ВВЕДЕНИЕ

В процессе развития человек создаёт много интересных и уникальных вещей, забывая при этом, что не все его действия идут на благо окружающей среде. Для того, чтобы создать что-то, часто необходимо, что-то уничтожить. И достаточно часто человек выбирает природу. Каждый день люди пользуются большим количеством различных благ, забывая, что где-то в другом месте из-за них, возможно, срубили целый лес деревьев. Большинство людей просто не осознаёт, что такие действия человека против природы в скором времени могут привести к печальным последствиям. Из-за выбросов заводов разрушается экосистема, гибнут деревья. Воздух в городах становится пыльным и тяжёлым. У множества людей начинают развиваться различные заболевания всех систем организма, так как воздух содержит кислород, который необходим для всего организма.

На данный момент человек не способен остановить вредное производство, поэтому был придуман ряд мероприятий, позволяющий уменьшить вред окружающей среде и улучшению экосистемы. К этим мероприятиям также относят насаждения различных лесных массивов, как вне города, так и непосредственно в городе.

Предприятия по озеленению появились совсем недавно и пользуются большим спросом. Большинство организаций, нуждающихся в такого рода услугах, устраивают так называемые «субботники», на которых трудовой коллектив этих организаций занимается озеленением и уборкой территории. На такие мероприятия люди приходят в основном добровольно, но таких людей обычно очень мало. Поэтому много работы откладывается на потом.

Озеленением городов занимаются исполкомы. Они составляют планы высадки деревьев и прочих зелёных насаждений. Но так как это достаточно занятая организация, были организованы предприятия по озеленению города. Работа данного предприятия заключается в получении заявки на озеленение от какой-либо государственной или частной организации, составлении плана работ для этой организации и исполнении в определённые сроки.

На данный момент таких организаций достаточно мало и они практически неизвестны. Планы работ составляются вручную, на что затрачивается достаточно много времени. Проекты и мероприятия разрабатываются достаточно большим количеством человек.

Разработка автоматизированной системы управления базами данных помогла бы существенно ускорить процессы разработки данных проектов и увеличить спрос данных организаций.

Данная разработка предполагает создание программного комплекса, который осуществляет управление данными в базе данных. Программный продукт сможет осуществлять анализ данных с выявлением наиболее ценной для пользователя информации, также представлять пользователю данные в наиболее информативном виде, для получения пользователем максимального количества информации. Данный программный комплекс даст возможность выполнить запросы к базе данных, результат выполнения которых имеет определенную важность для конечного пользователя.

Разрабатываемая автоматизированная информационная система даст пользователю такие преимущества, которые заключаются в сокращении времени на поиск и обработку нужных данных, также в анализе данных БД, который даст пользователю максимальную информацию о текущей ситуации.

Разработка автоматизированной информационной системы для учёта работы предприятия по озеленению является актуальной темой на сегодняшний день. Цель этой курсовой работы научится проектировать такую СУБД, которая пригодилась бы для организаций типа «Зеленстрой».

1. АНАЛИЗ РАЗРАБАТЫВАЕМОГО ПРОГРАММНОГО ПРОДУКТА

* 1. Анализ существующих информационных систем

При исследовании существующих программных продуктов, был сделан вывод, что используя комплексное программное обеспечение компании Microsoft, на базе электронных таблиц Excel или базы данных Access можно было создать такое же приложение, выполняющее поставленную задачу. При этом затрачивается минимум усилий на проработку интерфейса и процедур сортировки. Однако, большим минусом данного решения будет является необходимость обязательной предварительной установки программного обеспечении Microsoft Office, лицензионные версии которого требуют материальных затрат.

Фирма «Зеленстрой» - это узкоспециализированная организация, поэтому распространённых аналогов на подобии разрабатываемой программы выявлено не было.

Поэтому принято считать, что уже исходя из этого разрабатываемое Windows- приложение не является «бесполезным», потому как имеет одну четко сформировавшуюся цель и место применения. Вероятное использование данного продукта – автоматизированное рабочее место частных и государственных организаций, занимающиеся озеленением и благоустройством различных учреждений на базе операционных систем семейства Windows. Данная программа, используя стандартные компоненты, с успехом будет работать в ОС-ах Windows’98, 2000, ХР, не требуя при этом дополнительных инсталляций программного обеспечения.

* 1. Описание предметной области

Зеленстрой занимается высадкой зелёных насаждений, а также различными работами с ними. Помимо этого такое предприятие может предложить услуги, отвечающие за уборку территории, контролем над состоянием хозяйственных построек. Данная фирма принимает заказы от организаций и занимается их реализацией. Количество персонала такого предприятия зависит от масштабов его действий. Средняя фирма такого направления должна состоять из одного администратора (пользователя СУБД),исполнителей и руководителя, то есть опытного специалиста в данной сфере деятельности, который сможет объективно оценить ситуацию и управлять процессом создания проекта. Этот процесс подразумевает под собой – определение сроков начала работ и сдачи завершённого проекта заказчику, составление списка мероприятий, которые оптимально смогли бы решить поставленную задачу, назначение исполнителя, то есть работника, который сможет хорошо выполнить поставленную задачу.

Задача администратора СУБД заключается в своевременном принятии заявок, представлении их руководителю, контролем за работой, сроками сдачи, предоставлении отчётов по проектам, как руководству, так и заказчику. Такого рода работа будет отнимать много времени и ресурсов, если не будет автоматизирована.

Существующая автоматизированная система должна содержать список всех заявок, базы заказчиков, исполнителей, шаблонов мероприятий, что позволит быстрее определить сроки завершения проектов.

* 1. Общие положения

Наименование системы. Автоматизированная информационная система для учёта работы предприятия по озеленению («Зеленстроя»). Но для удобства можно использовать уславное наименование системы для этой АИС – «Зеленстрой».

Цель функционирования системы заключается в экономии материальных и нематериальных затрат оператора «Зеленстроя» на оформление заявки, разработки и согласовании проекта с руководством, облегчением поиска нужной информации по проектам.

Задачи для разрабатываемого программного обеспечения:

* возможность создать новую заявку;
* свободный доступ к информации;
* удобный и понятный интерфейс приложения;
* возможность редактировать базу данных;
* сортировка данных по выбранным полям;
* поиск информации с помощью запросов к системе;
* возможность печати отчётов.

В данной работе необходимо спроектировать «АИС учета Зеленстроя». Функции данной АИС – управление существующими базами данных, формирование новых записей, редактирование старых.

Также СУБД должна содержать список всех возможных выполняемых работ: список проектов, которые находятся в разработке, или в стадии «завершён». Для облегчения получения результатов предусмотрена печать всех важных данных на бумажном носителе. Результаты должны быть показаны на экране. Необходимо обеспечить резервное копирование данных и восстановление данных из резерва в случае утраты.

В ходе анализа данной предметной области были выделены следующие требования: определение функций, которые должен выполнять программный продукт, требования к надёжности, требования к программной документации.

Требования к программному продукту (ПП). ПП должен выполнять следующие функции:

* оформление заявок;
* хранение и обработка данных о проектах;
* поиск интересующей информации;
* выполнение запросов;
* печать отчётов.

Требования к надежности:

* ПП должен устойчиво функционировать и не приводить к сбоям операционной системы;
* ПП должен обеспечивать контроль входной и выходной информации на соответствие заданным форматам данных;
* ПП должен обеспечивать обработку ошибочных действий пользователя с выдачей соответствующих сообщений.

Требования к программной документации. Программная документация должна включать:

* исходные коды ПП с комментариями;
* контекстно–зависимую помощь в ПП;
* руководство пользователя;
* руководство по установке.

2. ПРОЕКТИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

* 1. Построение диаграмм ER-экземпляров и ER-типов

Для создания БД необходимо построить ER– диаграмму в виде совокупности связей, сущностей и атрибутов, изображенных в виде структуры (рис. 2.1).

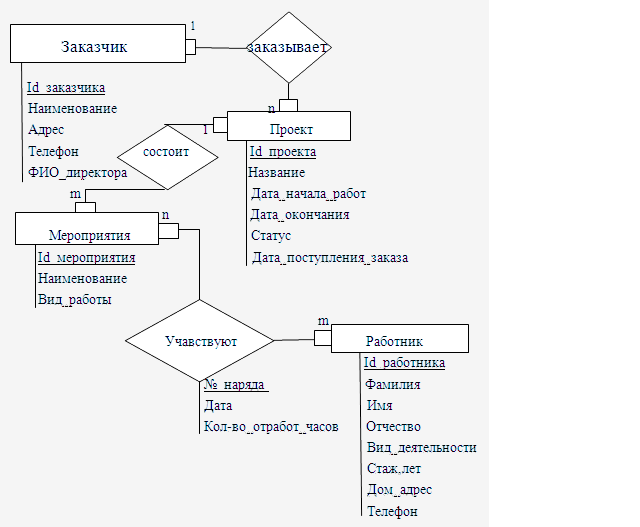


Рисунок 2.1 – ER-диаграмма

На данной ER-диаграмме видно, что для разработки данной системы необходимы 4 сущности: Заказчик, Проект, Работник, Мероприятия. Данные сущности соединяют 3 связи.

Для определения количества таблиц нужно построить диаграммы ER- экземпляров (рис.2.2 – 2.4).

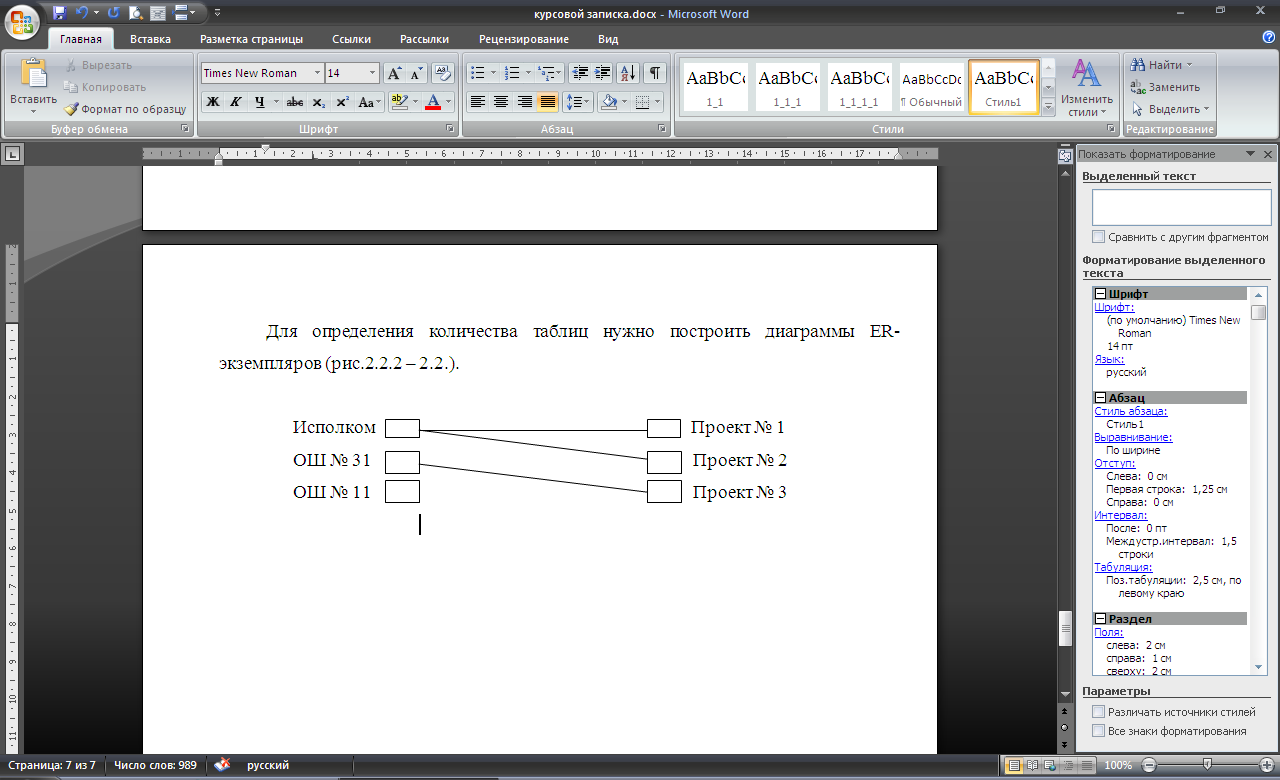


Рисунок 2.2 – Диаграмма ER-екземпляров для связи «Заказчик – Проект»

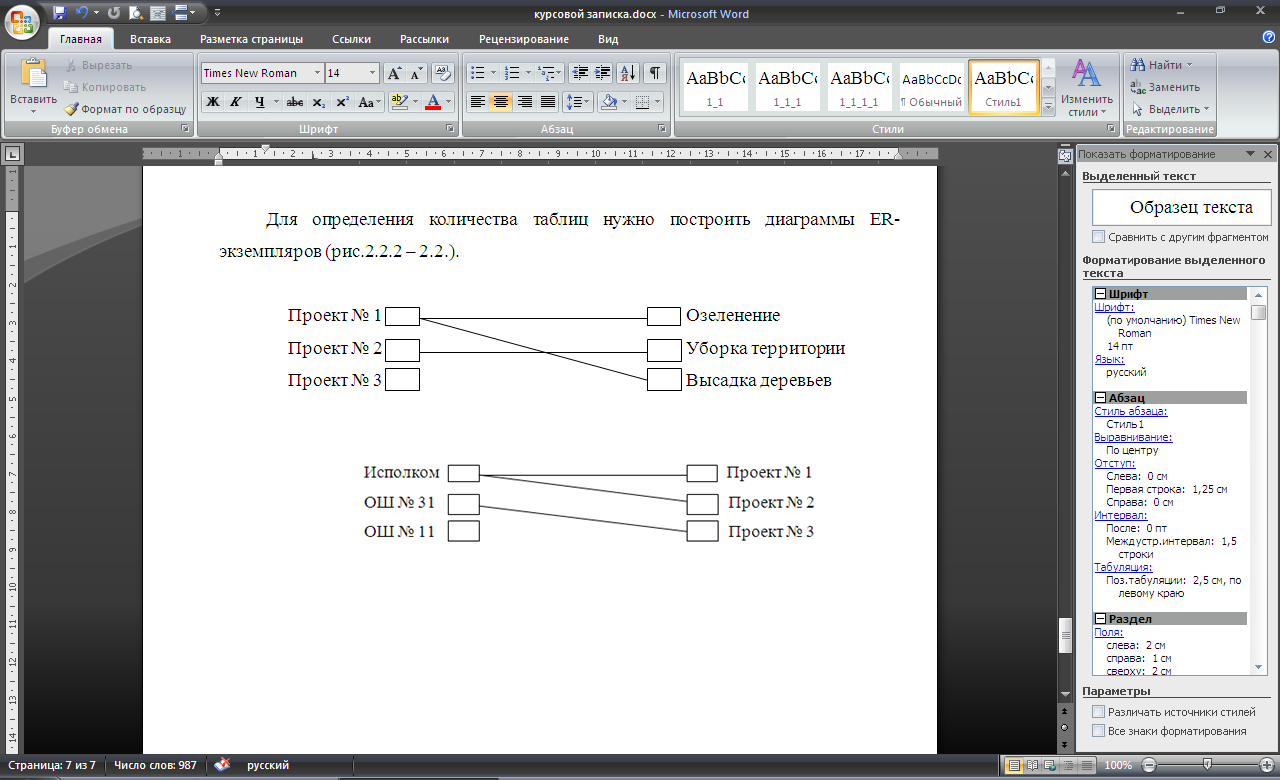


Рисунок 2.3 – Диаграмма ER-екземпляров для связи «Проект – Мероприятие»

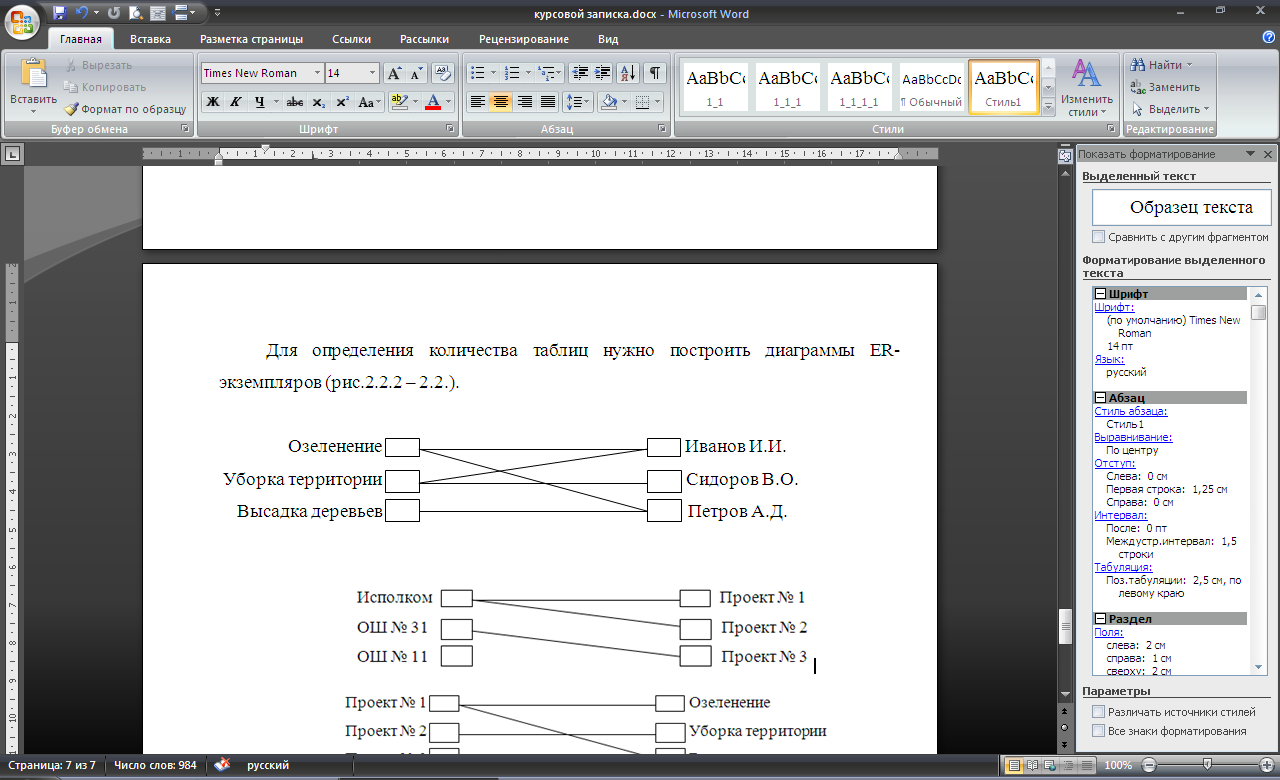


Рисунок 2.4 – Диаграмма ER-екземпляров для связи «Мероприятие – Работник»

По данным диаграммам и с учётом правил составления таблиц можно сделать вывод, что для разрабатываемого приложения нужны следующие таблицы.

Заказчик (Id\_заказчика, Наименование, Адрес, Телефон, ФИО директора).

Проект (Id\_проекта, Id\_заказчика, Название, Дата начала работ, Дата окончания, Статус, Дата поступления заказа).

Мероприятия (Id\_мероприятия, Id\_проекта, Наименование, Вид работы).

Участие (№ наряда, Id\_мероприятия, Id\_работника, Дата, Количество отработанных часов).

Работник (Id\_работника, Фамилия, Имя, Отчество, Вид деятельности, Стаж, Домашний адрес, Телефон).

После составления таблиц следует определить какого типа будут поля в данных таблицах. Данная процедура рассмотрена в табл.2.1.

Таблица 2.1

Типы полей таблиц разрабатываемого приложения.

|  |  |  |  |
| --- | --- | --- | --- |
| Заказчик | | Проект | |
| Название поля | Тип поля | Название поля | Тип поля |
| Id\_заказчика | Числовой (ключ) | Id\_проекта | Числовой (ключ) |
| Наименование | Текстовый | Id\_заказчика | Числовой |
| Адрес | Текстовый | Название | Текстовый |
| Телефон | Текстовый | Дата начала работ | Дата/время |
| ФИО директора | Текстовый | Дата окончания | Дата/время |
| Мероприятия | | Статус | Текстовый |
| Id\_мероприятия | Числовой (ключ) | Дата поступления заказа | Дата/время |
| Id\_проекта | Числовой | Участие | |
| Наименование | Текстовый | (№ наряда | Числовой (ключ) |
| Вид работы | Текстовый | Id\_мероприятия | Числовой |
|  |  | Id\_работника | Числовой |
|  |  | Дата | Дата/время |
| Работник | |  |  |
| Название поля | Тип поля |  |
| Id\_работника | Числовой (ключ) |  |  |
| Фамилия | Текстовый |  |  |
| Имя | Текстовый |  |  |
| Отчество | Текстовый |  |  |
| Вид деятельности | Текстовый |  |  |
| Стаж | Числовой |  |  |
| Домашний адрес | Текстовый |  |  |
| Телефон | Текстовый |  |  |

2.2 Концептуальное моделирование

Объектно-ориентированное проектирование программной системы выполнялось в виде построения UML-диаграмм [10]. Унифицированный язык моделирования UML (Unified Modeling Language) предназначен для описания, визуализации и документирования объектно-ориентированных программных систем и бизнес-процессов с ориентацией на их последующую реализацию в виде программного обеспечения. Проектирование системы на языке UML выполнялось с использованием CASE-средств IBM Rational Rose.

Для спецификации, визуализации, проектирования и документирования компонентов программного обеспечения была разработана диаграмма вариантов использования, представленная на рис. 2.5.

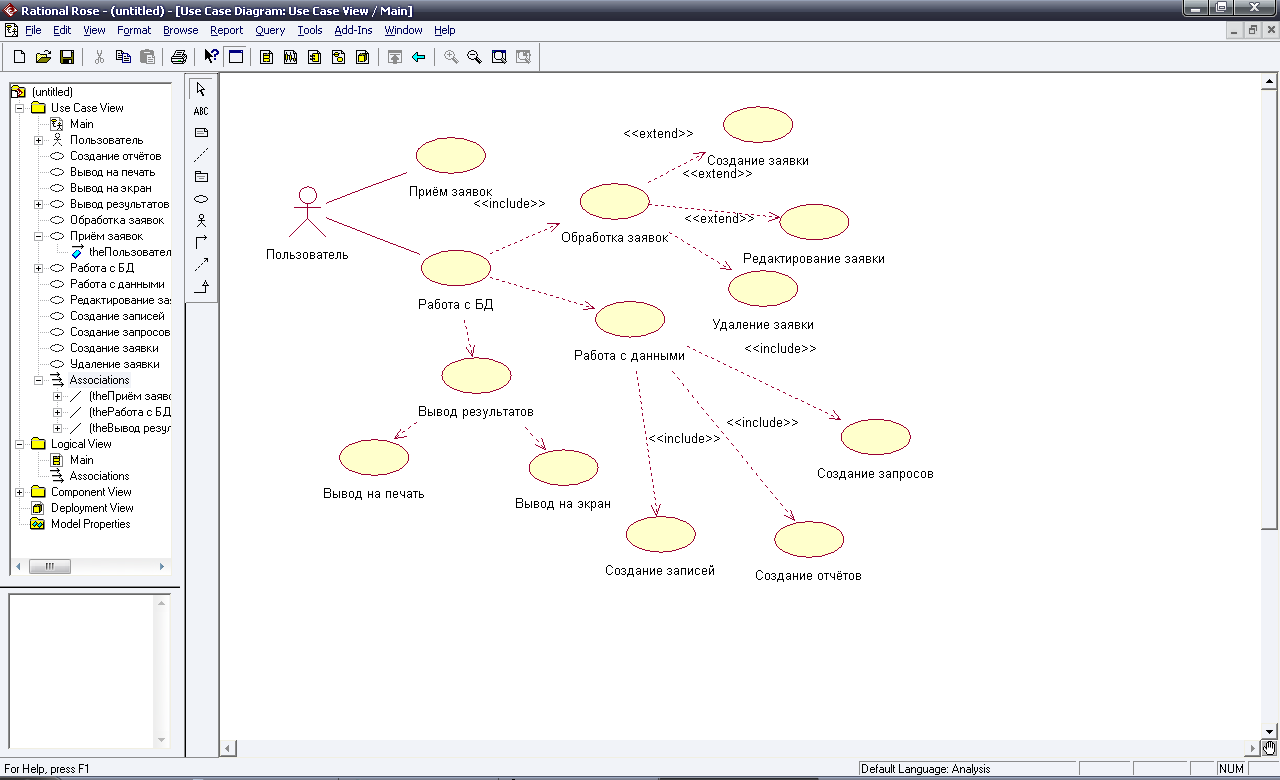


Рисунок 2.5 – Диаграмма вариантов использования

На данной диаграмме показано, что основной задачей пользователя разрабатываемой системы является приём заявок, работа с базами данных и получение результатов по этим данным. В работу с БД входит работа с данными и обработка заявок. Под обработкой заявок подразумевается возможность их создания, редактирования и удаления. В работу с данными входит возможность создания записей, запросов к системе и отчётов по ним.

3. РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

3.1 Разработка интерфейса приложения

Создание таблиц для АИС. Для этого шага использовался такой пакет как Microsoft Access. Таблицы создавать в этом приложении удобней, чем в Database Desktop, приложении на базе платформы для создания программ Delphi. Таблицы хранятся не отдельно каждая в своём файле, а сразу в одном, что обеспечивает целостность базы, поскольку для этого файла не нужны дополнительные компоненты.

Наряду с традиционными инструментами доступа к данным Borland Database Engine и ODBC в приложениях Delphi можно применять технологию Microsoft ActiveX Data Objects (ADO), которая основана на возможностяхСОМ, а именно интерфейсов OLE DB. OLE DB представляет собой набор специализированных объектов СОМ, инкапсулирующих стандартные функции обработки данных, и специализированные функции конкретных источников данных и интерфейсов, обеспечивающих передачу данных между объектами. Технология ADO в целом включает в себя не только сами объекты OLE DB, но и механизмы, обеспечивающие взаимодействие объектов с данными и приложениями. На этом уровне важнейшую роль играют провайдеры ADO, координирующие работу приложений с хранилищами данных различных типов. Так как технология ADO основана на стандартных интерфейсах СОМ, которые являются системным механизмом Windows, это сокращает общий объем работающего программного кода и позволяет распространять приложения БД без вспомогательных программ и библиотек. Данная АИС сделана с помощью ADO-технологии, что позволяет данным динамически обновляться во время работы приложения. База данных подключена через системный драйвер Microsoft.Jet.OLEDB.4.0. В данном ПП при разработке использовался компонент ADOConnection. К его классу относятся также используемые компоненты: ADOQuery и ADOTable.

Таблицы, созданные в Microsoft Access 2007, представлены на рис.3.1-3.5.

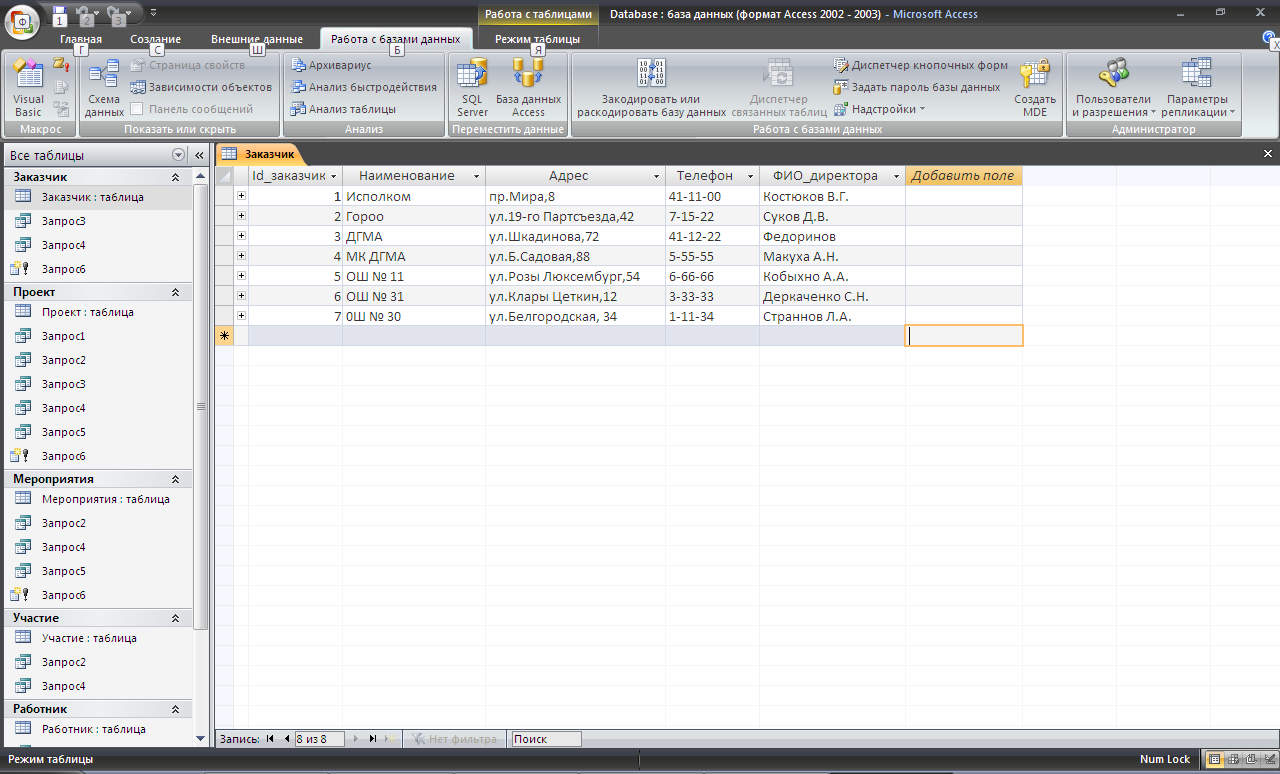


Рисунок 3.1 – Структура таблицы «Заказчик»

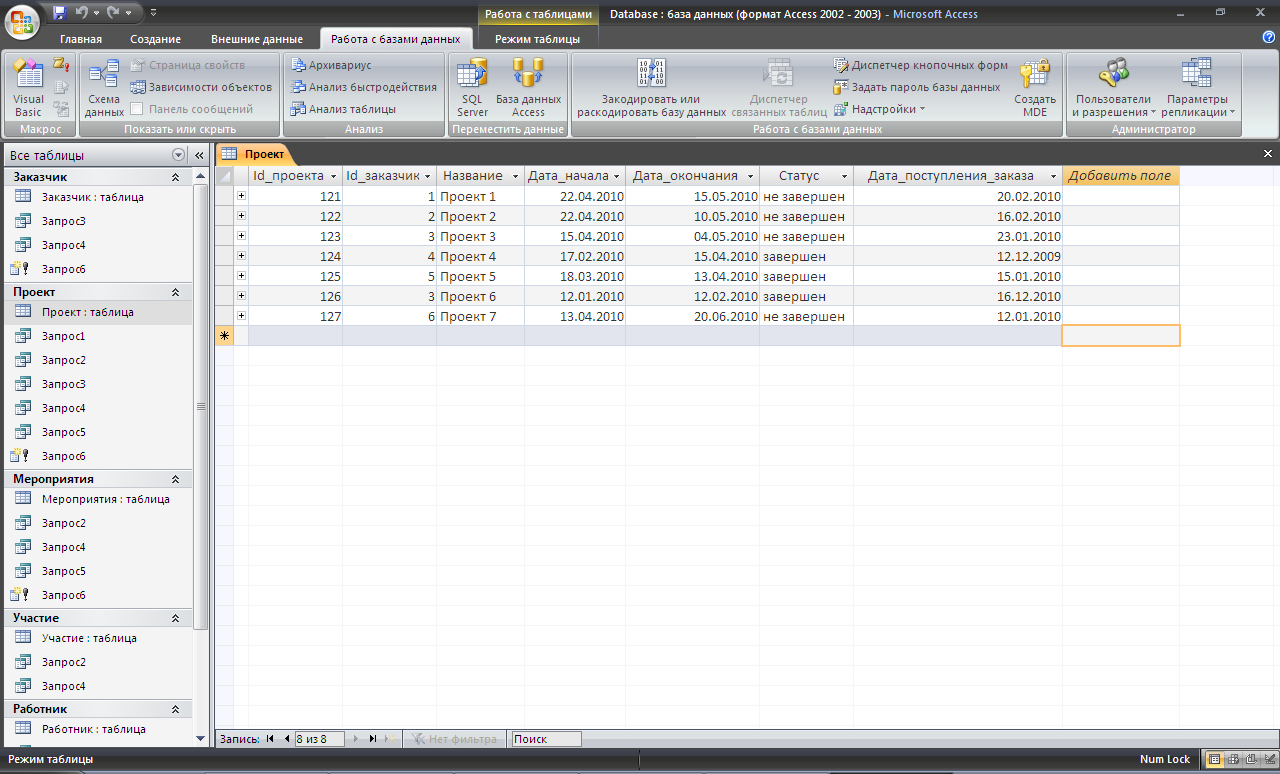


Рисунок 3.2 – Структура таблицы «Проект»

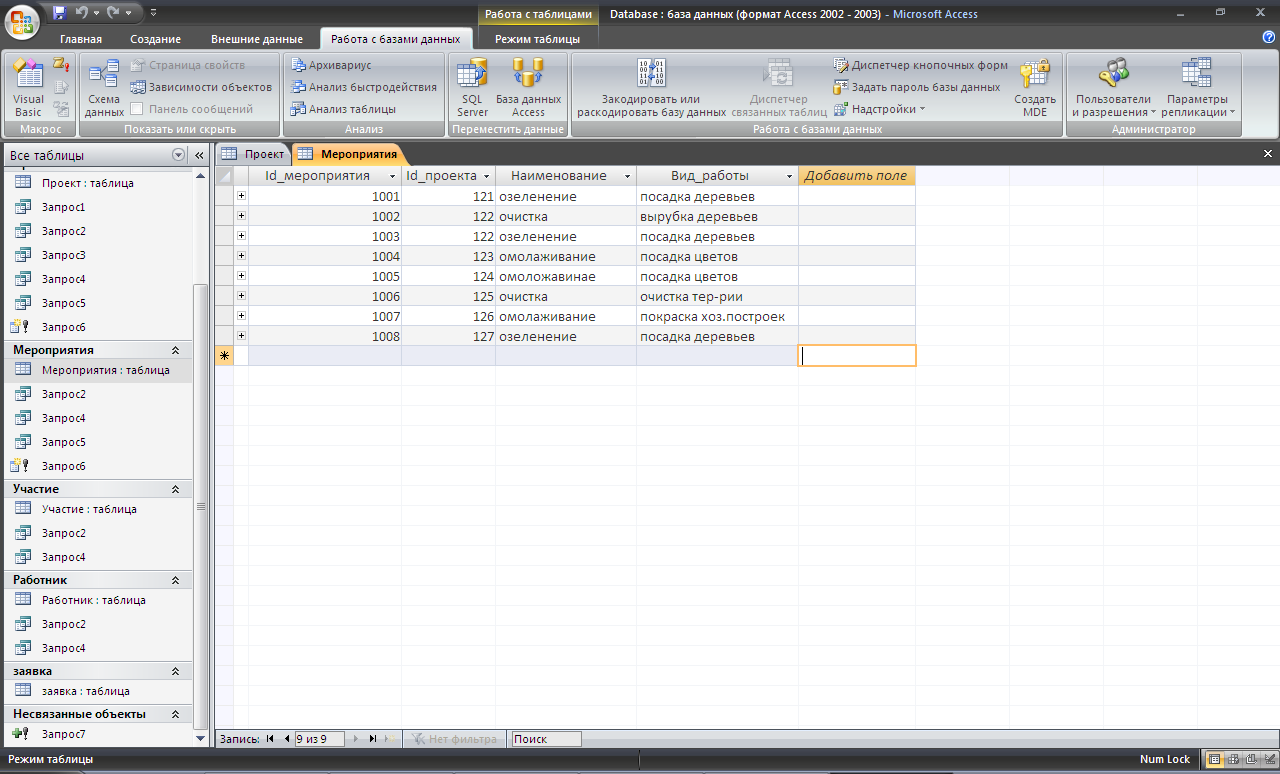


Рисунок 3.3 – Структура таблицы «Мероприятия»

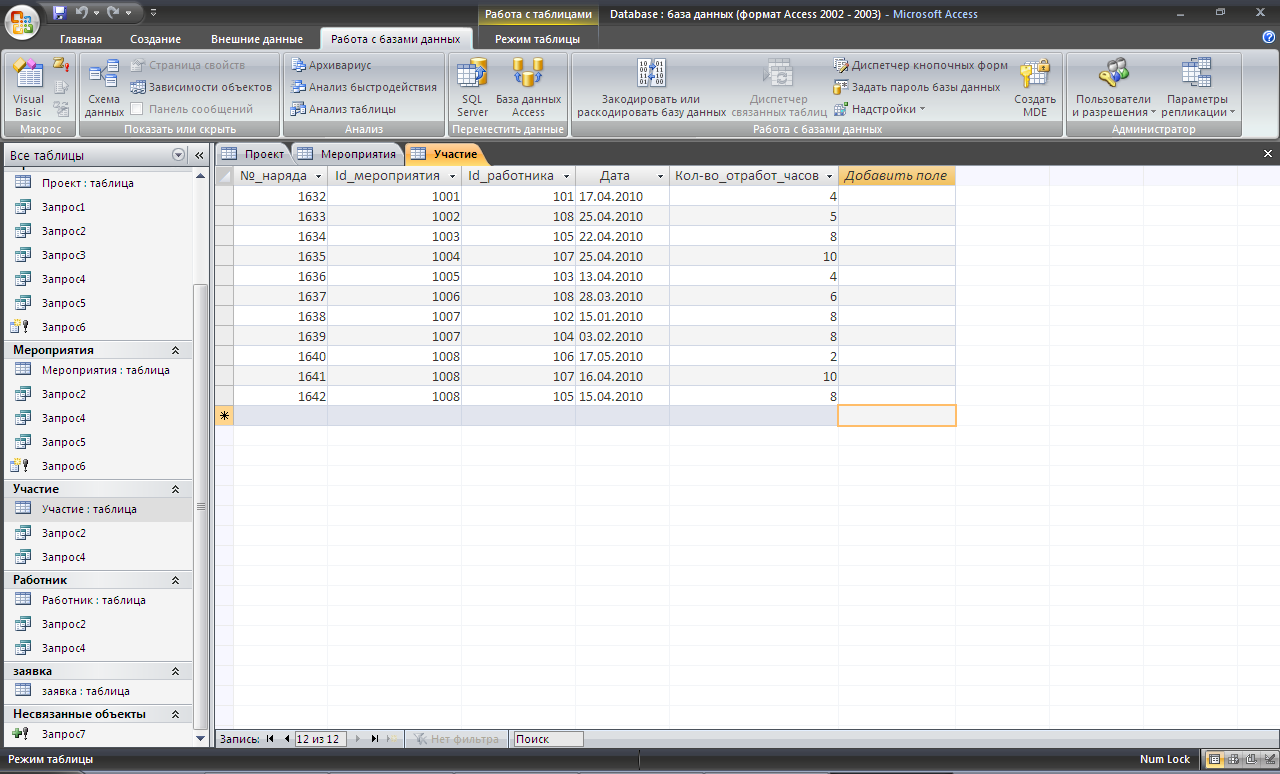


Рисунок 3.4 – Структура таблицы «Участие»

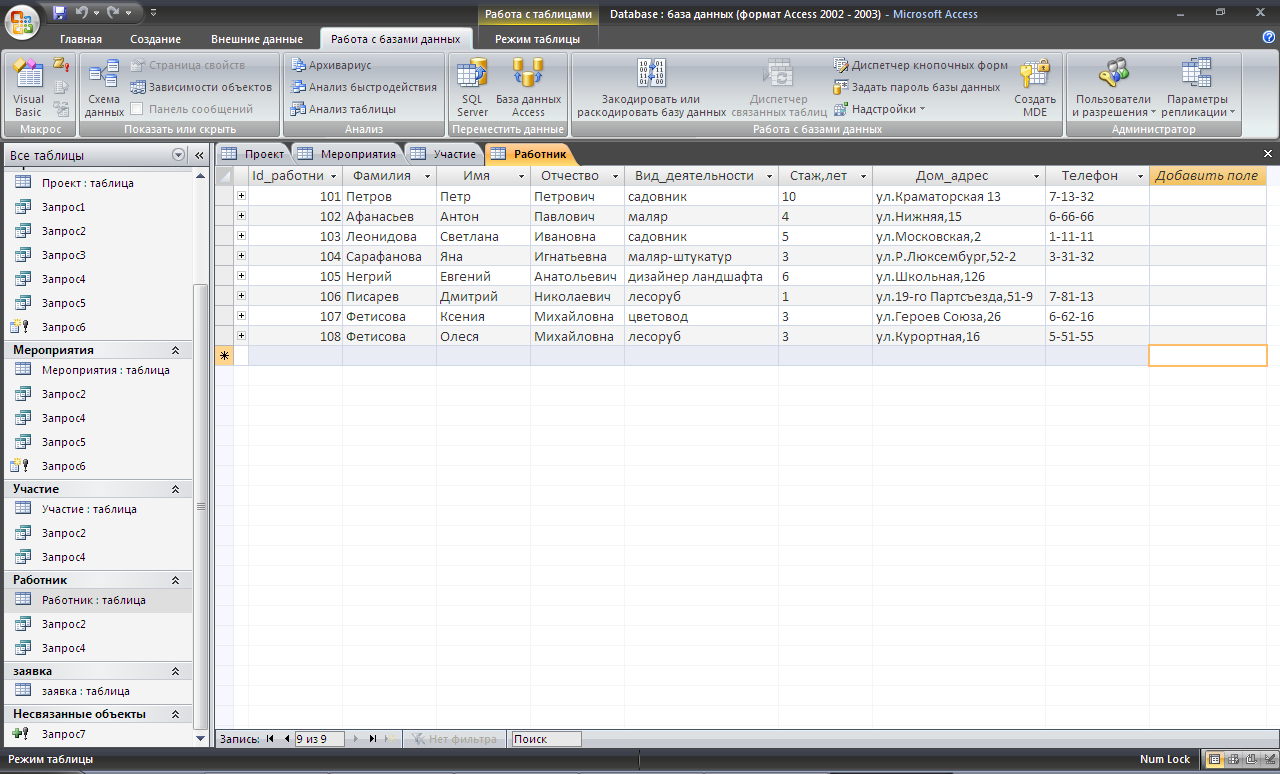


Рисунок 3.5 – Структура таблицы «Работник»

Данные таблицы соответствуют таблицам, полученным при проектировании данной автоматизированной информационной системы. Таблицы могут связываться в Access.

Ниже на рисунке 3.6 приведена логическая связь таблиц, спроектированных в Access.

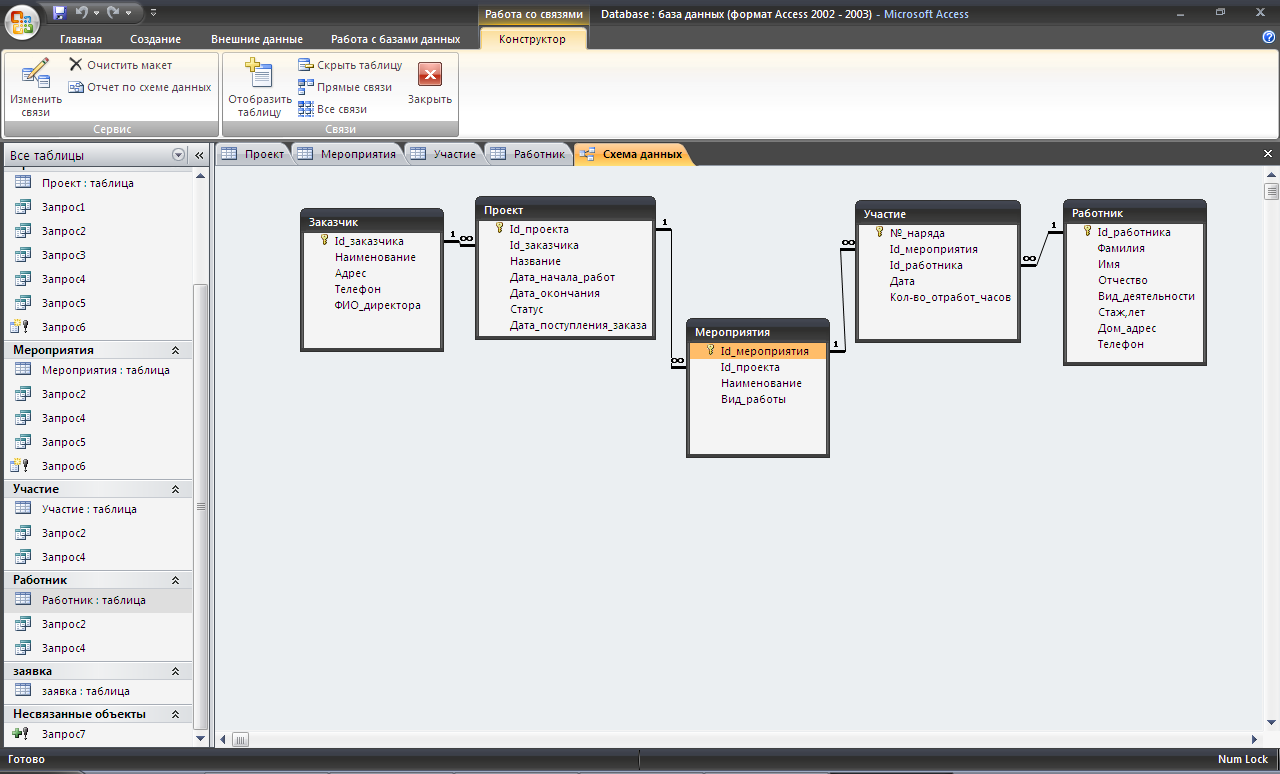


Рисунок 3.6 – Связи между таблицами

На данном рисунке (см.рис.3.6) детально показаны все таблицы, созданные для данного ПП. Связи в таблицах осуществляются через ключи в одних таблицах и идентичным им записям в других.

Создание интерфейса. В данном программном продукте разрабатывался интерфейс, который будет ориентирован на неопытного пользователя. Выбор нужной формы осуществляется с помощью контекстного меню (рис.3.7).

На главной форме предоставлены 2 таблицы: Проект и Мероприятия. Предусмотрена сортировка кликом мышки по на званию какого-либо столбца таблиц. Также система содержит справку пользователя.

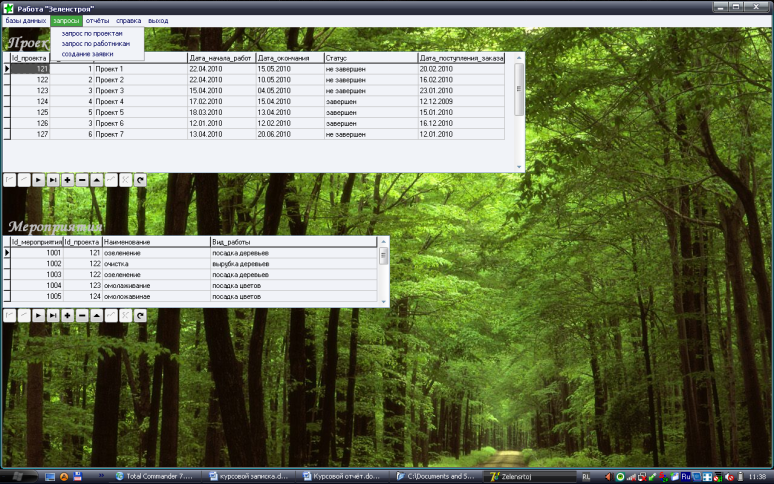


Рисунок 3.7 – Главная форма ПП

3.2 Создание SQL-запросов

В данном проекте были использованы различные SQL-запросы, для оптимизации работы и увеличения эффективности. Список этих запросов предоставлен ниже.

1) Сортировка

При нажатии на заголовок поля производится сортировка по этому полю, обработчики событий предоставлены ниже. Пример предоставлен на рис.3.8.

procedure TForm1.DBGid1TitleClick(Column: TColumn);

var s:string;

begin

s:=Column.Title.Caption;

ADOQuery1.SQL.Clear;

ADOQuery1.SQL.Add('select \* from проект order by '+s);

ADOQuery1.Active:=false;

ADOQuery1.Active:=true;

end;

procedure TForm1.DBGrid2TitleClick(Column: TColumn);

var s:string;

begin

s:=Column.Title.Caption;

ADOQuery2.SQL.Clear;

ADOQuery2.SQL.Add('select \* from Мероприятия order by '+s);

ADOQuery2.Active:=false;

ADOQuery2.Active:=true;

end;

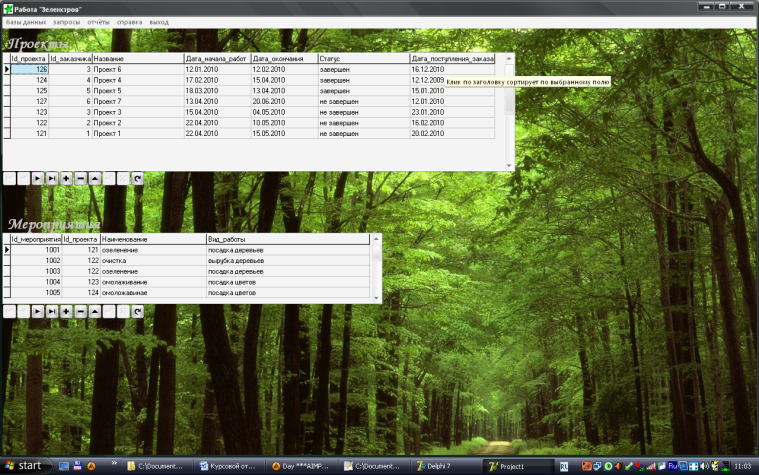


Рисунок 3.8 – Пример сортировки

2) Выбор завершённых проектов за период, определяемый пользователем.

procedure TForm7.Button1Click(Sender: TObject);

function DateConv(date: tdate):string;

var i: integer;

begin

result:=datetostr(date);

for i:=1 to length(result) do if result[i]='.' then result[i]:='/';

end;

begin

adoquery1.SQL.Clear;

OutputQuery:='SELECT \* FROM Проект WHERE ([Дата\_начала\_работ] Between #'+

DateConv(DateTimePicker1.Date)+'# And #'+DateConv(DateTimePicker2.Date)+

'#) And (статус='+#39+'завершен'+#39+')';

adoquery1.SQL.add(OutputQuery);

adoquery1.Active:=false;

adoquery1.Active:=true;

end;

Пример запроса приведен на рис.3.9.

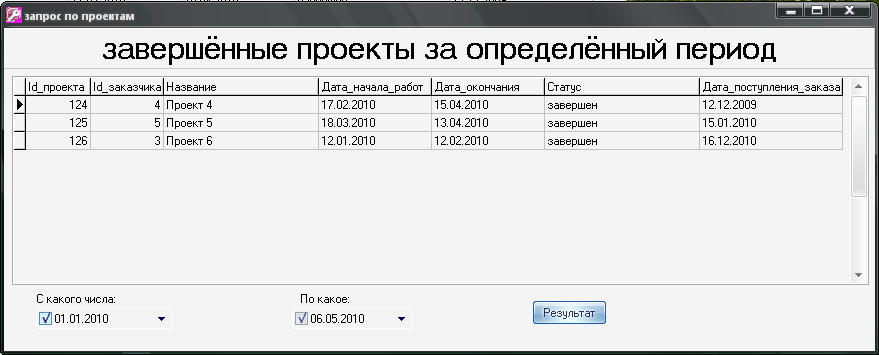


Рисунок 3.9 – Запрос по проектам

3) Создание заявки на новый проект в пустой таблице «Заявки»

Данный шаг сделан потому, что для разработки проекта требуется определённое время. Так как задача администратора заключается в том, чтобы принять заявку, то таким способом она регистрируется в базе данных. Поскольку специалисту необходимо просчитать время, за которое проект будет закончен, какие работники будут выполнять мероприятия по данному проекту, то есть разработать проект.

procedure TForm8.Button1Click(Sender: TObject);

var s:string;

function DateConv(date: tdate):string;

var i: integer;

begin

result:=datetostr(date);

for i:=1 to length(result) do if result[i]='.' then result[i]:='/';

end;

begin

s:='INSERT into Заявки values ('+#39+combobox1.text+#39+', '+#39+edit1.text+#39+','+#39+combobox2.text+#39+', #'+dateConv(DateTimePicker1.Date)+'#)';

form1.ExecSQL(s);

Form9.ShowModal;

Form8.Close;

end;

Пример регистрации заявки предоставлен на рис 3.10-3.11.

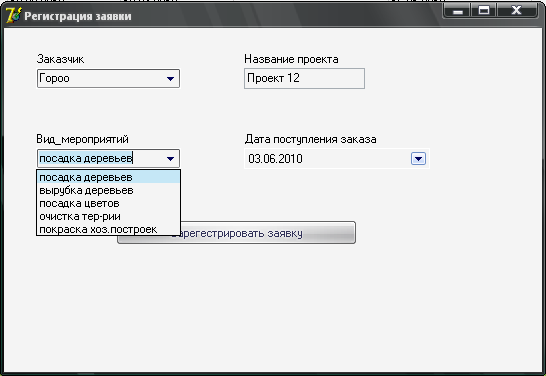


Рисунок 3.10 – Регистрация заявки

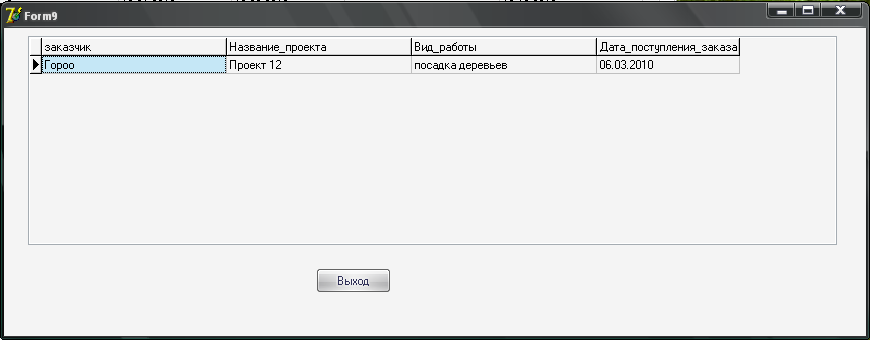


Рисунок 3.11 – Заполнение формы заявок

4) запрос, показывающий сколько часов и по какому проекту отработал определённый работник

procedure TForm10.Button1Click(Sender: TObject);

var s: string;

begin

adoquery1.SQL.Clear;

s:='SELECT Работник.Фамилия, Работник.Имя, Проект.Название, Участие.[Кол-во\_отработ\_часов] FROM '+

Работник INNER JOIN (Проект INNER JOIN (Мероприятия INNER JOIN Участие ON Мероприятия.Id\_мероприятия '+

'= Участие.Id\_мероприятия) ON Проект.Id\_проекта = Мероприятия.Id\_проекта) ON Работник.Id\_работника = Участие.Id\_работника';

adoquery1.SQL.add('select \* from ('+s+') where Фамилия ='+#39+edit1.text+#39+';');

adoquery1.active:=false;

adoquery1.active:=true;

end;

Вывод данного запроса на экран можно посмотреть на рис.3.12.

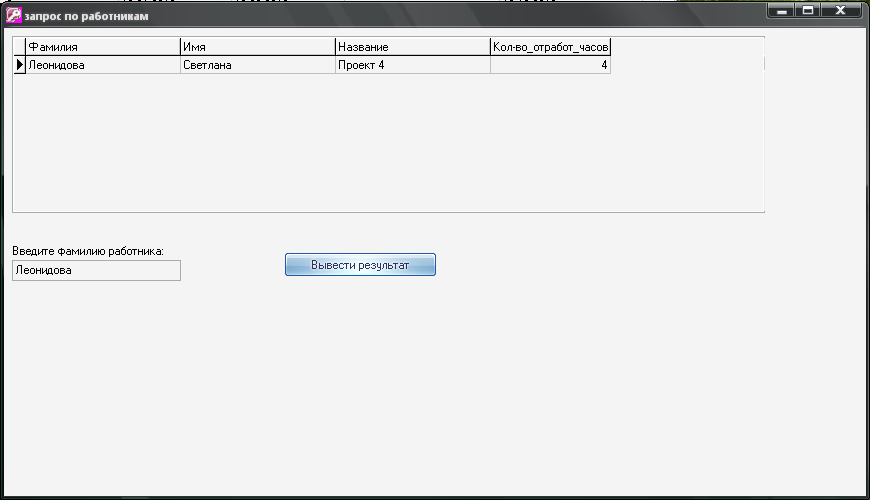


Рисунок 3.12 – Запрос по работникам

5) Вывод отчёта по завершённым проектам.

Такой отчёт может понадобиться руководству, затем чтобы оценить деятельность предприятия за определённый период. Пример предоставлен на рис.3.13.

SELECT Проект.Название, Заказчик.Наименование, Проект.Дата\_начала\_работ, Проект.Дата\_окончания, Проект.Статус

FROM Заказчик INNER JOIN Проект ON Заказчик.Id\_заказчика = Проект.Id\_заказчика WHERE Проект.Статус="Завершен"

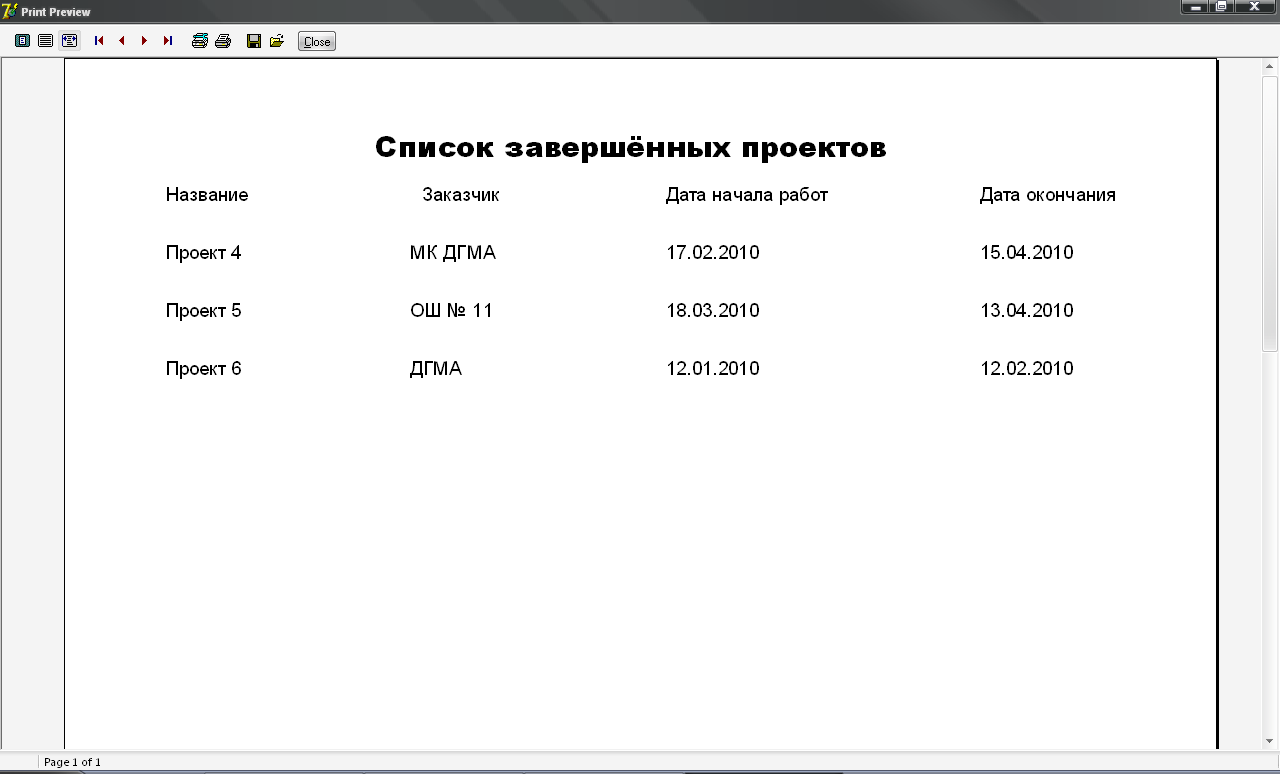


Рисунок 3.13 – Вывод отчёта перед печатью

6) Вывод отчёта по всем проектам

SELECT Проект.Название, Заказчик.Наименование, Мероприятия.Вид\_работы, Работник.Фамилия, Работник.Имя

FROM Работник INNER JOIN (((Заказчик INNER JOIN Проект ON Заказчик.Id\_заказчика = Проект.Id\_заказчика)

INNER JOIN Мероприятия ON Проект.Id\_проекта = Мероприятия.Id\_проекта)

INNER JOIN Участие ON Мероприятия.Id\_мероприятия = Участие.Id\_мероприятия) ON Работник.Id\_работника = Участие.Id\_работника;

Данный отчёт при печати показывает текущую дату, что позволяет убедиться в достоверности данных. Пример этого отчёта показан на рис.3.14.

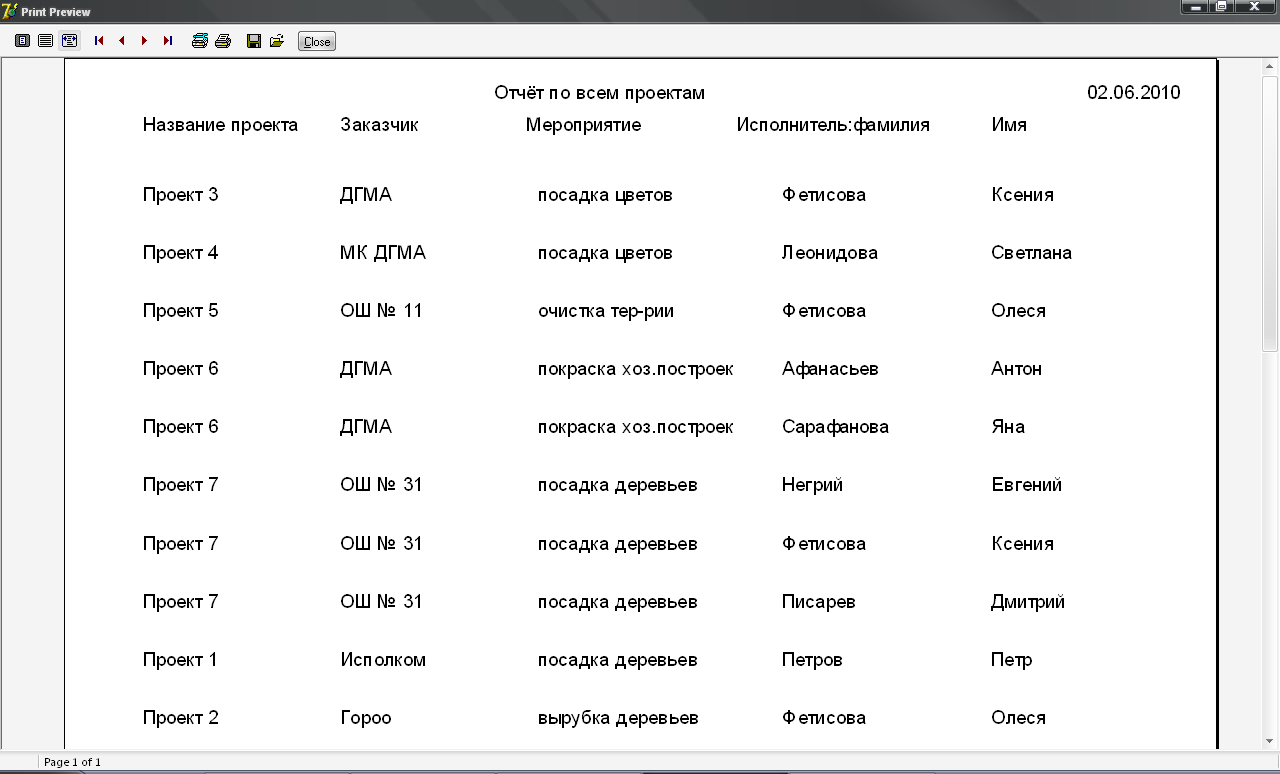


Рисунок 3.14 – Вывод отчёта по всем проектам

4. Руководство пользователя

* 1. Требования к аппаратным средствам

Техническое обеспечение АСОИ это комплекс технических средств - совокупность взаимосвязанных единым управлением и автономных технических средств, предназначенных для сбора, хранения, накопления, обработки, передачи, вывода информации, а также средств оргтехники и управления.

Техническое обеспечение должно выбираться по наиболее оптимальному соотношению цены и качества, что, в основном, зависит от финансовых возможностей предприятия. Также важнейший критерий, предъявляемый при приобретении нового технического средства – совместимость с ранее приобретенными.

Для эффективной работы программного продукта необходимо выполнение следующих требований к аппаратным и программным средствам:

* процессор 1000 MHz;
* видеокарта 32 Mb,
* 128 Mb оперативной памяти,
* Mb дискового пространства для минимальной конфигурации,
* операционная система Windows 2000/NT/Millenium/XP,
* лазерная мышь,
* клавиатура,
* принтер.
  1. Особенности запуска и работы с ПП

Программа имеет интуитивно–понятный интерфейс, обладает системой подсказок. Данные особенности программы устраняют трудности при ее запуске и работе.

После запуска программы выводится окно, в котором имеются таблицы БД, предназначенные для ввода данных пользователем. В программе поддерживается обновление данных и результатов. Это означает, что если пользователю необходимо изменить введенные данные, достаточно просто откорректировать их, после чего содержимое базы данных автоматически обновится.

Пользователю доступно главное меню программы, в котором реализованы следующие возможности Пользователю:

- создать заявку;

- редактировать базы данных работников, заказчиков, проектов, мероприятий;

- проконтролировать занятость работников;

- редактировать проект;

- выдавать отчёт.

ЗАКЛЮЧЕНИЕ

В ходе создания программного продукта был приобретен опыт разработки программного обеспечения с использованием знаний проектирования баз данных и системного программирования, что делает возможным разрабатывать программы для решения важных задач.

Были изучены и реализованы SQL–запросы, системные сообщения, фильтрация, поиск, сортировка записей файлов БД.

В данной работе SQL–запросы используются для того, чтобы из большого количества записей файлов базы данных отобразить только те, которые интересуют пользователя на данный момент.

В ходе выполнения курсовой работы были реализованы следующие функции:

* создание программного комплекса, который осуществляет управление данными в базе данных;
* возможность осуществлять анализ данных с выявлением наиболее ценной для пользователя информации;
* представление пользователю данных в наиболее информативном виде, для получения пользователем максимального количества информации;
* возможность выполнить запросы к базе данных, результат выполнения которых имеет определенную важность для конечного пользователя.

Разрабатываемая автоматизированная информационная система даёт пользователю такие преимущества, которые заключаются в сокращении времени на поиск и обработку нужных данных, также в анализе данных БД, который даёт пользователю максимальную информацию о текущей ситуации.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Кренке Д. Теория и практика построения баз данных. 8–е изд. / Д. Кренке. – СПб.: Питер, 2003. – 800 с.: ил.
2. Скотт Ф. Уилсон, Брюс Мэйплс Принципы проектирования и разработки программного продукта. Учебный курс MCSD/Пер. с англ. – 2–е изд. Издательско–торговый дом «Русская редакция», 2002 – 736 стр.
3. Федоров А.Г. Delphi 6.0 для всех.– 2–е изд., перераб. и доп.– М.:ТОО фирма «КомпьютерПресс»,1997.– 464с.
4. Архангельский А. Книга Delphi 7 Справочное пособие. – Спб: Бином–Пресс, 2007. – 1024 с.
5. Сагайда П.И. Розробка та організація баз даних у системах автоматизації проектування та управління: Навчальний посібник для студентів спеціальності 7.080402 / П.И. Сагайда. – Краматорськ: ДДМА, 2003. – 160 с.
6. Бобровский С. И. Delphi 7 Учебный курс – СПб.: Питер, 2005. – 736 с.: ил.
7. Фаронов В.В., Шумаков П.В. Delphi 5 Руководство разработчика баз данных – М.: «Нолидж», 2000. – 640., ил.
8. Скотт Ф. Уилсон, Брюс Мэйплс Принципы проектирования и разработки программного продукта. Учебный курс MCSD/Пер. с англ. – 2–е изд. Издательско–торговый дом «Русская редакция», 2002 – 736 стр.
9. Фаулер М., Скотт К. UML. Основые. – Пер. с англ. – Спб: Символ–Плюс, 2002. – 912 с., ил.
10. UML и Rational Rose / Под ред. У. Боггс, М. Боггс. – М.: Лори, 2001. – 608 с.
11. Коннолли Т., Бегг К. Базы данных.Проектирование, реализация и сопровождение. Теория и практика/Пер. с англ. – 3–е изд. М.:Издательский дом «Вильямс»,2003. – 1440 c, ил.
12. Дейт К. Дж. Введение в системы баз данных/Пер. с англ. — 8-е изд. М.: Издательский дом "Вильяме", 2005. — 1328 с., ил.

ПРИЛОЖЕНИЕ А

Листинг модуля Unit 1

var

Form1: TForm1;

implementation

uses Unit2, Unit3, Unit4, Unit5, Unit6, Unit7, Unit10,

Unit12,Unit8,unit11, Unit9;

{$R \*.dfm}

function TForm1.ExecSQL(Query: string): integer;

begin

ADOQuerySQL.SQL.Clear;

ADOQuerySQL.SQL.Add(Query);

result:=ADOQuerySQL.ExecSQL;

if result>0 then

begin

ADOQuery1.Active:=false;

ADOQuery1.Active:=true;

end;

end;

procedure TForm1.N7Click(Sender: TObject);

begin

Form1.Close

end;

procedure TForm1.N3Click(Sender: TObject);

begin

Form2.ShowModal;

end;

procedure TForm1.N2Click(Sender: TObject);

begin

Form3.ShowModal

end;

procedure TForm1.N5Click(Sender: TObject);

begin

Form4.ShowModal;

end;

procedure TForm1.N6Click(Sender: TObject);

begin

Form5.ShowModal;

end;

procedure TForm1.N9Click(Sender: TObject);

begin

AdoQuery3.Active:=True;

Form6.QuickRep1.Preview;

end;

procedure TForm1.N11Click(Sender: TObject);

begin

form7.showmodal;

end;

procedure TForm1.DBGid1TitleClick(Column: TColumn);

var s:string;

begin

s:=Column.Title.Caption;

ADOQuery1.SQL.Clear;

ADOQuery1.SQL.Add('select \* from проект order by '+s);

ADOQuery1.Active:=false;

ADOQuery1.Active:=true;

end;

procedure TForm1.n13Click(Sender: TObject);

begin

form10.showmodal

end;

procedure TForm1.DBGrid2TitleClick(Column: TColumn);

var s:string;

begin

s:=Column.Title.Caption;

ADOQuery2.SQL.Clear;

ADOQuery2.SQL.Add('select \* from Мероприятия order by '+s);

ADOQuery2.Active:=false;

ADOQuery2.Active:=true;

end;

procedure TForm1.FormShow(Sender: TObject);

begin

image1.Picture.LoadFromFile('MainFormPicture.jpg');

end;

procedure TForm1.N10Click(Sender: TObject);

begin

AdoQuery4.Active:=True;

form12.quickrep1.preview;

end;

procedure TForm1.N12Click(Sender: TObject);

begin

form8.ShowModal;

end;

procedure TForm1.N15Click(Sender: TObject);

begin

form11.showmodal;

end;

procedure TForm1.N16Click(Sender: TObject);

begin

form9.showmodal;

end;

end.