

Министерство образования и науки Российской Федерации  
**РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
НЕФТИ И ГАЗА ИМ. И.М. ГУБКИНА**

**Кафедра АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ**

**Г.Н. Малиновская**

**«Разработка приложений баз данных  
в среде MS Access 2007»**

Учебно-методическое пособие по курсу «Базы и банки данных»

**Москва 2010г.**

Рецензент – к.т.н., доцент В.В. Сидоров

**Малиновская Г.Н.** Учебно-методическое пособие «Разработка приложений баз данных в среде MS Access 2007» по курсу «Базы и банки данных», под редакцией Л.В. Кузнецовой М.: РГУ нефти и газа имени И.М. Губкина, 2010. - 38 с.

В учебно-методическом пособии рассмотрены основные приемы разработки информационных приложений с использованием баз данных.

Учебно-методическое пособие предназначено для:

- магистрантов специальностей: 220200 – Автоматизация и управление, 140600 – Электротехника, электромеханика, электротехнологии, 200100 – Приборостроение;
- магистрантов технологических специальностей университета, изучающих курс «Информационные технологии управления и интеллектуальный анализ данных» раздел Базы данных.

## **Введение**

Комплекс лабораторных работ состоит из 9 тем и отражает основные приемы разработки информационных приложений с использованием баз данных в среде Microsoft Access 2000. Комплекс основывается на пособии «Комплекс лабораторных работ в среде Delphi 3 по курсу «Базы Данных», автор Кузнецова Л.В., М.:РГУ нефти и газа, 1999г.

Учебное пособие содержит подробное описание 9 лабораторных работ.

Кроме данного методического пособия, целесообразно использовать литературу, приведенную в списке литературы или аналогичную.

### **Описание структуры базы данных**

В комплексе лабораторных работ предметной областью является торговая компания, которая работает с группой фирм (клиентов), выполняя их заказы на поставку товаров.

Клиенты имеют следующие характеристики (атрибуты):

- название фирмы;
- город;
- дата регистрации фирмы;
- почтовый адрес;
- пометку, если клиент постоянно работает с магазином.

Товары, которыми торгует магазин, имеют характеристики (атрибуты):

- название товара;
- цена;
- описание товара;
- фотография товара.

При оформлении заказа на покупку требуется зафиксировать:

- номер заказа;
- название клиента;
- название товара;
- количество товара;
- дату заказа;
- по какой цене товар продан.

База данных учебного примера состоит из четырех таблиц: KLIENT, TOVAR, ZAKAZ, GOROD. Справочник городов (GOROD) составлен для ускорения и устранения ошибок при вводе названий городов. Более подробная информация по структуре таблиц приведена в описании лабораторной работы №1.

## Лабораторная работа №1

### **ЦЕЛЬ:**

Создать базу данных Учебного примера в среде Access, содержащую 4 таблицы с данными.

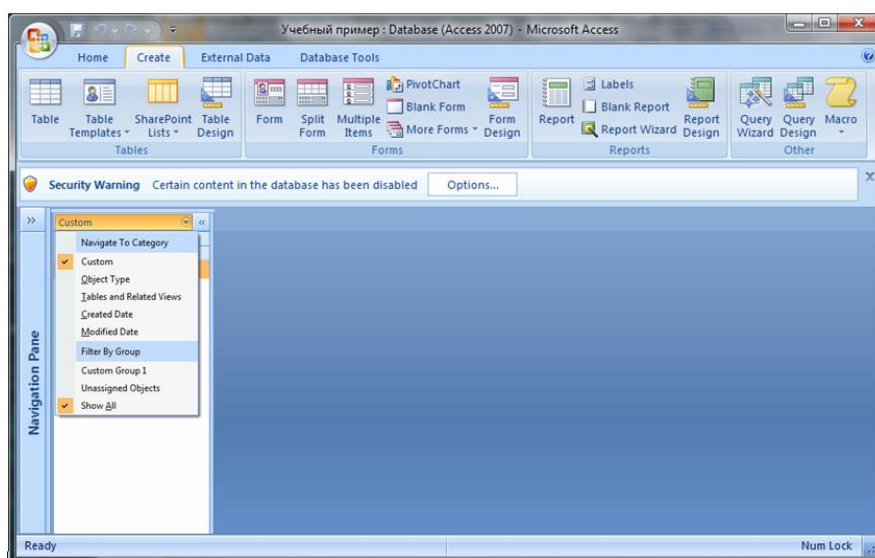
### **Порядок выполнения работы**

#### **Шаг 1**

Запустить Access. При первом запуске Access выводит на экран окно **Приступая к работе с Microsoft Office Access** (Getting Started with Microsoft Office Access). С его помощью можно выполнить:

- открытие одной из используемых ранее баз данных;
- создание базы данных путем выбора одного из готовых шаблонов представленных на вкладке **Шаблоны из Интернета** (Featured Online Templatest);
- создание новой базы данных кнопка **Новая база данных** (Blank Database) на вкладке **Новая пустая база данных** (New Blank Database).

Создаем и сохраняем на диске файл базы данных с именем **Учебный пример**.

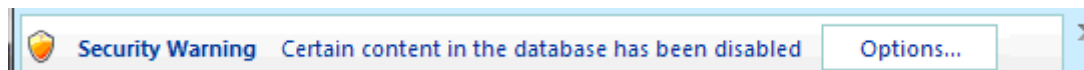


Все объекты базы данных сосредоточены в раскрывающейся/схлопывающейся области переходов, так называемой **навигационной панели** (Navigation Pane), обеспечивающей удобные средства их отображения по **категориям** (Navigate to Category) и **группам** (Filter by Group).

Все команды для работы с базой данных структурированы и представлены в контекстно-зависимых меню на вкладках ленты: - **Главная** (Home), **Создание** (Create), **Внешние данные** (External Data), **Работа с базами данных** (Database Tools).

### **ВСПОМОГАТЕЛЬНАЯ ИНФОРМАЦИЯ**

Вывод всех предупреждающих сообщений системы безопасности производится на панели сообщений. По умолчанию уже при открытии базы данных (если нет других настроек) отключаются все потенциально опасные, активные компоненты, такие как запросы на изменение, макросы и программы на VBA.



Чтобы включить отключенное содержимое щелкните **Параметры** (Options), а затем в появившемся диалоговом окне **Включить содержимое** (Enable this content).

## Шаг 2

Создать таблицы *TbKlient*, *TbTovar*, *TbZakaz*, *TbGorod* с помощью конструктора таблиц, для этого нажимаем кнопку **Конструктор таблиц** (Table Design) вкладки **Создание** (Create).

Структура таблицы **TbKlient**

Имя поля / Field Name	Тип данных / Data Type	Размер поля / Field Size	Индексированное поле / Indexed	Назначение поля / Description
Kod_kl	<b>Счетчик</b> / AutoNumber	<b>Длинное целое</b> Long Integer	<b>Да (совпадения не допускаются)</b> Yes (No Duplicates)	Первичный ключ формируется автоматически, используется для организации связи между клиентами и их заказами.
Klient	<b>Текстовый</b> / Text	25	<b>Да (совпадения не допускаются)</b> Yes (No Duplicates)	Название фирмы
Kod_gor	<b>Числовой</b> / Number	<b>Длинное целое</b> Long Integer	<b>Да (допускаются совпадения)</b> Yes (Duplicates OK)	Используется для организации связи между клиентами и справочником городов.
Data_reg	<b>Дата/время</b> / Data/Time	<b>Краткий формат даты</b> / Short Date	<b>Да (допускаются совпадения)</b> Yes (Duplicates OK)	Дата регистрации фирмы (дд.мм.гггг)
Post	<b>Логический</b> / Yes/No	<b>Да/Нет</b> / Yes/No		Признак постоянного клиента.
Adres	<b>Поле МЕМО</b> / Memo			Почтовый адрес выбран тип МЕМО, т.к. длина поля заранее не известна

Структура таблицы **TbTovar**

Имя поля / Field Name	Тип данных / Data Type	Размер поля / Field Size	Индексированное поле / Indexed	Назначение поля / Description
Kod_tov	<b>Счетчик</b> / AutoNumber	<b>Длинное целое</b> Long Integer	<b>Да (совпадения не допускаются)</b> Yes (No Duplicates)	Первичный ключ формируется автоматически, используется для организации связи между товарами и заказами.
Tovar	<b>Текстовый</b> / Text	25	<b>Да (совпадения не допускаются)</b> Yes (No Duplicates)	Название товара
Cena	<b>Денежный</b> / Currency	<b>Основной</b> / Currency		Цена товара.
Opis	<b>Поле МЕМО</b> / Memo			Подробное описание товара. Выбран тип МЕМО, т.к.. длина поля заранее не известна
Foto	<b>Поле объекта OLE</b> / OLE Object			Фотография товара.

Структура таблицы **TbZakaz**

Имя поля / Field Name	Тип данных / Data Type	Размер поля / Field Size	Индексированное поле / Indexed	Назначение поля / Description
Kod_zak	Счетчик / AutoNumber	Длинное целое Long Integer	Да (совпадения не допускаются) Yes (No Duplicates)	Первичный ключ формируется автоматически, используется для организации связи между клиентами и их заказами.
N_zak	Числовой / Number	Длинное целое Long Integer		Номер заказа
Kod_kl	Числовой / Number	Длинное целое Long Integer	Да (допускаются совпадения) Yes (Duplicates OK)	Код клиента
Kod_tov	Числовой / Number	Длинное целое Long Integer	Да (допускаются совпадения) Yes (Duplicates OK)	Код товара
Kol_vo	Числовой	Целое		Количество товара в заказе
Cena_prod	Денежный / Currency	Основной / Currency		Цена товара.
Data_zak	Дата/время / Data/Time	Краткий формат даты / Short Date		Дата оформления заказа (дд.мм.гггг)

Структура таблицы **TbGorod**

Имя поля / Field Name	Тип данных / Data Type	Размер поля / Field Size	Индексированное поле / Indexed	Назначение поля / Description
Kod_gor	Счетчик / AutoNumber	Длинное целое Long Integer	Да (совпадения не допускаются) Yes (No Duplicates)	Первичный ключ формируется автоматически, используется для организации связи между таблицами и списком городов
Gorod	Текстовый / Text	20	Да (совпадения не допускаются) Yes (No Duplicates)	Название города

Список свойств полей меняется в зависимости от типа данных. Необходимо ознакомиться со списком свойств каждого поля, и указать по необходимости:

- Размер поля (Field Size)
  - Формат поля (Format)
  - Маска ввода (Input Mask)
  - Подпись (Caption)
  - Значение по умолчанию (Default Value)
  - Условие на значение (Validation Rule)
  - Сообщение об ошибке (Validation Text)
  - Обязательно поле (Required)
  - Пустые строки (Allow Zero Length)
  - Индексированное поле (Indexed)
- и так далее.

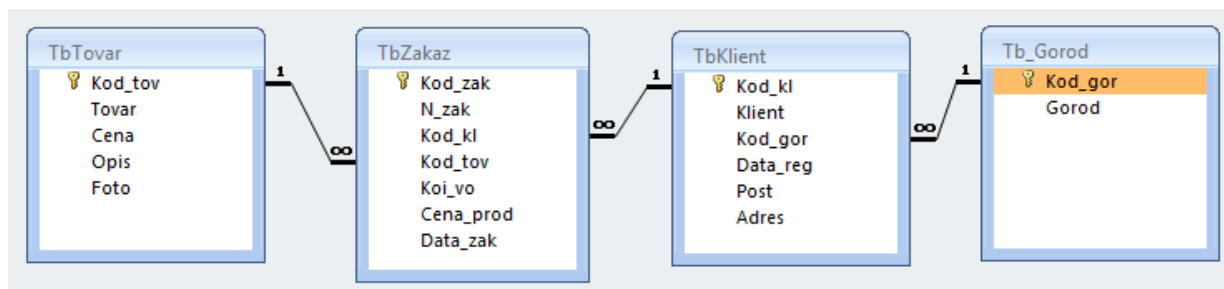
### Шаг 3

Необходимо определить связи между таблицами базы данных. После этого можно создавать запросы, формы и отчеты для одновременного отображения сведений из нескольких таблиц.

*Важнейшей задачей, которую позволяет решать схема данных, является обеспечение логической целостности данных в базе.*

- Закройте все открытые таблицы. Создавать или изменять связи между открытыми таблицами нельзя.
- Нажмите кнопку **Схема данных** (Relationships) на вкладке **Работа с базами данных** (Database Tools) ленты.
- Добавьте необходимые таблицы для создания связей. Для этого:
  - Нажмите кнопку **Отобразить таблицы** (Show Table) на вкладке ленты **Конструктор** (Design) в разделе **Схема данных** (Relationships) или же выберите **Отобразить таблицы** (Show Table) в контекстном меню, вызываемом нажатием правой клавиши “мыши” в области окна **Схема данных** (Relationships).
  - В раскрывшемся окне **Добавление таблицы** (Show Table) отображены все таблицы и запросы, содержащиеся в базе данных, размещенные на вкладках: **Таблицы** (Tables), **Запросы** (Queries) и **Таблицы и запросы** (Both).
  - Выбираем вкладку **Таблицы** (Tables), затем имя таблицы и нажимаем кнопку **Добавить** (Add).
  - В результате в окне **Схема данных** (Relationships) таблицы базы будут представлены окнами со списками своих полей и выделенными жирным шрифтом и ключами.
- Чтобы установить связи между полями таблиц, «мышкой» выберите поле в одной таблице и перетащите его на соответствующее поле во второй таблице.
- Будет открыто диалоговое окно **Изменение связей** (Edit Relationships), в котором необходимо проверить правильность имен полей связи и установив соответствующие флажки, обеспечить включение режимов:
  - **Обеспечение целостности данных** (Enforce Referential Integrity),
  - **Каскадное удаление связанных записей** (Cascade Delete Related Records),
  - **Каскадное обновление связанных записей** (Cascade Update Related Fields).
- Для создания связи нажмите кнопку **Создать** (Create).
- Для изменения или удаления связи необходимо правой клавишей мыши выделить линию связи и в контекстном меню выбрать необходимое действие.
- При закрытии окна **Схема данных** (Relationships) Microsoft Access выводит приглашение сохранить макет. *Независимо от выбранного ответа созданные связи будут сохранены в базе данных.*

Необходимо создать следующую схему данных:



#### Шаг 4

Необходимо осуществить проверку **автоматического поддержания целостности данных в базе данных** при изменении значений ключей связи в таблицах.

1. Убедимся, что невозможно включить новую запись в подчиненную таблицу *Tbklient* со значением ключа связи *Kod\_gor*, не представленным в главной таблице связи *TbGorod*.

2. Убедимся, что при изменении значения ключевого поля *Kod\_gor* в одной из записей в главной таблице связи *TbGorod*, повлечет за собой мгновенное автоматическое изменение значения поля *Kod\_gor* во всех записях в подчиненной таблице *Tbklient*, для клиентов этого города. Изменение происходит т.к. был установлен флажок **каскадное обновление связанных записей** (Cascade Update Related Fields).

3. Убедимся, что вместе с удалением записи в главной таблице связи *TbGorod* удаляются все подчиненные записи в таблицах *Tbklient* и *TbZakaz*, т.к. был установлен флажок **каскадное удаление связанных записей** (Cascade Delete Related Records).

*Заметим, если каскадное удаление не разрешено, то невозможно удалить запись в главной таблице связи, если имеются связанные с ней записи в подчиненной таблице.*



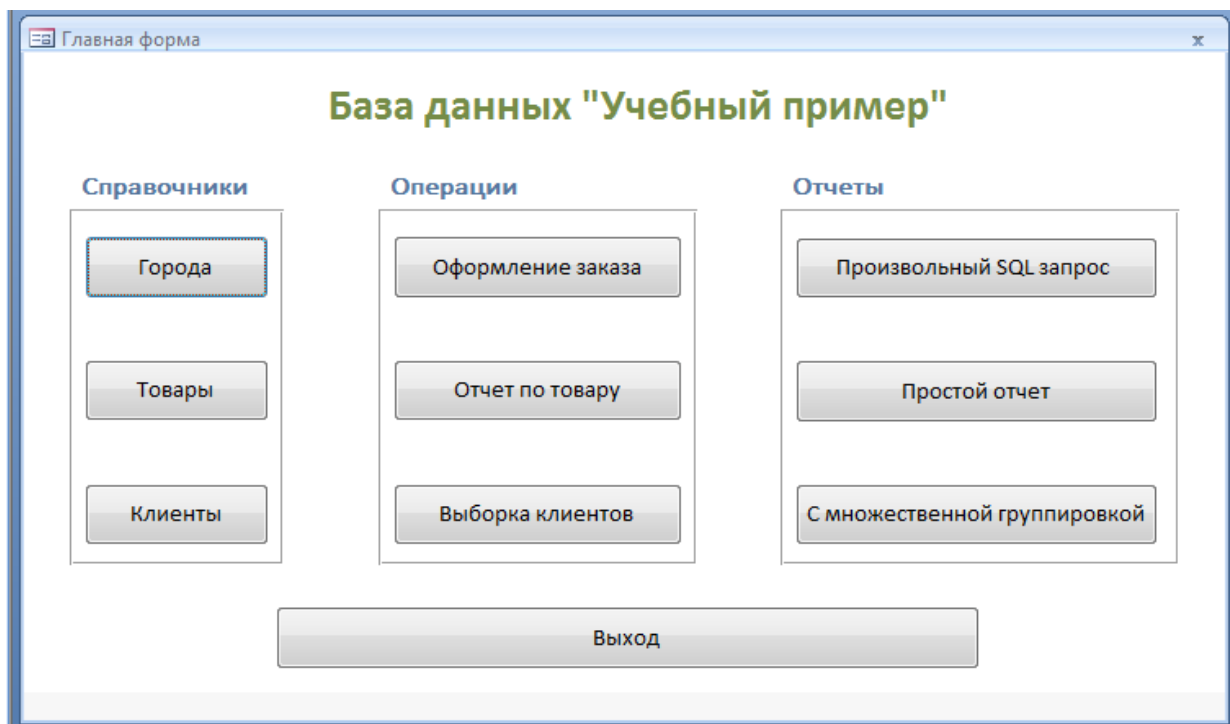
## Лабораторная работа №2

### ЦЕЛЬ:

Разработка главной кнопочной формы, на которой расположены кнопки вызова других форм.

Разработка формы *FrGorod*, позволяющей вводить и удалять названия городов в алфавитном порядке.

Разработка формы *FrTovar*, которая предназначена для ведения базы данных по товарам.



### Порядок выполнения работы

#### Шаг 1

На вкладке ленты **Создание** (Create) щелкаем кнопку **Конструктор форм** (Form Design). В режиме конструктора можно разрабатывать формы, модифицировать их структуру, изменять внешний вид, добавлять и удалять элементы управления. При работе с формой и ее элементами используются различные параметры, значения которых задаются с помощью диалогового окна **Свойства** (Field Properties ).

Далее устанавливаем следующие свойства формы:

- Подпись (Caption) – “Главная форма”
- Режим по умолчанию (Default View) – «Одиночная форма» (Single Form)
- Режим формы (Allow Form View) – “Да” (остальные режимы - нет)
- Режим таблицы (Allow Datasheet View) - “Нет”
- Режим сводной таблицы (Allow PivotTable View) - “Нет”
- Режим сводной диаграммы (Allow PivotChart View) - “Нет”
- Режим макета (Allow Layout View) - “Нет”
- Область выделения (Record Selectors) – “Нет”
- Кнопки перехода (Navigation Buttons) - “Нет”
- Разделительные линии (Dividing Lines)- “Нет”
- Кнопки размеров окна (Min Max Buttons) – “Отсутствуют”
- Полосы прокрутки (Scroll Bars) – “Отсутствуют”
- Тип границы (Border Style) - “Окна диалога” (Dialog)

## Шаг 2

Размещая на форме элементы: **Надпись** (Text Box), **Кнопки** (Button), расположенные на вкладке ленты **Конструктор** (Design), раздела **Элементы управления** (Controls), придаем форме законченный вид.

Кнопку «**Выход**» создаем с помощью мастера указав **Работа с формой** (Form Operation) в разделе **Категории** (Categories) и **Заккрыть форму** (Close Form) в разделе **Действия** (Actions). При создании остальных кнопок отменяем работу мастера.

Мастера создания элементов управления (кнопка, переключатель, список и т.д.) будут запускаться автоматически в том случае, если на вкладке **Конструктор** (Design), в разделе **Элементы управления** (Controls) будет активна кнопка **Использовать мастера** (Use Control Wizards).

Переключаясь в **Режим макета** (Layout View) настраиваем внешний вид формы, меняем расположение, размер и оформление объектов формы в соответствии с потребностями пользователя, наблюдая изменения этих объектов в реальном времени. Закончив оформление внешнего вида формы, отключаем возможность ее просмотра в этом режиме. Задаем форме свойство **Режиме макета** (Allow Layout View) – “*Нет*”.

### ВСПОМОГАТЕЛЬНАЯ ИНФОРМАЦИЯ

Access 2007 работает с моделью однодокументного интерфейса. Эта модель размещает все объекты базы данных (формы, отчеты, и т.д.) в одном окне и образует для каждого объекта свою вкладку.

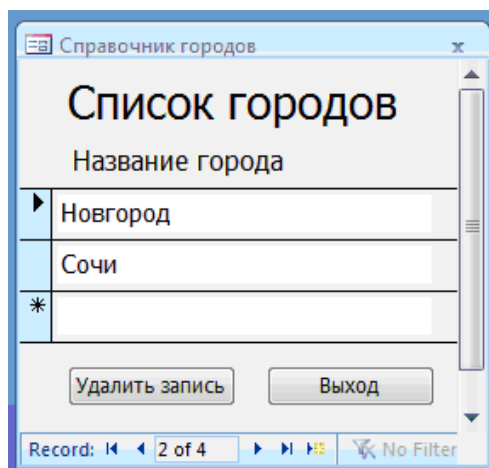
Для того чтобы открывать каждый объект в отдельном окне, необходимо в **окне параметров Access** (Access Options), на вкладке **Текущая база данных** (Current Database), в разделе **Параметры приложений** (Application Options), в группе **Параметры окна документов** (Document Window Options) установить флажок **Перекрывание окон** (Overlapping Windows), вместо флажка **Вкладки** (Tabbed Documents).

## Шаг 3

На вкладке ленты **Создание** (Create) в списке команд **Другие формы** (More Forms) выбираем команду **Мастер форм** (Form Wizard).

Создаем форму *FrGorod* с помощью мастера форм, следуя его указаниям:

- выбираем поля, которые разместим на форме (Мастер форм дает возможность выбора полей из разных таблиц и запросов (Tables/Queries)) - для нашего примера это одно поле *Gorod* из таблицы *TbGorod*;
- указываем вид формы (... layout ...) – «*Ленточный*» (*Tabular*);
- выбираем стиль формы (... style ...) – «*Стандартный*» (*Access 2003*);
- задаем имя форме (... title ...) – «*FrGorod*»;
- нажимаем **Готово** (Finish).



#### Шаг 4

Вносим изменения в форму *FrGorod*, созданную мастером. Для этого открываем форму в **режиме конструктора форм** (Design View).

- Устанавливаем желаемые размеры формы
- Изменяем размеры области примечания формы.
- Режим таблицы (Allow Datasheet View) – “Нет”
- Режим макета (Allow Layout View) - “Нет”
- Разделительные линии (Dividing Lines)- - “Да”
- Область выделения (Record Selection) – “Да”
- Модальное окно (Modal) – «Да»
- Размещаем в области примечания формы кнопки: «Удалить запись», «Выход» (кнопки создаем с помощью мастера).

#### Шаг 5

Запрограммируем вызов формы *FrGorod* из *Главной формы*.

Открываем *Главную форму* в **режиме конструктора** (Design View).

Выбираем кнопку «Города» и изменяем ее свойство **Нажатие кнопки** (On Click) (нажимаем на кнопку [...]). В диалоге **Построитель** (Choose Builder) выбираем **Программы** (Code Builder). В программе вводим команду открытия формы "FrGorod":

```
DoCmd.OpenForm "FrGorod"
```

и закрываем окно программного модуля и окно конструктора, подтвердив при необходимости сохранение изменений.

#### Шаг 6

Тестируем открытие формы *FrGorod* из *Главной формы*, и заполняем таблицу *TbGorod* через созданную форму.

#### Шаг 7

Создаем форму *FrTovar* аналогично форме *FrGorod*. Вид формы устанавливаем «Одиночная форма» (*Single Form*).

Придадим форме *FrTovar* более привлекательный вид, для этого в **режиме конструктора** (Design View) или **макета формы** (Layout View) отредактируем месторасположение полей и надписей, уточним их цвет, шрифт, размер и т.д.

Удобным средством настройки внешнего вида формы является группировка/разгруппировка элементов формы. Выделив предварительно необходимые элементы формы, нажимаем кнопку **Группировать** (Group) в группе **Макет элемента управления** (Control Layout) на вкладке **Упорядочить** (Arrange).

При создании формы с помощью **Мастера форм** (Form Wizard) автоматически создается **Макет элемента управления** (Control Layout). Макеты элементов управления – это направляющие, по которым выравниваются элементы по вертикали и горизонтали для единообразного оформления формы, в форме их может быть несколько.

Для образования нового макета элементов нужно выделить их и выбрать **В столбик** (Stacked) или **Табличный** (Tabular) на вкладке ленты **Упорядочить** (Arrange) в группе **Макет элемента управления** (Control Layout).

В рамках одного макета, созданного в форме, невозможно изменение местоположения и ширины отдельных элементов. Чтобы это сделать, нужно удалить элемент из макета. Для этого выделяем элемент/(ы) формы:

- и выбираем **Удалить** (Remove) на вкладке ленты **Упорядочить** (Arrange) в группе **Макет элемента управления** (Control Layout),
- или выбираем **Макет** (Layout)/**Удалить** (Remove) в контекстном меню, вызываемом нажатием правой клавиши “мыши”.

Товары

### База данных по товарам

**Карточка товара**

Название  
Стул

Цена  
85,00р.

Описание  
Деревянный

Фотография  
Фото отсутствует

Удалить запись      Выход

Record: 1 of 1    No Filter    Search

#### Шаг 9

Запрограммируем вызов формы *FrTovar* из *Главной формы*.

#### Шаг 10

Тестируем открытие формы *FrTovar* из *Главной формы*, и заполняем таблицу *TbTovar* через созданную форму.

### Лабораторная работа №3

#### ЦЕЛЬ:

Разработка формы *FrKlient*, которая предназначена для ведения базы данных по клиентам.

Данная форма работает в двух режимах: просмотра и правки записей по кнопке переключения режимов. Атрибут клиента *Город* выбирается из раскрывающегося списка городов.

База данных по клиентам

**РЕЖИМ ПРОСМОТРА**

Карточка клиента

Название фирмы: ООО Риск

Город: Сочи

Адрес: ул. Школьная, д.34, г. Сочи

Дата регистрации: 18.06.2004

☒ Постоянный клиент

Правка

Удалить запись

Выход

Record: 2 of 4 No Filter Search

#### Порядок выполнения работы

##### Шаг 1

Создаем форму *FrKlient* с помощью мастера, аналогично лаб. раб. №2.

Для удобства ввода даты меняем свойства поля *Дата регистрации*:

Формат поля (Format) – *Краткий формат даты (Short Date)*

Маска ввода (Input Mask)– следуя указаниям мастера, выбираем маску ввода.

##### Шаг 2

В созданной форме заменяем поле *Код города* на раскрывающийся список городов, значения для которого берутся из *TbGorod*:

- Открываем форму *FrKlient* в **режиме конструктора (Design View)** и удаляем поле *Код города*
- С панели инструментов переносим на форму элемент **Поле со списком (Combo Box)**.
- В появившемся мастере последовательно указываем:
  - «Значения из таблицы или запроса» (...the values in table or query);
  - таблицы (tables) *TbGorod*, поля (fields) *Kod\_gor*, *Gorod*;
  - по каким полям будем осуществлять сортировку, если это необходимо;
  - «Скрыть ключевой столбец» ( ☒ **Hide key column (recommended)** );
  - сохранить значение в поле (Store that value in this field) *Kod\_gor*;
  - подпись “*поля со списком*” (...label ... combo box) - Город.

##### Шаг 3

При открытии форма находится в режиме просмотра записей. Для того чтобы запретить изменение существующих записей (сделать форму доступной только для чтения), следует задать для ее свойств **Разрешить добавление** (AllowAdditions), **Разрешить удаление** (AllowDeletions) и **Разрешить изменение** (AllowEdits) значения «Нет».

#### Шаг 4

В режиме просмотра записей кнопка «Удалить запись» должна быть недоступна. Устанавливаем ей свойство **Доступ** (Enabled) – «Нет».

#### Шаг 5

Добавляем в форму кнопку «Смены режима» и надпись «РЕЖИМ ПРОСМОТРА». При нажатии на эту кнопку, вместе со сменой режима, будет меняться:

- текст надписи на форме «РЕЖИМ ПРОСМОТРА» / «РЕЖИМ ПРАВКИ»,
- цвет надписи на форме,
- свойство кнопки «Удалить запись» **Доступ** (Enabled) – «Да» / «Нет»,
- надпись на кнопке смены режима (Caption) – «Просмотр» / «Правка».

Для этого в программе обработки события «Нажатие» (On Click) кнопки «Смены режима» пишем:

```
'определяем текущий режим по надписи на кнопке
If Имя_кнопки_смены_режима.Caption = "Правка" Then
'меняем надпись
    Имя_кнопки_смены_режима.Caption = "Просмотр"
'разрешаем изменения данных
    Forms!FrKlient.AllowEdits = True
    Forms!FrKlient.AllowAdditions = True
    Forms!FrKlient.AllowDeletions = True
'меняем текст надписи на форме
    Имя_надписи.Caption = "РЕЖИМ ПРАВКИ"
'меняем цвет надписи
    Имя_надписи.ForeColor = 255
'разблокируем кнопку «Удалить запись»
    Имя_Кнопки_удаления.Enabled = True
Else
    Имя_кнопки_смены_режима.Caption = "Правка"
    Forms!FrKlient.AllowEdits = False
    Forms!FrKlient.AllowAdditions = False
    Forms!FrKlient.AllowDeletions = False
    Имя_надписи.Caption = "РЕЖИМ ПРОСМОТРА"
'меняем цвет надписи
    Имя_надписи.ForeColor = 155
'заблокируем кнопку «Удалить запись»
    Имя_Кнопки_удаления.Enabled = False
End If
```

#### Шаг 6

Запрограммируем вызов формы *FrKlient* из *Главной формы* (см. Лаб. раб. №2).

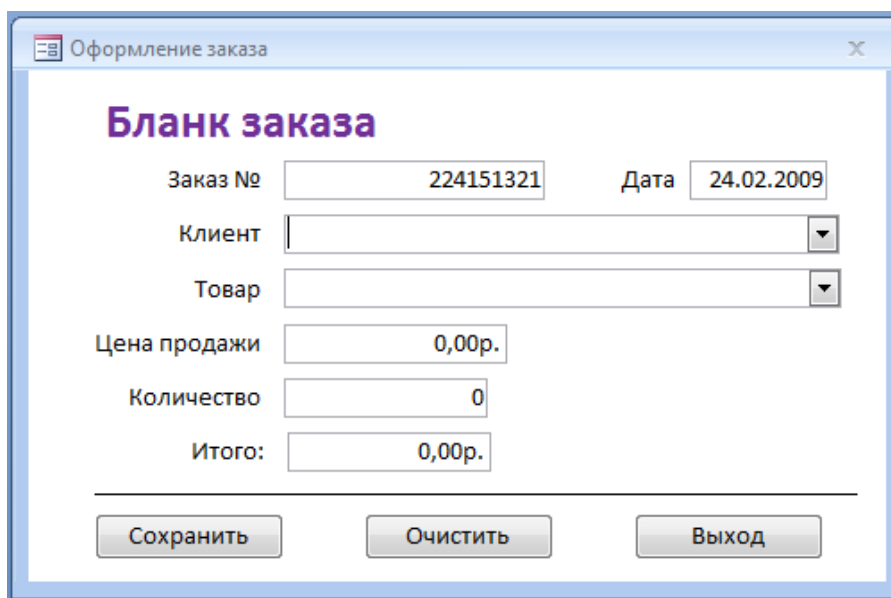
## Лабораторная работа №4

### ЦЕЛЬ:

Разработка формы *FrZakaz*, предназначенной для оформления нового заказа, с использованием непосредственного доступа к полям записей.

Поля *Дата* и *Номер заказа* формируются автоматически (без возможности корректировки). В поле *Дата* заносится текущая системная дата, поле *Номер заказа* будем формировать по схеме ММДДЧЧСС.

Атрибуты заказа *Клиент* и *Товар* берутся из раскрывающихся списков названий фирм (*TbKlient*) и товаров (*TbTovar*), в таблицу заказов (*TbZakaz*) вносятся коды Клиента и Товара.



### Порядок выполнения работы

#### Шаг 1

Создаем форму *FrZakaz* в режиме конструктора.

На вкладке ленты **Создание** (*Create*) выбираем **Конструктор форм** (Form Design).

- Устанавливаем желаемые размеры формы
- Изменяем свойства формы:
  - Подпись (Caption) – “Формирование заказа”
  - Режим по умолчанию (Default View)– «Одиночная форма» (*Single Form*)
  - Режим таблицы (Datasheet View) – «Нет»
  - Область выделения (Record Selectors) – «Нет»
  - Кнопки перехода (Navigation Buttons) - “Нет”
  - Кнопка закрытия (Close Button) – «Нет»
  - Разделительные линии (Dividing Lines)- “Нет”
  - Кнопки размеров окна (Min Max Buttons) – “Отсутствуют”
  - Полосы прокрутки (Scroll Bars) – “Отсутствуют”
  - Тип границы (Border Style) - “Окна диалога” (*Dialog*)
  - Модальное окно (Modal) – «Да»

#### Шаг 2

Помещаем на форму элементы управления:

- Поле (Text Box) – *Номер заказа*
  - Имя (Name) – «*pN\_Zak*»
  - Формат поля (Format) – «Основной» (*General Number*)
  - Блокировка (Locked) – «Да»
  - Переход по Tab (Tab Stop) – «Нет»

- Поле (Text Box) – *Дата оформления заказа*
    - Имя (Name) – «*pDate*»
    - Формат поля (Format) – «*Краткий формат даты*» (*Short Date*)
    - Блокировка (Locked) – «*Да*»
    - Переход по Tab (Tab Stop) – «*Нет*»
    - Данные (Control Source) - «*=Date()*» (функция возвращает текущую дату)
  - Поле со списком (Combo Box) – *Клиент* (имя *psKlient*), создаем с помощью мастера и следуя его рекомендациям последовательно указываем:
    - «Значение из таблицы» (...the values in table or query);
    - таблицы (tables) *TbKlient*, поля (fields) *Kod\_kl*, *Klient*;
    - по каким полям будем осуществлять сортировку, если это необходимо;
    - «Скрыть ключевой столбец» ( ☒ *Hide key column (recommended)* );
    - сохранить значение в поле (Store that value in this field) *Kod\_kl*;
    - подпись “*поля со списком*” (...label ... combo box) – Клиент.
  - Поле со списком (Combo Box) – *Товар* (имя *psTovar*), создаем аналогично полю со списком *Клиент* и указываем
    - Таблица – *TbTovar*,
    - Поля – *Kod\_tov*, *Цена* и *Tovar*,
    - Подпись “*поля со списком*” (...label ... combo box) – Товар.
- Задаем ширину столбцов (Column Widths) – *0см* для первых двух столбцов.

- Поле (Text Box) – *Цена продажи*.
  - Имя (Name) – «*pCena\_prod*»
  - Формат поля (Format) – «*Денежный*» (Currency)

После выбора товара, в это поле автоматически подставляется значение цены из таблицы *TbTovar*, с возможностью её изменения.

Для этого в процедуре обработки события «**После обновления**» (After Update) объекта поле со списком «*psTovar*» указываем:

```
Private Sub psTovar_AfterUpdate()  
    pCena_prod = CCur(psTovar.Column(1))  
End Sub
```

- Поле (Text Box) – *Количество*.
  - Имя (Name) – «*pKol\_vo*»
  - Формат поля (Format) – «*Основной*» (*General Number*)
  - Значение по умолчанию (Default Value) - 0
- Поле (Text Box) – *Итого* (вычисляемое поле, равное произведению цены продажи товара на его количество).
  - Имя (Name) – «*pItogo*»
  - Формат поля (Format) – «*Денежный*» (Currency)
  - Блокировка (Locked) – «*Да*»
  - Данные (Control Source) - «*=[pKol\_vo]\*[pCena\_prod]*» (используем построитель выражений [...] или же непосредственно вводим с клавиатуры).

### Шаг 3

Размещаем на форме кнопки: «Сохранить», «Очистить», «Выход».



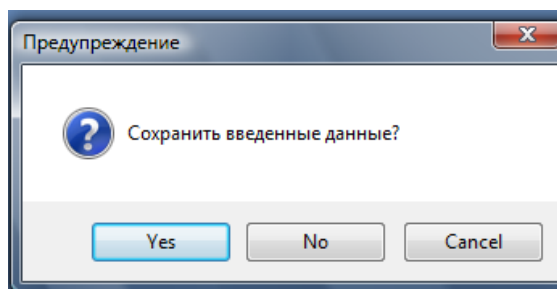
#### Шаг 4

Для отслеживания факта внесенных изменений вводим логическую переменную *NeedSave*, видимую из всех процедур модуля *FrZakaz*, принимающую значение True при необходимости сохранения введенных данных. В разделе (General)/(Declarations) указываем:

```
Dim NeedSave As Boolean
```

#### Шаг 5

При нажатии на кнопку «Выход» (имя – «*bExit*») перед закрытием формы *FrZakaz* будем осуществлять проверку значения переменной *NeedSave* и отображать соответствующее *Предупреждение*.



Для этого в процедуре обработки события «**Нажатие кнопки**» (On Click) указываем:

```
Private Sub bExit_Click()  
Dim Response As Integer  
On Error GoTo Err_bExit_Click  
If NeedSave Then  
    Response = MsgBox("Сохранить введенные данные ?", _  
        vbYesNoCancel + vbQuestion, "Предупреждение")  
    If Response = vbCancel Then Exit Sub  
    If Response = vbYes Then Call bSave_Click  
End If  
If NeedSave=False Then DoCmd.Close  
Exit_bExit_Click:  
Exit Sub  
Err_bExit_Click:  
MsgBox Err.Description  
Resume Exit_bExit_Click  
End Sub
```

#### Шаг 6

При нажатии на кнопку «Очистить» (имя – «*bClear*») формируем бланк нового заказа (что, в принципе, равноценно отмене внесенных изменений):

- в поле *pDate* заносим текущую системную дату,
- в поле *pN\_zak* формируем новый номер заказа,
- очищаем значения в полях со списками *Товар* и *Клиент*,
- обнуляем значения в полях *Количество* и *Цена продажи*,
- переменной *NeedSave* присваиваем значение *False*.

Для этого в процедуре обработки события «**Нажатие кнопки**» (On Click) указываем:

```
Private Sub bClear_Click()  
Dim CurDate As Date  
'получение текущих даты/времени  
CurDate = Now()
```

```

'формирование номера заказа
pN_Zak = Month(CurDate) * 100000000 + Day(CurDate) * 1000000 _
+ Hour(CurDate) * 10000 + Minute(CurDate) * 100 + _
Second(CurDate)
'отмена выбранного значения списка клиентов и товаров
psKlient.Value = -1
psTovar.Value = -1
'обнуление полей количество и цена
pKol_vo = 0
pCena_prod = 0
NeedSave = False
End Sub

```

### Шаг 7

При изменении значения поля *Количество* (*Kol\_vo*) товара, присваиваем переменной *NeedSave* значение *True*.

Для этого в процедуре обработки события **«Изменение»** (On Change) поля *Kol\_vo* указываем:

```

Private Sub pKol_vo_Change()
    NeedSave = True
End Sub

```

### Шаг 8

При загрузке формы *FrZakaz* будем задавать переменной *NeedSave* значение *False*, и вызывать процедуру формирования бланка нового заказа (*bClear\_Click*).

Для этого в процедуре обработки события **«Загрузка»** (On Load) указываем:

```

Private Sub Form_Load()
    NeedSave = False
    Call bClear_Click
End Sub

```

### Шаг 9

При нажатии на кнопку **«Сохранить»** (имя – *bSave*) выполняем процедуру сохранения данных из формы в таблицу *TbZakaz* и проверку корректности введенной информации.

Для этого в процедуре обработки события **«Нажатие кнопки»** (On Click) указываем:

```

Private Sub bSave_Click()
    Dim rst As ADODB.Recordset
    'Проверка правильности оформления заказа
    If psKlient.ListIndex = -1 Then
        MsgBox "Укажите клиента!", vbOKOnly + vbExclamation,
        "Проверка значений"
        Exit Sub
    End If
    'Аналогично выполняем проверку для остальных значений
    *
    *
    *
    'Добавление новой записи в таблицу TbZakaz
    'Иницилируем указатель на таблицу TbZakaz

```

```

Set rst = New ADODB.Recordset
rst.Open "TbZakaz", CurrentProject.Connection, adOpenKeyset, _
        adLockOptimistic, adCmdTableDirect
With rst
`Добавляем запись
    .AddNew
`Заносим значения в поля записи
    .Fields("N_zak") = pN_Zak.Value
    .Fields("Kod_kl") = psKlient.Value
    .Fields("Kod_Tov") = psTovar.Value
    .Fields("Kol_vo") = pKol_vo.Value
    .Fields("Cena_prod") = pCena_prod.Value
    .Fields("Data_zak") = pDate.Value
`Фиксируем внесенные изменения
    .Update
End With
rst.Close
Set rst = Nothing
Call bClear_Click
End Sub

```

### **ВСПОМОГАТЕЛЬНАЯ ИНФОРМАЦИЯ**

Для организации доступа к данным, начиная с Access 2000 используется библиотека объектов - *ActiveX Data Objects (ADO)*.

Для установления ссылки на библиотеку необходимо в редакторе **Visual Basic** выбрать пункт меню **Tools / References**.

Откроется диалоговое окно **References**, в котором перечислены все установленные на компьютере библиотеки объектов.

Мы будем использовать объекты ADO, по этому нам нужно подключить библиотеку Microsoft ActiveX Data Objects 2.1 Library..

### **Шаг 10**

Выполняем привязку вызова формы *FrZakaz* из *Главной формы* (см. Лаб. раб. №2).

## Лабораторная работа №5

### ЦЕЛЬ:

Разработка формы *FrTbZakaz* для просмотра и удаления заказов без возможности редактирования и добавления новых заказов.

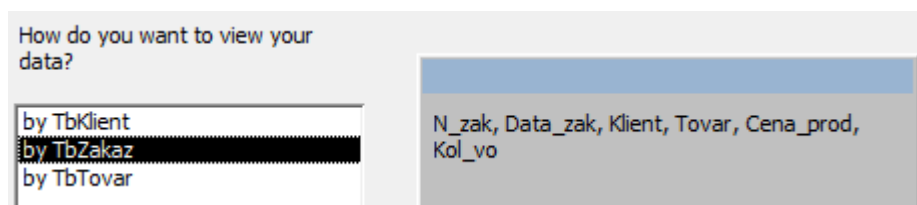
Разработка формы *FrKtoKupil* для просмотра информации по продажам выбранного товара на основе связи между таблицами типа 1:M и использовании подчиненной формы.

### Порядок выполнения работы

#### Шаг 1

Создаем форму *FrTbZakaz* с помощью мастера (*Form Wizard*) и следуя его указаниям последовательно выполняем:

- выбираем поля которые разместим на форме - для нашего примера это поля: *N\_zak*, *Data\_zak*, *Klient*, *Tovar*, *Cena\_prod*, *Kol\_vo*;
- вид представления данных – «Одиночная форма» (*Single form*);



- задаем вид формы (... layout...) – “Табличный” (*Datasheet*);
- задаем стиль формы (... style ...) – “Стандартный” (*Access 2003*);
- задаем имя форме (... title ...) – “*FrTbZakaz*”.

Номер заказа	Дата заказа	Клиент	Название товара	Цена продажи	Количество	Стоимость, руб
224130157	24.02.2009	ЗАО Строймаш	Диван	1 400,00р.	2	2 800,00р.
224131226	24.02.2009	ЗАО Строймаш	Стул	85,00р.	10	850,00р.
224145847	24.02.2009	ООО Риск	Кресло	800,00р.	5	4 000,00р.
224145912	24.02.2009	ООО Риск	Кресло	790,50р.	3	2 371,50р.
224150038	24.02.2009	ООО Риск	Стул	65,80р.	10	658,00р.
224150450	24.02.2009	ЗАО Строймаш	Кресло	799,60р.	3	2 398,80р.
224150510	24.02.2009	ЗАО Строймаш	Кресло	850,60р.	1	850,60р.
226113015	26.02.2009	ООО Риск	Диван	1 500,00р.	2	3 000,00р.
227111221	27.02.2009	ООО Риск	Кресло	850,60р.	5	4 253,00р.

#### Шаг 2

Открываем вновь созданную форму *FrTbZakaz* в режиме конструктора (*Design View*). Корректируем названия подписей полей, полю *Cena\_prod* задаем денежный (*Currency*) формат поля

Добавляем поле Стоимость заказа, в свойстве **Источник данных** (*Control Source*) указываем вычисляемое выражение “ = *[Cena\_prod] \* [Kol\_vo]*”, задаем денежный (*Currency*) формат поля.

Задаем следующие свойства для формы:

- Подпись (*Caption*) – “Список заказов”
- Режим по умолчанию (*Default View*) – “Режим таблицы” (*Datasheet*)
- Режим формы (*Allow Form View*) – “Нет” (и остальные режимы - нет)
- Разрешить изменение (*Allow Edits*) – “Нет”
- Разрешить добавление (*Allow Additions*) – “Нет”
- Разрешить удаление (*Allow Deletions*) – “Да”

### Шаг 3

Добавим кнопку вызова формы *FrTbZakaz* (“Список заказов”) в форму *FrZakaz*. Для этого открываем форму *FrZakaz* в режиме конструктора (Design View) и создаем кнопку “Список заказов” с помощью мастера указав **Работа с формой** (Form Operation) в разделе **Категории** (Categories) и **Открыть форму** (Open Form) в разделе **Действия** (Actions).

Изменим параметры открытия формы *FrTbZakaz*:

```
DoCmd.OpenForm "FrTbZakaz", acFormDS
```

### Шаг 4

Модифицируем процедуру сохранения заказа с целью обновления информации в форме *FrTbZakaz* при добавлении нового заказа.

Добавляем ниже приведенный код в конец процедуры сохранения заказа.

```
` Проверяем, загружена ли форма FrTbZakaz
If CurrentProject.AllForms("FrTbZakaz").IsLoaded Then
` Если да, то обновляем в ней информацию
Forms("FrTbZakaz").Requery
End If
```

### Шаг 5

Модифицируем процедуру закрытия формы *FrZakaz* с целью закрытия еще и формы *FrTbZakaz*.

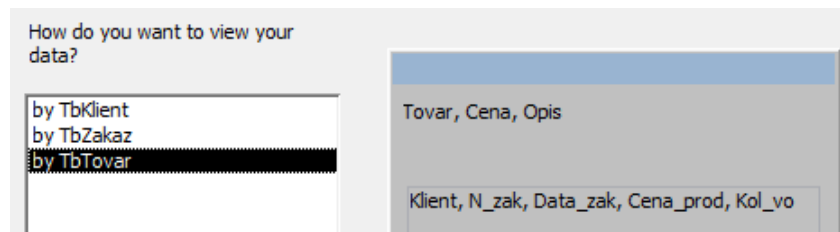
Для этого добавим в процедуру закрытия формы *FrZakaz* еще одну команду.

```
DoCmd.Close acForm, "FrTbZakaz"
```

### Шаг 6

Создаем форму *FrKtoKupil* с помощью мастера (Form Wizard) и следуя его указаниям выполняем:

- переключаясь между таблицами, задаем следующий порядок полей на форме *Tovar*, *Cena*, *Opis*, *Klient*, *N\_Zak*, *Data\_Zak*, *Cena\_prod*, *Kol\_vo*;
- выбираем вид представления данных – *TbTovar*;



- указываем – «Форма с подчиненной формой» (Form with *subform(s)*);
- вид подчиненной формы (... layout ... subform) – «Табличная» (Datasheet);
- стиль формы (Style) – «Стандартный» (Access 2003);
- задаем имя ( ... titles ...) формы (Form) – *FrKtoKupil*;
- задаем имя ( ... titles ...) подчиненной формы (Subform) - *FrKtoKupisl*.

## Шаг 7

Открываем форму *FrKtoKupil* в режиме конструктора (Design View) и задаем следующие свойства:

- Режим по умолчанию (Default View) – «Одиночная форма» (Single Form)
- Режим таблицы (Allow Datasheet View) – «Нем»
- Область выделения (Record Selectors) – «Нем»
- Разделительные линии (Dividing Lines) - «Нем»
- Разрешить изменение (Allow Edits) – «Нем»
- Разрешить добавление (Allow Additions) – «Нем»
- Разрешить удаление (Allow Deletions) – «Нем»

## Шаг 8

Для подчиненной формы *FrKtoKupisl* задаем свойства:

- Режим по умолчанию (Default View) – «Режим таблицы» (Datasheet)
- Режим формы (Allow Form View) – «Нем»
- Разрешить изменение (Allow Edits) – «Нем»
- Разрешить добавление (Allow Additions) – «Нем»
- Разрешить удаление (Allow Deletions) – «Нем»

Помещаем на форму *FrKtoKupisl* поле (Text Box) *Стоимость заказа*, задаем денежный формат поля (Currency), в качестве источника данных (Control Source) указываем вычисляемое выражение " $= [Cena\_prod] * [Kol\_vo]$ " (используем построитель выражений [...] или же непосредственно вводим с клавиатуры).

## Шаг 9

Размещаем на форме кнопку «Выход» и привязываем вызов формы *FrKtoKupil* из Главной формы (см. Лаб. раб. №2).

## Лабораторная работа №6

### ЦЕЛЬ:

Разработка формы для просмотра, поиска и фильтрации списка клиентов с возможностью вызова форм для редактирования информации.

При вызове формы для редактирования информации необходимо обеспечить автоматическое позиционирование на запись, которая была текущей в вызывающей форме, обновление информации в вызывающей форме после закрытия вызываемой.

Название фирмы клиента	Название города	Дата регистрации фирм	Постоянный клиент
ДОО Уникс	Москва	11.11.1999	Да
ЗАО Строймаш	Новосибирск	13.08.2008	
ООО Исток	Москва	12.09.2005	
ООО Петров	Чита	13.05.2004	
ООО Риск	Сочи	18.06.2004	Да

Record: 6 of 6   Unfiltered   Search

Отображать:  
☒ Все клиенты  
☐ Постоянные

Сортировка:  
☒ Клиент  
☐ Город  
☐ Город+Клиент  
☐ Дата регистрации

Найти:  
☒ Клиента  
☐ Город     

### Порядок выполнения работы

#### Шаг 1

Создаем форму *FrKlient2* в режиме конструктора (Form Design)

- Устанавливаем желаемые размеры формы
- Изменяем свойства формы:
  - Подпись (Caption) – “Списки клиентов”
  - Режим по умолчанию (Default View) – «Одиночная форма»
  - Режим таблицы (Datasheet View) – «Нет»
  - Область выделения (Record Selectors) – «Нет»
  - Кнопки перехода (Navigation Buttons) - «Нет»
  - Разделительные линии (Dividing Lines)- «Нет»
  - Кнопки размеров окна (Min Max Buttons) – «Отсутствуют»
  - Полосы прокрутки (Scroll Bars) – «Отсутствуют»
  - Тип границы (Border Style) - «Тонкая » (Thin)

#### Шаг 2

Помещаем на форму *FrKlient2* элемент управления **Подчиненная форма /отчет** (Subform/Subreport) - имя “*pfKlient*” и следуя рекомендациям мастера, указываем:

- использовать имеющиеся таблицы и запросы (Use existing Tables and Queries);
- переключаясь между таблицами, задаем следующий порядок полей на форме *Klient, Gorod, Data\_Reg, Post*;
- задаем имя подчиненной формы (...name... subform or subreport) - *FrKlient2sl*;
- Готово (Finish).

В результате на форме находится элемент управления **Подчиненная форма** *pfKlient*, у которой в свойстве **Объект-источник** данных (Source Object) указана форма *FrKlient2sl*.

### Шаг 3

Открываем форму *FrKlient2* в режиме конструктора (Form Design). Для подчиненной формы задаем свойства:

- Режим по умолчанию (Default View) – “Режим таблицы” (*Datasheet*)
- Режим формы (Allow Form View) – “Нет”
- Разрешить изменение (Allow Edits) – “Нет”
- Разрешить добавление (Allow Additions) – “Нет”
- Разрешить удаление (Allow Deletions) – “Нет”

### Шаг 4

Для увеличения наглядности, в подчиненной форме *pfKlient*, заменяем элемент управления «Постоянный клиент» (имя «*Post*») с **Флажка** на **Поле**, в котором будет выводиться надпись «ДА» для постоянных клиентов.

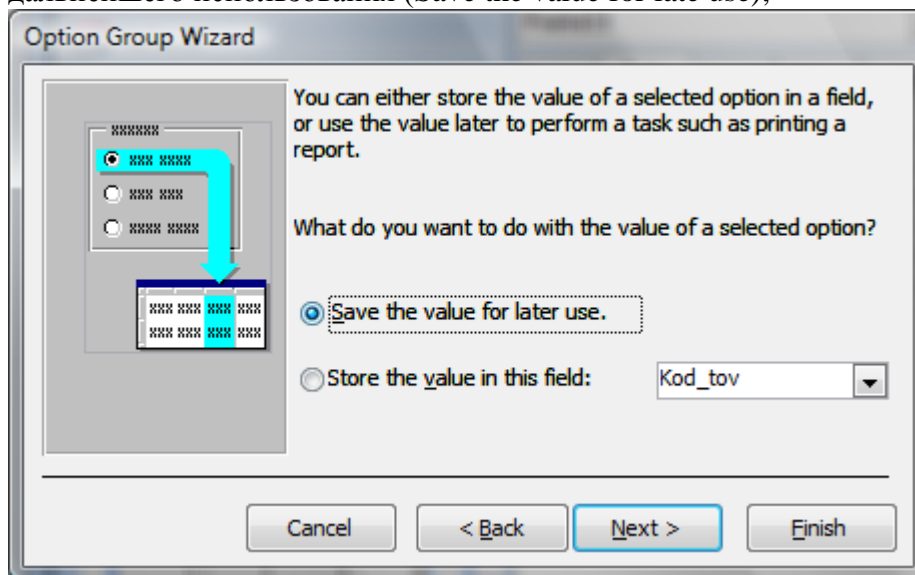
Для этого задаем полю *Post* свойство **Данные** (Control Source) равным “=if([*Post*];“ДА”;”)” (в зависимости от региональных установок в операционной системы вместо “;” следует использовать “,”).

### Шаг 5

Размещаем на форме *Frklient2* элемент управления **Группа переключателей** (Option Group) - имя “*gSort*”, служащий для выбора критерия сортировки списка клиентов в подчиненной форме.

Для этого, следуя указаниям мастера задаем:

- подписи (... label ...) для каждого переключателя (“Клиент”, “Город”, “Город + Клиент”, “Дата регистрации”);
- задаем значение переключателя используемого по умолчанию (... default choice) - “Клиент”;
- задаем числовые значения для каждого переключателя;
- указываем, что будем запоминать значение выбранного переключателя для дальнейшего использования (Save the value for late use);



- выбираем тип группы переключателей, это может быть набор кнопок, флажков или переключателей (... type of controls ...) - «Переключатель» (*Option buttons*) и стиль оформления (... style ...), например, «Приподнятый» (*Raised*),
- указываем подпись группы переключателей (... caption ...) – “Сортировка”.



При изменении значения группы переключателей *gSort* вызывается процедура, в которой осуществляется изменение свойств *OrderBy* (устанавливает критерий сортировки) и *OrderByOn* (активирует сортировку) подчиненной формы *pfKlient*.

Для формирования критерия объявляем строковую переменную, которой, используя функцию *Choose* и значение *gSort*, присваивается значение. Таким образом, процедура реакции на событие «После обновления» (AfterUpdate) элемента управления *gSort* будет иметь следующий вид:

```
Private Sub gSort_AfterUpdate()  
Dim str As String  
    str = Choose(gSort.Value, "Klient", "Gorod",  
                  "Gorod,Klient", "Data_Reg")  
'Задаем критерий сортировки  
    Forms("FrKlient2").Controls("pfKlient").Form.OrderBy = str  
'Включаем сортировку  
    Forms("FrKlient2").Controls("pfKlient").Form.OrderByOn = True  
End Sub
```

Соответственно, необходимо принудительно вызвать процедуру *gSort\_AfterUpdate* во время открытия формы:

```
Private Sub Form_Open(Cancel As Integer)  
    Call gSort_AfterUpdate  
End Sub
```

## Шаг 6

Аналогично создаем группу переключателей «Отображать» (имя - «*gFiltr*»). Эта группа переключателей предоставляет возможность показа всех или только постоянных клиентов. Для этого включаем/выключаем фильтрацию в подчиненной форме *pfKlient*.

Процедура реакции на событие «После обновления» (AfterUpdate) элемента управления *gFiltr* будет иметь следующий вид:

```
Private Sub gFiltr_AfterUpdate()  
    If gFiltr = 1 Then  
        Forms("FrKlient2").Controls("pfKlient").Form.FilterOn=False  
    Else  
        Forms("FrKlient2").Controls("pfKlient").Form.Filter=  
                                                    "Post=TRUE"  
        Forms("FrKlient2").Controls("pfKlient").Form.FilterOn=True  
    End If  
End Sub
```

## Шаг 7

Для возможности осуществления поиска помещаем на форму следующие элементы:

- группу переключателей *gFind* выбора области поиска;
- поле *pFind*, для ввода искомого фрагмента текста;
- кнопку *bFind*, инициирующую процедуру поиска.

В процедуре обработки события «Нажатие» (Click) кнопки *bFind*:

- вначале определяем, по какому полю осуществляем поиск (значение *gFind*) и делаем это поле активным в подчиненной форме;
- затем выполняем поиск фрагмента текста, введенного в поле *pFind* при помощи метода *FindRecord*.

Если запись найдена, то она становится активной, в противном случае, номер текущей записи не изменится.

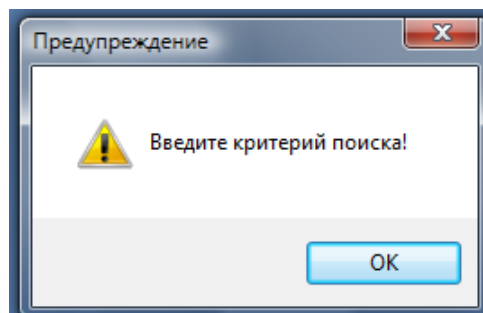
В методе *FindRecord* используем следующие параметры поиска:

- *pFind.Value* – что ищем;
- *acAnywhere* – искать по частичному совпадению;
- *False* – поиск без учета регистра;
- *acSearchAll* – поиск записи во всем наборе;
- *False* – без учета формата поля;
- *acCurrent* – ограничение поиска текущим полем;
- *True* – выполнять поиск с первой записи.

```
Private Sub bFind_Click()  
Dim str As String  
'сообщаем программе, что поиск осуществляем в подчиненной форме  
Forms("FrKlient2").Controls("pfKlient").SetFocus  
'выбираем столбец поиска  
If gFind.Value = 1 Then  
Forms("FrKlient2").Controls("pfKlient").Form.Klient.SetFocus  
Else  
Forms("FrKlient2").Controls("pfKlient").Form.Gorod.SetFocus  
End If  
DoCmd.FindRecord pCrit.Value, acAnywhere, False, _  
acSearchAll, False, acCurrent, True  
End Sub
```

## Шаг 8

Дополним процедуру обработки события «**Нажатие**» (On Click) кнопки *bFind* проверкой наличия критерия в поле *pFind* и в случае его отсутствия будем отображать соответствующее *Предупреждение*.



Для проверки наличия критерия можно использовать функцию *IsNull* (*Имя\_поля.value*) или сравнить значение поля *pFind* с пустой строкой.

## Шаг 9

Создадим на форме *FrKlient2* кнопку «*Редактировать*», при ее нажатии будет вызываться форма *FrKlient*, которая позволяет проводить операции с записями в таблице *TbKlient*. Для удобства работы выполним автоматическое позиционирование на запись при открытии формы *FrKlient*, совпадающую с текущей в форме *FrKlient2*.

Для этого используем возможность передачи аргумента в команде *OpenForm*, где строковая переменная *str* имеет значение названия клиента из текущей записи:

```
Dim str As String  
str = Forms("FrKlient2").Controls("pfKlient").Form.Klient  
DoCmd.OpenForm "FrKlient", , , , , str
```

Соответственно необходимо также модифицировать событийную процедуру «Открытие» (On Open) формы *FrKlient*. Вначале осуществляем проверку наличия переданного в форму аргумента и, в случае положительного результата, используем уже известный нам метод *FindRecord*:

```
Private Sub Form_Open(Cancel As Integer)
Dim str As String
    If Not IsNull(Forms("FrKlient").OpenArgs) Then
        str = Forms("FrKlient").OpenArgs
        DoCmd.GoToControl "Klient"
        DoCmd.FindRecord str, , True, , True, , True
    End If
End Sub
```

После закрытия формы *FrKlient* необходимо обновить информацию в форме *FrKlient2*, для этого используем метод *Requery* в ее процедуре обработки события «Включение» (On Activate).

```
Private Sub Form_Activate()
    Forms("FrKlient2").Controls("pfKlient").Requery
End Sub
```

#### **Шаг 10**

Размещаем на форме кнопку «Выход» и привязываем вызов формы *FrKlient2* из Главной формы.

## Лабораторная работа №7

### ЦЕЛЬ:

Построение произвольных SQL запросов к базе данных для формирования на их основе форм и отчетов.

Запросы используются для просмотра, анализа и изменения данных из одной или нескольких таблиц. Сам запрос не содержит данные, но позволяет осуществлять выборку данных из таблиц и выполнять над ними ряд операций.

### Порядок выполнения работы

#### Шаг 1

Создать SQL запрос - *zpSQL* в **режиме мастера** (кнопка **Конструктор запросов** / Query Wizard на вкладке ленты **Создание** (Create)) извлекающий из базы данных все заказы с атрибутами: товар, клиент, город, дата заказа, количество, цена продажи. Чтобы посмотреть результаты запроса достаточно открыть его в режиме таблицы.

Открываем сформированный мастером запрос в **режиме конструктора** (Design View) и добавляем вычисляемое поле «Итого».

Для этого в строке *Поле* (Field) указываем:

Итого: ([TbZakaz].[Kol\_vo]\*[TbZakaz].[Cena\_prod])

можно ввести текст непосредственно с клавиатуры или воспользоваться **построителем выражений** (Expression Builder) – кнопка **Построитель** (Builder) на вкладке **Конструктор** / **Настройка запросов** (Query Tools/ Query Setup).

#### Шаг 2

Создаем новую форму «*FrSQL*» в **режиме конструктора** (Design View) и размещаем на ней следующие элементы управления:

- группу переключателей “*gSort*” (для задания порядка сортировки),
- группу переключателей “*gKrit*” (для выбора критерия поиска),
- список “*Spisok*” (для выбора значения критерия поиска),
- поля “*pNData*” и “*pKData*” (для указания интервала времени),
- флажок “*Find*” (для включения режима поиска),
- кнопка запуска построения SQL запроса.

Товар	Клиент	Город	Дата заказа	Кол-во	Цена	Итого
Диван	ООО Риск	Сочи	26.02.2009	2	1 500,00р.	3 000,00р.
Кресло	ООО Риск	Сочи	03.04.2009	2	1 500,00р.	3 000,00р.
Кресло	ООО Риск	Сочи	27.03.2009	6	850,60р.	5 103,60р.
Кресло	ООО "Рога и копыта"	Сочи	27.03.2009	2	596,00р.	1 192,00р.
Кресло	ООО Риск	Сочи	24.02.2009	5	800,00р.	4 000,00р.
Органайзер	ООО Риск	Сочи	17.04.2009	6	1 200,00р.	7 200,00р.
Ручка шарик	ООО "Рога и копыта"	Сочи	17.04.2009	12	57,00р.	684,00р.
Стул	ООО Риск	Сочи	24.02.2009	10	65,80р.	658,00р.

### Шаг 3

Запрограммировать следующую логику работы формы (не забыв о начальной инициализации формы):

- если флажок “Find” не установлен, то скрыть элементы “gKrit”, “Spisok”, “pNData”, “pKData”;
- если флажок “Find” установлен то отобразить элемент “gKrit” и, в зависимости от его значения, отобразить либо поля “pNData” и “pKData”; либо “Spisok”, заполненный значениями из соответствующих таблиц БД (TbTovar, TbKlient, TbGorod);

### Шаг 4

Каждая строка списка значений критерия поиска “Spisok” будет содержать два столбца (свойство **Число столбцов** (Column Count) – 2). В первом содержатся коды, а во втором названия. Для скрытия первого столбца задаем свойство **Ширина столбцов** (Column Widths) – 0 см, 2,54 см.

При изменении критерия поиска (“gKrit”) изменяется список значений критерия поиска “Spisok”, для этого необходимо изменение свойства **Источник строк** (Row Source) элемента управления список “Spisok”. При этом свойство **Тип источника строк** (Row Source Type) должно иметь значение «Таблица или запрос» (Table/Query).

Если в качестве источника строк элемента управления список “Spisok” указывать названия таблицы, то будут выводиться данные из столбцов этой таблицы, число которых указано в свойстве **Число столбцов** (Column Count). В этом случае строки списка нельзя отсортировать по алфавиту. Поэтому в качестве источника строк будем использовать SQL-выражение.

Так, например, для получения списка городов, упорядоченного в алфавитном порядке, следует использовать следующее выражение:

```
Spisok.RowSource = "SELECT TbGorod.Kod_Gor, TbGorod.Gorod FROM  
TbGorod ORDER BY TbGorod.Gorod;"
```

После изменения источника строк необходимо также обнулить значение элемента:

```
Spisok.Value = Null
```

### Шаг 5

Добавляем на форму FrSQL подчиненную форму pfSQL с использованием мастера и следуя его рекомендациям последовательно указываем:

- использовать имеющиеся таблицы и запросы (Use existing Tables and Querries);
- в качестве источника данных (Tables/Queries) выбираем созданный на первом шаге запрос zpSQL и выбираем все поля;
- задаем имя подчиненной форме ( ... name ...) – pfSQL;
- нажимаем Готово (Finish).

### Шаг 6

Добавляем на форму FrSQL кнопку «Выполнить», которая будет вызывать процедуру формирования строки, содержащую текст запроса. Полученная строка будет использована в качестве источника данных (Record Source) подчиненной формы pfSQL.

### Шаг 7

Строку, содержащую *SQL* запрос, будем формировать последовательно, в строковую переменную **strSQL**, с областью видимости на уровне формы (объявленную в секции *Declarations*):

- в секции **SELECT** укажем выбираемые поля;  
strSQL = "SELECT TbTovar.Tovar, TbKlient.Klient, . . .
- в секции **FROM** указываем названия таблиц;  
str strSQL = str strSQL & "FROM TbZakaz, . . .
- в секции **WHERE** указываем условия объединения таблиц;  
strSQL = strSQL & "WHERE TbZakaz.Kod\_kl=TbKlient.Kod\_kl AND ...
- если выбран режим «Искать», то продолжаем формировать секцию **WHERE**, в зависимости от выбранного критерия *gKrit* и его значения *Spisok*;  
str\_SQL = str\_SQL & " AND TbTovar.Kod\_tov=" & Spisok.Value
- в случае выбора в качестве критерия даты, следует преобразовать значения дат в синтаксис *SQL* запроса;  
strSQL = strSQL & " AND Data\_zak>=#" & Format(pNData.Value, \_  
"m\ /d\ /yyyy") & "# "
- в завершении, в зависимости от выбранного пункта «Сортировка», формируем предложение **ORDER BY**.  
str\_SQL = str\_SQL & " ORDER BY TbTovar.Tovar;"

### Шаг 8

Размещаем на форме кнопку “Выход” и привязываем вызов формы *FrSQL* из Главной формы.

## Лабораторная работа №8

### ЦЕЛЬ:

Создание простого отчета.

Отчеты предназначены для выборки из БД необходимой информации и оформления ее в виде документов, которые можно просмотреть и/или вывести на печать. Отчеты могут содержать подчиненные отчеты. Создавать отчеты можно в режиме конструктора или использовать мастер.

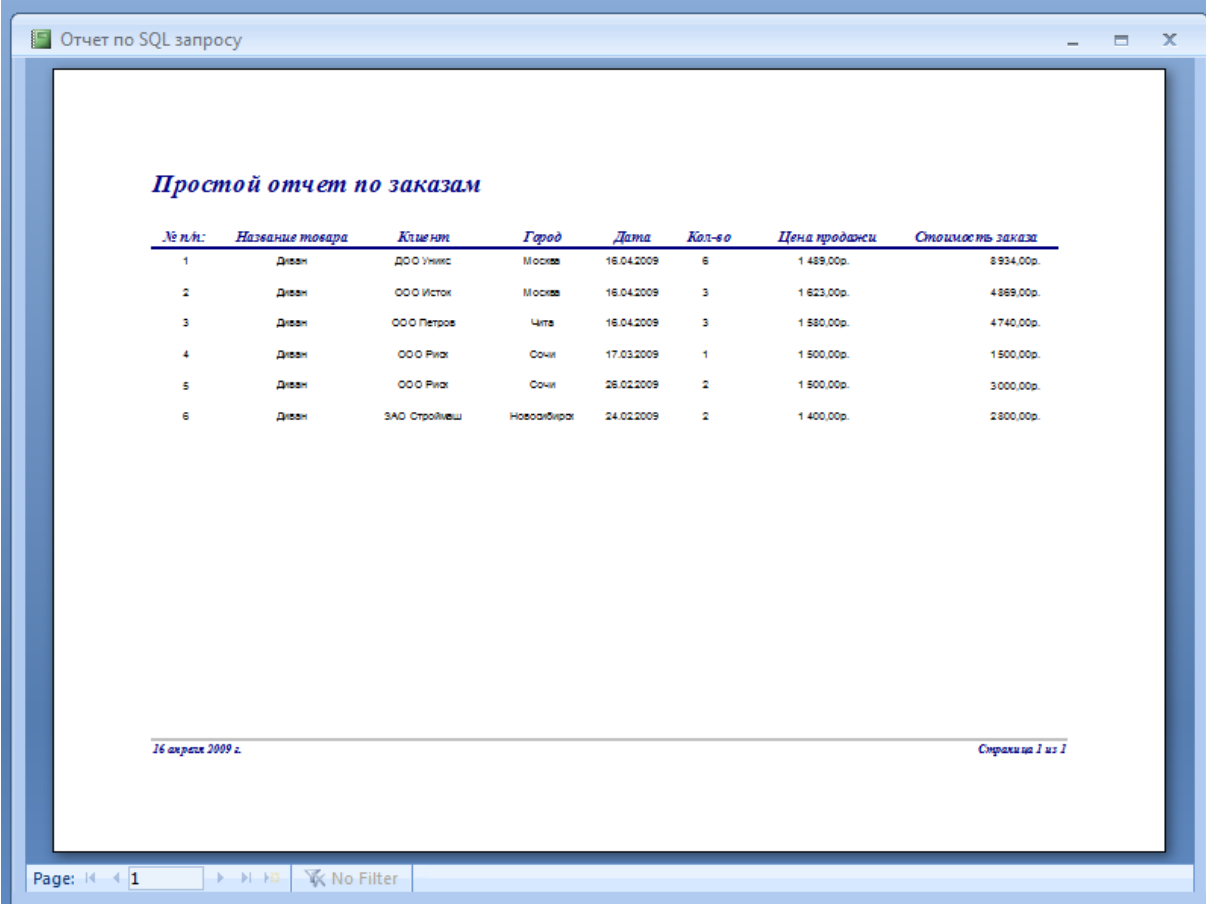
### Порядок выполнения работы

#### Шаг 1

Создадим с помощью мастера (кнопка **Конструктор отчетов** / Report Wizard на вкладке ленты **Создание** (Create)) отчет *otProstoy*, выводящий информацию по заказам.

Следуя указаниям мастера задаем:

- в качестве источника данных (Tables/Queries) для отчета зададим запрос *zpSQL*, созданный в лабораторной работе №7;
- выбираем все поля,
- представление данных (... to view ...) - по *TbZakaz*,
- уровни группировки (... grouping levels ...) - без группировки (*Tovar*);
- сортировка (... sort ...) - без сортировки;
- вид отчета (Layout) – *табличный* (*Tabular*);
- ориентация бумаги (Orientation) – *альбомная* (*Landscape*);
- стиль оформления (style) - «Стандартный» (*Access 2003*);
- задаем имя отчета (... titles ...) – «Отчет по SQL запросу»;
- нажимаем Готово (Finish).



The screenshot shows a report window titled "Отчет по SQL запросу". The report is titled "Простой отчет по заказам" and displays a table with 8 columns: № п/п, Название товара, Клиент, Город, Дата, Кол-во, Цена продажи, and Стоимость заказа. The table contains 6 rows of data. At the bottom of the report, it shows the date "16 апреля 2009 г." and "Страница 1 из 1". The status bar at the bottom indicates "Page: 1" and "No Filter".

№ п/п:	Название товара	Клиент	Город	Дата	Кол-во	Цена продажи	Стоимость заказа
1	Диван	ООО Уникс	Москва	16.04.2009	6	1 489,00р.	8 934,00р.
2	Диван	ООО Исток	Москва	16.04.2009	3	1 623,00р.	4 869,00р.
3	Диван	ООО Петров	Чита	16.04.2009	3	1 580,00р.	4 740,00р.
4	Диван	ООО Рика	Сочи	17.03.2009	1	1 500,00р.	1 500,00р.
5	Диван	ООО Рика	Сочи	26.02.2009	2	1 500,00р.	3 000,00р.
6	Диван	ЗАО Строймеш	Новосибирск	24.02.2009	2	1 400,00р.	2 800,00р.

## Шаг 2

Открываем созданный отчет в режиме конструктора (Design View). Корректируем название отчета и заголовки столбцов. В области нижнего колонтитула мастер создал два поля, выводящие текущую дату и номер страницы соответственно. Следует внимательно ознакомиться со значением свойства **Данные** (Control Source) этих полей.

## Шаг 3

Добавим столбец нумерации строк отчета. Помещаем в область данных отчета элемент управления *Поле* (Text Box) и в списке его свойств указываем:

Сумма с накоплением (Running Sum) – “Для всего” (Over All);

Данные (Control Source) – “=1”

## Шаг 4

Запрограммируем вызов данного отчета из *Главной формы* (необходимо указать режим предварительного просмотра, т.к., по умолчанию отчет отправляется на печать):

```
DoCmd.OpenReport "otProstoy", acViewPreview
```

## Шаг 5

Осуществим вызов данного отчета также и из формы *FrSQL*, для вывода информации согласно выбранным критериям. Для этого добавим на форму *FrSQL* кнопку вызова отчета «*Напечатать отчет*», и, при вызове, передадим отчету строку, содержащую SQL запрос.

```
DoCmd.OpenReport "otProstoy", acViewPreview, , , , strSQL
```

Товар	Клиент	Город	Дата заказа	Кол-во	Цена	Итого
Диван	ДОО Уникс	Москва	16.04.2009	6	1 489,00р.	8 934,00р.
Диван	ООО Исток	Москва	16.04.2009	3	1 623,00р.	4 869,00р.
Диван	ООО Петров	Чита	16.04.2009	3	1 580,00р.	4 740,00р.
Диван	ООО Риск	Сочи	17.03.2009	1	1 500,00р.	1 500,00р.
Диван	ООО Риск	Сочи	26.02.2009	2	1 500,00р.	3 000,00р.
Диван	ЗАО Строймаш	Новосибирск	24.02.2009	2	1 400,00р.	2 800,00р.

## Шаг 6

Модифицируем процедуру обработки события «Открытие» (On Open) отчета:

```
Dim str As String
If Not IsNull(Me.OpenArgs) Then
    str = Me.OpenArgs
    Me.RecordSource = str
End If
```



## Лабораторная работа №9

### ЦЕЛЬ:

Формирование «Отчета по заказам» с множественной группировкой.

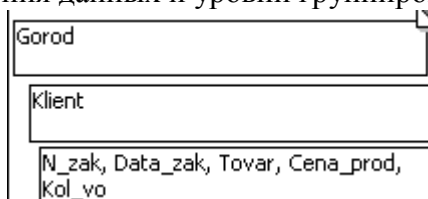
В данной работе необходимо сформировать отчет с информацией по заказам, сгруппированной по клиентам и городам, с промежуточным подсчетом количества заказов и их суммарной стоимости.

### Порядок выполнения работы

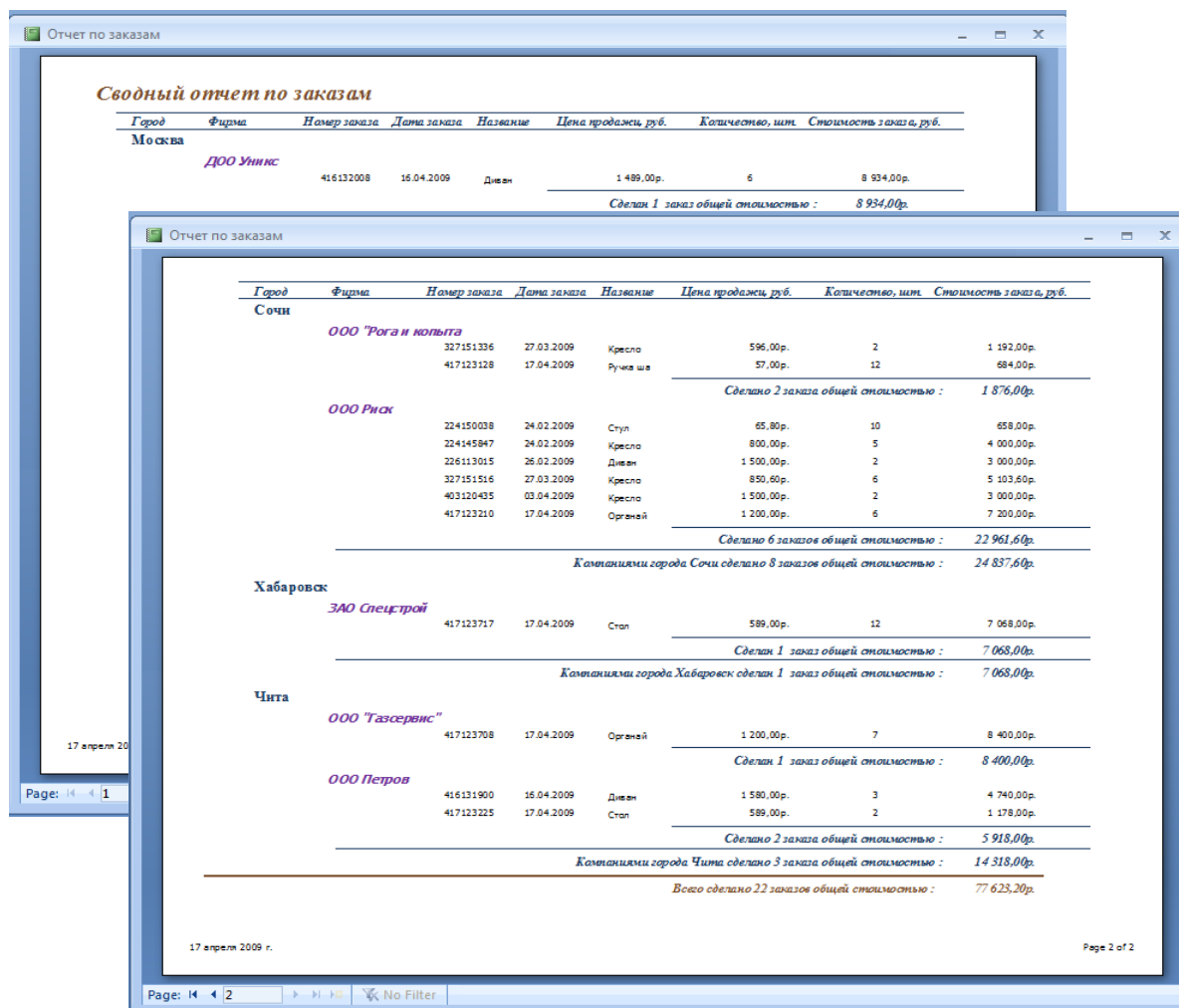
#### Шаг 1

С помощью мастера создаем «Отчет по заказам», (кнопка **Конструктор отчетов** (Report Wizard) на вкладке ленты **Создание** (Create)), и, следуя указаниям мастера, задаем:

- порядок полей: *Gorod, Klient, N\_zak, Data\_zak, Tovar, Cena\_prod, Kol\_vo*;
- вид представления данных и уровни группировки;



- порядок сортировки (... sort ...) для записей по полю - *Data\_zak*;
- вид отчета (Layout) – «Ступенчатый» (Stepped);
- ориентация бумаги (Orientation) – альбомная (Landscape);
- стиль оформления (style) - «Стандартный» (Access 2003);
- задаем имя отчета ( ... titles ...) – «Отчет по заказам»;
- нажимаем Готово (Finish).



**Сводный отчет по заказам**

Город	Фирма	Номер заказа	Дата заказа	Название	Цена продажи, руб.	Количество, шт.	Стоимость заказа, руб.
Москва	ООО Уникс	416132008	16.04.2009	Диван	1 489,00р.	6	8 934,00р.
Сделан 1 заказ общей стоимостью :							8 934,00р.
Сочи	ООО Рога и копыта	327151336	27.03.2009	Кресло	596,00р.	2	1 192,00р.
		417123128	17.04.2009	Рукава шв	57,00р.	12	684,00р.
Сделано 2 заказа общей стоимостью :							1 876,00р.
	ООО Риск	224150038	24.02.2009	Стул	65,80р.	10	658,00р.
		224145847	24.02.2009	Кресло	800,00р.	5	4 000,00р.
		226113015	26.02.2009	Диван	1 500,00р.	2	3 000,00р.
		327151516	27.03.2009	Кресло	850,60р.	6	5 103,60р.
		403120435	03.04.2009	Кресло	1 500,00р.	2	3 000,00р.
		417123210	17.04.2009	Органей	1 200,00р.	6	7 200,00р.
Сделано 6 заказов общей стоимостью :							22 961,60р.
Компаниями города Сочи сделано 8 заказов общей стоимостью :							24 837,60р.
Хабаровск	ЗАО Спектр-ой	417123717	17.04.2009	Стул	589,00р.	12	7 068,00р.
Сделан 1 заказ общей стоимостью :							7 068,00р.
Компаниями города Хабаровск сделан 1 заказ общей стоимостью :							7 068,00р.
Чита	ООО Газсервис	417123708	17.04.2009	Органей	1 200,00р.	7	8 400,00р.
Сделан 1 заказ общей стоимостью :							8 400,00р.
	ООО Петров	416131900	16.04.2009	Диван	1 580,00р.	3	4 740,00р.
		417123225	17.04.2009	Стул	589,00р.	2	1 178,00р.
Сделано 2 заказа общей стоимостью :							5 918,00р.
Компаниями города Чита сделано 3 заказа общей стоимостью :							14 318,00р.
Всего сделано 22 заказов общей стоимостью :							77 623,20р.

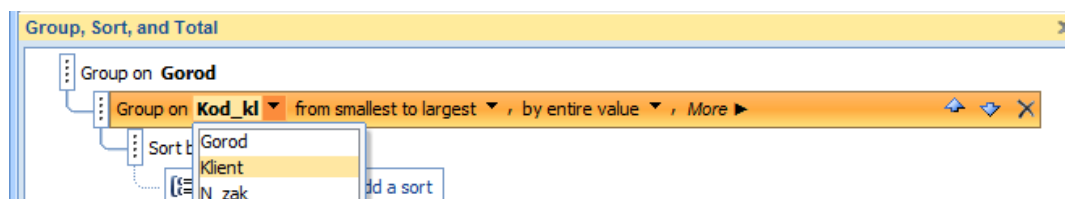
## Шаг 2

Придадим «Отчету по заказам» законченный вид. Для этого открываем отчет в режиме конструктора (Design View) и носим следующие изменения:

- полю *Номер заказа* задаем формат (Format) - «Фиксированный»(Fixed), и Число десятичных знаков (Decimal Places) - 0;
- в область данных добавим вычисляемое поле (Text Box) «Стоимость заказа», в качестве источника данных (Control Source) вводим выражение “= [Cena\_prod] \* [Kol\_vo]” (непосредственно с клавиатуры или используем построитель выражений [...]), указываем «Денежный» (Currency) формат для этого поля.

## Шаг 3

В окне «Группировка, сортировка и итоги» (Group, Sort, and Total) осуществляем замену полей, содержащих коды на поля *Gorod* и *Klient* соответственно.



Вызов окна «Группировка, сортировка и итоги» (Group, Sort, and Total) осуществляется нажатием кнопки «Группировка и сортировка» (Group & Sort) на вкладке **Инструменты конструктора отчетов / Конструктор** (Report Design Tools/Design) в разделе **Группировка и итоги** (Grouping & Totals) ленты или кнопки **Sorting and Grouping** контекстного меню, когда отчет открыт в режиме конструктора (Design View)).

Так же включаем отображение *Заголовка* и *Примечания* группы для этих полей. Для этого раскрываем отображение всех параметров группы, щелкнув **Больше** (More) в окне *Группировка, сортировка и итоги* (Group, Sort, and Total), далее выбрав **с разделом заголовка** (with a header section) и **с разделом примечания** (with a footer section).



## Шаг 4

В область **примечания групп** *Gorod* и *Klient* (Gorod Footer и Klient Footer соответственно) добавляем по два поля (Text Box), в первом будем отображать информацию о количестве сделанных заказов, во втором общую стоимость заказов, соответственно по всем компаниям указанного города и по всем заказам конкретной компании.

Аналогично, в области **примечания отчета** (Report Footer) будем выводить информацию об общем числе заказов и их общей стоимости.

Например, для подсчета количества заказов можно присвоить следующее выражение свойству **Данные** (Control Source):

```
=IIf(Count(*)=1;"Сделан 1 заказ"; ("Сделано заказов " &  
Count(*))) & " общей стоимостью :"
```

а, для подсчета суммарной стоимости заказов в группе:  
$$=Sum([Cena\_prod] * [Kol\_vo])$$

## Шаг 5

Привязываем к кнопке «Отчет по заказам с множественной группировкой» расположенной на Главной форме вызов отчета «Отчет по заказам».

## Шаг 6

Для придания приложению БД законченного вида требуется произвести настройку параметров запуска базы данных.

В окне **Параметры Access** (Access Options), на вкладке **Текущая база данных** (Current Database) следует указать:

в разделе **Параметры приложений** (Application Options)

**Форма просмотра** (Display Form) – *Главная форма*, которая будет появляться при открытии БД;

в разделе **Переходы** (Navigation)

убрать флажок **Область переходов** (Display Navigation Pane), для скрытия панели **Область переходов**.

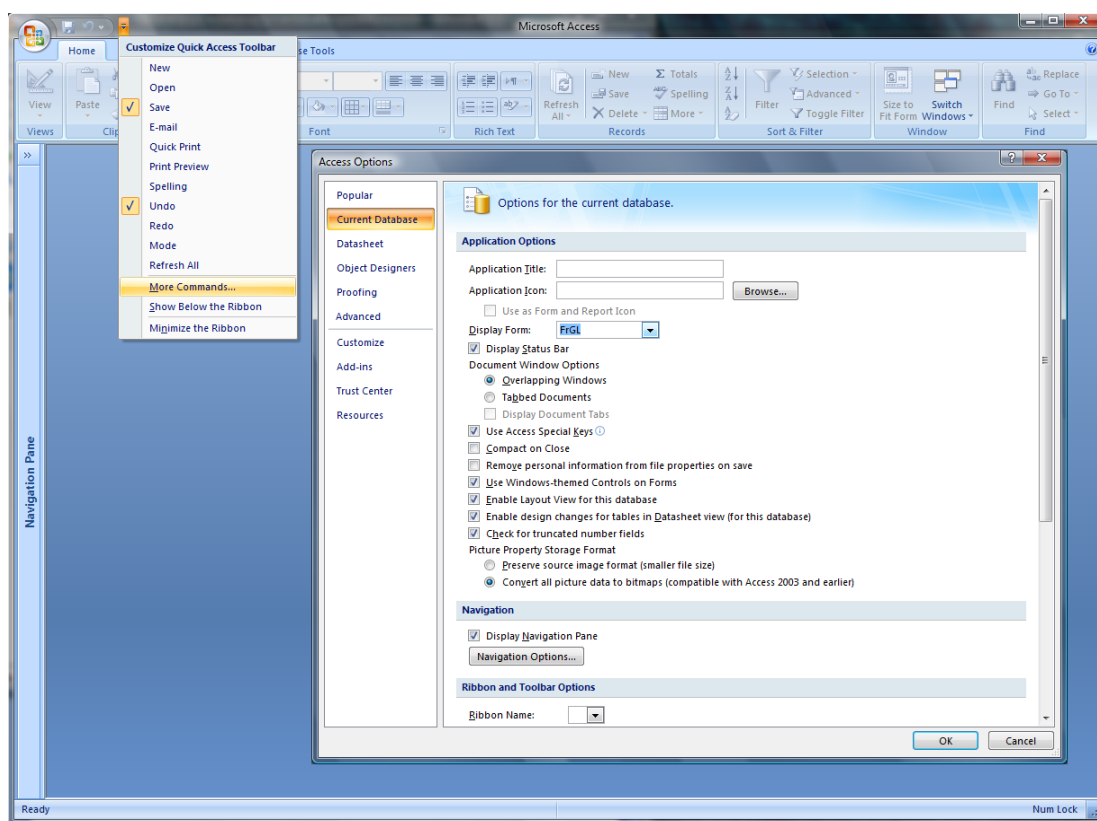
в разделе **Параметры ленты и панелей инструментов** (Ribbon and Toolbar Options) убрать флажки

**Полный набор меню Access** (Allow Full Menus)

**Контекстные меню по умолчанию** (Allow Default Shortcut Menus)

**Встроенные панели инструментов** (Allow Built-in Toolbars)

Вызов окна **Параметры Access** (Access Options) осуществляется путем выбора строки **Другие команды** (More Commands) списка команд раскрывающегося при нажатии на кнопку **Настройка панели быстрого доступа** (Customize Quick Access Toolbar) расположенной на **пользовательской панели инструментов быстрого доступа**



(Customize Quick Access Toolbar).

Теперь, для того чтобы открыть файл базы данных с возможностью внесения в него изменений, необходимо при его открытии удерживать нажатой клавишу <Shift>

### **Список литературы**

1. Харитонов И., Рудикова Л. Microsoft Office Access 2007 (+ CD-ROM). Серия В подлиннике. – БХВ-Петербург, 2008, 1280с
2. Бекаревич Ю., Пушкина Н. Самоучитель Access 2007(+ CD-ROM). – СПб.: БХВ-Петербург, 2007, 700с.
3. Сеннов А. Access 2007 учебный курс(+ CD-ROM). – СПб.: Питер, 2007, 266с;
4. Кренке Д. Теория и практика построения баз данных; учебное пособие. СПб.; Питер; 2003, 799 с., 64,5 п.л.;ил.
5. Кен Гетц, Пол Литвин, Майк Гилберт. Access 2000. Руководство разработчика Том 1 Настольные приложения: Пер. с англ. – К.: Издательская группа BHV, 2000, 1264с.

## СОДЕРЖАНИЕ

<b>Введение .....</b>	<b>3</b>
<b>Лабораторная работа №2 .....</b>	<b>9</b>
<b>Лабораторная работа №3 .....</b>	<b>13</b>
<b>Лабораторная работа №4 .....</b>	<b>15</b>
<b>Лабораторная работа №5 .....</b>	<b>20</b>
<b>Лабораторная работа №6 .....</b>	<b>23</b>
<b>Лабораторная работа №7 .....</b>	<b>28</b>
<b>Лабораторная работа №8 .....</b>	<b>31</b>
<b>Лабораторная работа №9 .....</b>	<b>33</b>
<b>Список литературы.....</b>	<b>36</b>