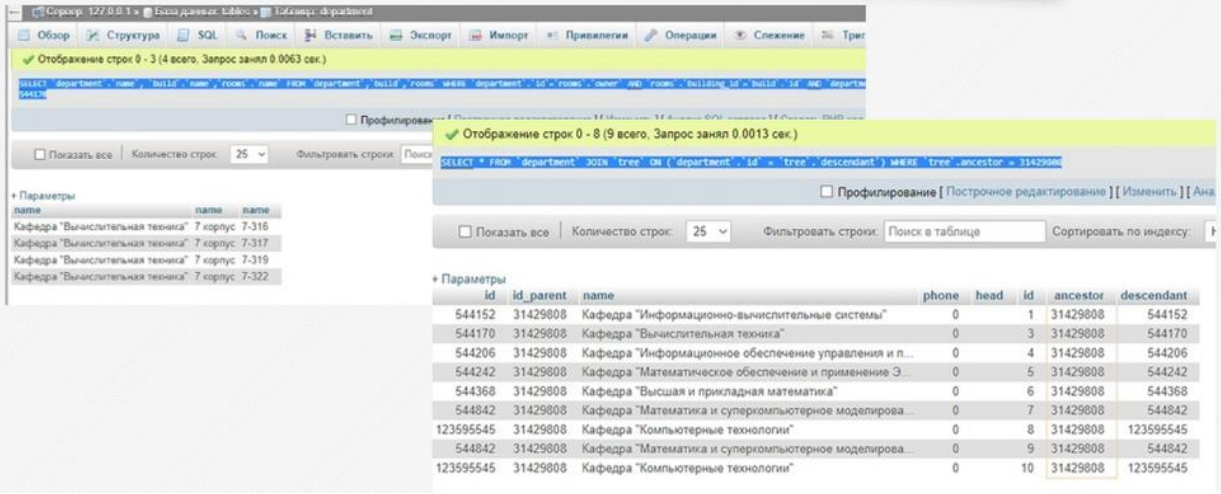
Модель базы данных



Интерфейсы



SQL-запросы



Заключение

В результате выполнения работы:

- Спроектирована и создана база данных для хранения информации
- Разработан интерфейс добавления данных в таблицы
- Разработаны SQL-запросы для выборки из таблиц

БЛАГОДАРЮ ЗА ВНИМАНИЕ