

1. Введение

Дисциплина «Базы данных» читается в шестом семестре студентам специальности 230101. Курсовая работа выполняется также в шестом семестре. Предполагается, что студент владеет основными методами проектирования информационных систем, изучаемыми в дисциплине.

2. Цель и задачи курсового проектирования

Цель курсового проектирования – научиться самостоятельно анализировать предметную область, которую предстоит автоматизировать, самостоятельно разработать информационную систему в среде промышленной системы управления базами данных (СУБД), например Visual FoxPro 9.0.

Задачами курсового проектирования являются:

- Анализ предметной области;
- Формальное описание результатов анализа с помощью ER – диаграммы;
- Отображение всех функциональных зависимостей;
- Разработка всех выходных документов для курсовой работы;
- Определение всех данных, необходимых для подготовки выходных форм;
- Разработка структуры базы данных: определение состава таблиц и их полей;
- Печать всех справок, определенных в задании, то есть, подготовлены отчеты (Report);
- Система должна быть эффективной, удобной для работы и иметь приятный внешний вид.

3. Варианты заданий

Вариант 1.

Разработать информационную систему, которая автоматизирует процесс резервирования билетов. Система должна: (а) содержать расписание рейсов (город вылета, город прилета, авиакомпания, номер рейса, время вылета, время прилета, тип самолета, дни недели выполнения рейса) и иметь возможность обновлять его. Между парой городов может быть несколько рейсов (места на заданный рейс могут продаваться на разные даты) (б) Список типов самолетов (тип самолета, план мест в салоне, например 12 рядов по 4 места в ряду: ABCD, какие места бизнес и экономического класса, места для курящих и нет); (в) показать наличие рейсов между парой городов (например, между Москвой и Петербургом); (г) список блюд, предлагаемых пассажирам (название блюда, его код); (д) иметь возможность резервировать

места (и отказаться от места) для пассажира на данный рейс с заказом блюд на питание из заданного списка. (е) выдать упорядоченный список пассажиров по заданному признаку: номер рейса, дата вылета, заказанное блюдо; (ж) выдавать список пассажиров, отказавшихся от места; (з) выдать список блюд и их количество, заказанное пассажирами; (и) иметь средства ввода и обновления всех данных.

Вариант 2.

Разработать информационную систему, которая автоматизирует обслуживание пассажиров в аэропорту агентом по регистрации. Система должна: (а) содержать расписание рейсов из Москвы; (б) списки пассажиров на рейсы разных авиакомпаний; (в) закрепить заданный рейс на стойку регистрации, (за два часа до времени вылета); (г) получить списки пассажиров на заданный рейс; (д) зарегистрировать подошедшего пассажира на рейс. Для этого надо указать номер пассажира в списке, номер паспорта, вес сдаваемого багажа и ручной клади; (е) снять пассажира с рейса, то есть отменить регистрацию, не удаляя его из списка; (ж) повторно его зарегистрировать; Завершить регистрацию, выдав списки пассажиров, прошедших регистрацию, неявившихся пассажиров, сводную ведомость регистрации с указанием количества пассажиров, с указанием пола, суммарный вес багажа и ручной клади и вес сверхнормативного багажа.

Вариант 3.

Разработать информационную систему, которая автоматизирует процесс резервирования билетов. Система должна: (а) содержать расписание рейсов (город вылета, прилета, авиакомпания, номер рейса, тип самолета, дни недели выполнения рейса) и иметь возможность обновлять его. Между парой городов может быть несколько рейсов (места на заданный рейс могут продаваться на разные даты) (б) Список типов самолетов (тип самолета, план мест в салоне, например 12 рядов по 4 места в ряду: ABCD); (в) показать наличие рейсов между парой городов (например, между Москвой и Петербургом); (в) реализовать транзакцию резервирования билетов для пассажира, для этого должна вводиться фамилия пассажира, дата выбранного им рейса, его номер и место в салоне самолета; (в) напечатать список пассажиров на данный рейс; (г) отказаться от места, то есть освободить место по заданной фамилии; (д) напечатать список пассажиров, отказавшихся от места. (е) иметь средства ввода, обновления и удаления всех данных. Напечатать результат регистрации в виде билета: фамилия, имя, рейс, откуда, куда, стоимость.

Вариант 4.

Разработать информационную систему, которая автоматизирует процесс резервирования билетов. Система должна: (а) содержать расписание рейсов (город вылета, прилета, авиакомпания, номер рейса, тип самолета, дни недели выполнения рейса) и иметь возможность обновлять его; (б)

Список типов самолетов (тип самолета, план мест в салоне, например 12 рядов по 4 места в ряду: ABCD, какие места бизнес и экономического класса, места для курящих и нет), их стоимость; (в) показать наличие рейсов между парой городов (например, между Москвой и Петербургом, рейсов может быть несколько); (г) иметь возможность продать места первого и бизнес классов, а также места для курящих и некурящих на тот же рейс на разные даты; (в) создавать список пассажиров, зарезервировавших места в бизнес и экономическом салоне; (г) найти пассажира по фамилии с выдачей номера рейса, класса и типа места (курящий- некурящий); (д) иметь средства ввода и обновления всех данных. Напечатать результат регистрации в виде билета: фамилия, имя, рейс, откуда, куда, класс бронирования, стоимость.

Вариант 5.

Разработать информационную систему, которая автоматизирует деятельность поликлиники. Система содержит: Списки врачей, (ФИО, номера, специальность врача, дату приема на работу, номер кабинета); Списки больных (ФИО, номер медкарты, номер страхового полиса, место работы, год рождения, пол); Списки болезней: название, ее номер, признак группы болезни (например 2 – кардиология, 3 – хирургия, 4 – терапия и т.д.); Списки посещений (номер больного, номер врача, номер болезни, дата). Стоимость приема в соответствии со страховой компанией. Система должна: проверить правильность приема врачом больного (номер группы болезни должен совпасть со специальностью врача) и выполняет следующие функции: (а) вводит данные по персоналу врачей; (б) вводит данные о больных, (в) регистрирует посещения; (г) выдает справки о посещении больного врачами по дате, месяцу, году; (д) выдает справки о приеме врачом больных за день, месяц, год; (ж) выдает справки по врачам данной специальности (например, терапевты) о числе принятых больных за месяц, год; (з) выдает справки по рейтингу наиболее популярных болезней.

Вариант 6..

Разработать информационную систему, которая автоматизирует учебный процесс и обрабатывает данные о преподавателях, учебных дисциплинах и группах. Система содержит (а) список преподавателей (ФИО, кафедра, должность, номер преподавателя, дата приема на работу); (б) Список дисциплин(название, код, семестр, специальность); (в) Список дисциплин кафедры (код, номер преподавателя, количество часов). Система должна: (г) учитывать возможность чтения одной дисциплины несколькими преподавателями и наоборот; (д) учитывать возможность изучения группой нескольких дисциплин и наоборот; (ж) иметь средства для ввода и модификации всех перечисленных данных; (з) выдавать справки по преподавателю - какие дисциплины он читает, для каких групп и сколько часов; (е) по запросу, указав код дисциплины, напечатать список

преподавателей, ее читающих, напечатать список дисциплин заданной кафедры.

Вариант 7.

Разработать информационную систему, которая автоматизирует учебный процесс и обрабатывает данные о преподавателях, кафедрах, дисциплинах, читаемых преподавателями кафедры. Система содержит (а) списки кафедр (название, номер, заведующий, факультет); (б) списки преподавателей выбранной кафедры (ФИО, дата приема на работу, должность, номер преподавателя,); (в) Список дисциплин (название, код, семестр, специальность). Система должна: (г) выдавать списки преподавателей заданной кафедры; (д) выдать название кафедры для заданного преподавателя и все сведения о нем; (е) выводить список дисциплин, читаемых выбранной кафедрой; (ж) напечатать список кафедр университета и список кафедр выбранного факультета; (з) иметь средства для ввода и модификации всех перечисленных данных.

Вариант 8.

Разработать информационную систему, которая автоматизирует работу кафедры по распределению учебной нагрузки. Система должна: (а) содержать список дисциплинах, читаемых кафедрой: (название, код, специальность, для которой она читается, количество лекций, практических и лабораторных занятий в учебных часах, признак экзамена или зачета, признак наличия курсовой работы, семестр) (б) список преподавателей кафедры (ФИО, дата приема на работу, должность, степень, звание, номер преподавателя); (в) список преподавателей и дисциплин (номер преподавателя, код дисциплины, специальность); (г) количество групп и студентов по специальностям; (в) выдавать количество часов по заданной дисциплине; (г) выдавать количество часов для заданного преподавателя; (д) напечатать список преподавателей, у которых количество часов меньше или больше заданного предела (например, 700); (г) иметь средства ввода и обновления всех данных.

Вариант 9.

Разработать информационную систему, которая автоматизирует обслуживание клиентов видеокассетами. Система должна: (а) содержать сведения о видеофильме: (компания-производитель, название, год выпуска, основные исполнители, характер фильма(боевик, триллер и т.д.)); (б) сведения о компании-производителе: страна, город, название, год основания; (в) данные о выдачах: номер фильма, фамилия и адрес клиента, дата выдачи, дата возвращения, залог, оплата; (г) выдавать справки о частоте выдачи фильма, прибыли, получаемой от проката фильма; (д) выдавать справки о выдаче за день (дату), названия фильмов и срок проката; (е) выдавать справки по компании производителе с названиями фильмов за заданный срок; (ж) выдавать справки по видам фильмов; (з) выдавать список (рейтинг)

самых популярных фильмов; (и) иметь средства ввода и обновления всех данных.

Вариант 10.

Разработать информационную систему, которая автоматизирует деятельность центра обучения фирмы. Сотрудники фирмы должны раз в год прослушать набор курсов, предназначенных для их отдела. Система должна: (а) содержать сведения о сотрудниках фирмы: (имя, фамилия, номер должности, номер отдела, дата приема на работу); (б) сведения о курсах, которые преподаются в центре: (название, номер, длительность в днях, предназначение (для какого отдела), краткое содержание); (в) сведения о прохождении обучения: (номер сотрудника, номер курса, дата прохождения); (г) выдавать справки о сотрудниках, проходящих обучение в данном месяце, году; (д) выдавать справки о сотрудниках, которые должны проходить обучение (срок больше года); (е) печатать справки о сотрудниках отделов с курсами, которые они прослушали; (ж) иметь средства ввода и обновления всех данных.

Вариант 11.

Разработать информационную систему о деятельности кирпичного завода. Система должна: (а) содержать данные о производимой продукции: кирпичи: размер (напр. Стандартный кирпич 25x12x6 см), стоимость и наличие на складе (штук); (б) данные о необходимом количестве кирпичей для домов стандартных размеров (напр., дом 6x6x2,5x0.5, дом 10x8x3 и т. д.)(один кубометр кирпичной кладки стандартного кирпича содержит около 400 кирпичей); (в) данные о заказах и заказчиках: дом (его размеры), фамилия, расстояние доставки кирпичей, адрес; (г) средства ввода данных о заказчике); (д) расчет стоимости заказа в зависимости от сорта кирпича, размера дома и расстояния (известна стоимость 1 км), возможность его модификации (например, изменить тип кирпича и, как результат пересчитать стоимость заказа) и дата поставки; (е) печать справок о количестве заказов на кирпичи и справок о наличии всех типов кирпичей на складе и проверки возможно ли выполнить заказ; (г) печать заказа на доставку кирпича с указанием количества и стоимости кирпича, фамилии заказчика, адрес, стоимость его доставки(г) иметь средства ввода и обновления всех данных.

Вариант 12.

Разработать информационную систему, которая автоматизирует деятельность авторемонтного предприятия. Система должна: (а) содержать сведения по рабочим мастерской: (имя, фамилия, номер, специальность (кузовщик, электрик, слесарь и т.д.), дата приема на работу, зарплата; (б) сведения о находящихся в ремонте машинах: (тип, регистрационный номер, дату поступления в ремонт, характер неисправности, порядковый номер,

присвоенный в мастерской, дату окончания ремонта (если он завершен) или планируемое окончание ремонта, стоимость завершенного ремонта; (в) сведения о ремонте: порядковый номер, регистрационный номер машины, тип ремонта, список рабочих, ремонтирующих машину; (г) печатать справки о доходах мастерской за месяц, год; (д) печатать справки о машинах, находящихся в ремонте и по типу ремонта; (е) печатать список рабочих, участвующих в ремонте данной машины; (ж) иметь средства ввода и обновления всех данных.

Вариант 13.

Разработать информационную систему, которая автоматизирует деятельность магазина. Система должна: (а) содержать сведения об отделах магазина: (название, номер отдела, имя заведующего); (б) сведения о сотрудниках: (имя, фамилия, номер, дата поступления на работу, номер отдела, должность, специализация); (в) сведения о товарах, поступающих в магазин на склад (название товара, его номер, отдел, в котором он продается, стоимость, количество единиц); (г) продажи товара – товар может продаваться в нескольких отделах: (номер товара, номер отдела, дата продажи, номер продавца); (д) печатать справки о продажах за месяц, год; (е) выдавать справки о продажах в отделе, заданного вида товара, заданного продавца за определенный период времени всех товаров; (ж) печатать справки о количестве единиц данного вида товара; (з) иметь средства ввода и обновления всех данных. При продаже товара его количество на складе должно уменьшаться, система должна проверять хватает ли товара на продажу.

Вариант 14.

Разработать информационную систему, которая автоматизирует деятельность гостиницы. Система должна: (а) содержать данные о номерах: (порядковый номер, номер, количество мест в номере, этаж, удобства (телевизор, холодильник, телефон, интернет и т.д.), стоимость (с завтраком или без); (б) данные о клиентах: (имя, фамилия, адрес, номер, им занимаемый, дата приезда, дата отъезда, состояние оплаты (оплачено все, часть); (в) выдавать справки о свободных номерах и о конкретном номере; (г) печатать справки о доходах гостиницы за год, месяц, день (задаваемые); (д) средства регистрации клиента и предложения ему номера (интерфейс пользователя, содержащий окна с данными о свободных номерах, их стоимости, удобствах); (е) печать документа для клиента с указанием названия гостиницы, фамилии имени, дат приезда, отъезда, стоимости за сутки, количество суток проживания и итоговую стоимость; (ж) иметь средства ввода и обновления всех данных.

Пример распечатки:

Отель Президент

Михаель Шумахер

Номер 113

Пребывание с 01.03.09

по 05.03.09

Стоимость за сутки 2000 руб
Всего 2000 x 4 = 8000 руб

Вариант 15.

Разработать информационную систему, которая автоматизирует деятельность отдела персонала. Система должна: (а) содержать сведения о сотрудниках: (имя, фамилия, номер отдела, номер должности, дату приема, семейное положение, образование, учебной заведение, пол, адрес); (б) список должностей: (название, номер должности, вилку оклада (напр. 30000-42000 р.)); (в) список отделов: (название, номер отдела, адрес, руководитель); (г) содержать список перемещений сотрудников: приказ (МЕМО поле), номер приказа, старая должность, новая должность, дата перемещения, старый оклад, новый оклад; (д) печатать справки о перемещениях за год, месяц, день; (е) печатать справки о заданном сотруднике: личные данные, перемещения; (ж) печатать справки о принятых или уволенных в заданном году сотрудниках, список перемещений в заданном году; (з) иметь средства ввода и обновления всех данных.

Вариант 16.

Разработать информационную систему, которая автоматизирует деятельность библиотеки. Система должна: (а) содержать данные о книгах: (название книги, автор, название издательства, год издания, номер библиотечный, номер тематики, количество экземпляров); (б) издателях: (название издательства, адрес, место издания); (в) читателей: (фамилия, адрес, город, номер карты в библиотеке); (г) выдачах: (номер карты в библиотеке, номер библиотечный, дата выдачи); (д) выдавать справки по количеству взятых книг по заданной тематике, за заданный год (упорядоченный по убыванию); (е) выдавать справки по заданному читателю и его книгам; (ж) выдавать справки по рейтингу книг по данной тематике; (з) справку по данной книге и взявшим ее читателям; (и) иметь средства ввода и обновления всех данных.

Вариант 17.

Разработать информационную систему, которая автоматизирует деятельность библиотеки. Система должна: (а) содержать данные о книгах: (название книги, автор, название издательства, год издания, номер библиотечный, шифр тематики, количество книг, признак – читальный зал - на дом); (б) издателях: название издательства, адрес, место издания; (в) читателей: фамилия, адрес, город, номер карты в библиотеке; (г) выдачах: номер карты в библиотеке, номер библиотечный, дата выдачи, дата возврата; (д) выдавать справки по книгам заданного издательства, включая количество экземпляров; (е) печатать справки по читателям, просрочившим заданный срок от заданной даты; (ж) печатать списки читателей по заданной тематике, а также списки должников (не вернувших книгу в срок); (з) иметь средства ввода и обновления всех данных.

Вариант 18.

Разработать информационную систему штурманского обеспечения полетов. Система должна: (а) содержать данные о промежуточных пунктах маршрута (ППМ): (название, номер ППМ, координаты ППМ - географическую широту и долготу) (смотри справочник); (б) данные о самолете: (тип самолета (напр. ТУ-134), его бортовой номер (напр: 65123), скорость взлета (320 км), скорость посадки (280), крейсерскую скорость (850 км), взлетную массу); (в) список трасс (код трассы, номерППМ1, номерППМ2, номерППМ3, и т. д.) выдавать план полета: по коду трассы рассчитать маршрут полета по трассе, состоящей из ППМ. Расстояние между ППМами рассчитывается по географическим координатам. Предполагается, что полет проходит по отрезкам прямых линий между ППМ. План полета самолета, указанного типа – это таблица содержащая название трассы (откуда куда), столбец со списком ППМ, расстояния между каждой парой ППМ и итогового расстояния; (г) иметь средства ввода и обновления всех данных. (Справочник по данным ППМ имеется)

Пример плана полета:

Трасса Москва – Иркутск	Тип самолета – ТУ-154
Название ППМ	Расстояние
Домодедово	0
Подольск	100
Екатеринбург	2300\
...	
Иркутск	1200
Расстояние	5600

Вариант 19.

Разработать информационную систему штурманского обеспечения полетов. Система должна: (а) содержать данные о промежуточных пунктах маршрута (ППМ): (название, номер ППМ, координаты ППМ - географическую широту и долготу (смотри справочник); (б) данные о самолете: тип самолета (напр. ТУ-204), его бортовой номер (5 цифр), скорость взлета, скорость посадки, крейсерскую скорость, взлетную массу; (в) список трасс (код трассы, номерППМ1, номерППМ2, номерППМ3, и т. д.); (г) печатать план полета: по коду и названию трассы рассчитать расстояние и время полета данного типа самолета по трассе, состоящей из списка ППМ. Время полета рассчитывается: между пунктом вылета и первым ППМ по скорости взлета, на последнем участке по скорости посадки, в остальных участках (между парами ППМ) по крейсерской скорости. Итог расчета - общее время полета по трассе: (г) иметь средства ввода и обновления всех данных. Пример плана полета

Трасса Москва – Иркутск	Тип самолета – ТУ-154	
Название ППМ	Расстояние	Время

Подольск	100	0.20
Всего	5400	5.30

Вариант 20.

Разработать информационную систему штурманского обеспечения полетов. Система должна: (а) содержать данные о промежуточных пунктах маршрута (ППМ): название, его номер, координаты ППМ - географическую широту и долготу (смотри справочник); (б) данные о самолете: тип самолета (напр. ТУ-134), его бортовой номер, крейсерскую скорость, взлетную массу, массу заправляемого топлива, километровый расход топлива; (в) список трасс (код трассы, номерППМ1, номерППМ2, номерППМ3, и т. д.) (г) выдавать план полета: по введенным названиям трассы и типа самолета. План полета содержит таблицу: Заголовок: название ППМ, расход топлива на участке, расход топлива по участкам с накоплением. Расход топлива рассчитывается: по километровому расходу топлива в зависимости от расстояния между ППМами. (д) иметь средства ввода и обновления всех данных. **Пример** плана полета:

Трасса Москва – Иркутск		Тип самолета – ТУ-154
название ППМ,	расход топлива на участке,	расход топлива по участкам с накоплением
Подольск	200	200
Климовск	170	370

Вариант 21.

Разработать информационную систему штурманского обеспечения полетов. Система должна: (а) содержать данные о промежуточных пунктах маршрута (ППМ): название, координаты ППМ - географическую широту и долготу (смотри справочник); (б) данные о самолете: тип самолета (напр. ИЛ-62), его бортовой номер (5 цифр, например 86123), крейсерскую скорость, взлетную массу, массу заправляемого топлива, километровый расход топлива; (в) печатать план полета: по введенным названиям трассы и типа самолета. рассчитать расход топлива и время полета по участкам трассы километровому расходу топлива и проверить, хватает ли на полет заправленного топлива; (г) иметь средства ввода и обновления всех данных.

Пример плана полета:

Трасса Москва – Иркутск		Тип самолета – ИЛ-62
название ППМ	расход топлива	время полета
Подольск	200	1.20
Всего	20000	5.40

Вариант 22.

Разработать информационную систему о деятельности книжного магазина. Система должна: (а) содержать данные о продаваемых книгах: название, автор, год выпуска, тематика, дата поступления в магазин, количество экземпляров; (б) ежедневный отчет о проданных книгах: дата, автор, название, количество экземпляров, время продажи; (в) предложения на

поставку книг: название, автор, год выпуска, тематика, количество экземпляров; (г) выдавать справки по количеству и сроках продажи книг по заданному автору, тематике, году выпуска за заданный срок (день, месяц, год); (е) отбор предложений по продажам в зависимости от проданных экземпляров (иметь возможность напечатать таблицу с процентами продаж по авторам, тематике); (ж) Построить график рейтинга самых популярных книг; (з) иметь средства ввода и обновления всех данных. При продаже нескольких экземпляров книги, количество имеющихся книг должно уменьшаться.

Вариант 23.

Разработать информационную систему о деятельности фирмы по продажам компьютеров. Система должна: (а) содержать данные о продаваемых типах компьютеров: марка, фирма-производитель, количество штук, технические характеристики (процессор, ОП, НЖМД и т.д.); (б) данные о программном обеспечении, прилагаемом для продажи: назначение, фирма-производитель, объем диска, стоимость; (в) заказы на покупку компьютеров: организация, адрес, количество штук, дата; (г) выдавать справки по продажам за определенный промежуток времени (день, месяц, год); (д) справки по популярности продаваемых типов компьютеров (упорядоченные по количествам продаж); (е) выдавать справки по сезонности продаж в виде гистограмм по месяцам; (ж) иметь средства ввода и обновления всех данных.

Вариант 24.

Разработать информационную систему о деятельности фирмы по продажам бытовой техники (телевизоров, магнитофонов и т. п.). Система должна: (а) содержать данные о наличии товаров по видам техники: вид, марка, фирма-производитель, дата выпуска, стоимость, количество; (б) данные о продаваемой за день технике по видам, фирмам, стоимости; (в) выдавать справки по проданной продукции по заданному типу, фирме за определенный период времени (год, месяц); (г) выдавать справки по популярности продаваемых изделий по типам, фирмам, стоимости (в виде гистограмм); (д) выдавать справки по сезонности продаж в виде гистограмм по месяцам и по типам изделий; (е) иметь средства ввода и обновления всех данных.

Вариант 25

Разработать информационную систему о деятельности фирмы по производству косметики (кремов, шампуней, зубной пасты и т.д.). Система должна: (а) содержать данные о производимой продукции: вид, название, стоимость, объем производства (единиц в день), наличие на складе (штук); (б) данные о наличии заказов на поставку продукции: название покупателя, адрес покупателя, вид продукции, срок поставки, количество штук; (в) печатать справки по наличию заказов по заданному виду продукции на

заданный срок (год, месяц, день); (г) справки по наличию на складе и спросу на заданный вид продукции за заданный период; (д) выдавать рекомендации по изменению производства в зависимости от заказов; (ж) печатать справки по сезонности продаж в виде гистограмм по месяцам и по видам продукции; (е) иметь средства ввода и обновления всех данных.

Вариант 26

Разработать информационную систему о деятельности ресторана. Система должна: (а) содержать данные о стоимости блюд: название, краткая характеристика, стоимость; (б) данные об официантах: фамилия, имя, дата приема на работу, его номер; (в) официант должен иметь возможность вводить данные о заказанных клиентом блюдах, номер столика, свой номер; (г) система должна подсчитать стоимость заказа с указанием номера столика и номера официанта; (д) печатать справки по деньгам, заработанным официантом на заданный срок (год, месяц, день) с указанием его фамилии и номера; (ж) печатать справки по популярности блюд, упорядоченных по убыванию; (е) иметь средства ввода и обновления всех данных.

Вариант 27

Разработать информационную систему о деятельности агентства по туризму. Система должна: (а) содержать данные о гостиницах в разных странах (Египет, Турция, Таиланд, Испания и т. д., около 10 стран), их стоимости в зависимости от сезона (лето, осень, зима), длительности пребывания (7, 10, 14 дней) и уровня номера (одиночный (SINGL), двойной (DBL) с ребенком и т.д.); (б) выдавать по запросу клиента информацию о стоимости поездки в заданную страну, в заданное время года и на определенную длительность пребывания; (в) выдавать справки о стоимости поездки в зависимости от заданного уровня стоимости (например 1000 долл на одного человека); (г) напечатать документ с подтверждением бронирования на заданное количество туристов с указанием фамилий, имен, паспортных данных, даты рождения; (д) иметь средства ввода и обновления всех данных.

Например:

Отель Royal Beach, Thailand	
Срок	10.10.10 - 20.10.10
Петров Иван	63 N 123456
Петрова Ирина	63 N 654321
Стоимость	1800 USD

Рекомендации по выполнению курсовой работы

Курсовая работа выполняется в среде СУБД Visual FoxPro 9.0. База данных создается средствами СУБД. Создание базы данных подробно описано в методических описаниях по выполнению лабораторных работ.

Основные компоненты курсовой работы

Структура базы данных;
Вход в разработанную систему через логин и пароль;
Наличие меню;
Формы и окна, отображающие все запросы;
Запросы на языке SQL
Программы, реализующие запросы

Структура базы данных

Структура базы данных реализуется в Database Designer. Пример базы данных приведен на рис. 1

Оценка эффективности базы данных

База данных является основной частью любой информационной системы, поэтому эффективная структура базы данных и рациональные методы обработки информации, находящейся в ней, оказывают большое влияние на эффективность всей системы в целом. Два основных показателя, характеризующие качество информационной системы, это время поиска данных в базе и физический объем данных на диске. Как правило, они конфликтуют между собой: чем больше объем данных, тем меньше время поиска и наоборот. Оба показателя зависят от организации данных в базе.

При одном и том же наборе данных объем базы может меняться существенно.

Рассмотрим пример. Мы хотим разработать информационную систему, которая по запросу выдает информацию:

- адрес жителя по его фамилии;
- список жителей заданного года рождения;
- список жителей заданной улицы;
- список работников заданного предприятия.

а также:

- * имеет средства для ввода данных о новых жителях;
- * позволяет менять названия улиц, фамилии жителей, название предприятия

Первое, что приходит в голову - сделать один файл базы данных и там хранить всю необходимую информацию. Поля базы данных представлены на ниже.

Название поля	Тип данных	Длина (байт)
Фамилия	Character	18
Имя	Character	12
Дата рождения	Date	8
Улица	Character	24
Номер дома	Numeric	3

Номер квартиры	Numeric	3
Предприятие	Character	48

Уже при вводе данных вы ощутите неудобство: название улицы и название предприятия вам придется вводить, многократно повторяя одни и те же названия. Еще большие проблемы возникнут, если вам придется изменить название улицы или предприятия: вам придется просматривать весь файл базы данных, чтобы убедиться, что все изменения выполнены. И, конечно, сразу возникнет проблема избыточности данных: одни и те же названия будут многократно присутствовать в базе данных, и, если вы ошиблись при вводе названия улицы, то эта запись уже не будет найдена, например, при поиске жителей одной улицы.

База данных будет намного проще в эксплуатации и занимать значительно меньший объем, если мы создадим три таблицы:

Таблица первая:

Таблица "Житель"	Тип данных	Длина (байт)
Фамилия	Character	18
Имя	Character	12
Дата рождения	Date	8
Номер улицы	Numeric	3
Номер дома	Numeric	3
Номер квартиры	Numeric	3
Номер предприятия	Numeric	3

Таблица вторая

Таблица "Улица"	Тип данных	Длина (байт)
Улица	Character	24
Номер улицы	Numeric	3

Таблица третья

Таблица "Предприятие"	Тип данных	Длина (байт)
Предприятие	Character	48
Номер предприятия	Numeric	3

При первоначальном вводе данных в файлы целесообразно сначала ввести улицы и предприятия с их номерами, а при вводе фамилий создать окна, где будут высвечиваться список улиц с их номерами и список предприятий с их номерами, и эти номера добавлять в файл фамилий. Такая база данных будет иметь значительно меньшую избыточность, и, как результат, занимать значительно меньше места на диске.

Оценим размер базы данных в первом и во втором случае для 1000 жителей, 10 улиц, 10 предприятий.

В первом случае: длина записи файла 116 байтов x 1000 жителей = 116000 байтов. Общий объем базы 116000 байтов

Во втором случае: длина записи файла “Житель” 50 байтов * 1000 = 50000 байтов, длина записи файла “Улица” 27 байтов * 10 = 270 байтов, длина записи файла “Предприятие” 51 байт * 10 = 510 байтов. Общий объем базы 50780 байтов.

Во втором варианте структуры базы данных объем уменьшился более чем в два раза.

Скорость поиска данных в базе будет более высокой за счет уменьшения количества записей в каждом файле.

Это означает, что даже простейший анализ прикладной области приводит к существенному улучшению показателей эффективности базы данных.

Примечание:

- В заданиях приведены ориентировочные данные (не содержимое таблиц), в случае, если выполняющий КП считает нужным, он может добавить недостающие данные.
- Все один раз введенные данные должны при их поиске отображаться в ComboBox!

Проверка правильности ввода логина и пароля

Для входа в систему необходимо указать логин и пароль. Система должна проверить, зарегистрирован ли данный пользователь в системе. Для этого система должна хранить данные о зарегистрированных пользователях.

Например, база данных состоит из 3 таблиц: операторов (`t_operators`), склада с товаром (`store`) и продаж (`sales`) (Рис.1.).

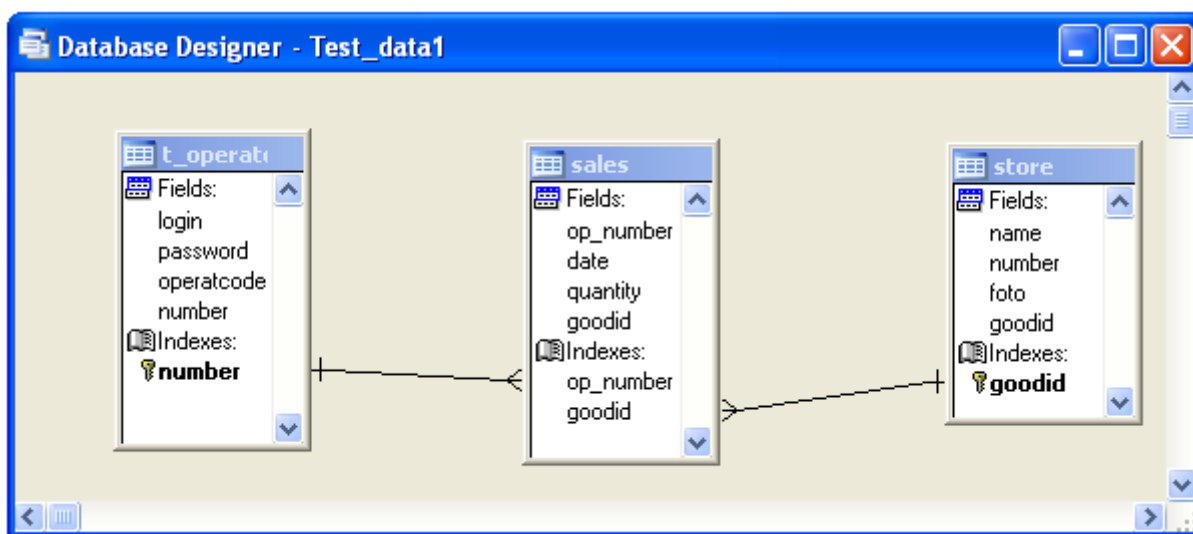


Рис. 1. Пример базы данных

Созданная таблица **t_operators** содержит данные об операторах. В ней хранятся данные об операторах, допущенных к работе с системой. В ней, как минимум, хранятся данные о допущенных к работе именах операторов (логинах) и паролях, кодах и порядковых номерах. Для защиты системы и ввода данных для проверки данных о пользователях создаем форму, которая проверяет допущенных к работе операторов. Первое поле **Имя оператора** (Логин), второе **Пароль**. Таблица служит только для ввода данных об операторах, данные из нее не заносятся ни в одну таблицу. Создаем в форме 2 поля: **Text1** и **Text2**

Определяем первое поле текста как **cOperator** на закладке **Data**

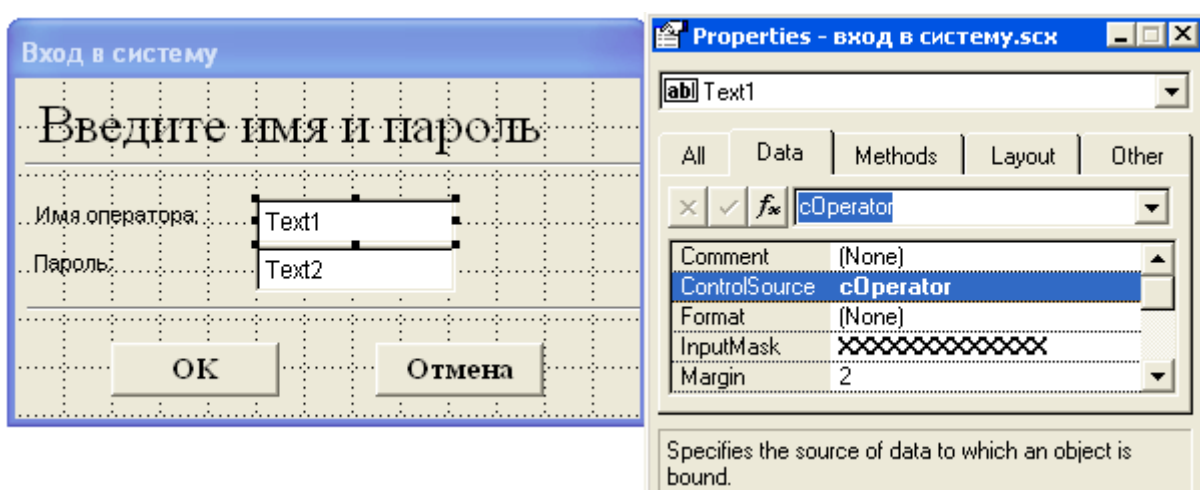


Рис. 2. Создание первого поля для ввода данных

Определяем второе поле текста как **cPassword** на закладке **Data**

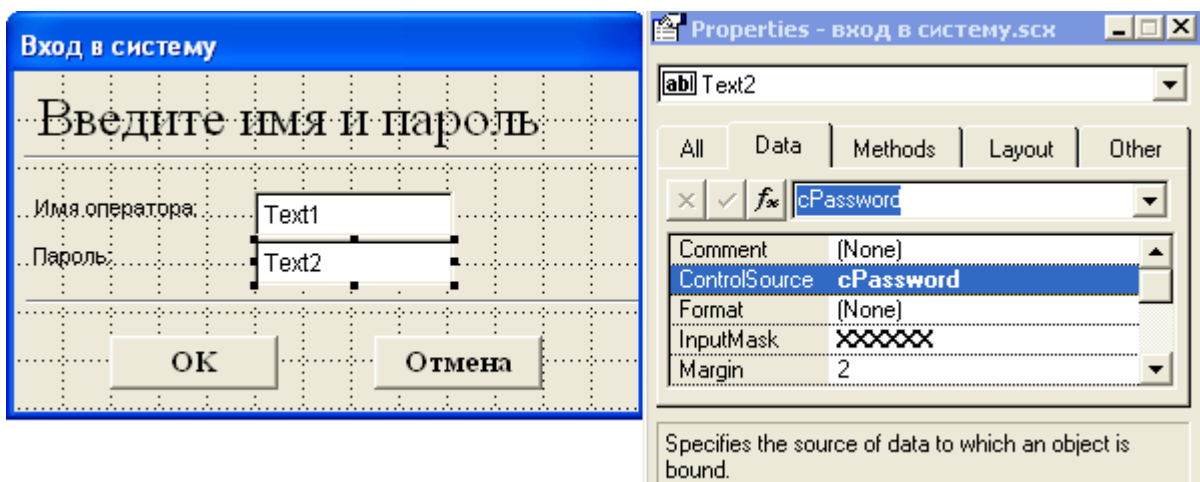


Рис.3. Создание второго поля для ввода данных

При вводе пароля можно использовать вместо пароля символы, которые скрывают пароль (например, звездочки *). Для этого в **Properties** поля ввода

Text2 на закладке **Layout** в свойстве **Password Char** указываем символ, который будет отображаться вместо вводимого пароля (например *).
Определяем название командной кнопки - **OK** на закладке **Layout**

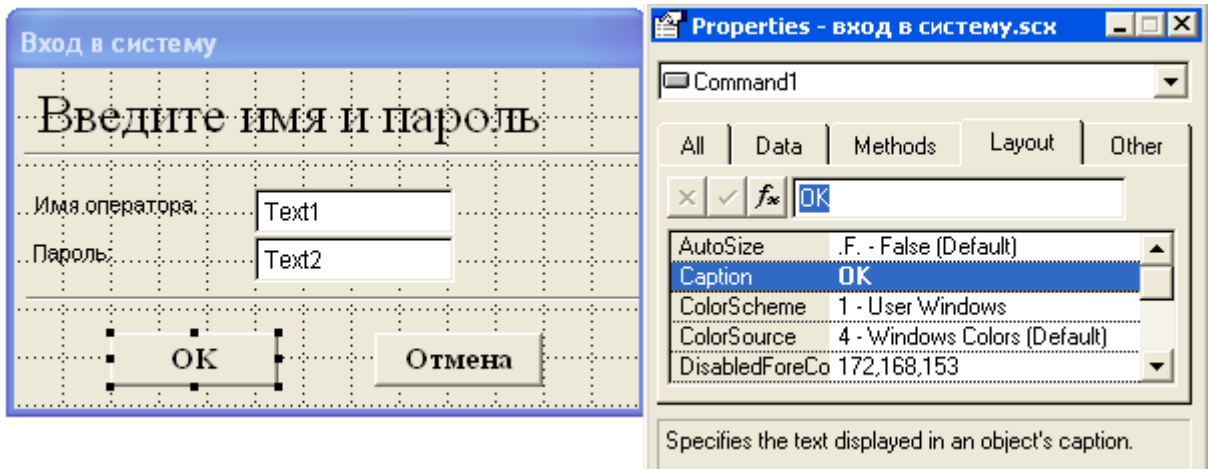


Рис.4. Задание имени для поля

Пишем процедуру пользователя при нажатии кнопки **OK** (**user Procedure** на закладке **Methods**, на методе **Click Event**).

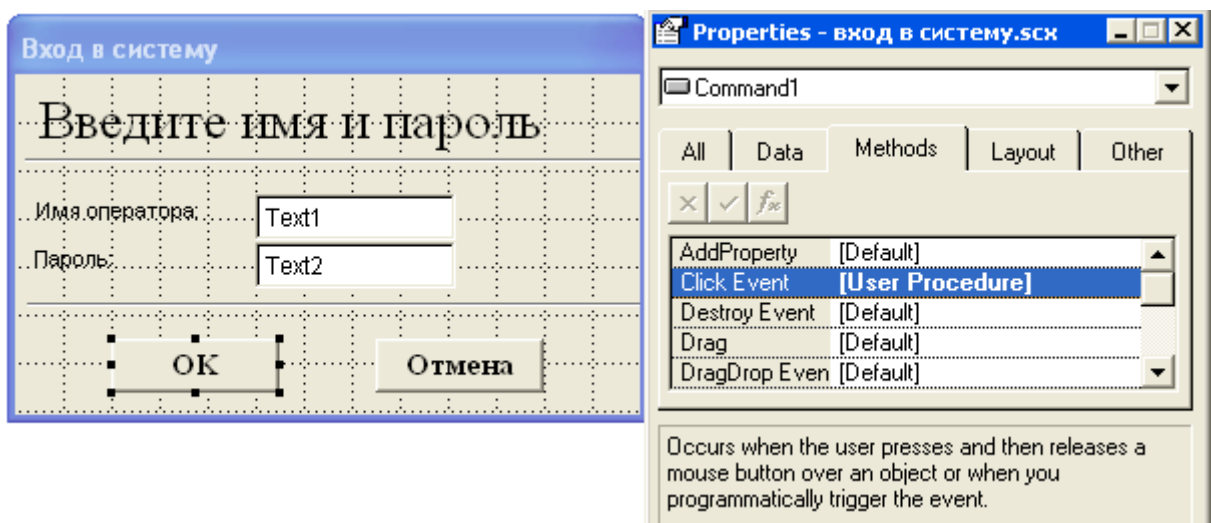


Рис.5. Написание процедуры, выполняемой при нажатии кнопки

Готовим также процедуру для кнопки **Отмена**, которая состоит просто в закрытии формы в случае, если мы передумали входить в систему.

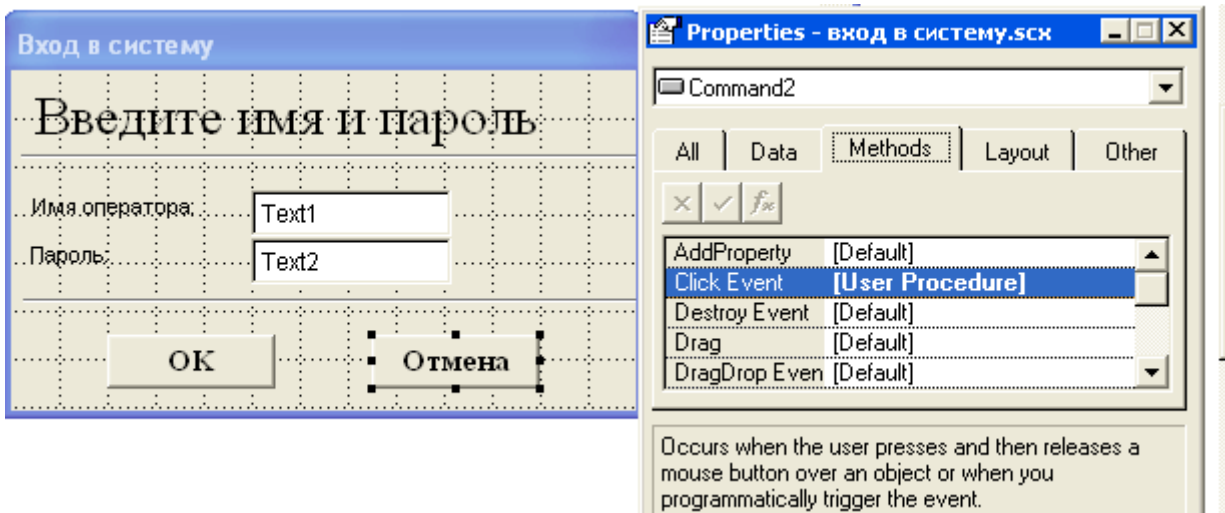


Рис.6. То же для другой кнопки.

User Procedure на кнопке **Отмена**:

thisform.release

*Закрывать форму

Процедура пользователя на кнопке ОК:

```

LOCAL cArr1[1], cArr2[1], t1[1], wasError
wasError = 0
    select login from t_operators where login ==
alltrim(cOperator) into array cArr1
    if vartype(cArr1(1)) == "L"
        MESSAGEBOX("неверное имя")
        cErrorMessage = "Неверное имя"
        cOperator = " "
        do form errorout
    else
        select password from t_operators where login ==
alltrim(cOperator) into array cArr1
        if alltrim(cArr1(1)) != alltrim(cPassword) or
len(alltrim(cPassword)) < 6
            MESSAGEBOX("неверный пароль")
            cErrorMessage = "Неверный пароль"
            cPassword = " "
            do form "ErrorOut"
        else
            *
            * if date() > {^2007-04-30}
            * cErrorMessage = "Обратитесь к
разработчику"
            * do form "ErrorOut"    ВЫВОДИТ сообщение
из cErrorMessage
            * quit
        endif
    endif

```

```

        select operatcode from t_operators where
login == alltrim(cOperator) into array cArr2
        cOperatcode = cArr2[1]
        MESSAGEBOX (cOperatcode)
        MESSAGEBOX ("Хорошо")
    endif
Endif
thisform.release      *закрывать форму

```

Картинку на форме можно вставить на свойствах формы Properties на закладке Layout, в свойстве Picture.

Проверка правильности ввода даты.

Например, дата repairdate не должна быть меньше 1950 года месяц не больше 12 и день не больше 31 и не больше текущей даты, которая задается функцией Date()

```

if year(repairdate) < 1950 or year (repairdate) >
year(date()) or month (repairdate) > 12 or day
(repairdate) > 31
    cErrorMessage = "Неверная дата"
*   lCF = .t.
endif

```

Функция Date () определяет текущую дату.

Процедура входа в систему:

Обычно на кнопке, запускающей систему, задаются некоторые параметры, общие для всей системы. Это можно сделать при запуске первой формы, набрав процедуру на методе Activate (Запуск формы):

Набор некоторых установок:

```

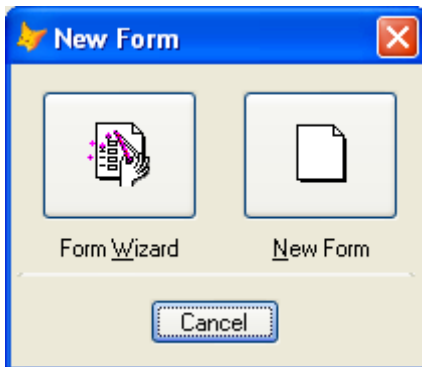
set sysmenu off          ;убрать системное меню
set date to british      &&установить нормальную дату
set exclusive off
set resource off
public lYesNo, lFO, cDir1 &&объявление переменных для
проекта
local sctr3, t1[1]      &&объявление локальных
переменных
iYesNo = .f.
lFO = .f.
on shutdown do p1      &&при завершении выполнить
программу p1

```

```
deactivate window("standard")
set deleted on
do form "Главная"           &&выполнить форму Главная
read event
```

Создание формы

Выбираем закладку Documents и кнопку New. Открывается окно. Выбираем Form Wizard.



Еще раз нажимаем **Form Wizard**. Открывается окно, в котором выбираем таблицу, для которой строим форму.

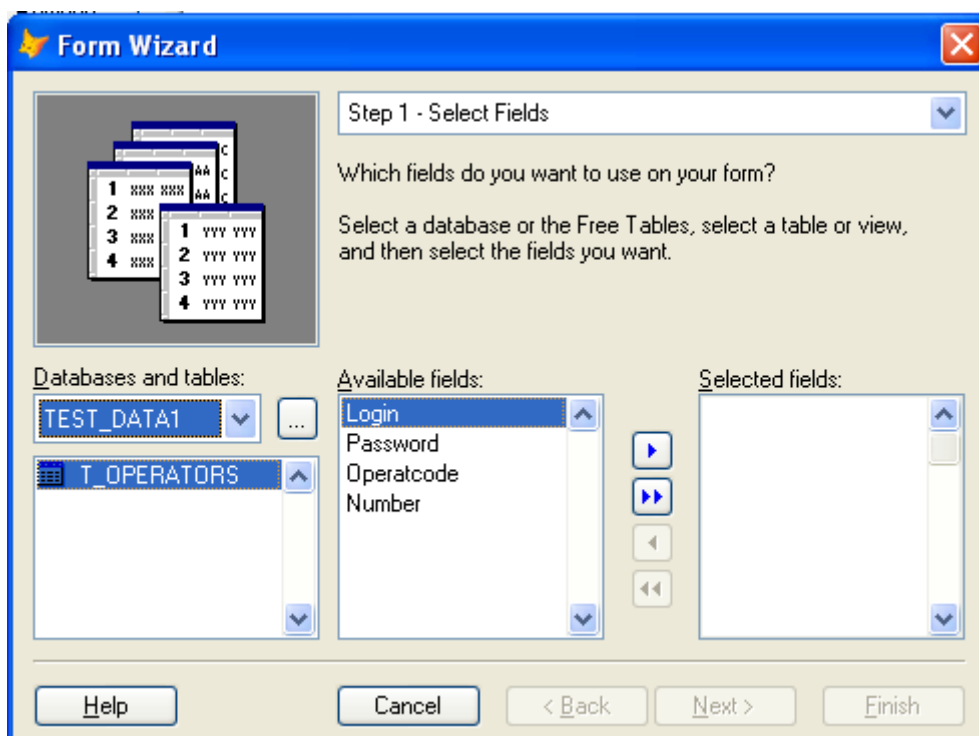


Рис.7. Подготовка формы

Ввод очередного номера записи в таблицу

При вводе очередных данных об операторе не надо вводить его номер вручную. Он должен вводиться автоматически. Для этого, например, можно использовать метод **Got Focus** в свойствах поля **NUMBER1**, или поле с автоматическим инкрементированием.

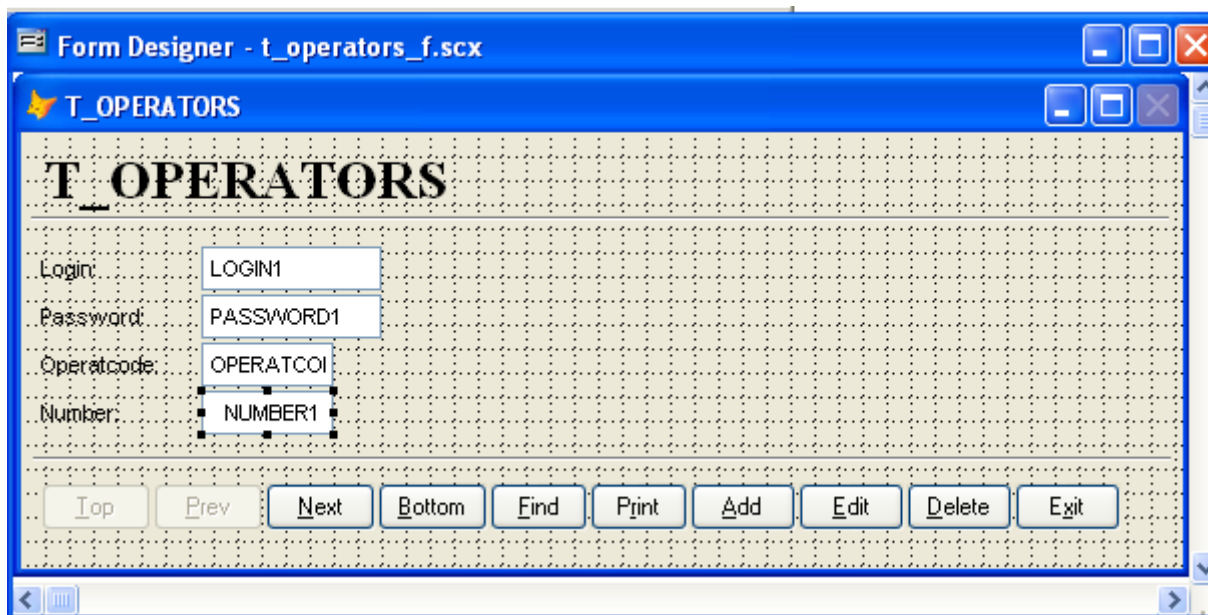


Рис.8. Подготовка процедуры для метода Got Focus

В закладке **Methods** свойств поля **Number1** выбираем **Got Focus** и набираем процедуру.

При попадании курсора (метод **Got Focus**) на поле **number1** выполняется процедура:

```
LOCAL RecNom, Nom[1]
RecNom = RECCOUNT()
thisform.number1.value = RECCOUNT()
MESSAGEBOX(RecNom)
```

Функция **RECCOUNT()** считает количество записей в таблице, которое присваивается полю **number1** в этой таблице (**thisform.number1.value**) **MESSAGEBOX(RecNom)** для отладки выводит это значение на экран.

Можно сделать совсем просто: полю счетчика присвоить тип данных **Integer (AutoInc)**. Тогда, указав начальное значение и шаг, каждая запись получит очередной номер автоматически.

Создание Combo Box на два столбца

Правой кнопкой на выбранном поле открываем **Combo Box Builder**.

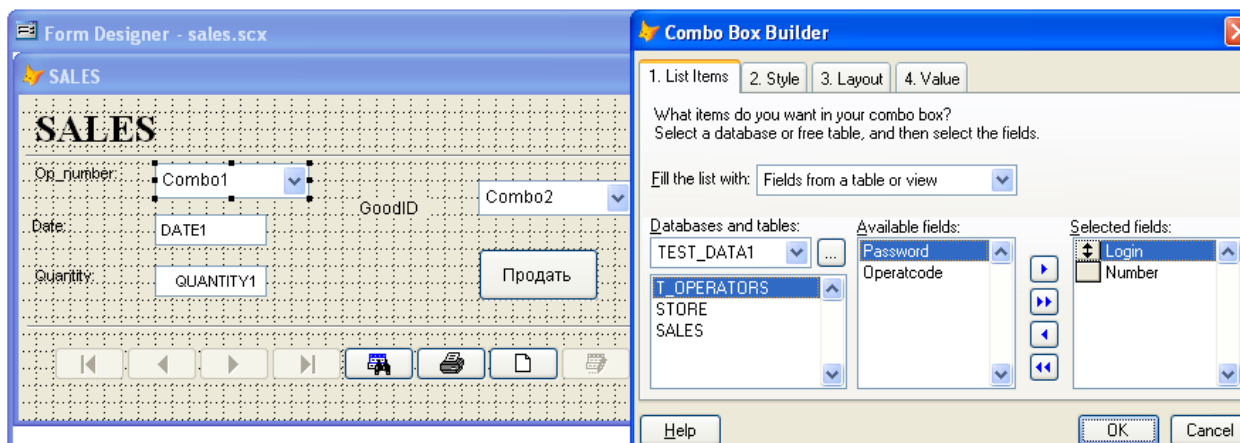


Рис.9. Подготовка ComboBox

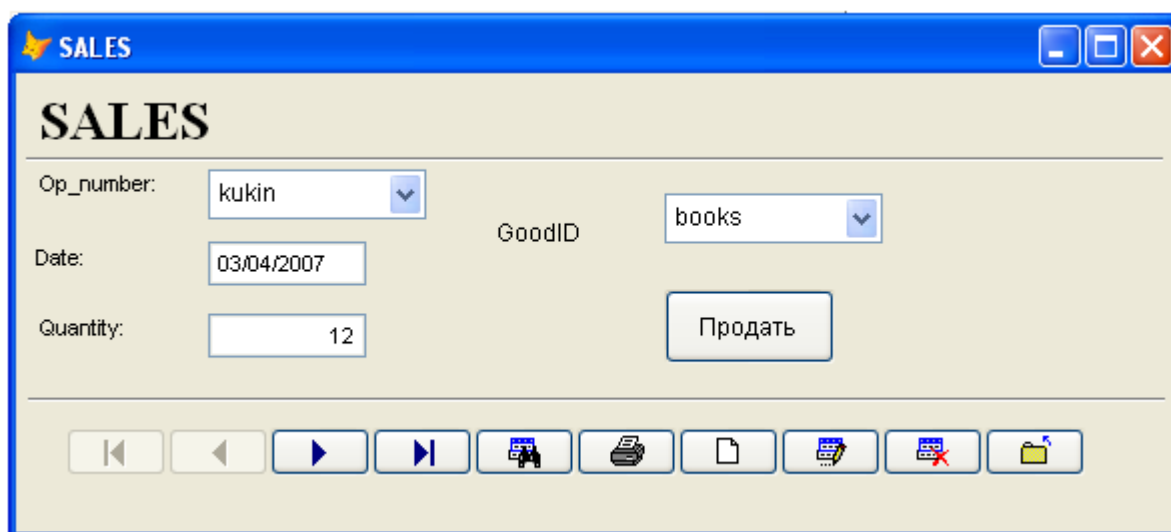
Для поля **Combo1** выбраны и будут отображаться 2 столбца: **Login** и **Number**. В таблицу вносится только номер **Number** (это указывается на закладке **Value**), а для справки выводится и имя. Аналогично сделано и для **Combo2**.

Изменение данных в таблице Store, в случае изменения данных в таблице Sales.

В форме **Sales** (продажи) создали кнопку **Продать**, на которой выполняется программа на методе **Click**:

При этом количество проданных единиц из таблицы **Sales** вычитается из количества имеющихся единиц на складе в таблице **Store**.

Процедура пользователя (метод **Click**) приведена ниже:



```

LOCAL ARRAY am[1]
LOCAL amount1, quant, gdd
GO bottom           переходим к последней записи

```

```

quant = quantity переменной quant присваиваем кол-во
проданного из последней записи
MESSAGEBOX(quant)          выводим для отладки на экран
MESSAGEBOX(goodid)        выводим на экран идентификатор
проданного товара
gdd = goodid              переменной gdd присваиваем
идентификатор
select number from store WHERE goodid = gdd    выбираем
сколько осталось на складе
MESSAGEBOX(number)        для отладки
amount = number - quant   вычитаем проданное из остатка
UPDATE store SET number = amount WHERE goodid = gdd
                          обновляем данные в Store для заданного
идентификатора товара
MESSAGEBOX(amount)        для отладки
CLOSE tables              закрываем таблицы

```

Добавление данных в таблицу

У нас созданы две таблицы Трассы и ППМ. Для каждой трассы мы должны набрать те ППМ, которые составляют данную трассу. Для этого создана таблица, в которой находятся номера трасс и ППМ, относящиеся к данному номеру трассы. Для связывания двух таблиц, например, Трассы и ППМы, создаем форму, в которой будут два поля типа **ComboBox** и кнопка **Добавить**. В первом поле отображается номер и название Трассы, во втором – номер ППМ, который мы хотим включить в данную трассу. Аналогично можно выполнить связывания товаров и продавцов,

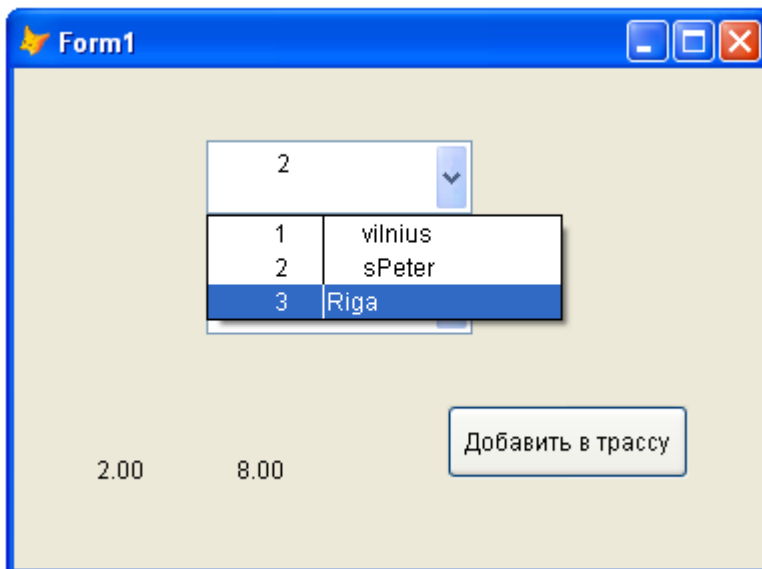


Рис. 10. Выбираем название и номер трассы

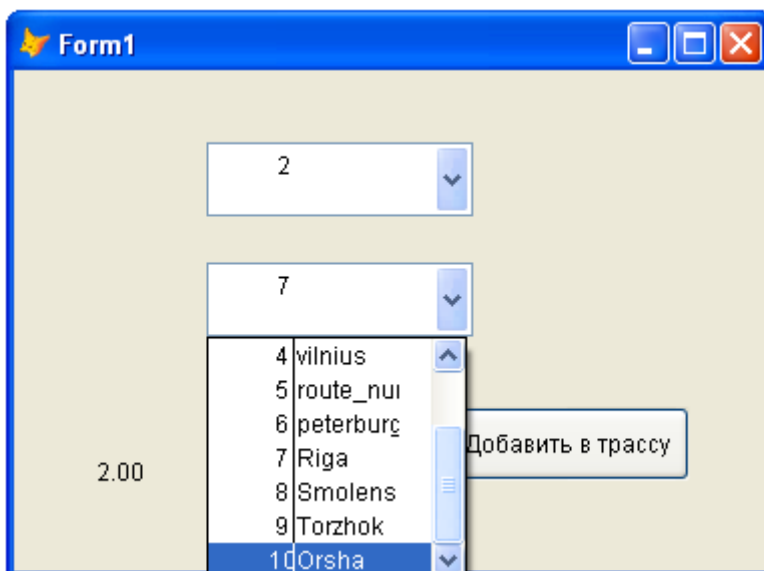


Рис. 11. Выбираем номер ППМ для выбранной трассы

На кнопке Добавить (метод Click – два раза щелкнули на кнопке) набрана программа:

```
@@ переменные nroute и nPpm для запоминания введенных значений
@@ функция VAL преобразует введенный в поле Combo текст в число.
nroute= VAL(thisForm.combo1.Text) @@ преобразуем
введенный текст в число, означающее номер маршрута
nPpm= VAL(thisForm.combo2.Text) @@ преобразуем
введенный текст в число, означающее номер ППМ.
? nroute, nPpm @@ для отладки и проверки выводим эти
значения на экран
@@ и дальше выполняем команду insert, выбрав из таблиц
Ppmdata, route данные и вставляем значения указанных в
операторе полей в таблицу Route_ppm.
INSERT INTO Route_ppm (route_num, Ppmnum,
Ppmname, Longit, Latitud);
SELECT route_num, Ppmnum, Ppmname, Longitude, Latitude
FROM Ppmdata, route;
WHERE ppmnum = nPpm AND route.route_num = nroute
```

Результаты можем посмотреть в созданной с помощью Grid таблицы:

Создание Grid

Часто удобно посмотреть данные из двух связанных таблиц. Это можно сделать используя метод Grid.

Закладка Documents → Forms → New → New form → Form Designer

Появится окно:

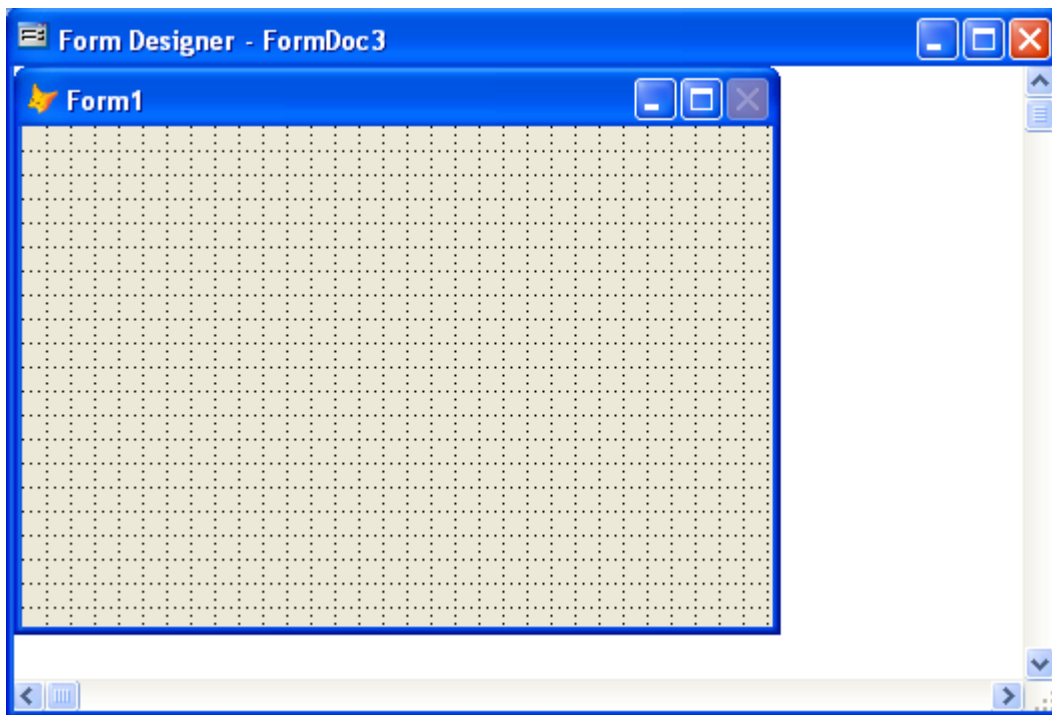
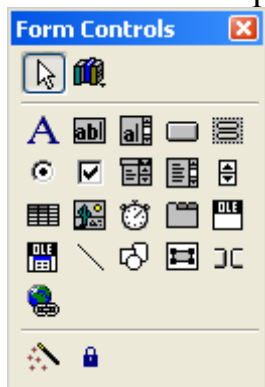


Рис.12. Открыли форму для создания Grid

Из панели инструментов выбираем иконку Grid



И дальше из контекстного меню (правая кнопка мыши) выбираем Builder и дальше формируем сетку для вывода полей таблицы:

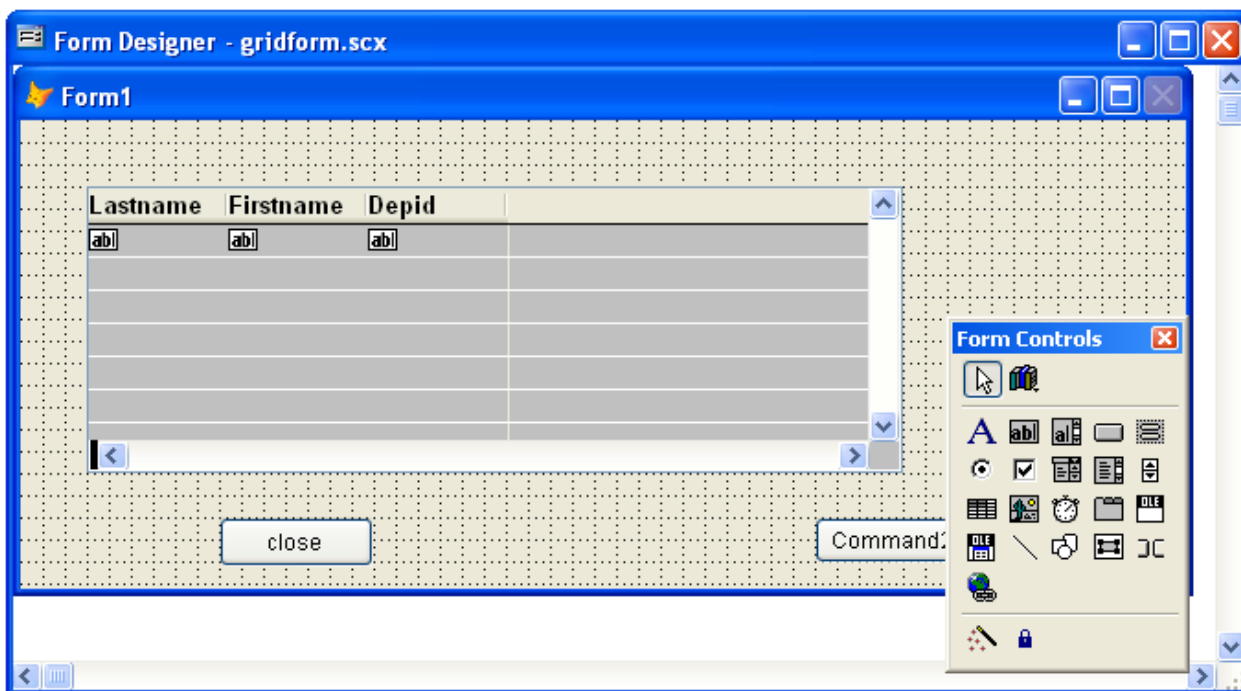
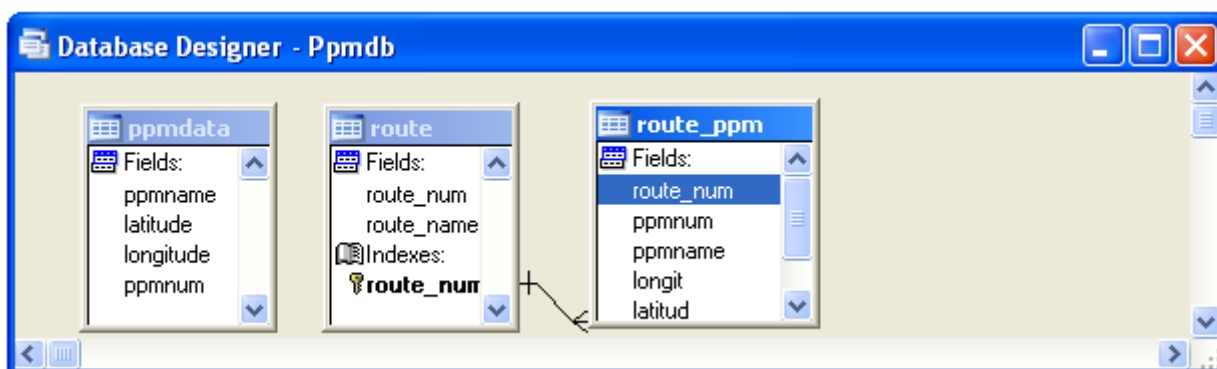


Рис.13.Выбираем поля отображения

Построение формы с Grid и раскрывающимся списком

Для базы данных, содержащей данные о трассах (route)и ППМах (routeppm) мы хотим посмотреть определенную трассу и ППМы, из которых она состоит. Общее поле для этих таблиц route_num.



На форму (Рис.12) вставляем элемент ComboBox.

Открываем всплывающее меню (правая кнопка на ComboBox), выбираем Builder.

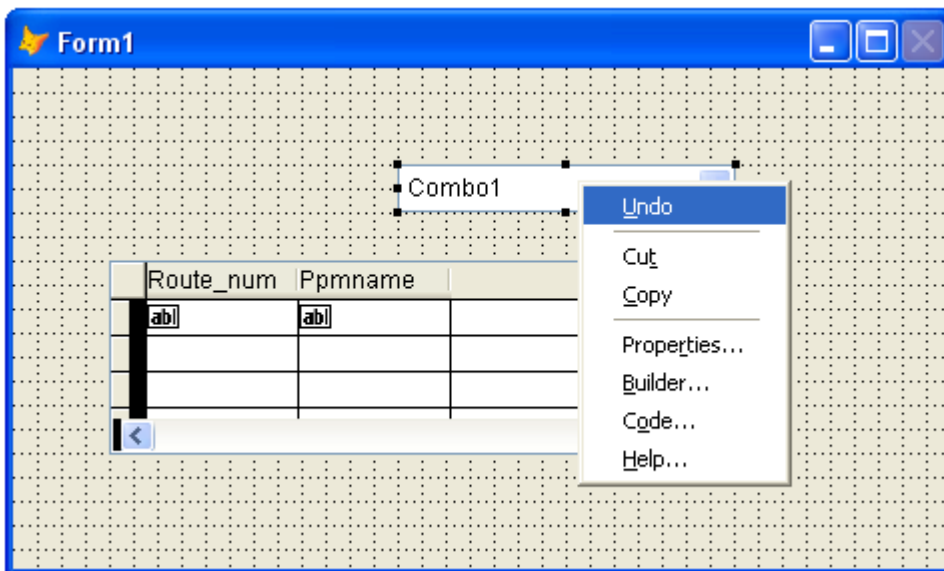


Рис.14.Выбираем Builder

и в List Items Builder указываем, что в раскрывающемся списке ComboBox будет отображаться (route_num).

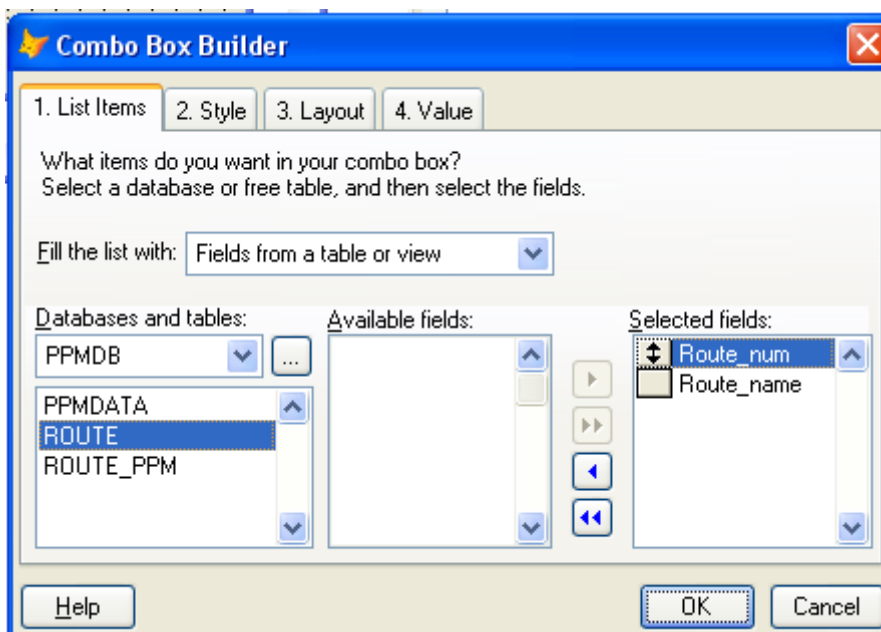


Рис.15. Продолжаем строить Grid

Аналогично Открываем всплывающее меню (правая кнопка на Grid), выбираем Builder

Для окна Grid в Builder выбираем поля для отображения и затем устанавливаем отношение между ComboBox и Grid (Закладка Relationship). Выбираем одинаковые поля для отношения, в нашем случае Route_num.

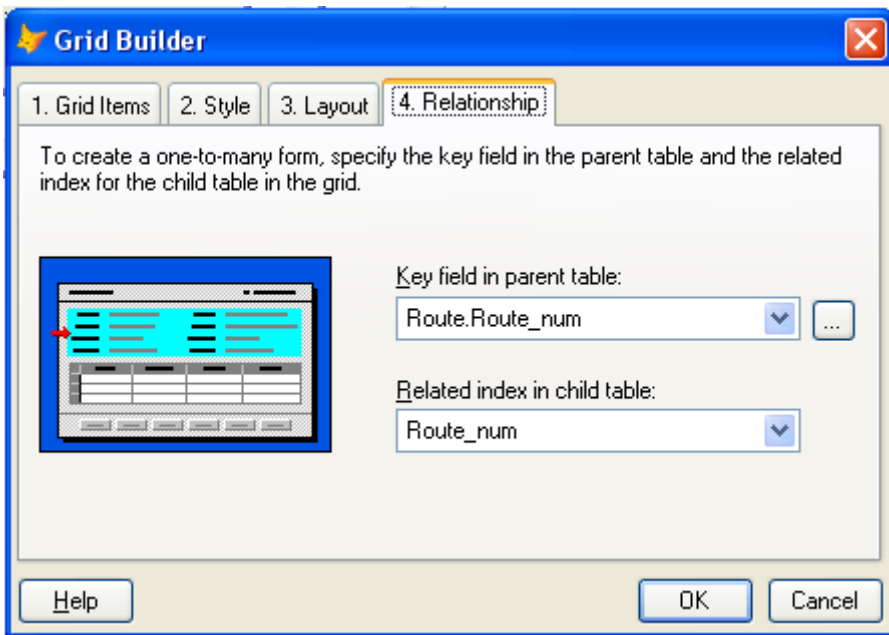


Рис.16. Устанавливаем отношение между таблицами

В результате при выборе в **ComboBox** номера трассы в **Grid** отображаются все ППМы, составляющие эту трассу.

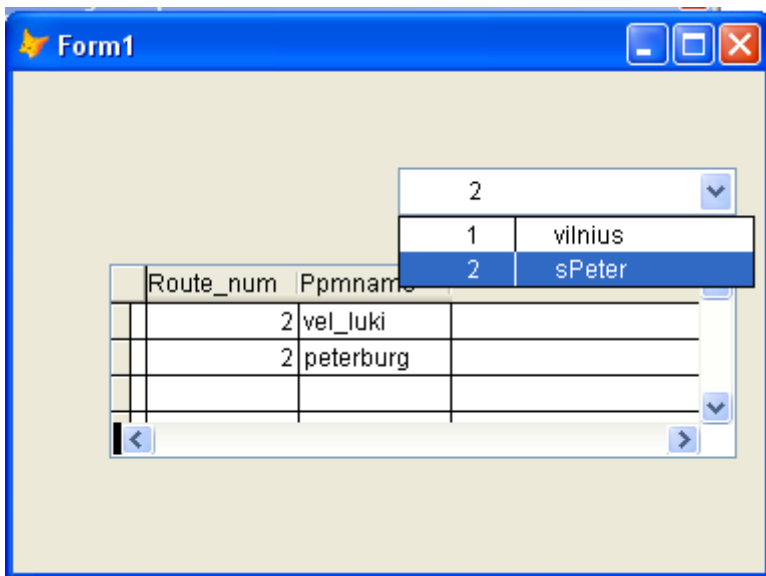


Рис.17. Результат работы Grid

Возможен вариант создания **Grid** с перебором записей трасс и отображением списка ППМ для каждой трассы. В этом случае на форме Трассы создаем **Grid**.

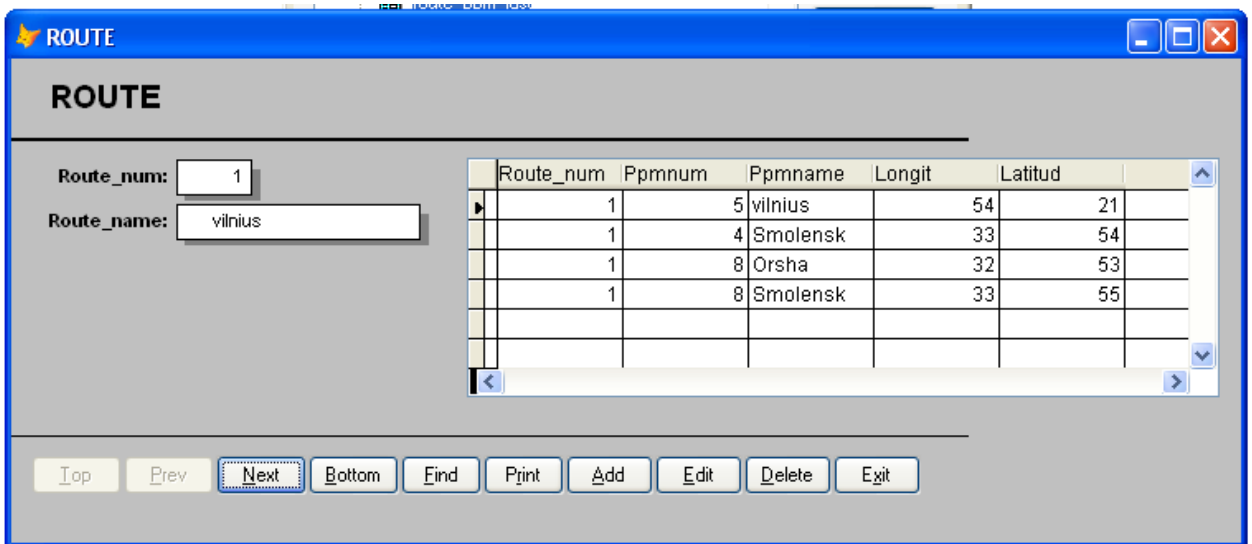
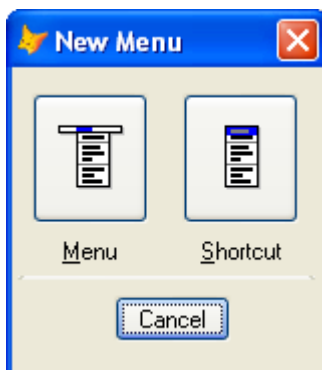


Рис.18. Другой вариант Grid

Создание меню

Для создания меню необходимо в окне Project Manager (Менеджер проектов), в закладке Other (Другие), щелкнуть мышью по папке Menus (Меню), а затем по кнопке New (Новый). Появится окно, показанное ниже:



Если мы создаем главное меню приложения, в окне следует выбрать кнопку Menu (Меню). В результате появится окно Menu Designer (Дизайнера меню), показанное на рис. 19.

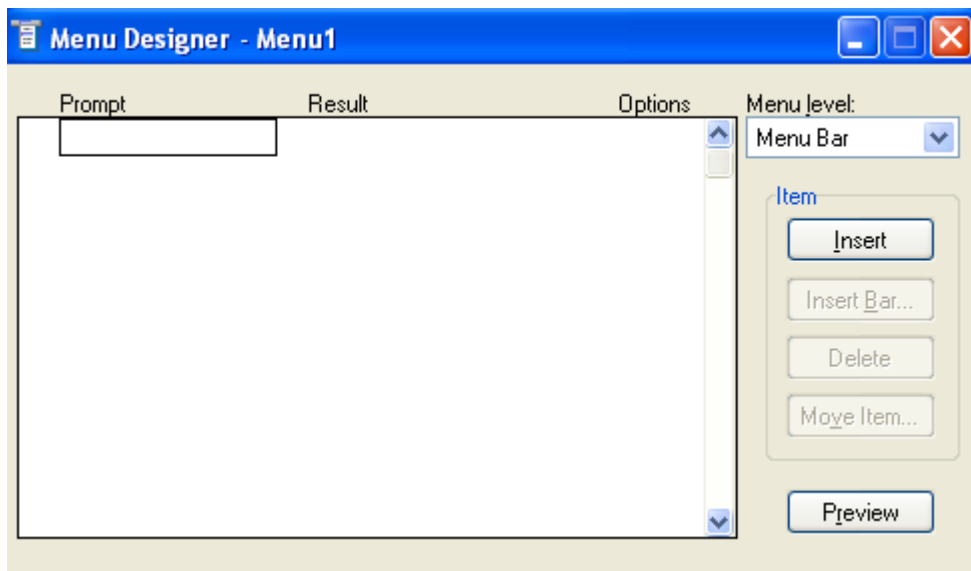


Рис.19.Начало создания меню.

В столбце **Prompt** вводится название пункта меню или команды. Его вы придумываете сами. При этом надо стремиться дать название краткое, но емкое и понятное по смыслу.

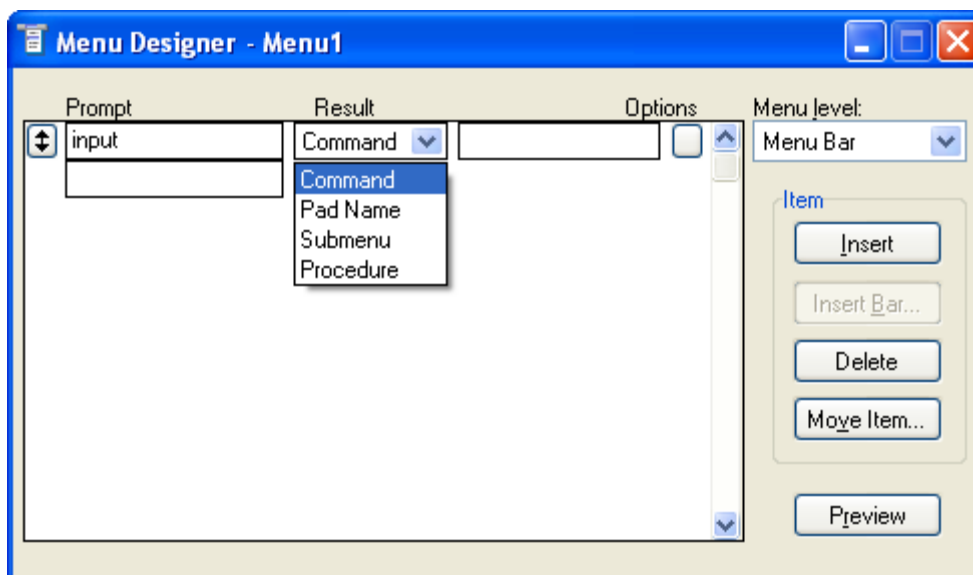


Рис.20. Выбираем тип пункта меню

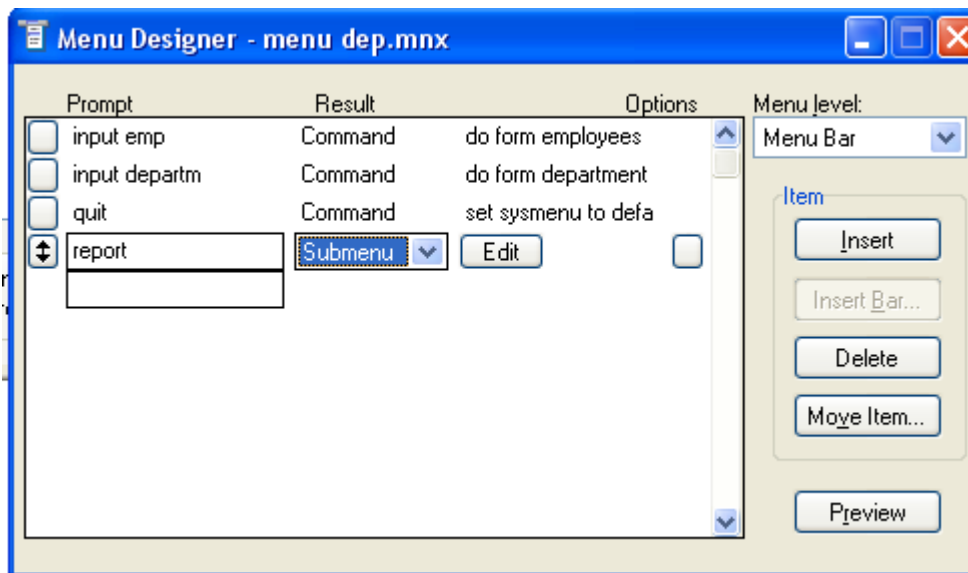


Рис. 21.Создание подменю

В столбце **Result** задается тип действия. Вы можете организовать подменю (**Submenu**), задать выполняемую при выборе пункта меню процедуру (**Procedure**) или определить команду, выполняемую при выборе пункта меню (**Command**).

Если вы выбрали тип действия **Submenu** (**Подменю**), появится кнопка **Edit** (**Редактировать**). При щелчке мышью по этой кнопке в окне **Menu Designer** (**Дизайнера меню**) будет показано меню, которое появится при выборе вашего пункта главного меню (более низкий уровень меню). Пункт главного меню, при выборе которого появится редактируемое вами в данный момент подменю, показано в раскрывающемся списке в правом верхнем углу окна **Menu Designer** (**Дизайнера меню**). При помощи этого раскрывающегося списка можно быстро переходить от редактирования одного уровня меню к редактированию другого. Таким образом, можно создавать меню любой вложенности и любого уровня. Конечно, лучше не увлекаться большим количеством уровней меню. Иначе вы запутаете пользователя, и он остановится в поисках НУЖНОЙ команды меню.

Если **Command**, то

Например:

```
D0 FORM myname
```

Или программу:

```
Do program_dowhile && где program_dowhile имя  
запускаемой программы
```

Или если отчет, то

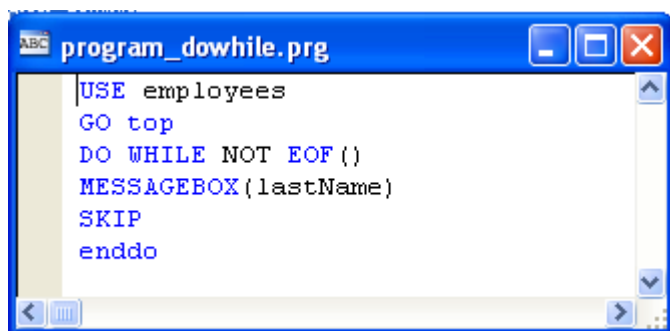
```
REPORT FORM myReport
```

Создание программы:

Для написания программы открываем закладку CODE → Programs → New .

Откроется окно программы, в него вводится текст программы, например:

Просмотр записей таблицы Employees:



```
USE employees
GO top
DO WHILE NOT EOF()
MESSAGEBOX(lastName)
SKIP
enddo
```

Рис.22. Пример программы

Программа в цикле просматривает записи. Цикл выполняется до конца файла (EOF() – функция определения конца файла). Skip – переход на следующую запись

Описание MessageBox()

MESSAGEBOX(*eMessageText* [, *nDialogBoxType*] [, *cTitleBarText*][, *nTimeout*])

Parameters

eMessageText

Содержит текст, который появляется в окне диалога

nDialogBoxType

Описывает кнопки и иконки, которые появляются в окне, кнопку по умолчанию. В следующих таблицах значения от 0 до 5 определяют какие кнопки появятся в окне. Значения иконки 16, 32, 48, and 64 определяют иконку, которая появится в окне. Значения 0, 256, and 512 определяют, какая кнопка будет по умолчанию. Опущенная ***nDialogBoxType*** означает 0.

Значение	Кнопки Dialog box
0	Только кнопка OK
1	Кнопки OK и Cancel
2	Кнопки Abort, Retry, и Ignore
3	Кнопки Yes, No, и Cancel
4	Кнопки Yes и No

5	Кнопки Retry и Cancel
Значение	Иконка
16	Знак Stop
32	Вопросит. знак
48	Восклицательный
64	Информация (i)
Значение	Кнопка по умолчанию
0	Первая
256	Вторая
512	Третья

nDialogBoxType может быть суммой трех значений, каждое из предыдущих таблиц. Например если *nDialogBoxType* is 290 (2+32+256), определяемое окно имеет значения

- Кнопки Abort, Retry, and Ignore.
- Отображается знак вопроса.
- Вторая кнопка Retry, кнопка по умолчанию.

Возвращаемое значение

MESSAGEBOX() возвращает значение, которое определяет какая кнопка была выбрана. Следующая таблица определяет возвращаемые значения **MESSAGEBOX()**.

Return value Button

1	OK
2	Cancel
3	Abort
4	Retry
5	Ignore
6	Yes
7	No

При наличии кнопки **Cancel**, нажатие на ESC возвращает то же значение 2, как и нажатие **Cancel**

MESSAGEBOX() возвращает значение -1 когда истекает время.

The **MESSAGEBOX()** тип параметра определяет какие параметры используются. Первый параметр *eMessageText* должен быть всегда. Однако необязательный второй параметр может быть *nDialogBoxType*, если его тип числовой **Numeric** или *cTitleBarText* если тип символьный **Character**.

Параметр *nTimeout* is always assumed for the second optional numeric parameter passed.

Примеры:


```
MESSAGEBOX ("HELLO", "MyTitle", 68, 6000)
MESSAGEBOX ("HELLO", 68, "MyTitle", 6000)
MESSAGEBOX ("HELLO", 68, 6000)
MESSAGEBOX ("HELLO", 68, 6000, "MyTitle")
```

Пример

Пример показывает диалоговое окно, определенное пользователем. Сообщение "Record not found. Would you like to search again?" отображается в окне, а "My Application" отображается в заголовке.

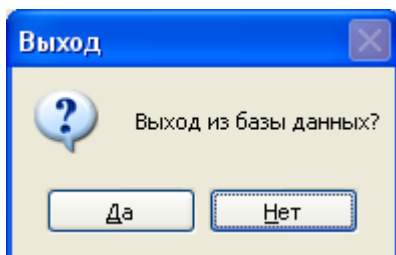
Окно содержит кнопки **Yes** и **No** и вопросительный знак, кнопкой по умолчанию является вторая кнопка (**No**). Когда вы нажимаете одну из кнопок, ваш выбор отображается.

```
eMessageTitle = 'My Application'
eMessageText = 'Record not found. Would you like to
search again?'
nDialogType = 4 + 16 + 256
* 4 = Yes and No buttons
* 16 = Stop sign icon
* 256 = Second button is default
```

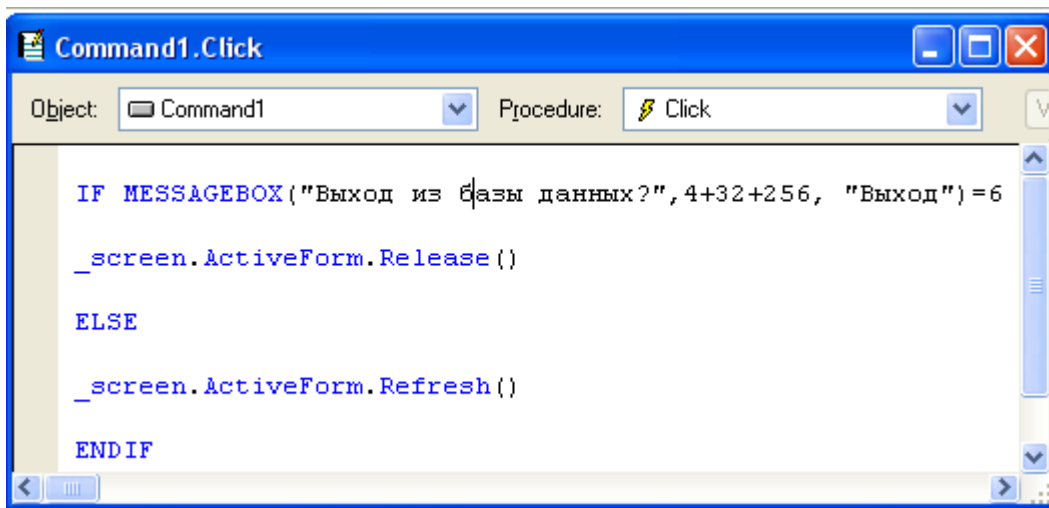
```
nAnswer = MESSAGEBOX(eMessageText, nDialogType,
eMessageTitle)
```

```
DO CASE
CASE nAnswer = 6
WAIT WINDOW 'You chose Yes'
CASE nAnswer = 7
WAIT WINDOW 'You chose No'
ENDCASE
```

Для выхода из проекта `MessageBox()` с двумя кнопками



Описание кнопки:



```
Object: Command1 Procedure: Click

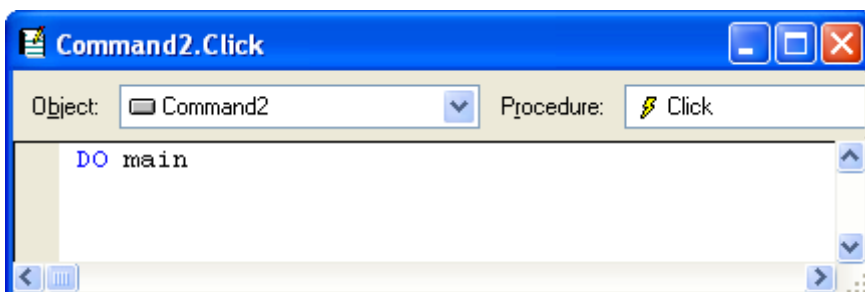
IF MESSAGEBOX("Выход из базы данных?", 4+32+256, "Выход")=6
    _screen.ActiveForm.Release()
ELSE
    _screen.ActiveForm.Refresh()
ENDIF
```

Запуск приложения

Чтобы определить какая программа является главной **Main** (с нее начинается работа системы), на закладке **Code** (программы) выберите программу и на ней нажмите правой кнопкой. В раскрывшемся меню выберите **Set main**.

Для запуска приложения можно на форме создать кнопку (Button) из панели инструментов (Form Controls)

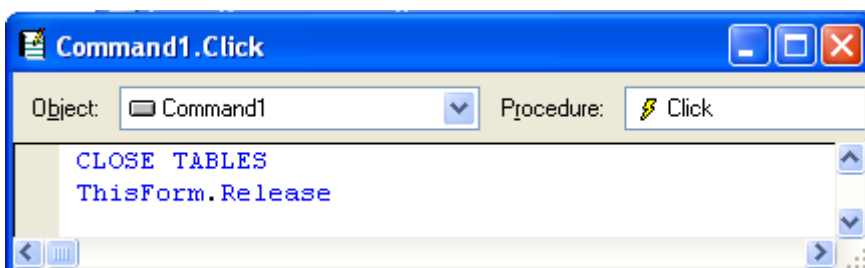
Запуск основной программы на кнопке Command (на методе Click), набрав DO main



```
Object: Command2 Procedure: Click

DO main
```

Закрытие формы и программы на кнопке Close



```
Object: Command1 Procedure: Click

CLOSE TABLES
ThisForm.Release
```

Пояснительная записка должна содержать:

- Задание на курсовую работу
- ER – диаграмму предметной области с функциональными зависимостями
- Структуру базы данных и оценку ее эффективности
- Структуру программных средств и меню
- Распечатку программ
- Результаты работы системы: примеры выполнения запросов, распечатки справок.

Объем записки 15 -20 страниц

Литература

1. А. Лебедев, Visual FoxPro 9.0. -М. «NT Press», 2005
2. В. Пирогов, SQL Server 2005 Программирование клиент – серверных приложений. – СПб. «БХВ_Петербург», 2006.