

*Джорджевич Н.,
студент 3 курс, институт
«Инженерных и цифровых технологий»
Белгородский Государственный национальный исследовательский
университет
Россия, г. Белгород*

*Гречишкина Е.И.,
студент 3 курс, институт
«Инженерных и цифровых технологий»
Белгородский Государственный национальный исследовательский
университет
Россия, г. Белгород*

*Гурьянова И.В.,
старший преподаватель кафедры
«Прикладной информатики и информационных технологий»
Белгородский Государственный национальный
исследовательский университет
Россия, г. Белгород*

**РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ "СИСТЕМА УЧЕТА
И КОНТРОЛЯ ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ"
С ИСПОЛЬЗОВАНИЕМ КЛИЕНТ-СЕРВЕРНОЙ ТЕХНОЛОГИИ**

Аннотация: Статья посвящена вопросу о правильном проектировании базы данных (БД) для учета и контроля выполнения лабораторных работ студентов, и разработке веб-приложения для получения доступа к хранящимся в ней данным, а также удаления, добавления и модификации данных в БД. Разработка такого приложения носит практический характер, поскольку решает задачу автоматизации учебного процесса в высших учебных заведениях.

Ключевые слова: БД, проектирование, СУБД, веб-приложение, студент, дисциплина, схема, интерфейс.

Annotation: The article is devoted to the question of the correct design of the database (DB) to account for and control the implementation of laboratory works students, and developing web applications for gaining access to the data stored in it, and also delete, add and modify data in the database. The development of such an application is practical, because it solves the problem of automation of the educational process in higher educational institutions.

Key words: DB, design, DBMS, web application, student, discipline, scheme, interface.

В настоящее время существует тенденция к внедрению информационных систем во все сферы общества, одной из наиболее важных, является сфера образования, в частности высшие учебные заведения. Автоматизация процессов отслеживания успеваемости студентов, сдачи лабораторных работ, осуществления доступа к учебным материалам и многих других является наиболее приоритетной. Для разработки таких систем существуют разные подходы и технологии. Наиболее простой и современной является клиент-серверная технология.

Главные идеи современных информационных технологий основываются на концепции, в соответствии с которой данные нужно организовывать в базы данных, целью которых является адекватное отображение изменяющегося реального мира и удовлетворение информационных нужд пользователя. Управление этими базами данных осуществляется под управлением специальных программных комплексов, называемых системами управления базами данных (СУБД).

Наиболее удобными и сравнительно простыми информационными системами, способными справиться с постоянным увеличением объема и структурной сложностью хранимых данных, а также с расширением круга пользователей являются реляционные СУБД. С целью обеспечения

одновременного доступа к данным множества пользователей, часто расположенных далеко друг от друга и от серверов баз данных, созданы сетевые многопользовательские версии БД, основанные на реляционной структуре.

Веб-приложение — это программа, которая запускается в интернет-браузере, основные алгоритмы которой обрабатываются удаленным веб-сервером. Для запуска веб-приложений пользователю не нужно устанавливать никаких дополнительных программ, оно запускается на практически любом устройстве с доступом в интернет или к веб-серверу. С его помощью посредством удобного и понятного интерфейса, достаточно легко осуществлять доступ к базе данных, находящейся на удаленном сервере.

На начальном этапе разработки веб-приложения проводится исследование предметной области, определение и описание его функционала.

Разрабатываемая система должна функционировать следующим образом. Студенты групп в рамках учебного плана изучают соответствующие дисциплины по которым выполняют разные виды работ согласно своему варианту. Виды работ могут быть следующие: лабораторные, курсовые и экзаменационные работы. По каждому из видов работ студентом пишется отчет, который в дальнейшем проверяется преподавателем, ведущим занятия по данной дисциплине. Также в рамках системы ведется контроль за посещаемостью студентами занятий.

Для организации вышеописанной системы выделены следующие базовые объекты:

- «Год поступления» - список годов поступления групп студентов;
- «Группа» - группы студентов, поступившие на обучение;
- «Студент» - содержит информацию о студентах групп;
- «Дисциплина» - список дисциплин, изучаемых студентами в рамках учебного плана.

- «Преподаватель» - содержит информацию о преподавателях;
- «Семестр» - список семестров обучения;
- «Задание» - тексты заданий по дисциплинам;

- «Отчет» - файл отчета о проделанной работе и оценка за него;
- «Номер варианта» - индивидуальный вариант студента, согласно которому тот выполняет задание;

- «Пропуски» - содержит информацию о количестве и дате пропуска занятий студентом;

- «Форма контроля» - несет информацию о форме контроля студентов.

Дальнейшее проектирование базы данных включает в себя несколько этапов:

1. Инфологическое моделирование. Его целью является обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Именно поэтому на этом этапе строится наглядная модель, доступная широкому кругу пользователей. Наиболее популярным является подход моделирования «сущность-связь».

Используя данный подход выделены следующие сущности и их атрибуты:

- Год поступления. Атрибуты: код года поступления, год поступления;
- Группа. Атрибуты: код группы, код года поступления группы;
- Студент. Атрибуты: номер студенческого, фамилия, имя, отчество, пароль, код группы, фотография;

- Дисциплина. Атрибуты: код дисциплины, название дисциплины;
- Преподаватель. Атрибуты: номер удостоверения, фамилия, имя, отчество;

- Семестр. Атрибуты: код семестра, название семестра;
- Задание. Атрибуты: код задания, срок сдачи, файл задания, код семестра-дисциплины, код формы контроля, порядковый номер задания, дополнительный текст задания;

- Отчет. Атрибуты: код отчета, код задания, код студента, дата и время сдачи, файл отчета, оценка;

- Номер варианта. Атрибуты: код студента, код семестра-дисциплины, номер варианта;
- Пропуски. Атрибуты: код пропуска, код студента, дата пропуска, количество пропущенных часов, код семестра;
- Форма контроля. Атрибуты: код формы контроля, название формы контроля.

2. Даталогическое проектирование. Его цель разработка логической структуры базы данных. Причем логическая структура базы данных, а также сама заполняемая данными база данных являются отображением реальной предметной области. Спроектировать логическую структуру базы данных означает определить все информационные единицы базы данных и связи между ними, задать их имена, типы и другие требуемые характеристики.

Некоторые зависимости между атрибутами отношений являются нежелательными из-за побочных эффектов и аномалий, которые они вызывают при модификации БД. При этом под процессом модификации БД мы понимаем внесение новых данных в БД или удаление некоторых данных из БД, а также обновление значений некоторых атрибутов.

Так как на данном этапе модель базы данных содержит связи «многие-ко-многим», то необходимо привести модель к третьей нормальной форме.

Для этого между таблицами «Преподаватель» и «Дисциплина», «Дисциплина» и «Семестр», создаются дополнительно таблицы «Преподаватель_Дисциплина», «Семестр_Дисциплина», а затем между таблицами «Группа» и «Семестр_Дисциплина» таблица «Группа_Семестр_Дисциплина» соответственно.

В результате приведения описанной модели к третьей нормальной форме получается модель, которая в последствии проектируется в среде ERwin. Спроектированная модель приведена на рисунке 1.

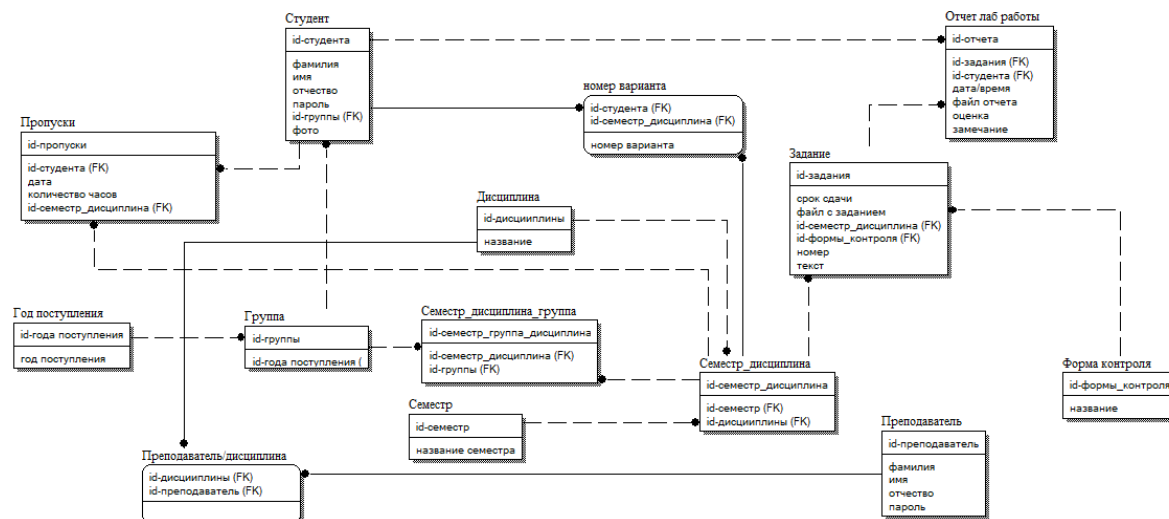


Рисунок 1. Логическая модель БД

3. Физическое проектирование — создание схемы базы данных для конкретной СУБД. Специфика конкретной СУБД может включать в себя ограничения на именованые объектов базы данных, ограничения на поддерживаемые типы данных и т. п. Кроме того, специфика конкретной СУБД при физическом проектировании включает выбор решений, связанных с физической средой хранения данных (выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, методов доступа к данным), создание индексов и т. д.

Определяется имя, тип и размер поля. Определяется первичный ключ, внешние ключи и поля индексирования (для быстрого поиска) в каждой из таблиц

Для каждой связи определяются условия целостности при операциях добавления, изменения или удаления данных в родительской и дочерних таблицах. Могут использоваться следующие варианты условий: отсутствие проверки; запрет операции; каскадное выполнение операции обновления или удаления данных сразу в нескольких связанных таблицах.

Физическая модель создана в соответствии с выбранной СУБД проиллюстрирована на рисунке 2.

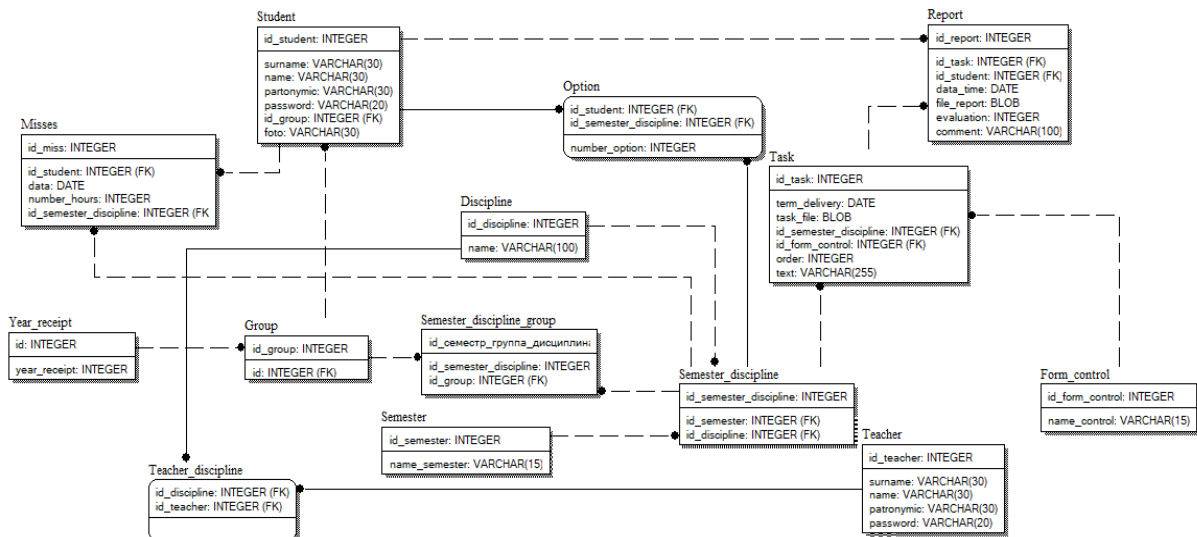


Рисунок 2. Физическая модель БД

После проектирования БД, проводится проектирование самого приложения. Первым шагом проектирования является создание функциональной схемы приложения. Разработанная схема представлена на рисунке 3.

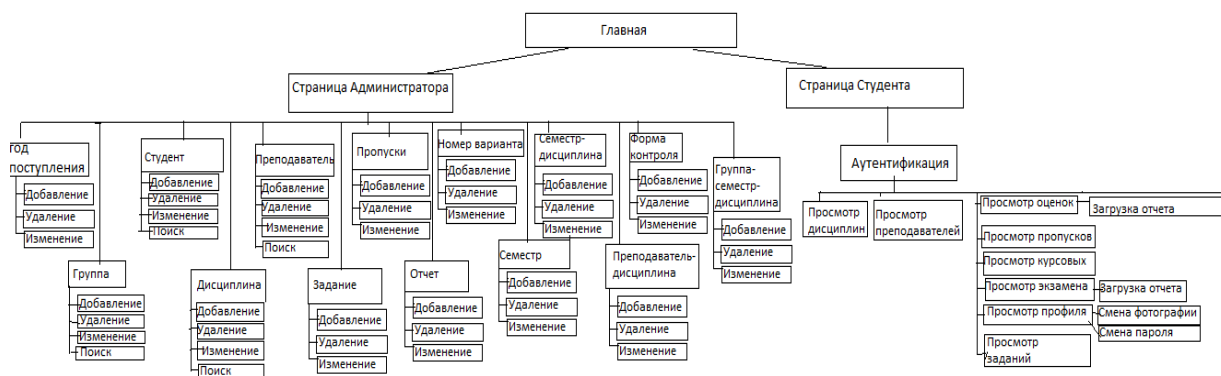


Рисунок 3. Функциональная схема веб-приложения

Вторым шагом проектирования является построение модульной схемы.

Веб-приложение содержит PHP-скрипты для поддержания работы всего его функционала, где реализованы запросы к базе данных, такие как вывод, удаление, добавление, изменение записей таблицы и т.п. Приложение создавалось с помощью фреймворка CodeIgniter, который использует подход Model-View-Controller. Главный контроллер приложения собрал в себя все функции обращения страниц к базе данных. В свою очередь все функции получения данных из базы данных собрала в себя модель. Модульная схема приложения представлена на рисунках 4.

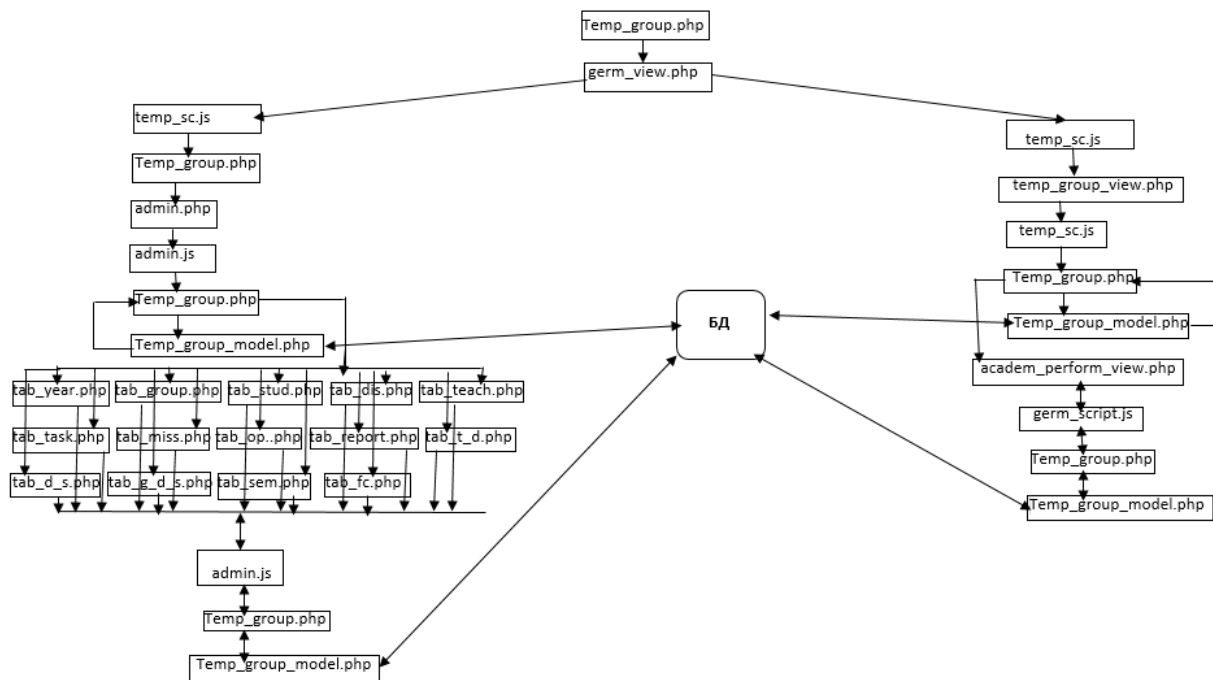


Рисунок 4. Модульная схема веб-приложения

Разработка интерфейс для Web-приложения.

В приложении будет два основных типа страниц для работы с двумя группами пользователей: администратор и студент. Первый тип страниц предназначен для ведения базы данных администратором. На страницах этого типа будут отображаться все записи таблиц и кнопки для добавления, удаления, вставки и поиска записей. Схематичное изображение представлено на рисунке 5.

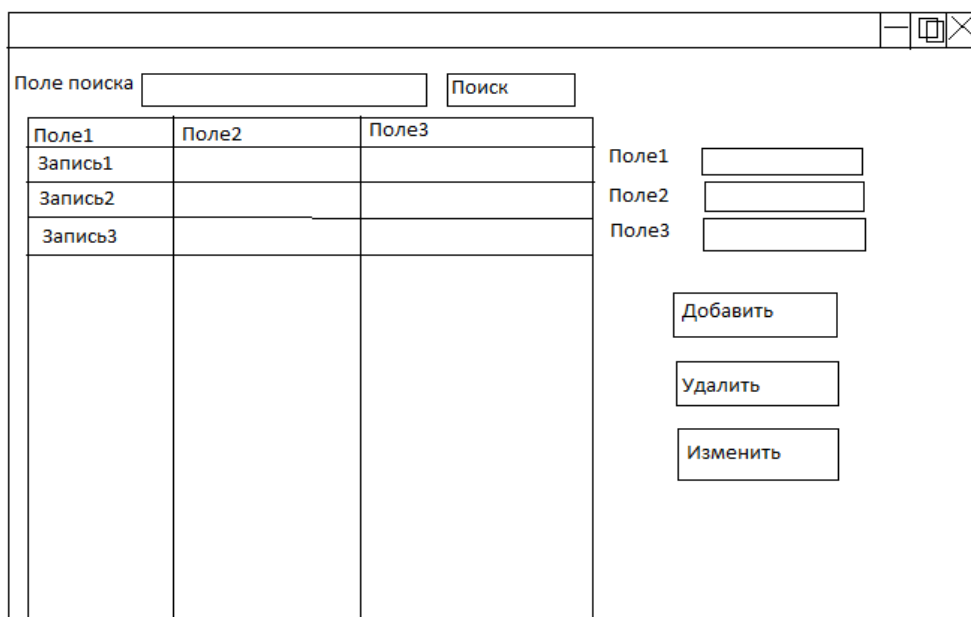


Рисунок 5. Схематичное изображение веб-приложения

Второй тип страниц предназначается для получения студентом данных о изучаемых дисциплинах, текстах работ, оценках за отчеты по работе, его пропусках и т.д. Страницы этого типа будут разделены на блоки: меню вкладок, дисциплины, преподаватели ведущие занятия по выбранному предмету, и центральный блок который будет отображать информацию в зависимости от выбранной вкладки.

На рисунке 6 схематично изображен макет Web-приложения на примере вкладки оценок.

Студент:

Оценки Пропуски Курсовые Тексты работ Экзамен Профиль

Дисциплины:

Дисциплина: Загрузка отчета

Поле1	Поле2			
Запись1				
Запись2				

Преподаватели:

Рисунок 6. Схематичное изображение Web-приложения

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Руководство по проектированию реляционных баз данных [Электронный ресурс]. URL: <https://habr.com/post/193136/> (дата обращения 20.11.2018).
2. PostgreSQL, Блог компании Mail.ru Group [Электронный ресурс]. URL: <https://habr.com/company/mailru/blog/248689/> (дата обращения: 22.11.2018).
3. Основы CSS [Электронный ресурс]. URL: <https://html5book.ru/osnovy-css/> (дата обращения: 05.12.2018).
4. Изучение веб-разработки. JavaScript [Электронный ресурс]. URL: https://developer.mozilla.org/ru/docs/Learn/JavaScript/Первые_шаги/What_is_JavaScript (дата обращения: 08.12.2018).

5. Основы JavaScript [Электронный ресурс].
URL:<https://html5book.ru/osnovy-javascript/> (дата обращения: 08.12.2018).
6. «Разработка веб-приложений на PHP и MySQL», Лаура Томсон, Люк Веллинг / Пер. с англ. – 2-е изд. – М.: ДиаСофт, 2003 – 674 с.
7. CodeIgniter и Ajax посредством jQuery: статья Ховарда Голдберга [Электронный ресурс]. <https://www.ibm.com/developerworks/ru/library/wa-aj-codeigniter/>(дата обращения: 11.12.2018).