Национальный Авиационный Университет

Институт Аэронавигации

Электронные устройства и системы



**Курсовая Работа**По Дисциплине «Информатика»

На тему: **«**. Автоматизация учёта отдела кадров.**»**

**Выполнил:**

Мелешко Олександр Игоревич

Группа СЕ106а

**Проверил:**

Моргун Алексей Анатольевич

**Киев 2015**

**Содержание**

|  |
| --- |
| 1. Введение. Постановка задачи .……………………………………..3 |
| 1. Базы данных. MySQL…………………………………………………4 |
| 1. Builder C++…………………………………………………………….7 |
| 1. Постановка задачи…………………………………………………….8 |
| 1. Описание программы…………………………………………………9 |
| 1. Заключения……………………………………………………………11 |
| 1. Список использованной литературы………………………………...12 |
| 1. Приложение…………………………………………………………...13 |

**Введение**

На каждом предприятии есть такое структурное подразделение как отдел кадров. Эффективность работы предприятия во многом зависит от профессионализма сотрудников, прием которого осуществляет именно это подразделение. Функциональная роль отдела кадров на предприятии очень велика по той причине, что именно в отделе кадров мы находим интересующую нас информацию о рабочих, служащих предприятия.

**Цель**: автоматизация учета кадров, что позволит быстрее сортировать и находить нужную нам информацию.

В данной программе я хочу упростить работу работников отдела кадров при помощи создании база данных в программе в Builder C++.

**Базы данных. MySQL.**

*База данных*- совместно используемый набор логически связанных данных. Это единое хранилище данных, которое однократно определяется, а затем используется одновременно многими пользователями.

*Система управления базами данных (СУБД)* - это программное обеспечение, с помощью которого пользователи могут определять, создавать и поддерживать базу данных, а также осуществлять к ней контролируемый доступ.

В *реляционных* базах данных (БД самого распространенного типа) данные хранятся в *таблицах*. На первый взгляд, эти таблицы подобны электронным таблицам Excel, поскольку они тоже состоят из строк и столбцов. Столбцы называются *полями* (fields) и содержат данные определенного типа. Строки именуются *записями* (records). В одной строке хранится один набор данных, описывающих определенный объект. Например, если в таблице хранятся данные о клиентах, она может содержать поля для имени, адреса, города, почтового индекса, номера телефона и т.д. Для каждого клиента будет создана отдельная запись.

Лучшие три СУБД на мое мнение это:  
1)Первое место, по праву принадлежит инструменту Workbench (разработка компании Sun Systems/Oracle), который может работать на платформах Microsoft Windows, Mac OS X и Linux. **MySQL Workbench** распространяется под свободной лицензией — Community Edition и с ежегодной оплачиваемой подпиской — Standard Edition. Последняя включает в себя дополнительные возможности, которые способны существенно улучшить производительность, как разработчиков, так и администраторов баз данных.

**Что делает Workbench популярным?**

* возможность представить модель БД в графическом виде, а также редактирование данных в таблице;
* наличие простого и функционального механизма по созданию связей между полями таблиц, среди которых реализована связь «многие-ко-многим» с возможностью создания таблицы связей;
* функция Reverse Engineering позволяет восстанавливать структуру таблиц и связей из той, которая была реализована ранее и хранится на сервере БД;
* наличие редактора SQL-запросов, который дает возможность при отправке на сервер получать ответ в табличном виде и другие возможности.

2) Второе место занимает **Navicat** (разработка компании PremiumSoft CyberTech Ltd) — инструмент для разработки и администрирования баз данных, который работает на любом сервере MySQL, начиная с версии 3.21. Для MySQL, Navicat доступен для работы на платформах Microsoft Windows, Mac OS X и Linux.

**Что делает Navicat популярным?**

* наличие визуального конструктора запросов;
* возможность импорта, экспорта и резервного копирования данных;
* возможность создавать отчеты;
* SSH и HTTP туннелинг;
* миграция и синхронизация данных и структуры;
* инструмент для планирования задач и другие возможности.

3) **PHPMyAdmin** — бесплатное приложение с открытым кодом, предназначенное для администрирования СУБД MySQL. PHPMyAdmin представляет собой веб-интерфейс с помощью которого можно администрировать сервер MySQL, запускать команды и просматривать содержимое таблиц и БД через браузер.

**Что делает PHPMyAdmin популярным?**

* возможность управлять СУБД MySQL без непосредственного ввода SQL команд;
* как панель управления PHPMyAdmin предоставляет возможность администрирования выделенных БД;
* интенсивное развитие;
* возможность интегрировать PHPMyAdmin в собственные разработки благодаря лицензии GNU General Public License и другие возможности.

**MySQL Workbench**

MySQL Workbench – программа нового поколения, позволяющая визуально проектировать базы данных. Приложение включает в себя все необходимые инструменты для того, чтобы пользователь мог моделировать, создавать и использовать БД MySQL. Если сравнивать данное приложение с другими подобными программами, то явное преимущество остается за MySQL Workbench, выгодно отличающейся быстродействием и мощностью.

   Это приложение можно использовать для того, чтобы разработать новый формат баз данных, а также с целью выполнения комплексного перехода. MySQL Workbench обеспечивает визуальное отображение таблиц базы данных, видов, хранимых процессов и внешних ключей.

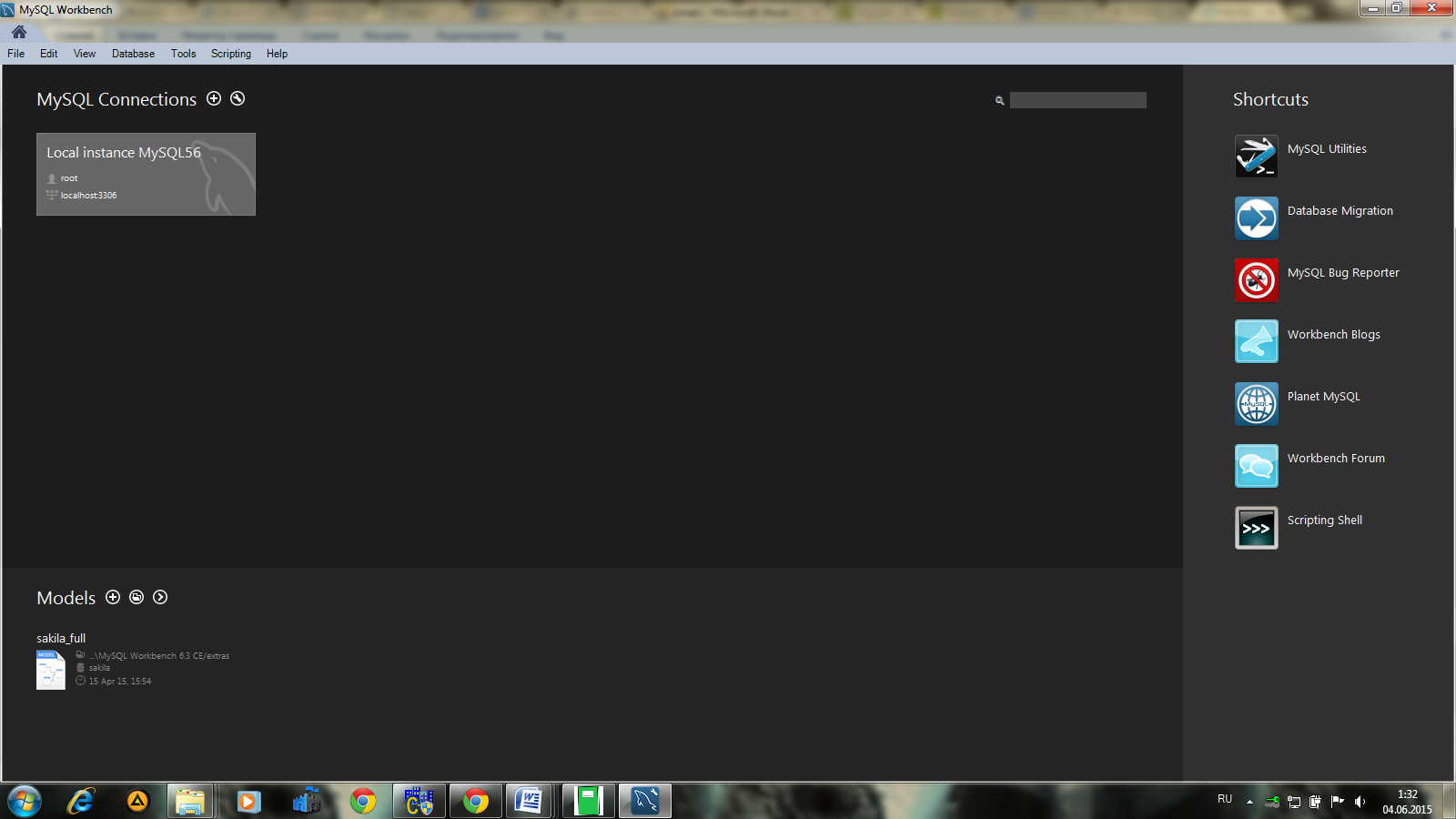


Рис.1(Начальная страница MySQL Workbench)

**Builder C++**

**C++ Builder** — программный продукт, инструмент быстрой разработки приложений(RAD), интегрированная среда программирования (IDE), система, используемая программистами для разработки программного обеспечения на языке программирования C++.

Изначально разрабатывался компанией Borland Software, а затем её подразделением CodeGear, ныне принадлежащим компании Embarcadero Technologies.

C++ Builder объединяет в себе комплекс объектных библиотек (STL, VCL, CLX, MFC и др.), компилятор, отладчик, редактор кода и многие другие компоненты.

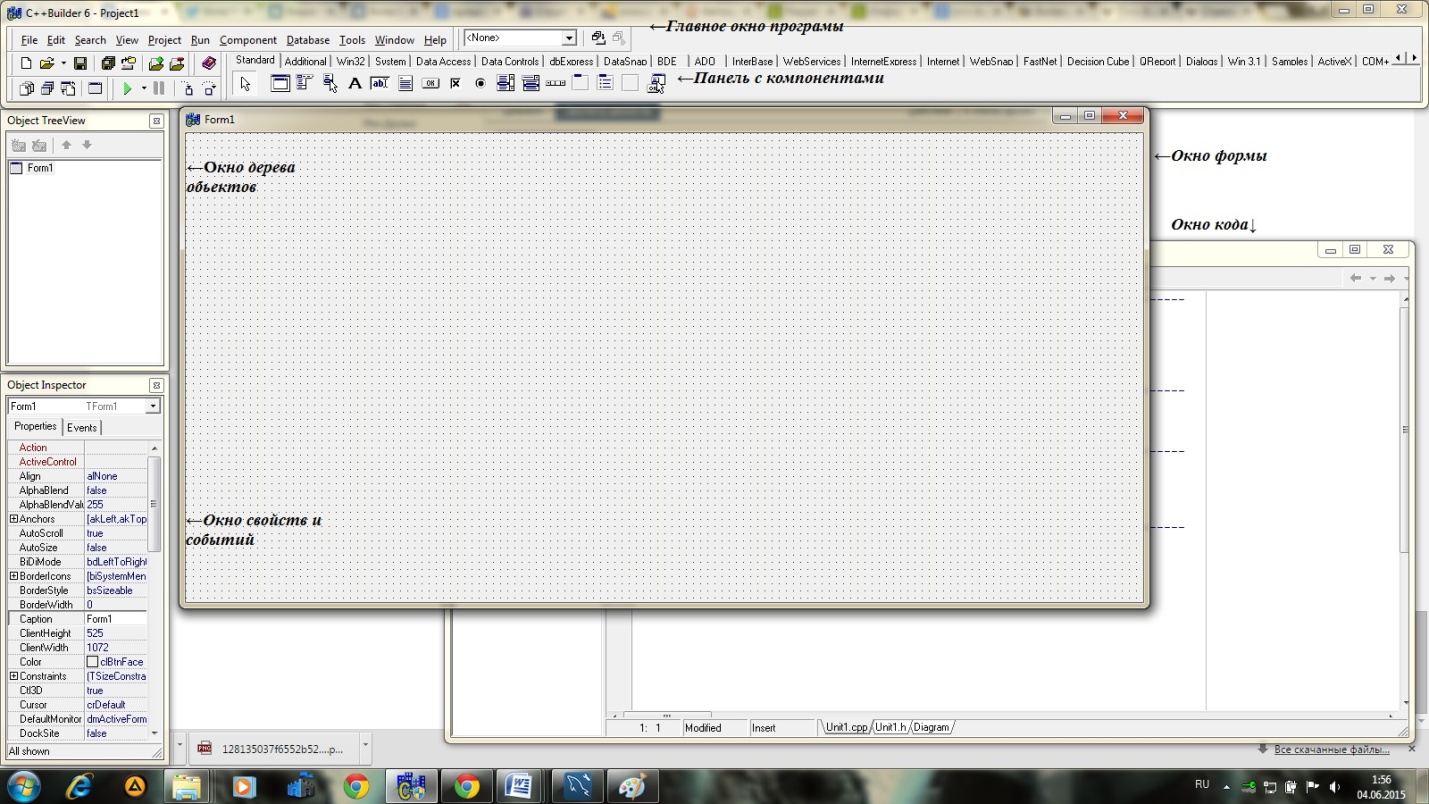


Рис.2(Главный вид программы)

**Постановка задачи**

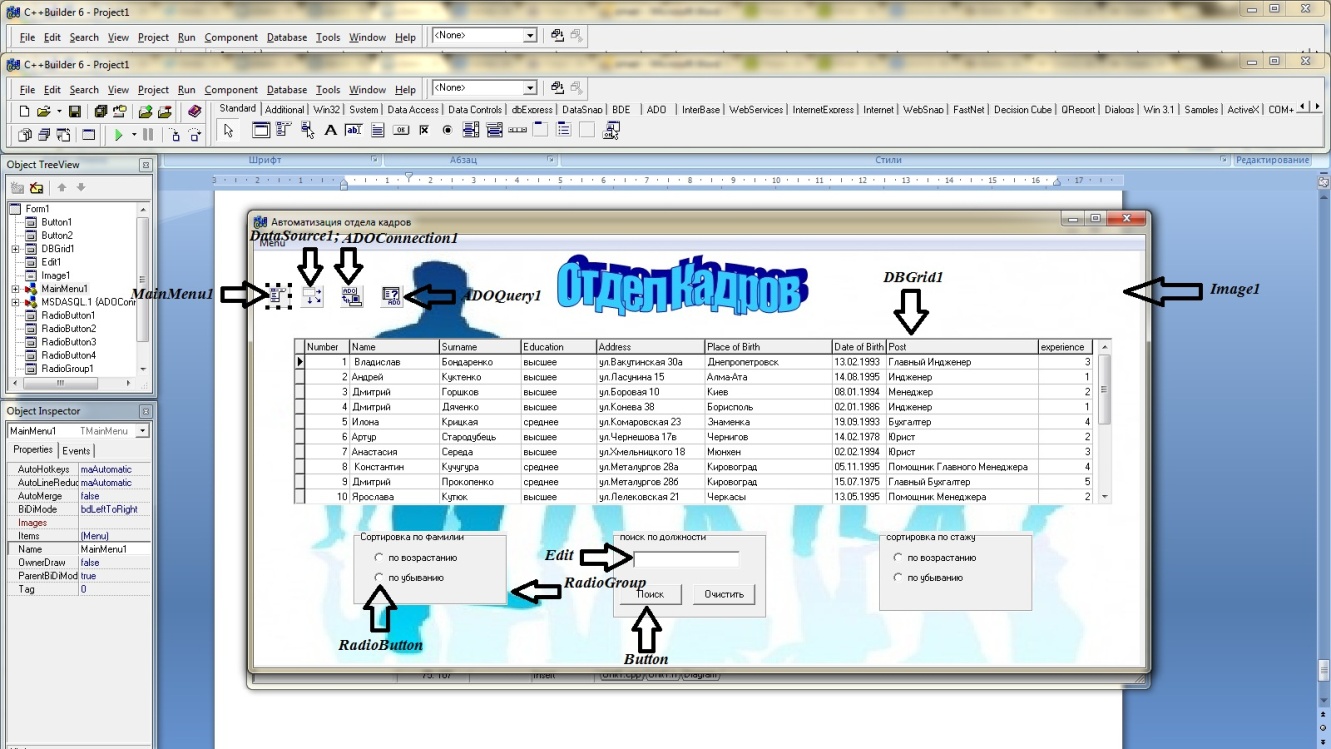
Создания программы в виде таблицы в которую можно вносить данные: о ФИО, Дату Рождения и Место Рождения, Должность и стаж работы на этой должности. А также сортировать для более комфортного поиска нужной информации и поиска людей по должности(рис.3)

****

**Рис.3**

**Описание программы**

Моя программа это база данных представлена в виде таблицы. Она предназначена для автоматизации учета отдела кадров. В ней имеються такие компоненты(Рис.4) :

****

**Рис.4**

1. **MainMenu**(Главное меню) — предназначен для создания главного меню программы.
2. **Button**(Кнопка) — является самым распространенным компонентом. Служит для создания в приложении различных прямоугольных кнопок с текстовой надписью.
3. **RadioGroup**(Группа радиокнопок) — позволяет создавать на форме контейнер в виде прямоугольной рамки для объединения группы взаимоисключающих радиокнопок.
4. **RadioButton**(Радиокнопка) — создает круглое поле с двумя состояниями (с точкой и без точки) и текстовой строкой, поясняющей ее назначение в программе. Обычно несколько таких компонентов, расположенных на форме, позволяют переключить только один элемент из группы.
5. **Image**(Образ) — предназначен для создания на форме невидимого контейнера, в который можно поместить один графический файл с битовым образом, пиктограммой или метафайл.
6. Компонент **DataSource** действует как посредник между компонентами TDataSet (TTable, TQuery, TStoredProc) и компонентами Data Controls - элементами управления, обеспечивающими представление данных на форме. Компоненты TDataSet управляют связями с библиотекой Borland Database Engine (BDE), а компонент DataSource управляет связями с данными в компонентах Data Controls.
7. **DBGrid** - табличное отображения данных;
8. **ADOConnection**-Используется для связи с набором данных ADO, позволяющим работать с несколькими компонентами набора данных и обеспечивает выполнение их команд.
9. **ADOQuery-** позволяет направить запрос серверу базы данных. Обычно он используется, если данные, которые надо получить из базы данных, распределены по нескольким таблицам
10. **DBGrid-**используется для просмотра и редактрирования базы данных в режиме таблице.

**Заключения**

Из-за того что в отделе кадров мы находим интересующую нас информацию о рабочих, служащих предприятия, то благодаря моей программе можно повысить эффективность работы работников отдела кадров из-за того что моя программа может сортировать и искать нужную информацию которая была занесена в таблицу.

**Список использованной литературы**

1) Корчагин Р.Н. Информационные системы в экономике. Методические рекомендации /Р.Н. Корчагин – М.: РАГС, 2009.

2) MySQL 5.0. Библиотека программиста: Питер; Санкт-Петербург; 2010

3) C++Builder 6. Справочное пособие. Язык C+. Архангельский А.Я;2002.

4) Программирование в С++ Builder 6 и 2006. Архангельский А.Я., Тагин М.А.2007.

**Приложение**

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::RadioButton1Click(TObject \*Sender)

{

if(RadioButton1->Checked)

{ADOQuery1->Sort="Surname ASC";}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button2Click(TObject \*Sender)

{

Form1->ADOQuery1->SQL->Clear();

Form1->ADOQuery1->SQL->Add("Select \* From otdel.new\_table where Post LIKE '%"+Edit1->Text+"%'");

Form1->ADOQuery1->Active=false;

Form1->ADOQuery1->Active=true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::RadioButton2Click(TObject \*Sender)

{

if(RadioButton2->Checked)

{ADOQuery1->Sort="Surname DESC";}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::RadioButton3Click(TObject \*Sender)

{

if(RadioButton3->Checked)

{ADOQuery1->Sort="experience ASC";}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::RadioButton4Click(TObject \*Sender)

{

if(RadioButton4->Checked)

{ADOQuery1->Sort="experience DESC";}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::close1Click(TObject \*Sender)

{

Form1->Close();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

Form1->ADOQuery1->SQL->Clear();

Form1->ADOQuery1->SQL->Add("Select \* From otdel.new\_table");

Form1->ADOQuery1->Active=false;

Form1->ADOQuery1->Active=true;

}

//---------------------------------------------------------------------------