# Департамент образования и науки Кемеровской области

Государственное образовательное учреждение

среднего профессионального образования

Профессиональный колледж г. Новокузнецка

Специальность: *230115 Программирование в компьютерных систе*мах

## КУРСОВОЙ ПРОЕКТ

**Дисциплина** МДК.02.02 Технология разработки и защиты баз данных

**Тема** «Расписания и учебные планы»

( КП.230115.13.01)

Шифр

Руководитель

И. С. Колмогорцев

Выполнил студент группы ПР 11-03

Имуков В.М.

Новокузнецк, 2014

Содержание

Введение

1. Постановка задачи

2. Проектирование базы данных

2.1 Инструменты используемые для создания базы данных

2.2 Скрипт на создание и заполнение базы данных

2.3 Главное окно приложения

3. Руководство пользователя

3.1 Навигация в программе

3.2 Добавление новой записи

3.3 Удаление записей

Заключение

Список литературы

Введение

Целью данного курсового проекта является собственная разработка базы данных, ее заполнение, создания пользовательского приложения, и подготовка руководства для пользователей.

Курсовой проект должен содержать:

* Описание структуры базы данных;
* Приведение ее к 3НФ;
* Создание базы на основе описанной структуры;
* Создание триггеров и генераторов для ключевых полей;
* Созданное приложение в IDE;
* Удобный пользовательский интерфейс;
* Добавление, удаление, и обновление записей;
* Фильтрацию.

В качестве темы курсового проекта, была выбрана «Расписание и учебные планы».

В данной базе будут храниться:

* Данные о группах, их дате основания;
* Предметы, и их количество часов на семестр;
* Расписания занятий для разных групп;
* Дни недели занятий для групп;
* Списки тем для разных групп, их дата, день недели.

1. Постановка задачи

Базу данных с начало необходимо продумать и спроектировать таким образом, чтобы уменьшить дальнейшие изменения в ней, т.е. на начальном этапе необходимо продумать, сколько полей и таблиц будет в базе, какого они будут типа, какой длинные будут поля, и сколько необходимо будет реализовать связей. Так же необходимо решить, какие поля не нужны, а какие позже нужно будет добавить. Для решения таких задач желательно записывать все наработки и задумки либо на листке бумаги, либо в текстовом редакторе или текстовом процессоре. Так же необходимо описать на какой СУБД будет реализована база данных. Желательно полям дать адекватные названия, и продумать их аналоги на английском языке.

СУБД должна быть простой, но эффективной, ее установка и настройка не должны отнимать много времени, а так же нужна поддержка основных стандартов баз данных.

Скрипт нужно написать на языке SQL, а так же необходимо реализовать авто заполнение ключевых полей.

Программа для работы с базой данных, служащая для ее создания, и начального заполнения должна быть простой, понятной, желательно на русском языке, с поддержкой SQL запросов, вставки, удаления, и обновления записей.

Для создания пользовательской программы нужно описать, на каком языке будет она написана, будет ли использоваться IDE, будет использоваться модель ООП или функций, как будут открываться таблицы, и в какой последовательности, а так же на какой СУБД была создана база, для использования нужного метода подключения и «общения» с ней. Так же нужно описать, требуется ли поиск и сортировка, а так же реализовать сложные запросы или вставки. Пользовательский интерфейс играет большую роль в написании программы. Если программа будет быстрой, но с трудным интерфейсом, то пользователь не сможет ее эффективно и качественно использовать. Так же необходимо реализовать хорошее быстродействие и запуск базы через путь указанный в Ini файле.

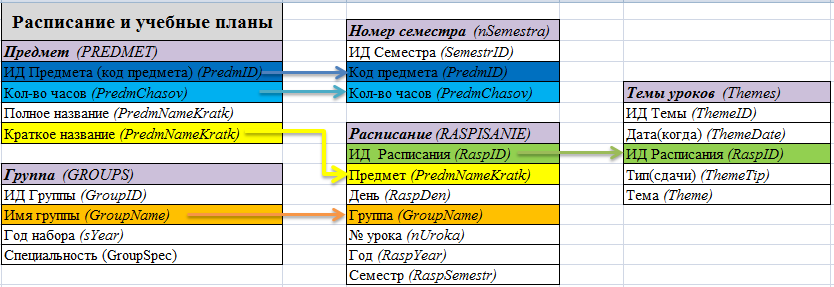
После создания приложения, его нужно протестировать и отладить. Потом нужно исправить найденные ошибки и провести рефакторинг кода, для удобной дальнейшей поддержки программы.

1. Проектирование базы данных

При проектировании и разработке базы данных использовалась схема связей, приведенная на Рисунке 1. На схеме показаны наименования полей и их же аналоги на английском языке для более удобной ориентации в базе.

Чтобы не засорять базу данных не нужными полями они были разбиты на отдельные таблицы. Связи были организованы за счет LookUp полей.

В каждой таблице имеется идентификатор, который будет присвоен записи, тем самым сделав ее индивидуальной.



*Рис. 1 – Схема 3-й нормальной формы*

2.1 Инструменты используемые для создания базы данных

В качестве редактора для написания скрипта использовался Notepad ++. Это бесплатный текстовый редактор с подсветкой синтаксиса.

Для удобного администрирования и быстрой корректировки базы данных использовался IBExpert. Он позволяет быстро подключиться к базе данных, создать или удалить пользователя, базы можно легко и удобно удалять и добавлять за счет «интуитивно» понятного интерфейса, а так же выполнение небольших SQL запросов с помощью Script Executive, которая встроена в эту программу.

В качестве сервера баз данных выбран Firebird. Он более простой в установке и настройке, нежели MySQL. Пользователи хранятся в отдельном зашифрованном файле. А файл с базой данных легко подключить в IBExpert или в пользовательском приложении.

Приложение для простой и удобной работы с базой данных написано на IDE Lazarus. В качестве языка программирования данная IDE использует ObjectPascal. А значит человек который до этого работал в Delphi 2007 или в TurboPascal без труда сможет самостоятельно разобраться и написать данное приложение.

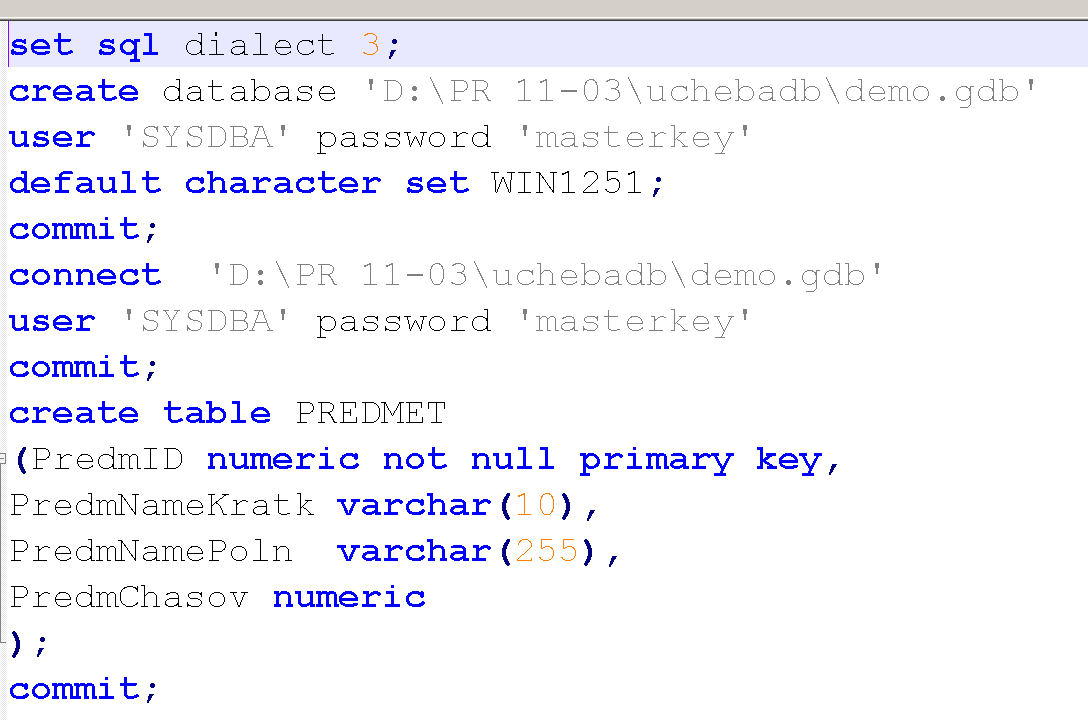
Вся сопутствующая документация к Lazarus, IBExpert, Firebird есть в интернете на официальном сайте в разделе с документацией, либо на сторонних ресурсах по разработке приложений и баз данных.

2.2 Скрипт на создание и заполнение базы данных

На Рисунке 2 приведена часть скрипта на создание и заполнение базы данных. Там указано местоположение файла базы, пользователь базы, пароль пользователя, а так же кодировка данных для базы.

После ее создания идет подтверждение на запись данных и подключение к вновь только что созданной базе.

Затем, выполняется запрос на создание таблицы, там же перечислены поля, их тип и длина поля.



*Рис. 2 – Часть скрипта на создание базы*

После создания базы и полей, необходимо создать триггеры и генераторы для авто-заполнения полей-идентификаторов.

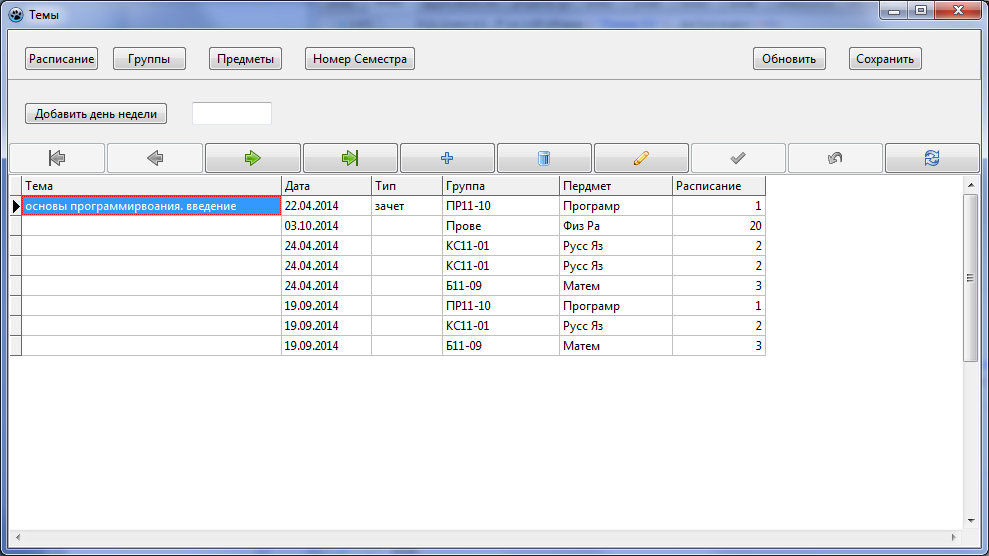
Как только все это будет создано и отлажено, следует начать заполнение базы. Для более простого решения данной задачи, можно написать небольшой скрипт или дополнить данный. Суть его в том, что он будет добавлять начальные данные в таблицы.

2.3 Главное окно приложения

После создания 3НФ и самой базы данных, для ее дальнейшего использования создается приложение. В данном случае оно написано в IDE Lazarus.

Приложение выполнено так, что каждое окно открываемое по нажатии определенной кнопки отвечает за отдельную таблицу.

Кнопки «Refresh» и «Commit» нужны для обновления и подтверждения записей базы данных.



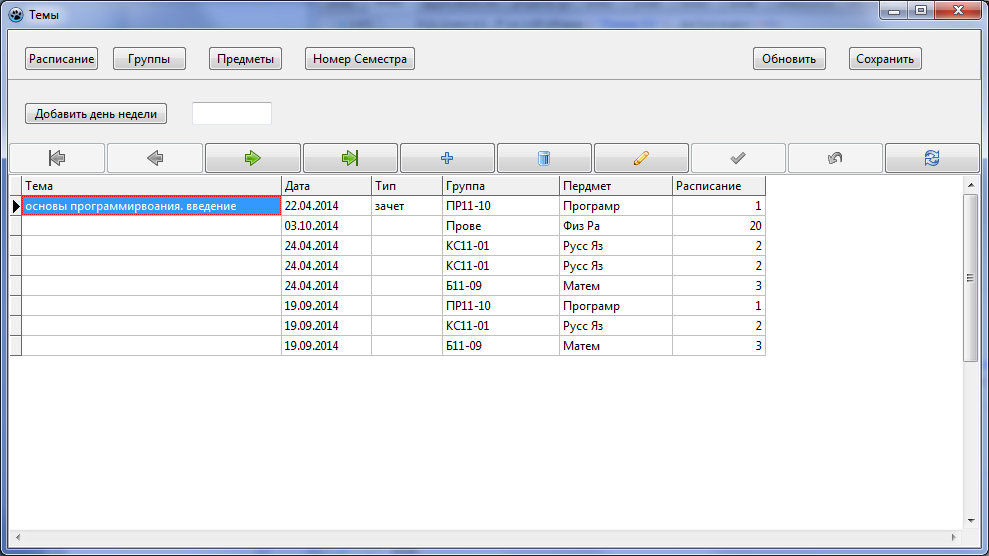
*Рис. 3 – Главное окно приложения, таблицы «Темы»*

3. Руководство пользователя

Запустив программу, и в случае успешного подключения и доступа к базе данных, появится главное окно программы.

В этом окне представлен интерфейс для работы с главной таблицей, а так же кнопки для перехода к другим окнам по работе с таблицами и сохранение, обновление текущей таблицы.

Главное окно представлено на рисунке 4.

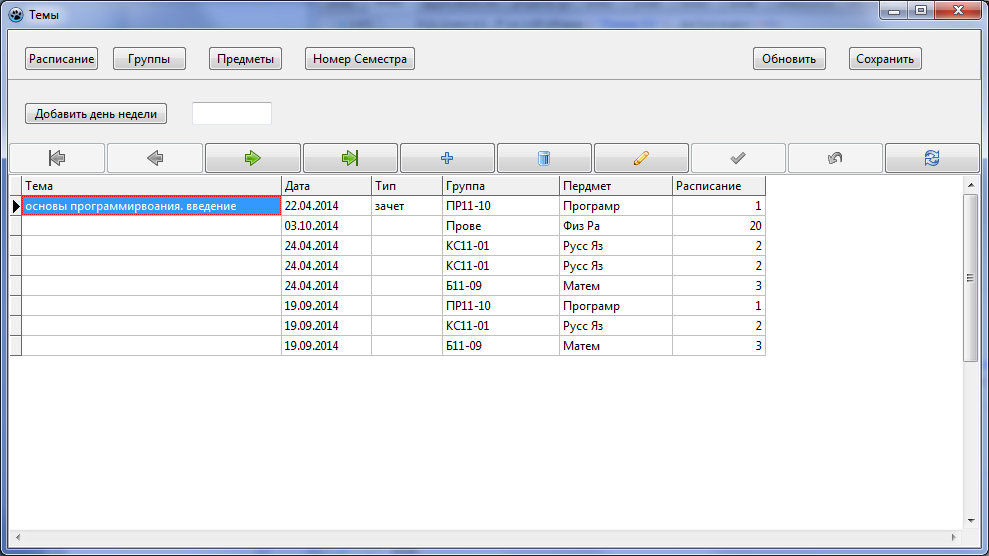


*Рис. 4 – Главное окно приложения, таблицы «Темы»*

3.1. Навигация в программе

Навигацию по записям таблицы можно осуществить либо с помощью щелчка по записи левой кнопкой мыши, либо с помощью кнопок с зелеными стрелками «Влево» или «Вправо».

Кнопки со стрелками «с вертикальной линией» служат для перехода либо к самой первой записи, либо к самой последней в данной таблице.

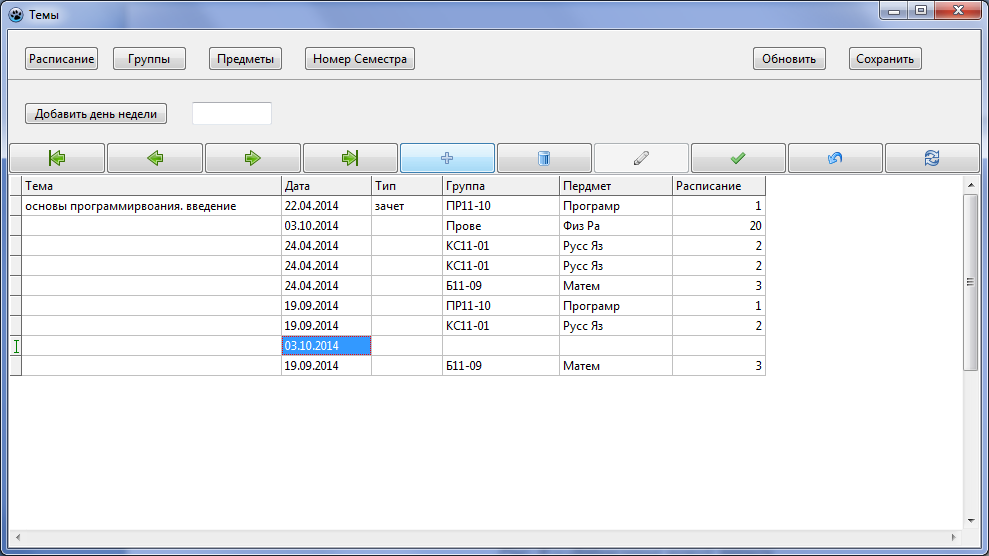


*Рис. 5 – Элементы навигации*

3.2. Добавление новой записи

Для добавления новой записи в данную таблицу нужно нажать на кнопку со значком синего «Плюса». После чего, в окне с записями появится новое место для заполнения информации. Заполнив все нужные поля, необходимо нажать на кнопку со значком зеленой «Галки». Далее, следует нажать на кнопку «Commit» и данная запись сохранится.

Для того, чтобы новая запись появилась в данной таблице, и мы смогли ее увидеть, следует нажать на кнопку «Refresh». Данная кнопка обновит записи.

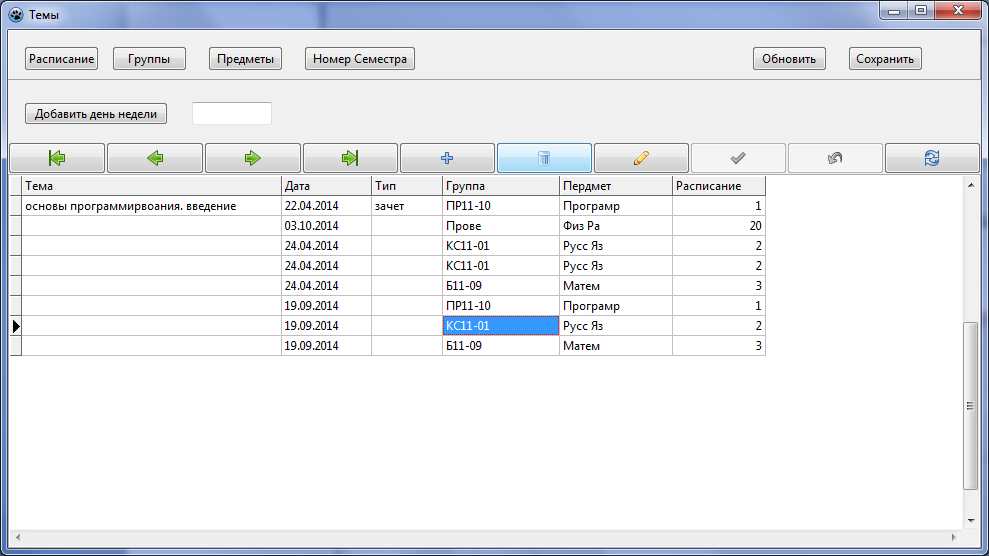


*Рис. 6 – Добавление новой записи*

3.3. Удаление записей

Для удаления определенной записи, нужно:

1. Выделить запись мышкой, либо кнопками навигации.
2. Если запись выделилась, следует нажать на кнопку с изображением «Урны».
3. Запись должна исчезнуть. Следует нажать кнопку «Commit», чтобы изменения сохранились в базе.
4. А для обновления следует нажать на кнопку «Refresh».



*Рис. 7 – Удаление записи*

Заключение

В процессе создания курсового проекта, задачи которые были поставлены в самом начале, а именно:

* Описание структуры базы данных;
* Приведение ее к 3НФ;
* Создание базы на основе описанной структуры;
* Создание триггеров и генераторов для ключевых полей;
* Созданное приложение в IDE
* Удобный пользовательский интерфейс;
* Добавление, удаление, и обновление записей.

Задачи были выполнены успешно, результат работы описан, в ходе работы все возникшие трудности, а именно:

* Создание SQL-запроса с параметром «left join»;
* Создание большого количества связей(LookUp полей);
* Сортировка таблицы «Расписание» относительно дня недели и идентификатора.

Трудности были решены с помощью технической литературы и сопутствующей документации к продуктам, которые использовались при разработке курсового проекта. Список документации написан в конце данного документа.

Список литературы

1. Документация к Firebird - <http://www.firebirdsql.org/manual/ru/>
2. Узнайте все о Firebird за 2 минуты - <http://www.firebirdnews.org/docs/fb2min_ru.html>
3. Документация к Lazarus - <http://wiki.freepascal.org/Lazarus_Documentation/ru>
4. Мартин Грайбер. «Понимание SQL»
5. А.Д. Хомоненко, В.М.Цыганков, М.Г.Мальцев. «Базы данных. Учебник для вузов»