



ДНЕПРОПЕТРОВСКИЙ УНИВЕРСИТЕТ  
ИМЕНИ АЛЬФРЕДА НОБЕЛЯ



---

# ИНФОРМАТИКА

**СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ**

---

ЛАБОРАТОРНЫЙ ПРАКТИКУМ



ДНЕПРОПЕТРОВСКИЙ УНИВЕРСИТЕТ  
имени АЛЬФРЕДА НОБЕЛЯ

# ИНФОРМАТИКА

Раздел *«СИСТЕМЫ УПРАВЛЕНИЯ  
БАЗАМИ ДАННЫХ»*

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Днепропетровск  
2014

УДК 004.65  
ББК 32.81  
И 74

*Рецензент:*

В.Е. Момот, доктор экономических наук, профессор,  
заведующий кафедрой инновационного менеджмента  
и международной логистики.

*Составители:*

Н.О. Ризун, канд. техн. наук, доц.,  
Т.В. Чумак, ст. преподаватель,  
А.И. Пасько, ст. преподаватель,  
Л.И. Ярмоленко, ст. преподаватель,  
М.А. Калинчук, ст. преподаватель.

Лабораторный практикум з дисципліни «Інформатика» (розділ «Системи управління базами даних») призначений для студентів економічних спеціальностей і передбачає набуття практичних навичок роботи із системою управління базами даних реляційного типу MS Access (модуль 3).

Інформатика. Раздел «Системы управления базами данных»:  
И 74 лабораторный практикум / Н.О. Ризун, Т.В. Чумак, А.И. Пасько  
и др. — Днепропетровск: Днепропетровский университет имени  
Альфреда Нобеля, 2014. — 92 с.

ISBN 978-966-434-291-6

Лабораторный практикум по дисциплине «Информатика» (раздел «Системы управления базами данных») предназначен для студентов экономических специальностей и предполагает получение практических навыков работы с системой управления базами данных реляционного типа MS Access (модуль 3).

УДК 004.65  
ББК 32.81

*Ответственный за выпуск:*

С.Б. Вакарчук, д-р физ.-мат. наук, профессор,  
заведующий кафедрой информатики и математических методов  
в экономике.

© Н.О. Ризун, Т.В. Чумак, А.И. Пасько,  
Л.И. Ярмоленко, М.А. Калинчук,  
составление, 2014

© Днепропетровский университет  
имени Альфреда Нобеля,  
оформление, 2014

ISBN 978-966-434-291-6

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ.....</b>	<b>4</b>
<b>Лабораторная работа № 1. СОЗДАНИЕ БАЗ ДАННЫХ MS ACCESS. ФОРМАТИРОВАНИЕ ТАБЛИЦ.....</b>	<b>5</b>
<b>I. СОЗДАНИЕ И ФОРМАТИРОВАНИЕ ТАБЛИЦ В РЕЖИМЕ МАСТЕРА ТАБЛИЦ И РЕЖИМЕ ТАБЛИЦЫ.....</b>	<b>5</b>
<b>II. СОЗДАНИЕ И ФОРМАТИРОВАНИЕ ТАБЛИЦ В РЕЖИМЕ КОНСТРУКТОРА.....</b>	<b>7</b>
<b>Лабораторная работа № 2. ПОИСК И ФИЛЬТРАЦИЯ ДАННЫХ. СОЗДАНИЕ ЗАПРОСОВ НА ВЫБОРКУ В MS ACCESS .....</b>	<b>17</b>
<b>I. ПОИСК И ФИЛЬТРАЦИЯ ДАННЫХ В РЕЖИМЕ ТАБЛИЦЫ.....</b>	<b>17</b>
<b>II. СОЗДАНИЕ ЗАПРОСОВ НА ВЫБОРКУ .....</b>	<b>22</b>
<b>Лабораторная работа № 3. ИСПОЛЬЗОВАНИЕ ФОРМ ДЛЯ ВВОДА И РЕДАКТИРОВАНИЯ ДАННЫХ В MS ACCESS.....</b>	<b>33</b>
<b>Лабораторная работа № 4. ПРОЕКТИРОВАНИЕ МНОГОТАБЛИЧНЫХ БАЗ ДАННЫХ В MS ACCESS. ТЕОРИЯ НОРМАЛИЗАЦИИ.....</b>	<b>39</b>
<b>Лабораторная работа № 5. СОЗДАНИЕ ЗАПРОСОВ К МНОГОТАБЛИЧНОЙ БАЗЕ ДАННЫХ В MS ACCESS.....</b>	<b>53</b>
<b>I. СОЗДАНИЕ МНОГОТАБЛИЧНЫХ, ИТОГОВЫХ И ПАРАМЕТРИЧЕСКИХ ЗАПРОСОВ .....</b>	<b>53</b>
<b>II. СОЗДАНИЕ ИТОГОВЫХ И ПАРАМЕТРИЧЕСКИХ ЗАПРОСОВ .....</b>	<b>56</b>
<b>III. СОЗДАНИЕ ПЕРЕКРЕСТНЫХ ЗАПРОСОВ .....</b>	<b>61</b>
<b>IV. СОЗДАНИЕ МОДИФИЦИРУЮЩИХ ЗАПРОСОВ.....</b>	<b>64</b>
<b>Лабораторная работа № 6. ОТОБРАЖЕНИЕ ИСХОДНЫХ ДАННЫХ И РЕЗУЛЬТАТОВ В MS ACCESS. ГЛАВНАЯ КНОПОЧНАЯ ФОРМА.....</b>	<b>74</b>
<b>I. СОЗДАНИЕ ИЕРАРХИЧЕСКИХ ФОРМ.....</b>	<b>74</b>
<b>II. СОЗДАНИЕ ОТЧЕТОВ.....</b>	<b>77</b>
<b>III. СОЗДАНИЕ ГЛАВНОЙ КНОПОЧНОЙ ФОРМЫ .....</b>	<b>83</b>

## ВВЕДЕНИЕ

В рамках дисциплины "Информатика" (раздел "Системы управления базами данных") студенты изучают одну из самых мощных и популярных в настоящее время систем управления базами данных (СУБД) реляционного типа – MS ACCESS.

СУБД MS ACCESS входит в состав пакета Microsoft Office и фактически стала промышленным стандартом профессиональной работы по созданию баз данных реляционного типа. Для взаимодействия пользователя с данными, хранящимися в базе, СУБД MS ACCESS содержит:

- набор средств поддержки таблиц и отношений между связанными таблицами;
- развитый пользовательский интерфейс, позволяющий вводить и модифицировать информацию, выполнять поиск и представлять информацию в текстовом или графическом виде;
- средства программирования высокого уровня, с помощью которых можно создавать собственные приложения.

Целью данного лабораторного практикума является получение теоретических и практических навыков самостоятельной работы в среде MS ACCESS при решении реальных экономических задач. При изучении функциональных возможностей MS ACCESS выделены следующие этапы:

- создание и организация работы однотабличных и многотабличных баз данных;
- использование форм для ввода и редактирования данных;
- создание итоговых, перекрестных, параметрических и модифицирующих запросов;
- формирование отчетов;
- автоматизация работы базы данных с помощью главной кнопочной формы.

# Лабораторная работа № 1. СОЗДАНИЕ БАЗ ДАННЫХ MS ACCESS. ФОРМАТИРОВАНИЕ ТАБЛИЦ

## I. СОЗДАНИЕ И ФОРМАТИРОВАНИЕ ТАБЛИЦ В РЕЖИМЕ МАСТЕРА ТАБЛИЦ И РЕЖИМЕ ТАБЛИЦЫ



База данных – это набор взаимосвязанных таблиц, содержащих информацию представленную в виде совокупности строк (записей) и столбцов (полей). Таблицы являются основой базы данных, поскольку в них хранится вся исходная информация.

Процесс создания таблицы состоит в задании ее полей и назначении их свойств. Существует несколько способов создания новой таблицы.

- *Мастер таблиц.* Это программа, ускоряющая создание структуры таблицы. Мастер задает ряд вопросов и, руководствуясь полученными ответами, создает структуру таблицы автоматически.
- *Режим таблицы.* Открывает заготовку, в которой все поля имеют имена по умолчанию: Поле1, Поле2... и т. д. и один стандартный текстовый тип. Таковую таблицу можно сразу наполнять информацией.
- *Конструктор.* В этом режиме можно самостоятельно задать имена полей, выбрать их тип и настроить свойства.

Для того чтобы вставить новое поле в *Режиме таблицы*, необходимо установить указатель мыши на маркер поля и выполнить команду *Вставка/Столбец*. Для удаления поля, его необходимо выделить и выполнить команду *Правка/Удалить столбец*. Для изменения свойств полей в *Режиме таблицы* необходимо перейти в *режим Конструктора (Вид/Конструктор)*. Закончив создание структуры, можно щелкнуть на кнопке *Вид* и перейти в *Режим таблицы* для заполнения ее данными.

## **Задания:**

1. Запустить программу MS Access.

2. Создать базу данных **КОНТАКТЫ** с помощью **Мастера баз данных** (команда меню **Файл / Создать / Общие шаблоны / вкладка Базы данных / Контакты**, далее следуйте указаниям **Мастера баз данных**).

3. Создать пустую базу данных **ТЕЛЕФОННО-АДРЕСНАЯ КНИГА** (команда меню **Файл / Создать / Новая база данных**).

4. С помощью **Режима таблиц** (вкладка **Таблицы / Создание таблицы путем ввода данных** или кнопка **Создать / Режим таблицы**) создать таблицу с именем **Адресная\_Книга**, которая состоит из следующих полей: *Фамилия, Имя, Отчество, РабочийТелефон, МобТелефон, Индекс, Страна, Область, Город, Адрес, ДатаРождения, Предприятие, Кредит, Примечание*.

5. Заполнить таблицу **Адресная\_Книга** не менее чем 15 записями так, чтобы:

- среди её абонентов были граждане разных стран (в том числе россияне) и жители разных городов (в том числе *Киева* и *Москвы*);
- несколько человек работали в *Университете имени Альфреда Нобеля, Днепропетровск*;
- были сотрудники предприятия *Айрит*, работающие в филиалах *Киева* и *Москвы*;
- были клиенты операторов мобильной связи *Киевстар* и *МТС*;
- были именинники, родившиеся в *марте*.

6. С помощью **Мастера таблицы** (вкладка **Таблицы / Создание таблицы с помощью мастера** или кнопка **Создать / Мастер таблиц**) создать таблицу с именем **Клиенты**, которая состоит из следующих полей: *КодКлиента, Фамилия, Имя, Отчество, Телефон, Индекс, Страна, Область, Город, Адрес, Предприятие, Кредит, Примечание*.

7. Выполнить следующие модификации структуры таблиц:

- добавить поле *Руководитель* после поля *Предприятие* и поле *Факс* после поля *РабочийТелефон* (таблица *Адресная\_Книга*);
- удалить поле *Кредит* (таблица *Клиенты*);
- поле *РабочийТелефон* поместить перед полем *Примечание* (таблица *Адресная\_Книга*).

## II. СОЗДАНИЕ И ФОРМАТИРОВАНИЕ ТАБЛИЦ В РЕЖИМЕ КОНСТРУКТОРА



Создание таблиц в окне *Конструктора* предоставляет наиболее широкие возможности по определению параметров создаваемой таблицы. При выборе *режима Создать/Конструктор* появится окно конструктора таблицы:

Имя поля	Тип данных	Описание
▶ КодКлиента	Текстовый	
	Текстовый	
	Поле МЕМО	
	Числовой	
	Дата/время	
	Денежный	
	Счетчик	
	Логический	
	Поле объекта OLE	
	Гиперссылка	
	Мастер подстано	

Свойства поля	
Общие	Подстановка
Размер поля	50
Формат поля	
Маска ввода	
Подпись	
Значение по умолчанию	
Условие на значение	
Сообщение об ошибке	
Обязательное поле	Нет
Пустые строки	Да
Индексированное поле	Нет
Сжатие Юникод	Да
Режим ИМЕ	Нет контроля
Режим предложений ИМЕ	Нет
Смарт-теги	

Тип данных определяет значения, которые можно сохр

*Имя поля* – может содержать до 64 символов, но не может начинаться с пробела. Два поля в одной таблице не могут иметь одинаковые имена.

*Тип данных:*

*Текстовый* – содержит до 255 символов по умолчанию.



*Числовой* – при вводе числового поля автоматически выполняется проверка на соответствие числовому типу; возможно выполнение математических операций.

*Размер поля* – используя значения Свойства *Размер поля*, можно установить следующий формат для полей *числового* типа:

<i>Значение</i>	<i>Описание</i>	<i>Число десятичных знаков</i>	<i>Размер</i>
Байт	Числа от 0 до 255	отсутств.	1 байт
Целое	Числа от - 32768 до 32768	отсутств.	2 байта
Дл. целое	(значение по умолчанию)	отсутств.	4 байта
Одинарное с пл. точкой (4 байт)		7	4 байта
Двойное с пл. точкой (8 байт)		15	8 байт

*Денежный* – по умолчанию принимается с двумя знаками после запятой, число знаков после запятой при желании можно изменить (от 1 до 4 знаков).

*Счетчик* – предназначен для хранения данных, значения которых не редактируются, а устанавливаются автоматически при добавлении каждой новой записи в таблицу. Их значения увеличиваются на 1 при добавлении каждой новой записи или могут быть любым случайным числом.

*Дата/время* – используется для хранения дат и времени полей, в которых данные предоставлены в специальном формате, предоставляет ряд дополнительных преимуществ: возможна установка различных *Форматов* поля в окне Свойств поля: 19 июля 2005г.; 19-июл-05; 19.06.05; при вводе даты будет выполняться проверка *корректности* данных.

*Логический* – используется для хранения данных, которые могут принимать одно из двух возможных значений. Свойство *Формат поля* позволит использовать один из встроенных форматов: Истина/Ложь, Да/Нет, Вкл/Выкл.

*Поле MEMO* – может содержать те же типы данных, что и простые текстовые поля. Отличием является то, что размер поля MEMO не ограничен 255 символами, а может содержать до 65535 символов.

*Поле объекта OLE* – MS Access позволяет хранить в таблицах изображения и другие двоичные рисунки (например, электронную таблицу MS EXCEL, документы MS Word, рисунок, звукозапись).

*Мастер подстановок* – выбор этого типа данных запускает Мастер подстановок, предназначенный для создания поля, в котором предлагается выбор значений из раскрывающегося списка, содержащего набор постоянных значений или значений из другой таблицы.

### **Свойства полей**

*Размер поля* – задает максимальный размер данных, для хранения которых предназначено данное поле. Это свойство зависит от типа данных.

*Число десятичных знаков* – используется для числовых полей, размер которых – одинарное или двойное с плавающей точкой.

*Формат поля* – позволяет определить формат представления данных на экране или на печате. Это свойство зависит от типа данных.

*Маска ввода* – для текстового, числового и денежного типов данных, а также для типа *Дата/Время* можно задать маску ввода, которую пользователь увидит на экране при вводе данных. Список шаблонов масок задан, шаблоны можно редактировать. Символ шаблона – это замещающий символ, появляющийся там, где будет вводиться текст.

*Подпись поля* – используется для того, чтобы в таблицах, отчетах, формах, имя поля отличалось от исходных (например, было более содержательным).

*Значение по умолчанию* – позволяет задать значение, которое появляется в текущем поле при добавлении каждой новой записи.

<b>Имя поля</b>	<b>Пример Значения по умолчанию</b>
Город	Днепропетровск
Страна	Украина

ДатаЗаказа	Date()
------------	--------

*Условие на значение* – позволяет ограничить ввод значений при заполнении данных в полях таблицы.

Имя поля	Пример Условия на значение
Кредит	$\geq 1000$ And $\leq 5000$ Between 1000 and 5000
ДатаРождения	$(\text{Year}(\text{Date}()) - \text{Year}([\text{ДатаРождения}])) \geq 18$
ДатаЗаказа	$\leq \text{Date}()$

*Сообщение об ошибке* – позволяет комментировать ошибки в случае нарушения *Условия на значение*.

Имя поля	Пример Сообщения об ошибке
Кредит	Кредит должен быть не менее 1000 у. е., но не более 5000 у. е.
ДатаРождения	Возраст сотрудника должен быть не менее 18 лет
ДатаЗаказа	Дата заказа не должна превышать текущую дату

*Обязательное поле* – если в данном поле не может быть 0 или NULL, то выбрать ДА.

*Индексированное поле* – совпадения не допускаются.

### Задания:

1. Запустить программу MS Access.
2. Открыть базу данных **ТЕЛЕФОННО-АДРЕСНАЯ КНИГА**.
3. С помощью окна **Конструктор** (кнопка **Конструктор** окна базы данных) проверить типы данных полей таблицы **Адресная\_Книга**. Обратить особое внимание на типы данных следующих полей: *ДатаРождения*, *Кредит*, *Примечание*. Откорректировать типы данных для полей, в которых отсутствует соответствие между значениями вводимых данных и их типом (раскрывающийся список столбца **Тип данных** в окне **Конструктор**).
4. Для полей, имена которых состоят из нескольких слов, добавить свойство **Подпись**, содержащее имя текущего поля с пробелом между словами

(вкладка **Поля / группа Свойства**). Например, для поля *РабочийТелефон* свойство **Подпись** должно иметь значение *Рабочий Телефон*;

5. Для облегчения ввода данных в таблицы:

– установить Маски ввода для полей *РабочийТелефон*, *Факс* и *Индекс*;

– для поля *Страна* ввести Значение по умолчанию – *Украина*, а для поля *Город* – *Днепропетровск*.

6. Организовать контроль данных при вводе их значений в таблицы. Для этого:

– в свойствах поля *Кредит* установить Условие на значение, полагая, что данные этого поля не могут быть отрицательными;

– в свойствах поля *ДатаРождения* установить контроль на принадлежность введенных данных некоторому разумному диапазону дат (Условие на значение), полагая, что даты рождения должны быть в интервале от 1.01.1940 г. до текущей даты;

– для всех заданных условий на значение сформулировать текст Сообщения об ошибке и ввести его в свойствах соответствующих полей.

7. Создать новую базу данных **ЗАКАЗЫ\_КЛИЕНТОВ** (команда меню **Файл / Создать / Новая база данных**).

8. В новой базе данных с помощью Конструктора (вкладка **Таблицы / Создание таблицы в режиме конструктора** или кнопка **Создать / Конструктор**) создать таблицу с именем **Заказы**, которая имеет следующую структуру:

### Структура таблицы Заказы

Имя поля	Тип данных	Описание
<i>КодКлиента</i>	Текстовый	Уникальное значение для каждого клиента
<i>КодТовара</i>	Текстовый	Уникальное значение для каждого товара

<i>ДатаЗаказа</i>	Дата/Время	
<i>Заказано</i>	Числовой	Количество единиц товара
<i>ДатаПродажи</i>	Дата/Время	
<i>Продано</i>	Числовой	Количество единиц товара
<i>Примечание</i>	Поле МЕМО	Дополнительная информация о заказе

9. Для всех полей, имена которых состоят из нескольких слов, добавить свойство **Подпись**, в тексте которого должно содержаться имя текущего поля с пробелом между словами.

10. Организовать контроль данных при вводе их значений в таблицу **Заказы**. Для этого:

- в свойствах поля *КодКлиента* создать Маску ввода, состоящую из *двух* обязательных и *одной* необязательной буквы, расположенных в начале кода (буквы должны автоматически переводиться в верхний регистр), и *пяти* обязательных цифр;

- в свойствах поля *КодТовара* создать Маску ввода, которая будет начинаться с буквы *T* и пробела и состоять из *двух* обязательных цифр и *двух* обязательных букв, разделенных знаком "-";

- в свойствах полей *Заказано* и *Продано* установить Условие на значение полагая, что данные этого поля не могут быть отрицательными и не могут принимать значения больше 1000000;

- в свойствах поля *ДатаЗаказа* и *ДатаПродажи* установить контроль введенных данных (Условие на значение) на принадлежность к некоторому разумному диапазону дат, полагая, что даты должны быть в интервале от 1.01.2010 г. до текущей даты;

- для всех заданных условий на значение сформулировать текст Сообщения об ошибке и ввести его в свойствах соответствующих полей.

11. Установить для числовых полей таблицы необходимое количество знаков после запятой, используя при этом **Свойства поля**: Размер поля, Формат поля, Число десятичных знаков.

12. В **Режиме таблицы** просмотреть Описание полей таблицы **Заказы** в *Строке состояния* окна Access.

13. Заполнить Условие на значение для всей таблицы (кнопка **Свойства** на панели инструментов окна **Конструктора**), обеспечив контроль данных по следующим условиям:

– значение поля **ДатаПродажи** не должно быть раньше значения поля **ДатаЗаказа**;

– значение поля **Продано**, не должно превышать, значение поля **Заказано**.

14. Сохранить структуру таблицы **Заказы** в базе данных.

15. Проверить работу созданных Свойств полей путем ввода значений в одну запись таблицы.

16. Удалить созданную запись.

### *Справочный материал к лабораторной работе № 1*

Специальные **числовые форматы** определяются с помощью следующих символов.

#### **Символ    Описание**

- Десятичный разделитель. Символ десятичного разделителя выбирается в окне **Язык и стандарты** панели управления Windows.
- , Разделитель групп разрядов.
- 0** Прототип разряда. Выводится цифра или 0.
- #** Прототип разряда. Выводится цифра или ничего не выводится.
- \$** Выводится символ \$.
- %** Процентный формат. Число умножается на 100 и к нему добавляется символ процентов.

Значение свойства **маска ввода** может содержать до трех разделов, разделяемых точкой с запятой (;).

**Первый** Представляет саму Маску ввода (например, !(999) 000-0000). Перечень символов, используемых для определения масок ввода, приводится ниже в таблице.

**Второй** Определяет режим занесения в таблицу строковых констант, добавляемых к символам, вводимым пользователем. Введенный в данный компонент символ 0 указывает, что постоянные символы (например, скобки и дефисы в маске ввода телефонных номеров) сохраняются вместе с введенными пользователем символами; значение 1 или пустое значение данного компонента указывает, что сохраняются только символы, введенные пользователем.

**Третий** Определяет символ, используемый для изображения пустых позиций в Маске ввода, в которые помещаются вводимые пользователем символы. В этом компоненте можно указать любой символ ANSI; пробел необходимо заключить в кавычки (" ").

Символы, которые следует вводить в **маску ввода**:

<b>Символ</b>	<b>Описание</b>
<b>0</b>	Цифра (обязательный символ; знаки (+) и (-) не разрешены).
<b>9</b>	Цифра или пробел (необязательный символ; знаки (+) и (-) не разрешены).
<b>#</b>	Цифра или пробел (необязательный символ; незаполненные позиции выводятся как пробелы в режиме редактирования, но удаляются при сохранении данных; знаки (+) и (-) разрешены).
<b>L</b>	Буква (обязательный символ).
<b>?</b>	Буква (необязательный символ).
<b>A</b>	Буква или цифра (обязательный символ).
<b>a</b>	Буква или цифра (необязательный символ).
<b>&amp;</b>	Любой символ или пробел (обязательный символ).

**С** Любой символ или пробел (необязательный символ).

. , : ; - / Десятичный разделитель, разделители групп разрядов, времени или даты (используемые символы разделителей определяются настройками, выбранными в окне Язык и стандарты панели управления Windows).

< Преобразует все символы к нижнему регистру.

> Преобразует все символы к верхнему регистру.

! Указывает, что маска ввода заполняется справа налево; этот символ следует использовать, если в левой части маски находятся позиции, заполнять которые не обязательно. Маски ввода обычно заполняются слева направо. Символ восклицательного знака можно помещать в произвольную позицию в маске ввода.

\ Указывает, что следующий символ следует воспринимать как постоянный (а не специальный) символ (например, \A представляет символ "А").

В следующей таблице приводятся **примеры** часто используемых масок ввода и образцы значений, соответствующих этим маскам.

Пример Маски ввода	Пример значений, соответствующих данной Маске
(000) 000-0000	(206) 555-0248
(999) 999-9999!	(206) 555-0248 ( ) 555-0248
(000) AAA-AAAA	(206) 555-TELE
>L0L 0L0	T2Ф 8M4
00000-9999	98115- 98115-3007
>L<??????????????	Кирилл Ирина



ISBN 0-#####-0	ISBN 1-55615-507-7 ISBN 0-13-964262-5
>LL00000-	DB51392-0493

Если пользователь определяет **Маску ввода** и одновременно задает для того же поля значение свойства **Формат поля**, то при выводе данных приоритет имеет свойство **Формат поля**. Это означает, что при форматировании данных сохраненная **Маска** ввода игнорируется. Данные в базовой таблице при этом не изменяются; свойство **Формат поля** определяет только режим отображения данных.



## ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И КОНТРОЛЯ ЗНАНИЙ

1. В каком случае для создания структуры таблицы целесообразно использовать *Мастер таблиц*?
2. Каким образом *Мастер таблиц* определяет типы полей новой таблицы?
3. Определите назначение кнопок выбора полей *Мастера таблиц*.
4. Как определяется тип данных полей при создании таблицы в *Режиме таблицы*?
5. Укажите правила образования имен полей таблицы.
6. Как в *Режиме таблицы* можно изменить наименование поля? Продемонстрируйте процедуру изменение имени поля.
7. Перечислите возможные варианты добавления новых полей в таблицу в *Режиме таблицы*.
8. Как осуществляется удаление поля таблицы в *Режиме таблицы*?
9. Что нужно сделать, чтобы переместить поле таблицы с одного места на другое?

10. Какие типы данных являются допустимыми в Access?

11. Как в окне *Конструктора* таблиц устанавливается тип данных для полей таблицы?

12. Для чего предназначен атрибут *Описание*? Что произойдет, если не указывать значений этого атрибута при создании структуры таблицы в окне конструктора?

13. Для чего предназначена *Маска ввода*, где она задается?

14. Перечислите основные символы, используемые в *Маске ввода*, охарактеризуйте их назначение.

15. Для чего предназначено свойство поля *Значение по умолчанию*? Приведите пример использования этого свойства.

16. Как связаны между собой такие свойства поля, как *Условие на значение* и *Сообщение об ошибке*? Приведите примеры использования этих свойств.

## **Лабораторная работа № 2. ПОИСК И ФИЛЬТРАЦИЯ ДАННЫХ. СОЗДАНИЕ ЗАПРОСОВ НА ВЫБОРКУ В MS ACCESS**

### **I. ПОИСК И ФИЛЬТРАЦИЯ ДАННЫХ В РЕЖИМЕ ТАБЛИЦЫ**

#### **Задания:**

1. Запустить программу MS Access.
2. Открыть базу данных **ЗАКАЗЫ\_КЛИЕНТОВ**.
3. Перенести в нее таблицу **Клиенты** из базы данных **ТЕЛЕФОННО-АДРЕСНАЯ КНИГА** (кнопка **Создать** / режим **Импорт таблиц**). Закрыть базу данных **ЗАКАЗЫ\_КЛИЕНТОВ**.
4. Открыть базу данных **ТЕЛЕФОННО-АДРЕСНАЯ КНИГА**.
5. Удалить из нее таблицу **Клиенты**, оставив там только таблицу **Адресная\_Книга**.

6. Открыть таблицу **Адресная\_Книга**.

7. Используя команды меню **Формат** или кнопки панели инструментов **Формат (Режим таблицы)**:

- зафиксировать поля *Фамилия, Имя, Отчество, РабочийТелефон* так, чтобы они всегда были видны на экране (**Формат/Закрепить столбцы**);
- сделать ячейки таблицы *приподнятыми* (**Формат/Режим таблицы**);
- скрыть поле *Примечание* (**Формат/Скрыть столбцы**);
- изменить размер шрифта, цвет букв, фон таблицы.



В MS Access существует множество способов отобразить требуемые данные при выполнении поиска конкретного значения, одной записи или группы записей.

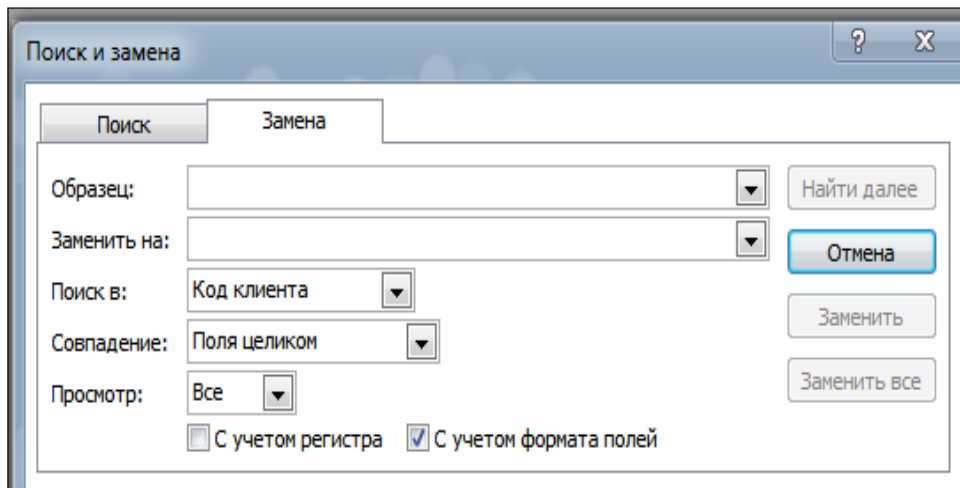
С помощью диалогового

В MS Access существует множество способов отобразить требуемые данные при выполнении поиска конкретного значения,

го окна *Поиск и Замена* вкладка *Поиск* можно найти конкретные записи или определенные значения в полях.

При обнаружении каждого требуемого элемента выполняется перемещение по записям с помощью кнопки *Найти далее*. При этом необходимо задать следующие опции:

- *Образец* – указываются данные, которые нужно найти;
- *Найти далее* – позволяет найти первое значение по условию, а также перейти к следующим записям с этим условием;
- *Поиск в:* – определяет место поиска, по умолчанию содержит название выбранного столбца;
- *Совпадение* – содержит три установки, которые определяют тип сравнения с образцом: с любой частью поля; поле целиком; с начала поля;



- *Просмотр* – определяет направление поиска: Все, Вверх, Вниз;
- флажок *С учетом регистра* – позволяет при поиске учитывать начертание букв (прописные или строчные);
- флажок *С учетом формата полей* – позволяет выполнять поиск данных в указанном формате отображения.

Для того чтобы выполнить не только поиск, но и изменение данных в найденном поле, необходимо в том же диалоговом окне *Поиск и Замена* открыть вкладку *Замена* и в поле *Заменить на:* указать значение, на которое нужно заменить искомое значение. После этого нажать кнопку *Заменить* или *Заменить все*.

8. Найти данные об абонентах (кнопка **Найти**):

- работающих на предприятии *Айрит*;
- у которых день рождения сегодня;
- у которых в этом году юбилей (возраст, кратный 10).

9. Сформировать различные варианты критерия поиска для определения номера *РабочегоТелефона* и найти в таблице абонента, в шестизначном номере телефона которого Вы помните:

- первые две цифры;
- четвертую и последнюю цифры.

10. С помощью команды **Заменить** (пункт меню **Правка / Заменить**) выполнить следующую корректировку данных таблицы:

- изменить значение поля *Фамилия* для абонента – женщины, которая вышла замуж;
- изменить поле *Предприятие* для абонента из *Киева*, который временно остался без работы.



*Фильтры* позволяют временно изолировать часть записей с целью просмотра конкретного набора записей в открытой форме или в любом объекте MS Access, открытом в *Режиме таблицы*. В MS Access предусмотрено четыре способа отбора записей с помощью фильтров: *Изменить фильтр*, *Фильтр по выделенному*, *Исключить выделенное* и *Расширенный фильтр*.

*Изменить фильтр* – используется для отбора записей по значениям нескольких полей.

*Фильтр по выделенному* – осуществляет поиск всех записей, содержащих выделенное значение непосредственно в форме или в таблице.

*Исключить выделенное* – осуществляет поиск всех записей, не содержащих выделенное значение в форме или в таблице.

*Расширенный фильтр* – для создания сложных фильтров создается в окне расширенного фильтра с помощью бланка запроса.

11. С помощью **Фильтра по выделенному** показать данные только по абонентам:

- которые проживают в *Украине*. Отменить результаты фильтрации;
- которые проживают за пределами *Украины*. Отменить результаты фильтрации;
- у которых *Имя* начинается с буквы "*Н*". Отменить результаты фильтрации;
- которые родились в *Марте*. Не отменять результаты фильтрации;
- женщинам. Проанализировать полученный результат.

12. С помощью команды **Изменить фильтр** показать данные только по абонентам:

- из *Днепропетровска* ИЛИ *Киева*. Отменить результаты фильтрации;
- являющимся клиентами оператора мобильной связи *Киевстар*;
- из *Днепропетровска* И являющимся клиентами оператора мобильной связи *МТС*. Отменить результаты фильтрации;
- без *Кредита* И у которых *Фамилия* содержит букву "о". Отменить результаты фильтрации;
- без *Мобильного Телефона* ИЛИ у которых *Фамилия* начинается с букв "А" ИЛИ "В" ИЛИ "С";
- женщинам с суммой *Кредита* больше либо равной 10000 у. е. Отменить результаты фильтрации;
- у которых нет телефона (выражение *Is Null* в строке условия окна **Изменить фильтр** для поля *Телефон*). Отменить результат;
- возраст которых в диапазоне от 20 до 30 лет. Отменить результаты фильтрации;
- которые имеют работу (выражение *Is not Null* в строке условия окна **Изменить фильтр** для поля *Предприятие*). Отменить фильтр;
- проживающим в *Киеве* ИЛИ в *Москве* и работающим на предприятии *Айрит*.

13. По данным своей таблицы **Адресная\_Книга** придумать и реализовать условие поиска с использованием критериев *И* и *ИЛИ*.

14. Отсортировать таблицу так, чтобы в пределах записей об абонентах из одной и той же *Страны* *Фамилии* абонентов были отсортированы по алфавиту.

### ***Справочный материал к лабораторной работе № 2.1***

Для **поиска** значений полей, записей или имен файлов допускается применение следующих символов в диалоговых окнах **Поиск** и **Замена** или в запросах, командах и выражениях.

<b>Знак</b>	<b>Использование</b>	<b>Пример</b>	<b>Найденные значения</b>
*	Соответствует любой цифре	*55*	553-3486

	или любому символу		123-5561
?	Соответствует любому текстовому символу	Снег?рев	“Снегирев” “Снегерев” “Снегарев”
[ ]	Соответствует любому одному символу из тех, что заключены в скобки	Снег[ие]рев	“Снегирев” “Снегерев”
!	Соответствует любому одному символу кроме заключенных в скобки	b[!a]ll	bill bell <u>но не</u> ball
-	Соответствует любому символу из диапазона Необходимо указывать этот диапазон по возрастанию	b[a-c]d	bad bbd bcd
#	Соответствует любой цифре	1#3	103 113 123

## II. СОЗДАНИЕ ЗАПРОСОВ НА ВЫБОРКУ



**Запрос** – это средство выбора необходимой информации из базы данных. Вопрос, сформированный по отношению к базе данных, и есть запрос. Применяются два типа запросов: по образцу (QBE – Query by example) и структурированный язык запросов (SQL – Structured Query Language).

**QBE-запрос по образцу** – средство для отыскания необходимой информации в базе данных. Он формируется путем заполнения бланка запроса в окне *Конструктора* запросов.

**SQL-запросы** – это запросы, которые состоят из последовательности SQL-инструкций. Эти инструкции задают, последовательность действий, которую необходимо выполнить с входным набором данных для генерации выходного набора. Все запросы Access строит на основе SQL-запросов – для того чтобы посмотреть их необходимо в активном окне проектирования запроса выполнить команду *Вид/SQL*.

Существует несколько *Типов запросов*: на выборку, на обновление, на добавление, на удаление, перекрестный запрос, создание таблиц.

Наиболее распространенным является *Запрос на выборку*. Запрос на выборку позволяет сформировать пользовательское представление о данных, не обязательно отвечающее требованиям нормализации.

Результат выполнения запроса на выборку – это новая, чаще всего временная, *таблица*, которая существует до закрытия запроса. Структура такой таблицы определяется выбранными из одной или нескольких взаимосвязанных таблиц полями. Записи формируются путем объединения записей таблиц, участвующих в запросе. Способ объединения записей различных таблиц указывается при определении их связи.

*Условия отбора*, сформулированные в запросе, позволяют фильтровать записи, составляющие результат объединения таблиц. Простейшие запросы могут быть созданы с помощью *Мастера*. Любой запрос можно создать в режиме конструктора. *Конструктор* предоставляет удобное для пользователя диалоговое графическое средство формирования, с помощью которого легко может быть построен сложный запрос.

Запрос может получать данные из одной или нескольких таблиц, из существующих запросов или из комбинаций таблиц и запросов, объединяя данные, содержащиеся в них. Таблицы и запросы, объединенные для получения данных, образуют источник записей для запроса. Запрос, построенный на основе другого запроса, использует его временную таблицу с результатами в качестве источника записей.

### **Создание запроса на выборку с помощью Мастера**

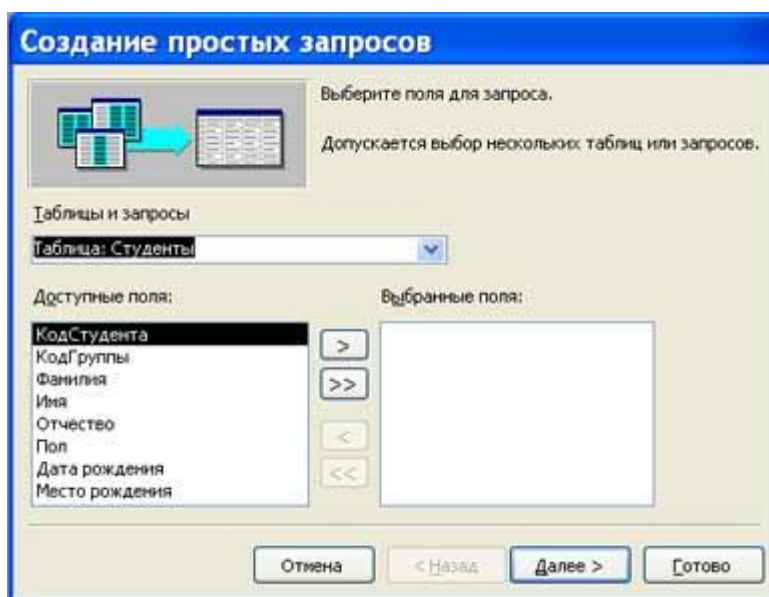
При создании *Запроса* необходимо определить:

- поля в базе данных, по которым будет выполняться поиск информации;
- предмет поиска в базе данных (критерии отбора);
- перечень полей, отображаемых в результате выполнения запроса.



Для создания запроса в окне база данных выбрать вкладку *Запросы* и дважды щелкнуть на пиктограмме *Создать / Простой запрос*.

В окне *Мастера* выбрать необходимую таблицу (или запрос – источник данных) из оп-



ции *Таблицы и запросы* и выбрать поля данных. Если *Запрос* формируется на основе нескольких таблиц, необходимо повторить действия для каждой таблицы – источника.

Затем в окне *Мастера* надо выбрать подробный или итоговый отчет и щелкнуть на кнопке *Далее*. После этого необходимо задать имя *Запроса* и выбрать один из вариантов дальнейшего действия: *Открыть* запрос для просмотра данных или *Изменить* макет запроса и нажать кнопку *Готово*.

### Задания:

1. Открыть базу данных **ТЕЛЕФОННО-АДРЕСНАЯ КНИГА**.

2. С помощью *Мастера запросов* создать следующие запросы (кнопка **Создать / Простой запрос**):

– **Телефоны** – запрос, содержащий Подробную информацию по полям *Фамилия, Имя* и *МобТелефон*;

– **Кредиты** – запрос содержащий Итоговую информацию об общем размере *Кредита* абонентов из каждой *Страны* (использовать поля *Страна, Кредит*).

## Создание запроса на выборку с помощью Конструктора

С помощью *Конструктора* можно создать следующие виды запросов на выборку:

- простой;
- по условию;
- параметрический;
- итоговый;
- с вычисляемыми полями.

Для того, чтобы вызвать *Конструктор* запросов, необходимо во вкладке *Запросы* вызвать *Создать / Конструктор*. Появится активное окно *Добавление таблицы* на фоне неактивного окна "Запрос: запрос на выборку".

В окне *Добавление таблицы* следует выбрать таблицу-источник или несколько таблиц из представленного списка таблиц, на основе которых будет проводиться выбор данных, и щелкнуть на кнопке *Добавить*. После этого закрыть окно *Добавление таблицы*, окно "Запрос: запрос на выборку" станет активным.

Окно *Конструктора* состоит из двух частей – верхней и нижней. В верхней части окна размещается схема данных запроса, которая содержит список таблиц-источников и отражает связь между ними. В нижней части окна находится *Бланк* построения запроса QBE (Query by Example), в котором каждая строка выполняет определенную функцию:

*Поле* – содержит имена полей, которые участвуют в запросе.

*Имя таблицы* – имя таблицы, с которой выбрано это поле.

*Сортировка* – указывает тип сортировки.

*Вывод на экран* – устанавливает флажок просмотра поля на экране.

*Условия отбора* – задаются критерии поиска.

*Или* – задаются дополнительные критерии отбора.

3. С помощью *Конструктора запросов* создать следующие *Запросы на выборку без использования вычисляемых полей* (кнопка **Создать / Конструктор**):

- **Украина** – запрос, содержащий информацию только абонентах из *Украины* (поле *Страна* не выводить на экран);
- **Зарубежные** – запрос, содержащий информацию только о зарубежных абонентах, у которых *нет рабочего телефона*;
- **МТС** – запрос, содержащий информацию о абонентах, которые живут в *Днепропетровске*, являются абонентами *МТС* и не имеют *кредита*;
- **Сложный\_1** – запрос, содержащий информацию об абонентах, которые: работают в *Университете имени Альфреда Нобеля* или *Айрит*; живут за пределами *Днепропетровска*; имеют *кредит* более 10000 у. е.;
- **Сложный\_2** – запрос, содержащий информацию о абонентах, которые работают на предприятиях, названия которых начинаются с букв "А", "В", "С" или "Д"; проживают в *Киеве* или *Днепропетровске*; родились в *апреле* (использовать два варианта создания условия отбора).

### Вычисляемые поля



В запросе для каждой записи могут производиться вычисления с числовыми, строковыми значениями или значениями дат с использованием данных из одного или нескольких полей. Результат вычисления образует в таблице запроса новое *вычисляемое поле*. В исходных таблицах базы данных новых полей не создается. При каждом выполнении запроса производится вычисление на основе текущих значений полей.

В выражениях вычисляемых полей помимо имен полей могут использоваться константы и функции. В результате обработки выражения может получаться только одно значение.

Имя вычисляемого поля становится заголовком столбца в таблице с результатами выполнения запроса. Это имя можно изменить.

### Функции категории Дата/Время

Функция *Date* ( ) – возвращает значение типа Variant (Date), содержащее текущую системную дату.

Функция *Day* (дата) – возвращает значение типа Variant (Integer), содержащее целое число (от 1 до 31 включительно), которое представляет день месяца.

Функция *Month* (дата) – возвращает значение типа Variant (Integer), содержащее целое число (от 0 до 12 включительно), которое представляет месяц в значении даты.

Функция *Year* (дата) – возвращает значение типа Variant (Integer), содержащее целое число, представляющее год.

Функция *DateSerial*(год, месяц, день) – возвращает значение типа Variant (Date) для указанных года, месяца и дня.

Синтаксис функции *DateSerial* включает в себя следующие аргументы:

Аргумент	Описание
<i>год</i>	Обязательный аргумент типа <b>Integer</b> . Число от 100 до 9999 включительно или числовое выражение
<i>месяц</i>	Обязательный аргумент типа <b>Integer</b> . Любое числовое выражение
<i>день</i>	Обязательный аргумент типа <b>Integer</b> . Любое числовое выражение

4. С помощью *Конструктора запросов* создать следующие *Запросы на выборку с использованием вычисляемых полей* (см. справочный материал к лабораторной работе №2):

– **День\_рождения\_1** – запрос, отображающий только абонентов, родившихся в *Октябре* (данное задание выполнить с использованием функции `DateSerial` и без нее);

– **День\_рождения\_2** – запрос, отображающий только абонентов с *днем рождения* в ближайшие 10 дней (без учета возможности перехода на следующий месяц);

– **Возраст** – запрос, позволяющий рассчитать *возраст* абонентов;

– **День\_рождения\_3** – запрос, отображающий только абонентов с *днем рождения* в ближайшие 20 дней (с учетом возможности перехода на следующий месяц).

5. Добавить в таблицу **Адресная\_Книга** данные о новых абонентах и проанализировать изменения в результатах выполнения всех созданных Вами запросов.

6. Сохранить запросы в базе данных.

### *Справочный материал к лабораторной работе № 2.2*

Примеры использования функции **DATESERIAL**:

**Пример 1:**

- **Day (Date())** – текущий день
- **Month (Date())** – текущий месяц
- **Year (Date())** – текущий год

Контакты, <i>ДатаРождения</i> которых приходится на 14 Марта	<b>DateSerial(Year([Адресная_Книга]![ДатаРождения]);3;14)</b>
Контакты, <i>ДатаРождения</i> которых приходится на текущий день	<b>DateSerial(Year([Адресная_Книга]![ДатаРождения]);Month(Date());Day(Date()))</b>
Последний день текущего месяца	<b>DateSerial(Year(Date());Month(Date()) + 1;0)</b>
Последний день следующего месяца	<b>DateSerial(Year(Date());Month(Date()) + 2;0)</b>
Первый день предыдущего месяца	<b>DateSerial(Year(Date());Month(Date())-1;1)</b>

Последний день предыдущего месяца	<b>DateSerial(Year(Date()); Month(Date());0)</b>
-----------------------------------	--

Примеры использования функций категории *DATE/TIME*, для создания запросов:

**Пример 1:**

Создать запрос, позволяющий получить информацию обо всех абонентах, которые родились в первые 5 дней января.

Пример бланка данного запроса с использованием функций **Month()** и **Day()**:

Поле: Имя таблицы: Сортировка: Вывод на экран: Условие отбора:		
Фамилия	Месяц: Month([Адресная_Книга]![ДатаРождения])	День: Day([Адресная_Книга]![ДатаРождения])
Адресная_Книга		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
1		>=1 And <=5

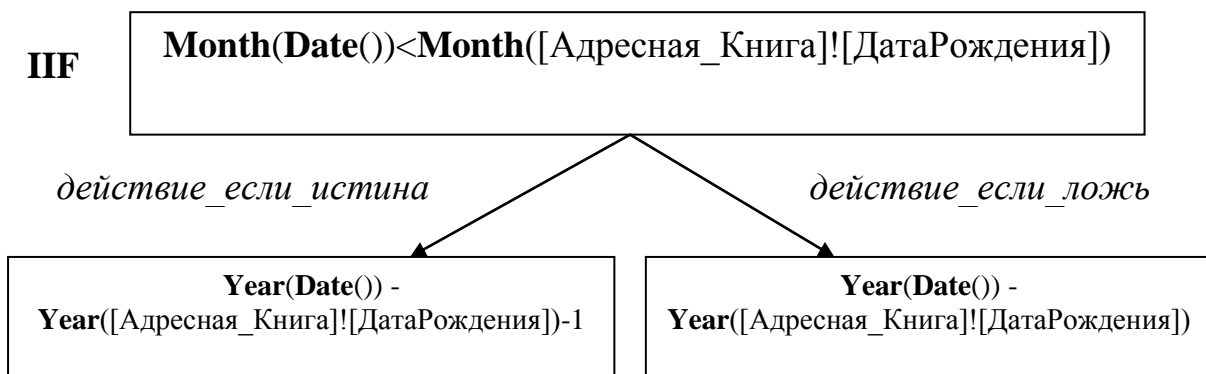
**Пример 2:**

С целью повышения точности определения возраста абонентов **Адресной\_Книги** при создании вычисляемого поля **Возраст** целесообразно использовать функцию:

**ИФ** (логическое выражение; действие\_если\_истина; действие\_если\_ложь).

Данная функция позволит выполнить сравнение текущего месяца с месяцем рождения абонента и в зависимости от результата сравнения определить количество его полных лет со дня рождения:

*Логическое выражение*



Тогда вычисляемое выражение **Возраст** будет иметь вид:

**Возраст: ИФ(Монх(Дате())<Монх([Адресная\_Книга]![ДатаРождения]);  
(Year(Date())-Year([Адресная\_Книга]![ДатаРождения]))-1; Year(Date())-  
Year([Адресная\_Книга]![ДатаРождения])).**

**Пример 3:**

Создать запрос, позволяющий отобразить абонентов, день рождения которых приходится на следующие 10 дней (с учетом возможности перехода на следующий месяц).

Алгоритм реализации данного задания:

**Шаг 1.** Добавить первое вычисляемое поле – для вычисления количества дней, которые остались до конца текущего месяца:

**Осталось: Day(DateSerial(Year(Date());Month(Date())+1;0))-Day(Date())**

**Шаг 2.** Добавить второе и третье вычисляемые поля – для извлечение дня и месяца рождения:

**День: Day([Адресная\_Книга]![ДатаРождения])**

**Месяц: Month([Адресная\_Книга]![ДатаРождения])**

**Шаг 3.** Создать три критерия для первого вычисляемого поля с целью анализа количества дней, которые остались до конца текущего месяца – для случаев, когда значение этого поля больше или равно или меньше 10 (анализируемый диапазон):

Остаток: Day(DateSerial(Year(Date());Month(Date())+1;0))-Day(Date())
<input checked="" type="checkbox"/>
>=10
<10
<10

**Шаг 4.** Создать три группы критериев для второго и третьего вычисляемых полей

➤ **первая** группа критериев – для ситуации, когда количество дней, которые остались до конца текущего месяца, **больше или равно 10**

– позволяющая отобразить все контакты, дата рождения которых в ближайšie 10 дней приходится на текущий месяц:

Остаток: Day(DateSerial(Year(Date());Month(Date))+1;0))-Day(Date())	День: Day([Адресная_Книга]![ДатаРождения])	Месяц: Month([Адресная_Книга]![ДатаРождения])
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
>=10	>=Day(Date()) And <=Day(Date()+10	Month(Date())

➤ **вторая** группа критериев – для ситуации, когда количество дней, которые остались до конца текущего месяца, **меньше 10** – позволяющая отобразить **часть** контактов, дата рождения которых в ближайšie 10 дней приходится на текущий месяц (от текущего дня до конца месяца):

Остаток: Day(DateSerial(Year(Date());Month	День: Day([Адресная_Книга]![ДатаРождения])	Месяц: Month([Адресная_Книга]![ДатаРождения])
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
>=10	>=Day(Date()) And <=Day(Date()+10	Month(Date())
<10	>=Day(Date()) And <=Day(DateSerial(Year(Date());Month(Date))+1;0)	Month(Date())

*от текущего дня до последнего дня текущего месяца*

➤ **третья** группа критериев – для ситуации, когда количество дней, которые остались до конца текущего месяца, **меньше 10** – позволяющая отобразить **часть** контактов, дата рождения которых в ближайšie 10 дней приходится на следующий месяц: от *первого* числа следующего месяца и до даты, соответствующей разнице между заданным количеством дней (10) и количеством дней от текущего дня до последнего дня текущего месяца (поле *Остаток*):

Остаток: Day(DateSerial(Year(Date());Month	День: Day([Адресная_Книга]![ДатаРождения])	Месяц: Month([Адресная_Книга]![ДатаРождения])
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
>=10	>=Day(Date()) And <=Day(Date()+10	Month(Date())
<10	>=Day(Date()) And <=Day(DateSerial(Year(Date());Month(Date))+1;0)	Month(Date())
<10	>=Day(DateSerial(Year(Date());Month(Date))+1;1) And <=Day(DateSerial(Year(Date());Month(Date))+1;1)	Month(Date()+1

**>=Day(DateSerial(Year(Date()),Month(Date))+1,1) And  
<=Day(DateSerial(Year(Date()),Month(Date))+1,1)+(10-  
(Day(DateSerial(Year(Date()),Month(Date))+1,0))-Day(Date()))**



где

$(10 - (\text{Day}(\text{DateSerial}(\text{Year}(\text{Date}()), \text{Month}(\text{Date}()) + 1, 0)) - \text{Day}(\text{Date}()))$  –

разность между заданным количеством дней (10) и количеством дней от текущего дня до последнего дня текущего месяца (поле *Остаток*)



## ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И КОНТРОЛЯ ЗНАНИЙ

1. Перечислите подстановочные символы, которые можно использовать в командах *Найти* и *Заменить*.
2. Охарактеризуйте различия в использовании подстановочных символов #, ?, \*. Приведите примеры.
3. Охарактеризуйте назначение дополнительной опций поиска – *С учетом формата полей* и *С учетом регистра*.
4. Каким образом с помощью команды *Найти* можно осуществлять поиск данных по всей таблице, а не только в текущем столбце.
5. Чем отличается команда *Найти* от команды *Заменить* и что у них общего?
6. Охарактеризуйте особенности *Сортировки* данных одновременно по двум полям таблицы.
7. Приведите пример использования *Фильтр по выделенному* в таблице **Адресная\_Книга**.
8. В каком случае необходимо использовать команду *Фильтр по форме* при поиске данных в таблице?
9. Охарактеризуйте отличия в использовании критерия *И* и критерия *ИЛИ* при применении *Фильтр по форме*.
10. Перечислите команды меню *Формат*, доступные в режиме *Конструктора таблиц*.
11. Что называется *Запросом* в Access? Перечислите возможные типы запросов.
12. Определите особенности создания *Запроса на выборку* с помощью *Мастера запросов*.

13. Охарактеризуйте элементы окна *Конструктора запросов*.

14. Приведите пример *Запроса на выборку* с условием точного несовпадения значений одного из полей.

15. Приведите пример *Запроса на выборку* с использованием условий не-точного совпадения со значением одного из полей.

16. Приведите пример *Запроса на выборку* с условием выбора записей по диапазону значений.

17. Приведите пример *Запроса на выборку*, в котором Вы использовали операторы *OR* и *AND*.

18. Охарактеризуйте особенности создания в *Запросах на выборку* вычисляемых полей, приведите пример такого запроса в лабораторной работе.

### **Лабораторная работа № 3. ИСПОЛЬЗОВАНИЕ ФОРМ ДЛЯ ВВОДА И РЕДАКТИРОВАНИЯ ДАННЫХ В MS ACCESS**

#### **Задания:**

1. Запустить программу MS Access.

2. Открыть базу данных **АДРЕСНАЯ\_КНИГА**.

3. Создать следующие варианты простых форм для таблицы **Адресная\_Книга** (кнопка **Создать** окна **Формы** базы данных):

–*Автоформа*: в столбец;

–*Автоформа*: ленточная;

–*Автоформа*: табличная.

Сравнить полученные варианты автоформ, оценить их достоинства и недостатки.

4. Создать форму с помощью **Мастера форм** для таблицы **Адресная\_Книга** (кнопка **Создать** окна **Формы** базы данных). Сравнить возможности Мастера форм с автоформами.



*Формы* предоставляют собой более удобный способ просмотра и правки данных в таблицах. *Формы* содержат так называемые *Элементы управления*, с помощью которых осуществляется доступ к данным в таблицах. Элементами управления являются текстовые поля для ввода и правки данных, кнопки, флажки, переключатели, списки, надписи, а также рамки объектов для отображения графики и объектов OLE. Создание форм, содержащих необходимые элементы управления, существенно упрощает процесс ввода данных и позволяет предотвратить ошибки.

MS Access предлагает несколько способов создания форм. Самым простым из них является использование средств автоматического создания форм на основе таблицы или запроса.

Автоматически создаваемые формы (автоформы) бывают нескольких видов, каждый из которых отличается способом отображения данных:

*Автоформа* – организованная "в столбец". В такой форме поля каждой записи отображаются в виде набора элементов управления, расположенных в один или несколько столбцов.

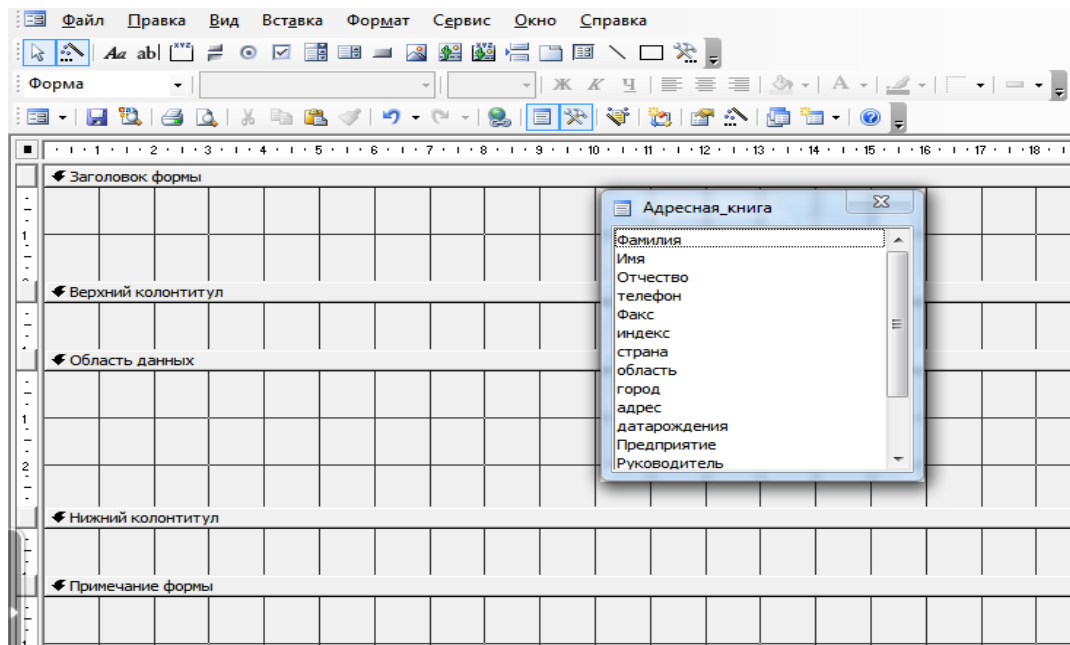
*Табличная* – форма будет выглядеть так же, как обычная таблица MS Access.

*Ленточная* – в такой форме поля каждой записи располагаются в отдельной строке. Это удобно для работы с большими массивами данных, поскольку данные располагаются в таком же порядке, как в простой таблице. Преимуществом именно этого представления формы по сравнению с табличным является то, что каждое поле представлено в виде отдельного элемента управления, которое можно оформить в любом стиле.

Другим простым и быстрым способом создания формы является использование *Мастера форм*.

С помощью *Мастера форма* создается в четыре этапа: выбор полей, данные для которых можно будет вводить форме; выбор внешнего вида формы; выбор фонового рисунка формы; задание имени формы.

Для проектирования собственной формы используется *Конструктор*. При открытии *Конструктора* форма содержит одну область – *Область данных*. Помимо этого форма может содержать область заголовка и примечаний, нижний и верхний колонтитулы.



Данные, размещаемые в *заголовке*, *области данных* и *области примечания*, отображаются в форме. Области верхнего и нижнего колонтитула отображаются только при печати. На экране обязательно должны находиться:

*Список полей* – содержит все поля таблицы или запроса, на основе которых задается форма.

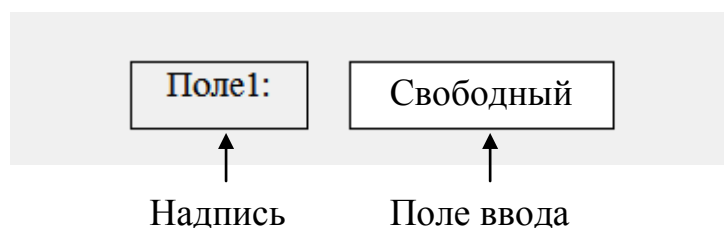
*Панель элементов* – используется для построение *Форм*.

Размещение текста в экранной форме осуществляется с помощью инструмента *Надпись* на *Панели элементов*.

Разместить *Поля ввода* в форме можно двумя способами:

1) Выбрать инструмент *Поле* на *Панели элементов* и щелкнуть мышью в том месте формы, где должно появиться поле.

Выделив данное поле и превратив указатель мыш-



ки в *ручку*, его можно перемещать. Для установления *связи* данного поля с полем таблицы, или запроса нужно:

- выделить поле ввода и в *Свойствах поля* во вкладке *Данные* указать в строке *Источник записей* поле таблицы запроса, данные которого должны выводиться на экран;
- в случае необходимости использования арифметических выражений, в строке *Нет записей* нужно вызвать *Построитель*. В формулах и математических выражениях не должны присутствовать имена Таблиц.

2) Выбрать в окне *Список полей* имя поля и перетащить его в область данных формы. Поле ввода будет связано с полем таблицы, а надпись к нему – соответствовать имени поля (данный способ нельзя применять для вычисляемых полей).

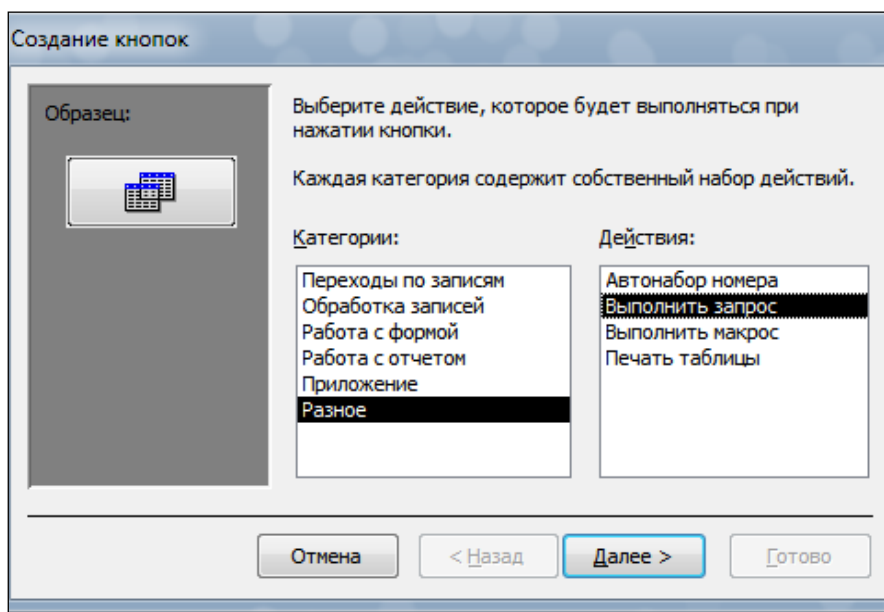
### Создание Кнопок управления

#### Кнопки

используются в формах для выполнения определенного действия или ряда действий

(например, открытие другой формы, запро-

са). Для того чтобы кнопка выполняла какое-либо действие, необходимо создать макрос или процедуру обработки события и связать их со свойством кнопки *Нажатие кнопки*. Предусмотрена возможность создания более 30 различных стандартных кнопок с использованием *Мастер кнопок*. Для этого нужно: установить режим использования мастера на *Панели элементов*; выбрать элемент *Кнопка*; установить курсор в то



место, где должна появиться кнопка. При этом появится окно *Создание кнопок*.

5. В режиме *Конструктора* создать форму для таблицы **Адресная\_Книга** (образец макета приведен на рис. 1).

*Рисунок 1*

The screenshot shows a form titled "Адресная книга" (Address Book) within a software application. The form is divided into several sections:

- Personal Information:**
  - Фамилия: Параев
  - Имя: Леонид
  - Отчество: Павлович
- Address:**
  - Индекс: 69120
  - Страна: Украина
  - Город: Киев
  - Адрес: ул. Ленина, 17
- Дополнительные сведения (Additional Information):**
  - Раб телефон: 78-87-78
  - Дата рождения: 10.03.1969
  - Факс: 69-78-65
  - Предприятие: Угольпром
  - Возраст: 44
- Information about subscribers from Ukraine:**
  - Информация об абонентах из Украина

The application window title bar includes "Файл", "Правка", "Вид", "Вставка", "Формат", "Записи", "Сервис", "Окно", "Справка". The status bar at the bottom indicates "Запись: 6 из 12" and "Режим формы".

Выполнить следующие работы:

- добавить в бланк формы *Область заголовка* и *Область примечаний* (пункт меню **Вид / Заголовок / примечание формы**);
- вывести на экран *Список полей* таблицы **Адресная\_Книга** и *Панель элементов* (кнопки с аналогичными названиями на панели инструментов **Конструктор форм**);
- в соответствии с макетом формы переместить *поля* из *Списка полей* в бланк формы;
- создать *Заголовок* формы с помощью кнопки **Надпись** *Панели элементов*. В области *Заголовка* указать название формы;

–создать **Кнопку управления**, щелчок по которой позволил бы вызывать на выполнение запрос **Украина** (информация обо всех абонентах из Украины);

–для созданной **Кнопки управления** заполнить свойство *Всплывающая подсказка* (вкладка *Другие окна* диалога *Свойства* выбранного объекта). Всплывающая подсказка должна давать краткое пояснение функций, выполняемых кнопкой управления;

–создать вычисляемое поле для расчета *Возраста* абонентов (обратите внимание на то, что в *Выражении* не должно присутствовать название таблицы);

–вставить в форму рисунок (кнопка **Вставить / Рисунок / C:\ Program files \ Microsoft office \ Media \ Cagat10**) и изменить его размеры *По размеру рамки* (кнопка **Свойства рисунка / вкладка Макет / Установка размеров / По размеру рамки**);

–проверить *порядок обхода объектов* формы при нажатии клавиши *Tab*. В случае необходимости изменить порядок обхода (меню **Вид / Последовательность перехода**);

–выполнить *редактирование* формы в соответствии с макетом, а именно:

- a) изменить размеры объектов;
- b) переместить их в нужное место формы;
- c) выполнить выравнивание объектов;
- d) изменить установки цвета объектов формы;
- e) использовать линии и прямоугольники для объединения в группы схожих по смыслу объектов и для улучшения внешнего вида формы;

– ввести в таблицу **Адресная\_Книга** несколько новых записей с помощью созданной формы;

–сохранить форму в базе данных.



## ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И КОНТРОЛЯ ЗНАНИЙ

1. Перечислите и определите назначение *Кнопок перемещения* по записям в автоформах.
2. Как осуществляется переход из *Формы* в *Режим таблицы*? Перечислите возможные варианты.
3. Перечислите альтернативные способы создания *Автоформ*.
4. Перечислите области формы *Конструктора форм*.
5. Как осуществляется выделение, изменение размеров, перемещение и удаление объектов в форме?
6. Для чего и как выполняется изменение *Порядка обхода* объектов формы?
7. Определите назначение *Панели элементов*? Какие кнопки *Панели элементов* и для чего Вы использовали при создании формы?

## Лабораторная работа № 4. ПРОЕКТИРОВАНИЕ МНОГОТАБЛИЧНЫХ БАЗ ДААННЫХ В MS ACCESS. ТЕОРИЯ НОРМАЛИЗАЦИИ

### Задания:

1. Запустить программу MS Access.
2. Открыть базу данных **ЗАКАЗЫ\_КЛИЕНТОВ**.
3. Открыть таблицу **Клиенты** в режиме **Конструктора**.





*Ключ* или *потенциальный ключ* – это поле или набор полей, однозначно (уникально) идентифицирующих запись. Ключ должен быть минимально достаточным: в нем не должно быть полей, удаление которых из первичного ключа не отразится на его уникальности.

Ключи можно разделить:

*а) по признаку общности:*


➤ *простой* – сформирован из значений единственного поля, которые однозначно определяют каждую запись;

➤ *составной* – сформирован из значений нескольких полей.

*б) по ограничению целостности данных:*

➤ *первичный ключ* – служит как ограничение целостности в рамках одной таблицы и для однозначной идентификации данных таблицы. Значения поля первичного ключа не может повторяться или быть пустым:

➤ *внешний ключ (вторичный)* – служит как ограничение целостности связей нескольких таблиц и создается в подчиненной таблице. Подчиненная таблица не может ссылаться на несуществующие записи главной таблицы.

Самый простой способ создания первичного ключа – добавить в таблицу поле типа *Счетчик* и затем нажать кнопку с изображением ключа на панели инструментов .

В результате этого:

➤ свойство счетчика *Обязательное поле* примет значение *Да*,  
 ➤ свойство *Индексированное поле* – *Да* (совпадения не допускаются).

Признаком первичного ключа является иконка с изображением ключа слева от наименования поля. Использование *Счетчика* в этом случае является очень удобным, поскольку Microsoft Access сам заботится о правильном заполнении этого поля.

В качестве *первичного* ключа можно также определить любое подходящее поле (кроме поля типа объект OLE ), содержащее уникальные непустые значения. Такой первичный ключ, состоящий из одного единственного поля, и является *простым*.

Если ни одно из полей таблицы не содержит или по сути не может содержать уникальных значений (например, фамилии могут повторяться), то можно вместо простого ключа создать *составной* ключ, состоящий из нескольких полей.

Для того чтобы создать составной ключ, необходимо в режиме *Конструктора* таблиц выделить требуемые поля (если удерживать нажатой клавишу <Ctrl>, то при помощи мыши можно выделять несмежные поля) и затем нажать кнопку с изображением ключа на панели инструментов.

4. Создать **первичный** ключ таблицы **Клиенты** – поле *КодКлиента* (кнопка **Ключевое поле** на Панели инструментов **Конструктор таблиц** или команда **Ключевое поле** контекстного меню).

5. В таблице **Клиенты** создать *Маску ввода* для поля *КодКлиента*, состоящую из *двух* обязательных и *одной* необязательной буквы, расположенных в начале кода (буквы должны автоматически переводиться в верхний регистр), и *пяти* обязательных цифр.

6. В режиме **Конструктора** создать таблицу **Поставщики**, которая имеет следующую структуру:

#### Структура таблицы Поставщики

Имя поля	Тип данных
<i>КодПоставщика</i>	Текстовый
<i>НаименованиеПоставщика</i>	Текстовый
<i>Руководитель</i>	Текстовый

7. Создать **первичный** ключ таблицы **Поставщики** – поле *КодПоставщика*.

8. В режиме **Конструктора** создать таблицу **Товары**, которая имеет следующую структуру:

#### Структура таблицы **Товары**

<b>Имя поля</b>	<b>Тип данных</b>
<i>КодТовара</i>	Текстовый
<i>Категория</i>	Текстовый
<i>НаименованиеТовара</i>	Текстовый
<i>Цена</i>	Числовой
<i>Поставщик</i>	Текстовый

9. Для таблицы **Товары**:

– создать Маску ввода для поля *КодТовара*, которая будет начинаться с буквы *T* и пробела и состоять из *двух* обязательных цифр и *двух* обязательных букв, разделенных знаком "–";

– создать **первичный** ключ таблицы – поле *КодТовара*.

10. Создать Раскрывающийся Список для поля *Поставщик* таблицы **Товары** (таблица **Товары** / режим **Конструктора** / поле **Поставщик** / **Тип данных** / **Мастер подстановок** / в качестве источника данных выбрать поле *КодПоставщика* из таблицы **Поставщики**).

11. Открыть таблицу **Заказы** в режиме **Конструктора** и выполнить следующие действия:

– создать **внешний составной** ключ, состоящий из полей *КодКлиента*, *КодТовара* и *ДатаЗаказа*;

– создать раскрывающийся список для поля *КодКлиента* (таблица **Заказы** / режим **Конструктора** / поле **Поставщик** / **Тип данных** / **Ма-**

**стер подстановок** / в качестве источника данных выбрать поле *КодКлиента* из таблицы **Клиенты**);


– создать раскрывающийся список для поля *КодТовара* (таблица **Заказы** / режим **Конструктора** / поле **Поставщик** / **Тип данных** / **Мастер подстановок** / В качестве источника данных выбрать поле *КодТовара* из таблицы **Товары**).



Между двумя или более таблицами базы данных могут существовать отношения подчиненности. Отношения подчиненности определяют, что для каждой записи *главной* таблицы (*родительской*) может существовать одна или несколько записей в *подчиненной* таблице (*дочерней*).

### Схема данных

**Схема данных** является графическим образом базы данных. Она используется различными объектами Access для определения связей между несколькими таблицами. Например, при создании формы, содержащей данные из нескольких взаимосвязанных таблиц, схема данных обеспечивает автоматический согласованный доступ к полям этих таблиц. Она же обеспечивает целостность взаимосвязанных данных при корректировке таблиц.

После того, как созданы таблицы базы данных, можно установить связи между ними, обратившись к схеме данных. Запуск схемы данных выполняется через значок **Схема данных**  на панели инструментов Access.

Связь между таблицами устанавливает отношения между совпадающими значениями в ключевых полях, обычно между полями, имеющими одинаковые имена в обеих таблицах. В большинстве случаев с **первичным** ключом одной (**главной**) таблицы, являющимся уникальным идентификатором каждой записи, связывается **внешний ключ** другой (**подчиненной**) таблицы.

## Типы связей

Тип отношения в создаваемой Microsoft Access связи зависит от способа определения связываемых полей.

Существует три разновидности связей между таблицами базы данных:

- «1 – ∞» («*один-ко-многим*»);
- «1 – 1» («*один-к-одному*»);
- «∞ – ∞» («*многие-ко-многим*»).

Отношение «*один-ко-многим*» имеет место, когда одной записи родительской таблицы может соответствовать несколько записей в дочерней таблице. Связь «*один-ко-многим*» является самой распространенной для реляционных баз данных.

Отношение «*один-к-одному*» имеет место, когда одной записи в родительской таблице соответствует одна запись в дочерней таблице.

Отношение «*многие-ко-многим*» имеет место, когда:

- а) записи в родительской таблице может соответствовать больше одной записи в дочерней таблице;
- б) записи в дочерней таблице может соответствовать больше одной записи в родительской таблице.

Большинство СУБД (в частности Access) не поддерживают связи «*многие-ко-многим*» на уровне индексов и ссылочной целостности. Считается, что всякую связь «*многие-ко-многим*» можно заменить на одну или более связей «*один-ко-многим*».

### Определение связей между таблицами

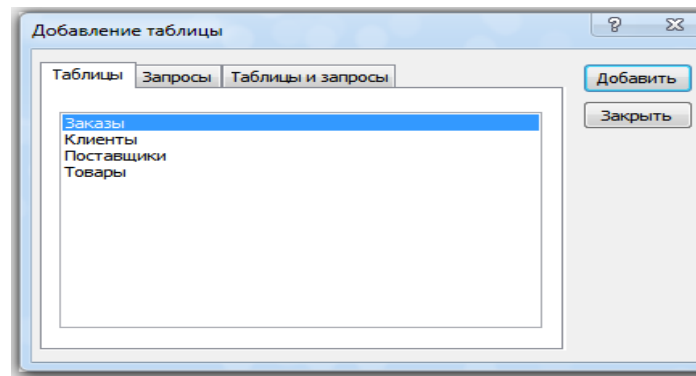
Связь между таблицами определяется путем добавления связываемых таблиц в окно *Схема данных* с последующим перетаскиванием ключевого поля из одной таблицы в другую.

Рассмотрим пример связывания таблиц.

Требуется установить связь между таблицами **Товары** и **Поставщик** через поля *Поставщик* и *КодПоставщика*. В таблице **Поставщик**

это поле является уникальным *первичным* ключом, а в таблице **Товары** – *внешним* ключом.

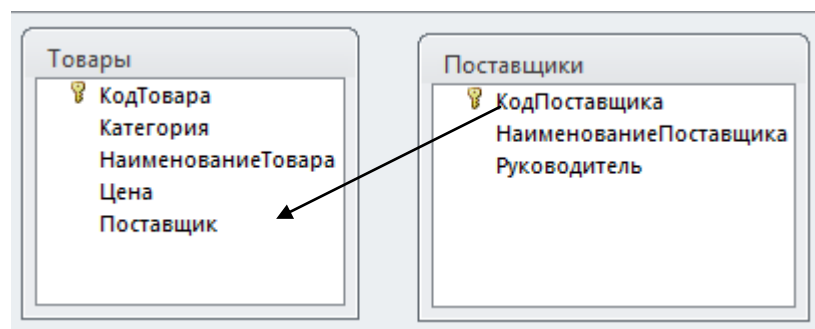
Если схема данных создается первый раз, то при нажатии на кнопку *Схема данных* поверх окна схемы данных появится окно *Добавление таблицы*. В этом окне следует выделить требуемые таблицы и нажать *Добавить*.



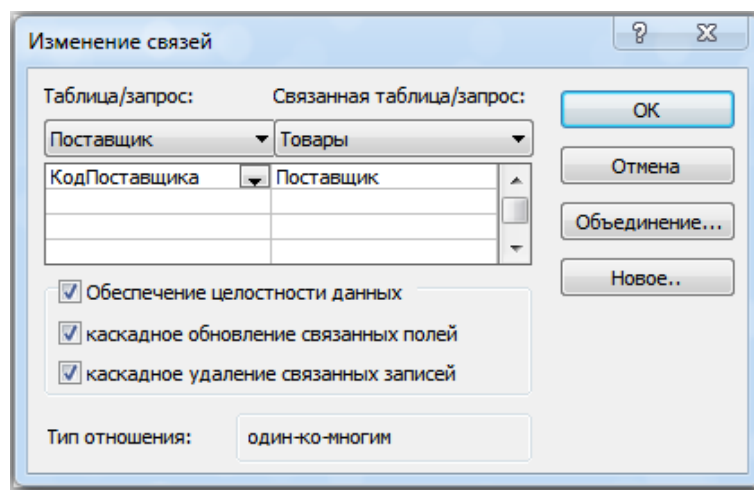
В результате в окно схемы данных будут добавлены графические образы двух таблиц.

Таблица **Поставщики** является *главной*, а **Товары** – *подчиненной*.

Необходимо перетащить мышью поле *КодПоставщика* таблицы **Поставщик** на поле *Поставщик* таблицы **Товары**.



В открывшемся окне *Изменение связей* следует установить флажок *Обеспечение целостности данных*. В этом случае Access будет выдавать предупреждающие сообщения о неправильном вводе данных, если, например, в поле *Поставщик* подчиненной таблицы **Товары** будет введено значение, отсутствующее в поле *КодПоставщика* главной таблицы **Поставщики**.



Обратите внимание, что Access автоматически определил тип связи как **«один-ко-многим»**.

Обязательным условием при установлении связи является совпадение связываемых полей по типу и формату.

Если перетащить поле, не являющееся ключевым и не имеющее уникального индекса, на другое поле, которое также не является ключевым и не имеет уникального индекса, создается неопределенное отношение. В запросах, содержащих таблицы с неопределенным отношением, Microsoft Access по умолчанию отображает линию объединения между таблицами, но условия целостности данных при этом не накладываются и нет гарантии уникальности записей в любой из таблиц.

### **Обеспечение целостности данных**

*Обеспечение ссылочной целостности* данных означает выполнение для взаимосвязанных таблиц следующих условий корректировки базы данных:

- в подчиненную таблицу не может быть добавлена запись, для которой не существует в главной таблице ключа связи;
- в главной таблице нельзя удалить запись, если не удалены связанные с ней записи в подчиненной таблице;
- изменение значений ключа связи главной таблицы должно приводить к изменению соответствующих значений в записях подчиненной таблицы.

В том случае, если пользователь нарушил эти условия в операциях обновления или удаления данных в связанных таблицах, Access выводит соответствующее сообщение и не допускает выполнения операции.

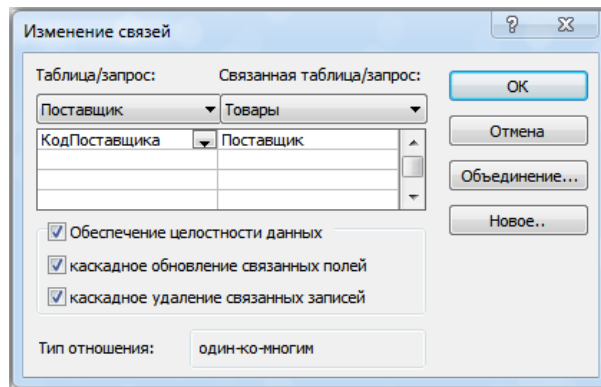
Access не позволяет создавать связи с параметрами обеспечения ссылочной целостности в схеме данных, если ранее введенные в таблицы данные не отвечают требованиям целостности.

Задание параметров целостности данных возможно только при следующих условиях:

- связываемые поля имеют одинаковый тип данных, причем имена полей могут быть различными;
- обе таблицы сохраняются в одной базе данных Access;
- главная таблица связывается с подчиненной по первичному простому или составному ключу (уникальному индексу) главной таблицы.

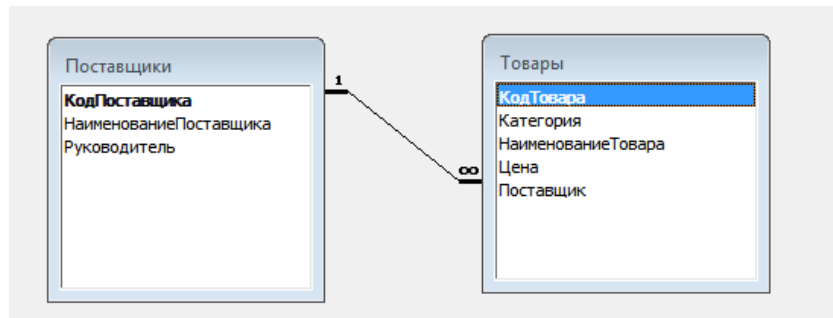
Если для выбранной связи обеспечивается поддержание ссылочной целостности, то можно задать режимы *каскадного обновления* и *удаления* связанных записей:

- в режиме *каскадного обновления* связанных записей при изменении значения в поле связи главной таблицы Access автоматически изменит значения в соответствующем поле в подчиненных записях;
- в режиме *каскадного удаления* связанных записей при удалении записи из главной таблицы Access выполняет каскадное удаление подчиненных записей на всех уровнях.



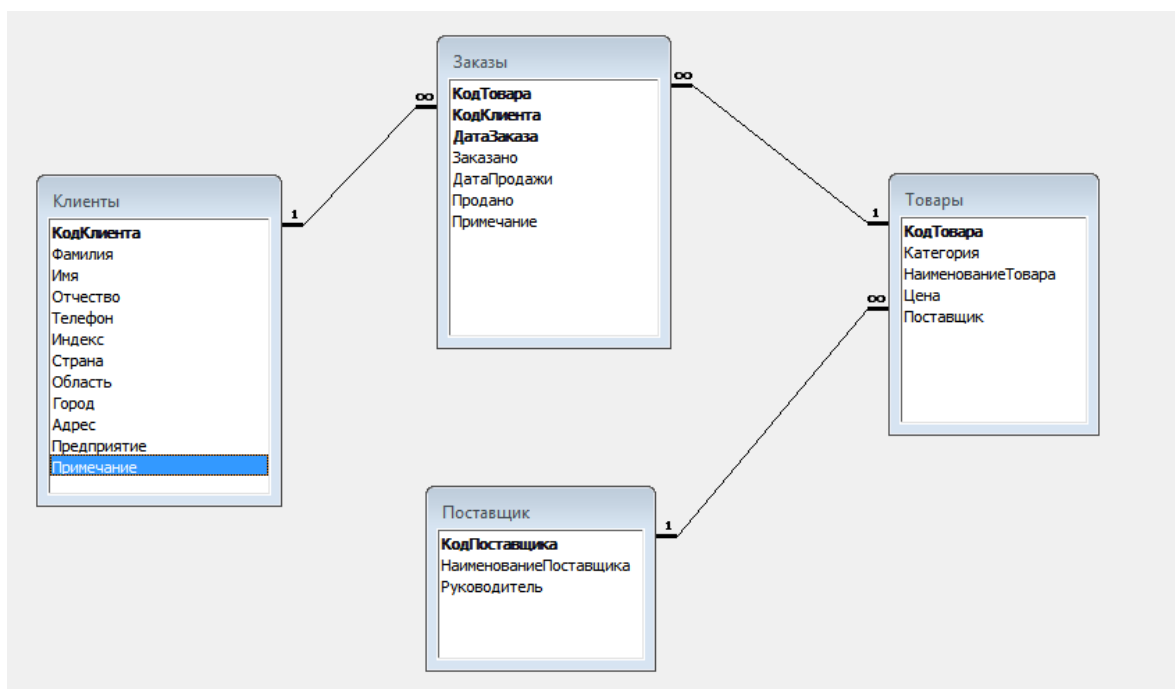


После нажатия на кнопку *Создать*, образы таблиц будут соединены связями как показано на рисунке (появятся символы 1 и ∞ над линией):



Ключевые поля в данных таблицах выделяются **жирным** шрифтом.

12. Посредством *внешних* и *первичных* ключей установить *связи* между таблицами базы данных согласно приведенной **Схеме данных** (пункт меню **Сервис/Схема данных** или кнопка **Схема данных** на Панели инструментов **База данных**). Определить *тип отношения* и установить *условия ссылочной целостности* базы данных.



13. Для всех таблиц базы данных **ЗАКАЗЫ\_КЛИЕНТОВ** создать **автоформы**.

14. Организовать ввод данных в таблицы базы **ЗАКАЗЫ\_КЛИЕНТОВ** с использованием созданных автоформ, соблюдая следующую последовательность:

- таблица **Поставщики** (не менее пяти записей);
- таблица **Клиенты** (не менее пяти записей);
- таблица **Товары** (не менее пяти записей);
- таблица **Заказы** (не менее десяти записей).

15. Проанализировать результаты установленных параметров ссылочной *целостности* базы данных **ЗАКАЗЫ\_КЛИЕНТОВ**, выполнив следующие ситуационные задания по корректировке информации, хранящейся в таблицах:

- один из клиентов ушел на обслуживание в другую фирму (у Вас он выбыл);
- на обслуживание поступил заказ от нового клиента;
- в продажу поступил новый товар высшей категории взамен старого;
- клиент с кодом (например код – 1) получил новый код (например код – 101).

#### *Справочный материал к лабораторной работе № 4*

### **Теория нормализации**

Под *нормализацией* отношения подразумевается процесс приведения отношения к одной из так называемых нормальных форм (или в дальнейшем НФ).

При проектировании баз данных упор в первую очередь делается на *достоверность* и *непротиворечивость* хранимых данных, причем эти свойства не должны утрачиваться в процессе работы с данными, т.е. по-

сле многочисленных изменений, удалений и дополнений данных по отношению к первоначальному состоянию БД.

Для поддержания БД в устойчивом состоянии используется ряд механизмов, которые получили обобщенное название средств поддержки целостности. Эти механизмы применяются как статически (на этапе проектирования БД), так и динамически (в процессе работы с БД).

Приведение структуры БД в соответствие ограничениям, которым она должна удовлетворять в процессе создания, независимо от наполнения ее данными – это и есть *нормализация*.

В целом суть этих ограничений весьма проста: каждый факт, хранимый в БД, должен храниться один-единственный раз, поскольку дублирование может привести к несогласованности между копиями одной и той же информации. Следует избегать любых неоднозначностей, а также избыточности хранимой информации.

Процесс *нормализации* – последовательное преобразование исходной БД к НФ, при этом каждая следующая НФ обязательно включает в себя предыдущую (что, собственно, и позволяет разбить процесс на этапы и производить его однократно, не возвращаясь к предыдущим этапам).

Основными в реляционной теории являются первые три нормальные формы:

### ***Первая нормальная форма (1НФ):***

Таблица в первой нормальной форме должна удовлетворять следующим требованиям:

1. Таблица не должна иметь *повторяющихся записей*.
2. В таблице должны отсутствовать *повторяющиеся группы полей*.
3. Каждое поле *неделимо (атомарно)*.
4. Столбцы и строки должны быть *не упорядочены*.

Для удовлетворения условия 1 каждая таблица должна иметь, по крайней мере, один уникальный ключ.

Поэтому для выполнения условия необходимо создать *индекс* (определить *первичный ключ*), который включал бы совокупность всех полей таблицы, однозначно определяющих запись.

Требование 2 должно устранять повторяющиеся группы.

### ***Вторая нормальная форма (2 НФ):***

О таблице говорят, что она находится во второй нормальной форме, если:

1. Она удовлетворяет условиям *первой нормальной формы*.
2. *Первичный* ключ однозначно определяет *всю запись*.
3. *Все* поля зависят от *первичного* ключа.
4. *Первичный* ключ не должен быть *избыточным*.

Понятие второй нормальной формы применимы только к таблицам, имеющим составной индекс.

Все таблицы, имеющие только одно индексное поле, автоматически удовлетворяет условию второй нормальной формы.

### ***Третья нормальная форма (3 НФ):***

О таблице говорят, что она находится в третьей нормальной форме, если:

1. Она удовлетворяет условиям *второй нормальной формы*.
2. Ни одно из *неиндексных* полей таблицы не идентифицируется с помощью другого *неиндексного* поля (каждое *неключевое* поле не должно зависеть от другого *неключевого* поля).

Сведение таблицы к третьей нормальной форме предполагает разделение таблицы с целью помещения в отдельную таблицу столбцов, которые не зависят от полного индекса.

В результате такого разбиения каждое из *неиндексных* полей должно оказаться независимым от какого-либо другого *неиндексного* поля.



## ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И КОНТРОЛЯ ЗНАНИЙ

1. Что такое ключ? Какие виды *ключей*, классифицированных по признаку общности, Вы знаете?
2. Каковы основные отличия *первичного ключа* от *внешнего*?
3. Перечислите основные требования к полям, между которыми устанавливаются *связи*. Могут ли быть установлены связи между полями с различными именами, с различными типами данных?
4. Перечислите возможные *типы отношений* между таблицами базы данных. Определите типы отношений между таблицами Вашей базы данных.
5. Как *изменить* существующую *связь* между таблицами базы данных?
6. Как *удалить* существующую *связь* между таблицами базы данных?
7. Как добавить (удалить) еще одну таблицу в *Схему данных*?
8. Как вызвать окно *Конструктора* таблицы, не выходя из окна *Схемы данных*?
9. Дайте определение *ссылочной целостности* базы данных. Перечислите ограничения, действующие после определения целостности данных в базе.
10. Для чего применяется *нормализация* реляционных баз данных?
11. Каким требованиям должна удовлетворять таблица в первой нормальной форме?
12. Каким требованиям должна удовлетворять таблица в второй нормальной форме?
13. Каким требованиям должна удовлетворять таблица в третьей нормальной форме?

14. Для чего используются режимы *каскадного удаления* и *каскадного обновления* связанных полей реляционной базы данных? При выполнении какого задания Вы использовали эти режимы?

15. Что необходимо сделать для того, чтобы при удалении клиента из списка клиентов информация о его заказах не удалялась?

## **Лабораторная работа № 5. СОЗДАНИЕ ЗАПРОСОВ К МНОГОТАБЛИЧНОЙ БАЗЕ ДАнных В MS ACCESS**

### **I. СОЗДАНИЕ МНОГОТАБЛИЧНЫХ, ИТоговых И ПАРАМЕТРИЧЕСКИХ ЗАПРОСОВ**



*Многотабличный запрос* позволяет сформировать результат путем объединения взаимосвязанных записей из таблиц и запросов базы данных и выбора из них нужных полей и записей. Многотабличный запрос часто осуществляет объединение данных, которые на этапе проектирования были разделены на множество объектов в соответствии с требованиями *нормализации*.

При проектировании и конструировании запроса важнейшим условием является правильное представление о том, как идет объединение записей таблиц при формировании результата.

### **Задания:**

1. Запустить **Access**.
2. Открыть базу данных **ЗАКАЗЫ\_КЛИЕНТОВ**.
3. Создать следующие **многотабличные запросы** (вкладка **Запросы** / кнопка **Создать** / **Конструктор** / **Добавить** необходимые для создания запроса таблицы):

- **Фамилия** – запрос на получение информации обо всех клиентах, не имеющих телефона, фамилии которых начинаются с буквы "Д", но не содержат букв "А", "С" и "М";
- **Телефон** – запрос на получение информации обо всех клиентах, телефон которых начинается на одну или две семерки;
- **Месяц** – запрос на получение информации о заказах, сделанных клиентами-женщинами в преддверии 8 Марта;
- **Цена** – запрос на получение информации о товарах, цены на которые для клиентов, не проживающих в Украине, меньше 1000 у.е.;
- **Вып\_заказ** – запрос на получение информации обо всех выполненных заказах;
- **Остаток** – запрос на получение информации о товарах, проданных в апреле текущего года. В запросе создать для каждого клиента вычисляемое поле *Остаток*, определяющее разность между значениями полей *Заказано* и *Продано* как в количественном, так и в денежном выражении (кнопка **Построитель выражений** на панели инструментов окна **Конструктора запросов**);



**Объединения** устанавливают правила, которым должны соответствовать включаемые в операцию запроса данные. В отличие от фильтров, объединения также определяют, что каждая пара строк, удовлетворяющая условию объединения, в наборе записей может быть представлена в виде одной строки.

В зависимости от решаемых задач *Объединения* между таблицами в запросах может быть нескольких видов. Наиболее распространенным является **Внутреннее объединение**.

Запросы, созданные на основе Внутреннего объединения предполагают, что *строки одной из объединяемых таблиц соответствуют строкам другой таблицы по значениям в связанных полях*. То есть, при выпол-

нении такого запроса в операцию включаются только строки, имеющие одинаковые значения в обеих связанных таблицах.

Подобного рода объединения между таблицами Access создает автоматически, если:

- в таблицах имеются поля с одинаковыми именами и согласованными типами, причем одно из полей является ключевым;
- соединение было явно задано в окне Схема данных.

Запросы, созданные на основе **внешнего объединения** представляют собой запросы, в которые включаются *все строки одной таблицы, а также строки другой таблицы, значения которых в первой и второй таблице совпадают, несмотря на то, что некоторые из строк с обеих сторон полностью совпадают.*

Внешние объединения являются направленными, они могут быть **Левыми** или **Правыми**:

- запрос, в котором участвуют таблицы с **Левым** внешним объединением, выводит все записи таблицы "один", в независимости от того, имеются ли соответствующие им записи в таблице "многие";
- запрос, в котором участвуют таблицы с **Правым** внешним объединением, выводит все записи таблицы "многие", в независимости от того, имеются ли соответствующие им записи в таблице "один".

Например, при создании запроса, который отображает клиентов, не сделавших ни одного заказа, необходимо использовать *Левый* запрос. Для этого в окне *Кон-*  
*структора* запросов щелкните правой кнопкой мыши на линии, соединя-



ющей таблицы *Клиенты* и *Заказы*, и выберите из контекстного меню команду **Параметры объединения**.

В появившемся диалоговом окне таблица на стороне "один" считается *Левой* (независимо от того, как она расположена в окне *Конструктора*), а таблица на стороне "многие" – *Правой*. Ниже расположены три переключателя:

"1" – задает *внутреннее* соединение;

"2" – *левое* внешнее соединение;


"3" – *правое* внешнее соединение.

Задайте левое внешнее соединение, выбрав значение "2".

– запрос **Без\_заказа** на получение информации обо всех товарах, которые не были заказаны. Поле *Код товара* в данном случае целесообразно перемещать в *бланк запроса* из таблицы **Товары**. Проверьте, как будет выглядеть запрос, если переместить данное поле из таблицы **Заказы**.

## II. СОЗДАНИЕ ИТОГОВЫХ И ПАРАМЕТРИЧЕСКИХ ЗАПРОСОВ



Запросы позволяют не только отбирать нужную информацию из таблиц и обрабатывать ее путем создания новых (вычисляемых) полей, но и производить так называемые итоговые вычисления. Для создания **Итоговых запросов** используется кнопка  *Групповые операции*.

В результате в бланк запроса добавляется строка *Групповая операция*, в которой для каждого поля можно выбрать из списка соответствующую групповую операцию:

- Группировка – поле, определяющее группу записей, по которой вычисляются статистические параметры. К одной группе относятся все записи, для которых значение поля с режимом Группировка одинаковы;
- Sum – сумма значений некоторого поля для группы;
- Avg – среднее от всех значений поля в группе;

- Max, Min – максимальное, минимальное значение поля в группе;
- Count – число значений поля в группе без учета пустых значений;
- StDev – среднеквадратичное отклонение от среднего значения поля в группе;
- Var – дисперсия значений поля в группе;
- First, Last – значение поля из первой или последней записи в группе.
- Условие – режим задания условия отбора для поля, по которому не выполняется группировка;
- Выражение – вычисляемое поле, значение которого рассчитывается по сложной формуле.

Пример итогового запроса по определению количества заказов, сделанных каждым клиентом:

ФИО: [Клиенты]![Фамилия]+" "+Left([Клиенты]![Имя];1)+" "+Left([Клиенты]![Отчество];1)+" "	Заказано
	Заказы
Группировка	Count
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

В данном запросе для формирования вычисляемого поля "ФИО" используется операция сложения строк (+) и функция *Left(строка, n)* – возвращает n левых символов строки:

**ФИО: [Клиенты]![Фамилия]+" "**  
 **"+Left([Клиенты]![Имя];1)+" "+Left([Клиенты]![Отчество];1)+" "**

### Правила создания Итоговых запросов

1. Если для некоторого поля в строке *Групповая операция* выставлены значения *Группировка* или *Выражение*, то по этому же полю в строке *Условие отбора* можно указать условие.

*Пример* итогового запроса для вывода информации о количестве заказов, сделанных клиентами, фамилии которых начинаются с буквы "А":

ФИО: Клиенты!Фамилия+" "+Left(Клиенты!Имя;1)+" "+Left(Клиенты!Отчество;1)+" "	Заказано
	Заказы
Группировка	Count
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Like "A**"	

2. Если для некоторого поля в строке *Групповая операция* выставлена любая итоговая *операция* (*Sum*, *Count*, *Avg*, и др.), то по этому полю в строке *Условие отбора* нельзя указывать условие. Для выполнения отбора по такому полю его помещают в бланк запроса дважды.

*Пример* итогового запроса для вывода информации о количестве заказов по каждому клиенту, причем учитываются только заказы, объем которых превышает 200 шт:

ФИО: Клиенты!Фамилия+" "+Left(Клиенты!Имя;1)+" "+Left(Клиенты!Отчество;1)+" "	Заказано	Заказано
	Заказы	Заказы
Группировка	Count	Условие
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		>200

3. Если для некоторого поля в строке *Групповая операция* выставлено *Условие*, то в *Конструкторе запросов* автоматически убирается флажок *Вывода на экран*.

4. Если по некоторому полю необходимо выполнить *несколько итоговых операций*, то его помещают в бланк запроса несколько раз.

*Пример* итогового запроса с условием отбора: информация о количестве и сумме заказов по каждому клиенту, причем учитываются только заказы, объем которых превышает 200 шт:

ФИО: Клиенты!Фамилия+" "+Left(Клиенты!Имя;1)+" "+Left(Клиенты!Отчество;1)+" "	Заказано	Заказанс	Заказано
	Заказы	Заказы	Заказы
Группировка	Count	Sum	Условие
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
			>200

### Задания:

1. Создать следующие **итоговые запросы** (в окне **Конструктора** запросов выбрать кнопку **Групповые операции** на Панели инструментов):

- **Итог\_1** – запрос на получение информации о максимальном и среднем размере заказа за каждый месяц;
- изменить запрос **Итог\_1** так, чтобы он позволял получить информацию о максимальном и среднем размере заказа только за текущий месяц;
- **Итог\_2** – запрос на получение информации об общем количестве заказанного и проданного товара по каждому клиенту. Клиент должен идентифицироваться с помощью поля ФИО;
- изменить запрос **Итог\_2** так, чтобы он позволял получить информацию об общем количестве заказанного и проданного товара по каждому клиенту, причем количество заказанного товара в заказе должно быть не меньше 1000 шт;
- **Итог\_3** – запрос на получение информации о количестве невыполненных заказов и их общей сумме в у. е.



Любой тип запроса может выполняться в одном из двух режимов:

- Обычном;
- Параметрическом.

*Параметрический запрос* создается в том случае, когда пользователю необходимо работать с запросами, построенными одинаково, но имеющими некоторые различия в поставленных условиях отбора.

В том случае, для того чтобы не создавать множество одинаковых по структуре запросов, следует сформировать запрос, в котором его окончательная настройка будет выполняться непосредственно *при его выпол-*

нении и заключатся в процедуре ввода с клавиатуры конкретного значения параметра условия отбора.

Для создания *Параметрического запроса* необходимо в строке *Условие отбора* в квадратных скобках ввести текст вопроса, который будет появляться при запуске запроса для уточнения параметров отбора.

Фамилия	Имя	Страна
Клиенты	Клиенты	Клиенты
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
		[Введите название страны]

Текст вопроса

Тогда, например, для уточнения *Страны*, гражданином которой является интересующий Вас клиент фирмы, появиться диалоговое окно:

Введите название страны

OK Cancel

Примеры условий отбора Параметрических запросов:

Условие отбора	Пояснение
[Название города 1] or [Название города 1]	При запуске запроса необходимо последовательно ввести названия двух интересующих Вас городов
>=[Введите начальную дату]	При запуске запроса необходимо ввести интересующую Вас начальную дату продаж
Between [Минимальная цена] and [Максимальная цена]	При запуске запроса необходимо последовательно ввести минимальную и максимальную интересующую Вас цену товара
Like [Первая буква фамилии]&*	При запуске запроса необходимо ввести первую букву отбираемых из базы данных фамилий клиентов

2. Создать следующие *параметрические запросы*:

– **Парам\_2** – запрос, позволяющий определить общую сумму (в денежном выражении) невыполнения заказов по любым двум товарам (по одному из двух товаров);



Функция **MonthName(номер\_месяца)** возвращает строку, показывающую текстовое название месяца для выбранного номера месяца от 1 до 12.

Входной параметр также может быть выражением, например:

**MonthName (Month([Заказы]![ДатаЗаказа])** – функция возвращает название месяца, в котором был сделан заказ.

– запрос **Итог\_2** сделать параметрическим (назовите его **Парам\_1**), позволяющим получить информацию о максимальном и среднем размере заказа за любой месяц текущего года. Для идентификации названия месяца создать вычисляемое поле с использованием функции *MonthName*;

### III. СОЗДАНИЕ ПЕРЕКРЕСТНЫХ ЗАПРОСОВ



*Перекрестный запрос* – представляет собой выборку данных, записанную в виде двумерной матрицы и созданную на основе таблиц и/или запросов Access. Этот запрос представляет выбранные данные в формате, напоминающем формат электронной таблицы. С точки зрения типа запроса *Перекрестный запрос* представляет собой специальный тип группового запроса, для которого строка **Групповая операция** всегда включена.

Для создания перекрестного запроса необходимо указать:

- поле заголовков *строк*;
- поле заголовков *столбцов*;
- поле для подсчета *итоговых* значений на пересечении строк и столбцов.

Для создания перекрестного запроса необходимо определить необходимые поля в режиме **Конструктора**, а затем перейти в режим **Пере-**

## крестный запрос

*Пример* простого перекрестного запроса, в котором на пересечении строк и столбцов выводится суммарное количество единиц купленного товара каждым из клиентов.

Поле:	Код клиента	Код товара	Заказано
Имя таблицы:	Клиенты	Заказы	Заказы
Групповая операция:	Группировка	Группировка	Sum
Перекрестная таблица:	Заголовки строк	Заголовки столбцов	Значение
Сортировка:			
Условие отбора:			
или:			

### Задания:

1. Запустить **Access**.
2. Открыть базу данных **ЗАКАЗЫ\_КЛИЕНТОВ**.
3. Создать **перекрестные** запросы для следующих условий:

3.1. Необходимо получить информацию об итоговых суммах заказов (в у. е.) по каждому клиенту на каждый вид товара (для изменения типа запроса используется кнопка *Тип запроса* на Панели инструментов окна **Конструктора** запросов). В качестве *заголовков строк* в создаваемой *Перекрестной таблице* будут выступать **Фамилии** клиентов, а в качестве *заголовков столбцов* – **Наименования товаров** (запрос **Перекр\_1**).

3.2. Изменить условия перекрестного запроса **Перекр\_1** так, чтобы он выводил информацию об итоговых суммах заказов (в у. е.) на каждый вид товара только по клиентам, работающим на предприятии *Айрит*.



Для обеспечения группировки по отдельным элементам Даты в качестве заголовков строк можно использовать функцию **Format** (**дата; формат\_даты**) которая одновременно определяет:

*формат вывода даты:*

- d* – день в формате 1, 2 , 3;
- dd* – день в формате 01,02,03;
- mm* – месяц в формате 01, 02, 03;
- mmm* – месяц в формате апр, май, июн, июл, авг;
- mmmm* – месяц в формате апрель, май, июнь, июль, август;
- yy* – год в формате 98, 99, 00;
- yyyy* – год в формате 1998, 1999, 2000,

и *уровень группировки* в перекрестной таблице.

Например:

- функция **Format([Заказы]![ДатаПродажи];"dd-mmm-yyuu")** позволяет выполнять группировку по датам в их полном формате, (в качестве заголовков столбцов будут выступать выражения 01-янв-2000, 02-янв-2000 и т. д.);

- функция **Format([Заказы]![ДатаПродажи];"yyuu")** позволяет выполнять группировку по годам (в качестве заголовков столбцов будут выступать выражения 2012, 2013, 2014 и т. д.);

- функция **Format([Заказы]![Дата продажи];"mmm-yyuu")** позволяет выполнять группировку по месяцам каждого года, по годам (в качестве заголовков столбцов будут выступать выражения янв-2012, фев-2012, янв-2012и т.д.).

Если в вашей таблице содержится информация о заказах за один год, то в функции **Format()** информацию о формате *года* можно опустить.

3.3. Необходимо получить информацию об *объемах продаж* (в у. е.) по каждому *виду товара* (заголовки строк) за каждый *месяц* (заголовки столбцов) (запрос **Перекр\_2**).

3.4. Изменить условия перекрестного запроса **Перекр\_2** так, чтобы он выводил информацию об *объемах продаж* (в у. е.) по каждому *виду товара* за каждый *месяц* только по полностью *выполненным* заказам. Переименуйте новый запрос в **Перекр\_21**.



3.5. Самостоятельно сформулировать условие и с помощью **Мастера перекрестных запросов** создать перекрестный запрос **Перекр\_3** (вкладка **Запросы**/ кнопка **Создать / Перекрестный запрос**). Обязательным условием является добавление в перекрестный запрос **Перекр\_3** итоговых значений по каждой строке.

3.6. Выполнить *Экспорт* перекрестной таблицы, сформированной в результате выполнения запроса **Перекр\_3** в *MICROSOFT EXCEL* (кнопка **Связи с Office / Анализ в Microsoft Excel**).

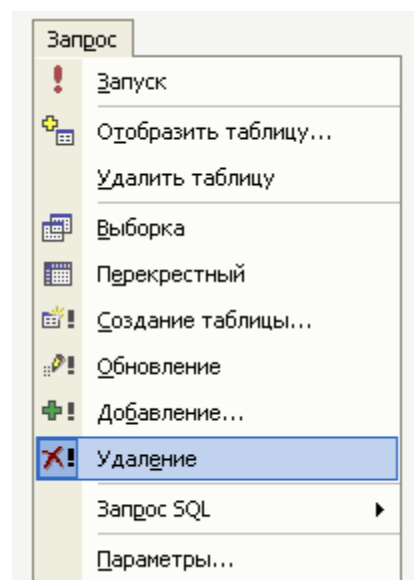
#### IV. СОЗДАНИЕ МОДИФИЦИРУЮЩИХ ЗАПРОСОВ



**Модифицирующие запросы** – это запросы на выполнение действий, в результате выполнения которых изменяется содержимое таблиц базы данных.

Существует четыре вида модифицирующих запросов:

- запросы *Удаления* позволяют удалить записи из таблицы БД, удовлетворяющие определенным условиям;
- запросы *Добавления* используются для добавления записей, удовлетворяющих определенным условиям, из одной таблицы в другую;
- запросы *Обновления* позволяют изменить значения выбранных полей во всех записях таблицы, удовлетворяющих определенным условиям;
- запросы *Создания* таблиц позволяют создать новые таблицы в базе данных на основе информации, полученной из других таблиц или запросов.

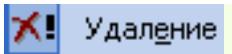


## Запросы удаления

При необходимости быстро удалить большой объём данных или регулярно удалять определенные данные можно воспользоваться запросом **Удаления**: он позволяет указывать условия, по которым можно быстро найти и удалить данные. Использование такого типа запроса экономит время, поскольку его можно сохранить и затем применять необходимое количество раз.

Последовательность создания и выполнения запроса *Удаления*:

1. В режиме Конструктора запросов *Добавить* таблицу, из которой будут удаляться записи и изменить *Тип запроса* на запрос




2. В результате в бланке запроса появится строка *Удаление*.


3. В первый столбец бланка запроса из списка полей таблицы добавляем символ *звездочка* (\*), обозначающий, что будут удаляться все поля интересующих нас записей таблицы. При этом в строке *Условие* появится опция *Из*.

4. В следующих столбцах бланка запроса необходимо добавить названия полей таблицы, *Условия отбора* по которым ограничивают список удаляемых записей. При этом в строке *Условие* появится опция *Условие*.

Пример бланка запроса *Удаления* всех записей о клиентах из *России*:

Поле:	Клиенты.*	Страна
Имя таблицы:	Клиенты	Клиенты
Удаление:	Из	Условие
Условие отбора:		"Россия"

5. Для *проверки* корректности установленных условий отбора записей, предназначенных для удаления, используйте кнопку *Вид* .

6. Для *выполнения* запроса *Удаления* необходимо использовать кнопку  **Запуск**. При этом последовательно отображаются: сообщение с приглашением подтвердить удаление записей и информация о количестве записей, которые будут удалены.

Если при выполнении запроса *Удаления*, включающего только таблицу на стороне один отношения «*один-ко-многим*» для данного отношения была включена опция *Каскадное удаление связанных записей*, то будут также удаляться и связанные записи на стороне «*многие*». При чем, при выполнении запроса предупреждения об этом не будет.

Перед тем, как удалить данные или выполнить запрос на удаление, необходимо убедиться, что была создана *резервная* копия базы данных.

### Запрос Создания таблицы

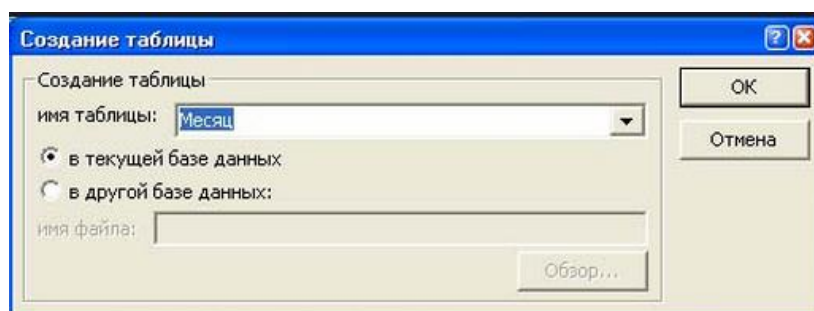
Запрос *Создания* таблицы получает данные из одной или нескольких таблиц, а затем помещает набор результатов в новую таблицу. Эта новая таблица может располагаться в базе данных, открытой в данный момент, или создается в другой базе данных.

Последовательность создания и выполнения запроса *Создания* таблицы:


1. В режиме Конструктора запросов *Добавить* таблицу, из которой будут удаляться записи и изменить *Тип запроса* на запрос


 **Создание таблицы...**

2. Ввести имя создаваемой таблицы и указать ее местоположение – в текущей или другой базе данных.



3. Добавить все необходимые для новой таблицы *поля*, а также создать *вычисляемые* поля и *Условия* отбора.

4. Для *проверки* корректности выбранных полей, а также установленных условий отбора записей, используйте кнопку *Вид* .

5. Для *выполнения* запроса *Создания таблицы* необходимо использовать кнопку  *Запуск*. При этом последовательно отображаются: сообщение с приглашением подтвердить создание таблицы и информация о количестве записей, которые будут в неё добавлены.

6. Полученная в результате выполнения запроса новая таблица не наследует свойства полей (маска ввода, поле подстановки и т. п.) и первичные ключи.

4. Создать **модифицирующие** запросы **Создания таблицы** и **Удаления** для выполнения следующего задания: Вам необходимо удалить из таблицы **Заказы** все записи о сделках, совершенных до 1-го апреля текущего года. Причем удаленные записи нужно сохранить в специальной таблице **Удаленные**.

Для этого:

4.1. Создать резервную копию таблицы **Заказы**: для этого (находясь во вкладке **Таблицы** выделить таблицу **Заказы** и последовательно выполнить операции **Копировать** и **Вставить**, а затем в появившемся диалоговом окне задать новое Имя таблицы **Копия\_заказы** и установить параметры вставки **Структура и данные**).

4.2. Создать **Мод\_1** – запрос создания новой таблицы с именем **Удалённые**, в которой будут храниться записи со значением поля *ДатаПродажи* раньше 1-го апреля текущего года (кнопка **Тип запроса/Создание таблицы** на Панели инструментов окна **Конструктора** запросов. В появившемся диалоговом окне необходимо задать Имя создаваемой таблицы и указать, что она будет находиться в *текущей базе данных*).

4.3. Просмотреть выбранные для помещения в новую таблицу записи и убедиться в правильности заданного условия (кнопка **Вид / режим Таблицы** на Панели инструментов окна **Конструктора** запросов).

4.4. Выполнить запрос *Создания* таблицы **Удалённые** (кнопка **Запуск** на Панели инструментов окна **Конструктора** запросов).

4.5. Открыв запрос **Мод\_1** в режиме **Конструктора** изменить его тип на запрос **Удаления** (кнопка **Тип запроса /Удаление**).


4.6. Выполнить запрос **Удаления** (кнопка **Запуск** на Панели инструментов окна **Конструктора** запросов).

### Запрос Обновления



Запрос **Обновления** записей предназначен для внесения изменений в группу записей одной или нескольких таблиц за одну операцию.

Последовательность создания и выполнения запроса *Обновления*:

1. В режиме Конструктора запросов *Добавить* таблицу, из которой будут удаляться записи и изменить *Тип запроса* на запрос  **Обновление**.


2. В бланк запроса добавить те поля таблицы, которые необходимо обновить. При этом в строке *Обновление* указываются выражения, значения которых будут присвоены выбранным полям.


Примеры допустимых выражений для обновления полей таблицы

Поле	Выражение	Результат
Город	"Днепропетровск"	Заменяет значения текстового поля записей, удовлетворяющих условиям отбора, на строку "Днепропетровск"
ДатаПродажи	Date()	Задаёт новое значение для поля типа дата/время (Date/Time) – текущую дату
КодКлиента	"PN" & [Код]	Для каждой записи, подлежащей обновлению, значением поля будет комбинация префикса "PN" и

		строки из поля Код той же записи
Заказано	[Цена] *[Количество]	Значением поля каждой записи, подлежащей обновлению, будет произведение полей Цена и Количество той же записи
Зарплата	[Зарплата]* 1.1	Если задано для поля Зарплата, то увеличивает зарплату на 10% в записях, удовлетворяющих условиям отбора

3. В последнюю очередь добавляются *поля*, для которых предполагается указать *Условия* отбора.

4. Для *проверки* корректности выбранных полей, а также установленных условий отбора записей, используйте кнопку *Вид* .

5. Для *выполнения* запроса *Обновления* таблицы необходимо использовать кнопку  *Запуск*. При этом последовательно отображаются: сообщение с приглашением подтвердить обновление записей и информация о количестве записей, которые будут обновлены.

Перед запуском запросов *Обновления* таблицы не забудьте сделать копии изменяемых таблиц, чтобы можно было вернуть базу данных в первоначальное состояние.

Пример запроса *Обновления*, позволяющего для еще не проданных товаров с кодом 111 установить дату продажи – сегодняшнее число.

Поле:	ДатаПродажи	КодТовара	Продано
Имя таблицы:	Заказы	Заказы	Заказы
Обновление:	Date()		
Условие отбора:		"111"	0

5. Создать следующие **модифицирующие** запросы **Обновления**:

5.1. **Мод\_2** – запрос, позволяющий увеличить на 20% цены на товары с любыми двумя кодами.

5.2. **Мод\_3** – запрос, позволяющий для всех заказов, выполненных сегодня, заполнить поле *Продано* значением соответствующего ему поля *Заказано*.

### Запрос Добавления



Запрос **Добавления** записей добавляет набор записей из одной или нескольких исходных таблиц (или запросов) в одну или несколько результирующих таблиц. Обычно исходные таблицы и конечная таблица находятся в одной и той же базе данных, но это не обязательно.

Последовательность создания и выполнения запроса на *Добавление*:

1. В режиме *Конструктора* запросов *Добавить* таблицу, из которой будут удаляться записи и изменить *Тип запроса* на запрос

 **Добавление...**


2. Ввести имя таблицы, в которую будет выполняться добавление, и указать ее местоположение – в текущей или другой базе данных.


3. Добавить из списка полей в бланк запроса *поля*, которые необходимо *добавить* или которые будут использоваться при определении *условия отбора*. При добавлении полей в бланк запроса необходимо учитывать следующие особенности:

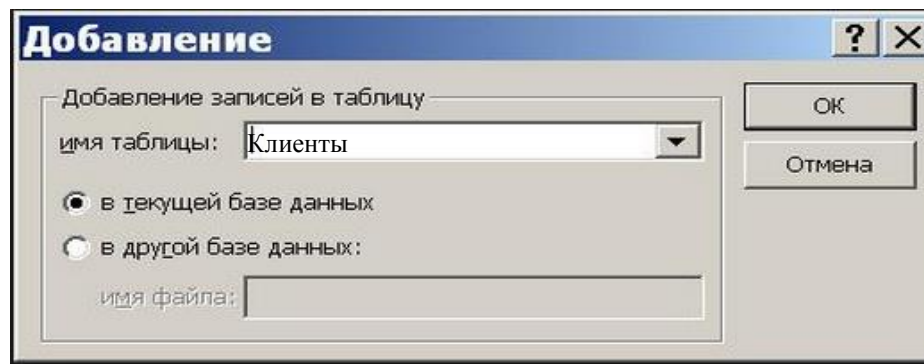
- если все поля в обеих таблицах имеют одинаковые имена, то можно просто перетащить знак «звездочка» (\*) в бланк запроса;
- для автоматического добавления значений Счетчика не следует при создании запроса перетаскивать поле счетчика в бланк запроса. В этом случае при добавлении записей значения в поле счетчика вставляются автоматически;
- если в обеих таблицах выделенные поля имеют одинаковые имена, то соответствующие имена автоматически вводятся в строку *Добавление*. Если имена полей двух таблиц отличны

друг от друга, нужно будет указать в строке **Добавление** имена полей таблицы-получателя.

4. В последнюю очередь добавляются *поля*, для которых предполагается указать *Условия* отбора.

5. Для *проверки* корректности выбранных полей, а также установленных условий отбора записей, используйте кнопку *Вид* .

6. Для *выполнения* запроса *Добавления* необходимо использовать кнопку  **Запуск**. При этом последовательно отображается информация о количестве записей, которые будут добавлены.



Перед запуском запросов *Добавления* не забудьте сделать копии изменяемых таблиц, чтобы можно было вернуть базу данных в первоначальное состояние.

Пример добавления записей о клиентах из *Украины* в таблицу *Клиенты* из таблицы *Контакты*:

Поле:	Фамилия	Имя	Страна
Имя таблицы:	КОНТАКТЫ	КОНТАКТЫ	КОНТАКТЫ
Сортировка:			
Добавление:	Фамилия	Имя	Страна
Условие отбора:			"Украина"

6. Создать **модифицирующие** запросы **Добавления** – **Мод\_4** и **Мод\_5**:

6.1. **Мод\_4** – запрос, позволяющий добавить в таблицу **Удаленные** данные о заказах, дата продажи, которых находится в промежутке от 1 до



30 сентября текущего года (после изменения типа запроса с помощью кнопки *Тип запроса/Добавление* в появившемся диалоговом окне необходимо задать Имя таблицы, в которую будут добавляться записи – **Удаленные**, и указать, что данная таблица находится в *текущей базе данных*). Обратите внимание на то, что записи, добавленные в таблицу **Удаленные** таким способом, не удаляются из таблицы **Заказы**, как это происходило при выполнении пунктов 3.3-3.5.

6.2. **Мод\_5** – запрос, позволяющий добавить из таблицы **Адресная\_Книга** базы данных **ТЕЛЕФОННО-АДРЕСНАЯ КНИГА** в таблицу **Клиенты** базы данных **ЗАКАЗЫ\_КЛИЕНТОВ** информацию обо всех клиентах, работающих в СП. Для этого:

6.2.1. В базе данных **ЗАКАЗЫ\_КЛИЕНТОВ** в таблице **Клиенты** отменить ключевое поле, а также в окне **Схемы данных** удалить связи таблицы **Клиенты** с таблицами **Заказы** и **Предприятие**.

6.2.2. В базе данных **ТЕЛЕФОННО-АДРЕСНАЯ КНИГА** во вкладке **Запросы** создать запрос **Добавления**. В появившемся диалоговом окне указать, что таблица находится в *другой базе данных*, выбрать имя базы данных **ЗАКАЗЫ\_КЛИЕНТОВ.mdb**, а затем *Имя таблицы*, в которую будут добавляться записи – **Клиенты**.

7. Ответить на контрольные вопросы.

### *Дополнительное задание*

Создать **модифицирующие** запросы, позволяющие *формировать* и *помесячно обновлять* сводную таблицу **Мес\_Итоги**, содержащую информацию об итоговых суммах продаж (в денежном выражении) по каждому виду товара за каждый месяц.



## ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И КОНТРОЛЯ ЗНАНИЙ

1. Определите последовательность действий при создании *много-табличных* запросов.
2. Перечислите запросы, в которых Вы использовали *логические* операторы AND/OR.
3. Определите назначение *Итоговых* запросов.
4. Как в бланк конструктора запросов ввести строку *Групповая операция*?
5. Перечислите типы операций, доступных в строке *Групповые операции* бланка запроса.
6. Назовите правила формирования имени поля в запросе, по которому выполняется *Групповая операция*.
7. Как можно задать условие выборки в *Итоговом* запросе? Приведите пример из лабораторной работы.
8. Как осуществляется включение в *Итоговый* запрос вычисляемого поля? Приведите пример из лабораторной работы.
9. Определите преимущества в использовании *Параметрических* запросов, приведите пример из лабораторной работы.
10. Дайте определение *Перекрестным запросам*? Как выполнить экспорт перекрестной таблицы?
11. Перечислите типы *Модифицирующих* запросов.
12. Определите последовательность действий при создании *Модифицирующих* запросов.
13. Каким должен быть запрос *Удаления*, чтобы он позволил полностью очистить содержимое таблицы?

14. В чем разница между режимом *Вид / Таблица* и *Запуск* запроса в модифицирующих запросах? Имела ли место данная разница при выполнении запросов *На выборку*?

15. Как создать копию таблицы, содержащую только структуру данных исходной таблицы?

16. Что изменяется в бланке запроса при выборе команд *Вид запроса / Удаление*, *Вид запроса / Добавление* и *Вид запрос / Обновление*?

## Лабораторная работа № 6. ОТОБРАЖЕНИЕ ИСХОДНЫХ ДАННЫХ И РЕЗУЛЬТАТОВ В MS ACCESS. ГЛАВНАЯ КНОПОЧНАЯ ФОРМА

### I. СОЗДАНИЕ ИЕРАРХИЧЕСКИХ ФОРМ



К **многотабличным** относят формы, построенные на основе запроса, объединяющего несколько таблиц, а также формы с внедренными в них другими формами.

**Иерархической** называется форма, представляющая собой комбинацию *главной* и *подчиненной* (внедренной) формы. Подчиненные формы особенно удобны для отображения данных из таблиц или запросов с отношением «*один-ко-многим*». В этом случае подчиненность заключается в том, что для каждой записи главной формы в подчиненной отражаются все записи с тем же значением *внешнего* ключа, которое имеет *первичный* ключ в главной форме.

Иерархическая форма может быть создана в режиме *Конструктора* или *Мастера* форм. Наиболее технологичным является первоначальное создание формы с помощью *Мастера* с последующей доработкой в режиме *Конструктора*.

Для создания формы в режиме *Мастера* необходимо во вкладке *Формы* выбрать *Создать / Мастер форм* и последовательно выполнить следующую последовательность операций:

1. Выбрать источник данных *главной* формы – таблицу или запрос (сторона «*один*» отношения «*один-ко-многим*»). Добавить необходимые поля из выбранной таблицы.

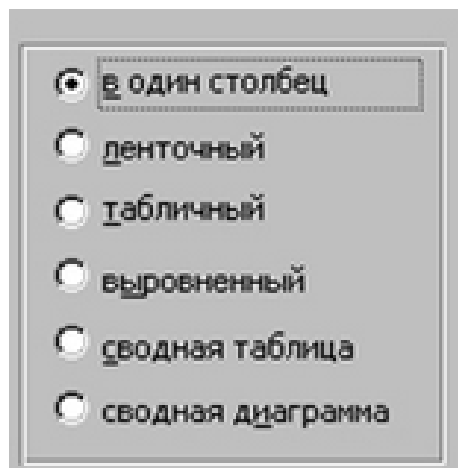
2. На той же странице выбрать источник данных *подчиненной* формы – таблицу или запрос (сторона «*много*» отношения «*один-ко-многим*»). Добавить необходимые поля из выбранной таблицы и нажать *Далее*.

3. На следующей странице Мастера появится сообщение *Выберите вид представления данных*, в котором необходимо выбрать таблицу или запрос, который представляет сторону «*один*» отношения «*один-ко-многим*».

Пример создания иерархической формы, содержащей информацию о заказах по каждому товару. В данном случае таблица *Товары* представляет сторону «*один*».

4. Установить в нижней части страницы мастера переключатель *Подчиненные формы* и нажмите кнопку *Далее*.

5. На странице *Выберите внешний вид подчиненной формы*: выбрать нужный вариант макета и нажмите кнопку *Далее*.



6. На последней странице *Мастера* ввести заголовки форм. Имена автоматически присваиваются формам на основе введенных заголовков, а надписи подчиненных форм — на основе введенных заголовков подчиненных форм.

7. Указать режим, в котором нужно открывать форму: режим формы (чтобы можно было просмотреть или ввести данные) или режим конструктора (чтобы можно было изменить ее структуру). Затем нажать кнопку *Готово*.

### Задания:

1. Запустить **Access**.
2. Открыть базу данных **ЗАКАЗЫ\_КЛИЕНТОВ**.
3. Создать **иерархическую форму** для связанных таблиц **Клиенты** и **Заказы**, имеющих отношение «*один-ко-многим*». Для этого выполнить следующие операции:
  - создать черновой вариант формы с помощью **Мастера форм** согласно предложенному макету:
  - использовать **Конструктор форм** для добавления в область *Примечания* подчиненной формы **Заказы** вычисляемых полей, позволяющих определить итоговые суммы по полям *Заказано* (=sum([Заказано]),) и *Продано* (=sum([Продано])).

The screenshot shows a Microsoft Access form titled 'Клиенты' (Clients) and 'Заказы' (Orders). The 'Клиенты' section contains fields for client information: Код клиента (68), Имя (Андрей), Индекс (43005), Страна (Украина), Город (Днепропетровский), Адрес (пл. Кирова, 89), Фамилия (Степанюк), Отчество (Иванович), Телефон (34-45-56), Факс (67-67-88), Предприятие (АО 'Астра'), and Возраст (33). The 'Заказы' section is a table with columns: Код товара, Дата заказа, Заказано, Дата продажи, Продано, and Цена. It contains three rows of data, with the first two rows having values and the third row being a blank template row with asterisks. The status bar at the bottom indicates 'Запись: 14 | 1 | 3 | из 7' and 'Режим формы'.

4. Для связанных таблиц **Поставщики** и **Товары**, имеющих отношение «*один-ко-многим*», в режиме Конструктора создать **иерархическую форму**. Для этого выполнить следующие операции:

- создать и спроектировать главную форму **Товары\_много**, используя в качестве источника данных поля таблицы **Товары**;
- создать и спроектировать форму подчиненную **Поставщики**, используя в качестве источника данных поля таблицы **Поставщики**;
- с помощью кнопки *Подчиненная форма* на *Панели элементов* определить местонахождение подчиненной формы **Поставщики** в главной форме **Товары\_много**;
- вызвав кнопку *Свойства* данного объекта во вкладке *Данные* в строке *Объект-источник*, указать имя подчиненной формы **Товары\_много**.

## II. СОЗДАНИЕ ОТЧЕТОВ



**Отчёт** – это объект базы данных, который используется для вывода на экран, в печать или файл структурированной информации. Отчёт позволяет извлечь из таблиц или запросов базы данных необходимую информацию и представить ее в виде удобном

для восприятия. Отчёт содержит заголовок, область данных, верхний и нижний колонтитулы.

В Microsoft Access можно создавать отчеты с использованием: Конструктора; Мастера отчетов; Автоотчета: в столбец; Автоотчета: ленточный; Мастера диаграмм; Почтовых наклеек.

### **Структура отчета в режиме Конструктора**

*Заголовок.* Этот раздел печатается только в верхней части первой страницы отчета. Используется для вывода таких данных, как текст заголовка отчета, дата или констатирующая часть текста документа, которые следует напечатать один раз в начале отчета. Для добавления или удаления области заголовка отчета необходимо выбрать в меню *Вид* команду *Заголовок/примечание отчета*.

*Верхний колонтитул.* Используется для вывода таких данных, как заголовки столбцов, даты или номера страниц, печатающихся сверху на каждой странице отчета. Для добавления или удаления верхнего колонтитула необходимо выбрать в меню *Вид* команду *Колонтитулы*. Microsoft Access добавляет верхний и нижний колонтитулы одновременно. Чтобы скрыть один из колонтитулов, нужно задать для его свойства *Высота* значение 0.

*Область данных.* Расположена между верхним и нижним колонтитулами страницы. Содержит основной текст отчета. В этом разделе появляются данные, распечатываемые для каждой из тех записей в таблице или запросе, на которых основан отчет. Для размещения в области данных элементов управления используют список полей и панель элементов. Чтобы скрыть область данных, нужно задать для свойства раздела *Высота* значение 0.

*Нижний колонтитул.* Этот раздел появляется в нижней части каждой страницы. Используется для вывода данных, таких как итоговые значения, даты или номера страницы, печатающихся снизу на каждой странице отчета.

*Примечание.* Используется для вывода данных, таких как текст заключения, общие итоговые значения или подпись, которые следует напечатать один раз в конце отчета. Несмотря на то, что в режиме *Конструктора* раздел "*Примечание*" отчета находится внизу отчета, он печатается над нижним колонтитулом страницы на последней странице отчета. Для добавления или удаления области примечаний отчета необходимо выбрать в меню *Вид* команду *Заголовок/примечание отчета*. Microsoft Access одновременно добавляет и удаляет области заголовка и примечаний отчета

*Заголовок группы.* Отображается в начале группы и обычно содержит название поля, по которому выполняется группировка.

*Примечание группы.* Отображается в конце группы и содержит итоговые значения (сумму, среднее и т.д.) по группе.

*Пример* Конструктора отчета, позволяющего отобразить информацию о продажах с группировкой по месяцам:

Объем продаж по месяцам : отчет		
☛ Заголовок отчета		
Объем продаж по месяцам		
☛ Верхний колонтитул		
Дата продажи - по месяцам	Дата продажи	Итого продано
☛ Заголовок группы 'Дата продажи'		
=Format\$([Дата продажи]; "mmmm")		
☛ Область данных		
	Дата продажи	Итого продано
☛ Примечание группы 'Дата продажи'		
="Итого для " & ""[Дата продажи]" & " " & [Дата продажи] & " (" & Count(*) & " " & If(Count(*)=1;"запись";"записей") & ""		
Sum		=Sum([Итого продан])
☛ Нижний колонтитул		
=Now()		"Страница " & [Page] & " из " & [Pages]
☛ Примечание отчета		
ИТОГО		=Sum([Итого продан])

В данном примере:



➤ содержимое *Заголовка отчета* будет отображаться только в начале первой страницы отчета:

☛ Заголовок отчета  
Объем продаж по месяцам

➤ содержимое *Верхнего колонтитула* (шапка таблицы) будет отображаться вверху каждой страницы отчета:

☛ Верхний колонтитул  
Дата продажи - по месяцам    Дата продажи    Итого продано

➤ содержимое *Заголовка группы* (текстовое название месяца) будет отображаться перед каждой группой – месяцем продаж:

☛ Заголовок группы 'Дата продажи'  
=Format\$([Дата продажи];"mmmm")

➤ содержимое *Области данных* (конкретная дата продажи и объем продаж) будет сгруппировано по месяцам:

☛ Область данных  
Дата продажи    Итого продано

➤ содержимое *Примечания группы* будет отображаться после окончания информации по месяцу продаж и содержать суммарное значение продаж по каждому месяцу:

☛ Примечание группы 'Дата продажи'  
="Итого для " & "'Дата продаж' = " & " " & [Дата продаж] & " (" & Count(\*) & " " & If(Count(\*)=1;"запись";"записей") & ")"  
Sum    =Sum([Итого продан])

➤ содержимое *Нижнего колонтитула* (текущая дата, номер и количество страниц) будет отображаться в конце каждой страницы отчета:

☛ Нижний колонтитул  
=Now()    ="Страница " & [Page] & " из " & [Pages]

➤ содержимое *Примечания отчета* (итоговая сумма продаж по всем месяцам) будет отображаться в конце отчета:

☛ Примечание отчета  
ИТОГО    =Sum([Итого продан])

## Задания:

### 1. Создать отчеты:

1.1. **Информация о заказах.** Для этого выполнить следующие операции:

- с помощью **Мастера отчетов** создать первоначальный вариант отчета;
- использовать **Конструктор отчетов** для добавления общего итога по полям *Заказано* и *Продано*. Для этого в области **Примечания отчета** добавить вычисляемые поля `=sum([Заказано])` и `=sum([Продано])`.

Фамилия	Имя	Отчество	Наименование товара	Заказано	Продано
Елисеев	Петр	Андреевич	Монитор	14	14
			Принтер	2	2
Кондратьев	Павел	Викторович	Клавиатура	21	21
				3	2
Степанов	Андрей	Иванович	Монитор	2	1
			Клавиатура	10	10
Федоров	Олег	Георгиевич	Клавиатура	2	2

1.2. **Вып\_заказы.** Для этого выполнить следующие операции:

- проверить, чтобы в запросе **Вып\_заказ** было поле *Цена*;
- создать пустой бланк отчета, выбрав в качестве источника данных запрос **Вып\_заказ** (вкладка **Отчеты / Создать / Конструктор**);
- в **Области данных** создать вычисляемое поле, позволяющее сформировать из полей *Фамилия*, *Имя* и *Отчество* единое поле (`= [Фамилия] + " " + [Имя] + " " + [Отчество]`). В соответствующую область **Верхнего колонтитула** поместить *Надпись*, содержащую текст "ФИО";

**Информация о выполненных заказах**

Наименование товара	ФИО	Сумма продано	
		Количество	\$ (долл.)
Клавиатура	Федоров Олег Георгиевич	2	64
	Кондратьев Павел Викторович	21	840
	Степанов Андрей Иванович	10	350
<b>Итого</b>		<b>33</b>	<b>1254</b>
Монитор	Елисеев Петр Андреевич	14	2184
<b>Итого</b>		<b>14</b>	<b>2184</b>
Принтер	Елисеев Петр Андреевич	2	1600
<b>Итого</b>		<b>2</b>	<b>1600</b>
<b>Итого по всем товарам</b>		<b>49</b>	<b>5038</b>

Страница: 1  
Готово

- переместить в **Область данных** все остальные поля из **Списка полей** (кроме поля **Цена**), расположив их горизонтально в одну линию, после чего *надписи* данных полей переместить в область **Верхнего колонтитула**;
- для поля *Продано* создать дополнительное вычисляемое поле, позволяющее определить его значение в денежном выражении ( $=[\text{Продано}] * [\text{Цена}]$ );
- определить группировку по полю *НаименованиеТовара* (кнопка **Сортировка и группировка**), а также включить вывод на экран области **Примечания группы** (в окне **Группировка и сортировка** в Свойстве **Примечание группы** установить режим **Да**);
- отключить вывод повторяющихся значений поля *Наименование товара* (кнопка **Свойства** данного объекта / вкладка **Макет**, в строке **Не выводить повторы** установить режим **Да**);
- в области **Примечания группы** и **Примечания отчета** добавить вычисляемые поля  $=\text{sum}([\text{Продано}])$ ,  $=\text{sum}([\text{Продано}] * [\text{Цена}])$ .

### III. СОЗДАНИЕ ГЛАВНОЙ КНОПОЧНОЙ ФОРМЫ



**Главная кнопочная форма** создается с целью навигации по базе данных, т.е. она может использоваться в качестве главного меню БД. Элементами главной кнопочной формы являются объекты форм и отчетов.

Запросы и таблицы не являются элементами главной кнопочной формы. Поэтому для создания кнопок Запросы или Таблицы на кнопочной форме можно использовать макросы. Сначала в окне базы данных создают макросы *Открыть Запрос* или *Открыть Таблицу* с уникальными именами, а затем в кнопочной форме создают кнопки для вызова этих макросов.

Для одной базы данных можно создать несколько кнопочных форм. Кнопки следует группировать на страницах кнопочной формы таким образом, чтобы пользователю было понятно, в каких кнопочных формах можно выполнять определенные команды (запросы, отчеты, ввода и редактирования данных). На подчиненных кнопочных формах должны быть помещены кнопки возврата в главную кнопочную форму.

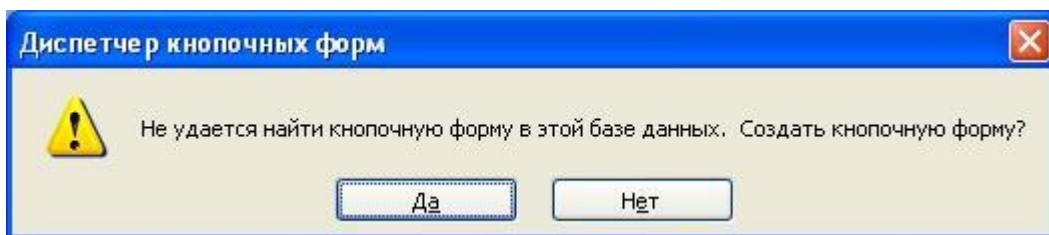
Технология создания *Кнопочных форм* следующая:

- 1) Создать страницу *Главной кнопочной формы*.
- 2) Создать необходимое количество страниц *подчиненных кнопочных форм* (например, формы для ввода данных, для отчетов, для запросов и т.д.).
- 3) Создать *элементы* главной кнопочной формы.
- 4) Создать *элементы* для кнопочных форм отчетов и форм ввода или изменения данных.
- 5) Создать *макросы* для запросов или для таблиц с уникальными именами.
- 6) Создать *элементы* для кнопочных форм запросов или таблиц.

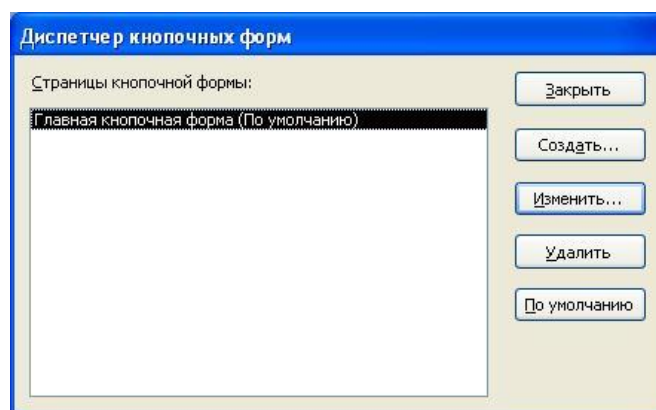
Структура кнопочных форм может быть представлена в следующем виде:



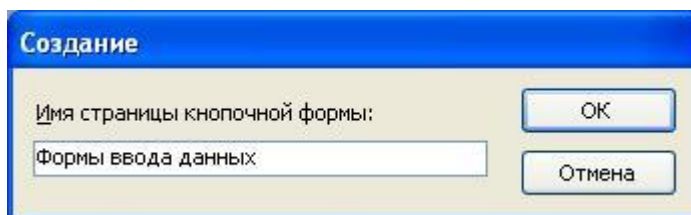
Для создания *Главной кнопочной формы* и ее элементов необходимо открыть базу данных (например, **Заказы**) и выполнить команду *Сервис / Служебные программы / Диспетчер кнопочных форм*. Если кнопочная форма ранее не создавалась, то откроется окно диалога:



В результате будет создана страница *Главной кнопочной формы*.

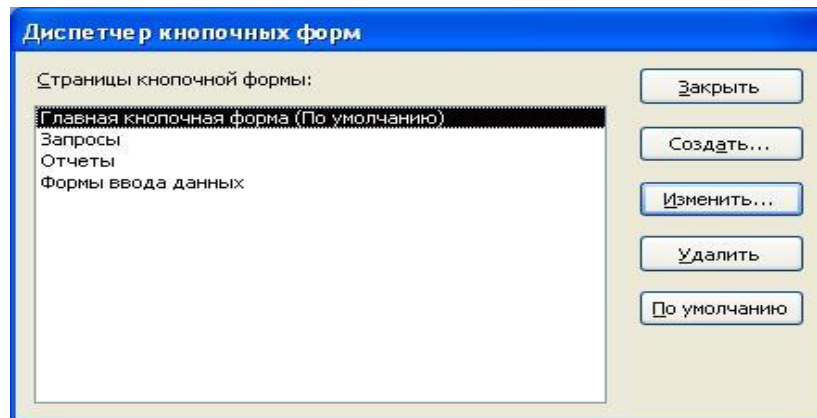


Далее можно создать еще три страницы кнопочной формы: *Формы ввода данных*, *Отчеты* и *Запросы*. Для этого следует щелкнуть



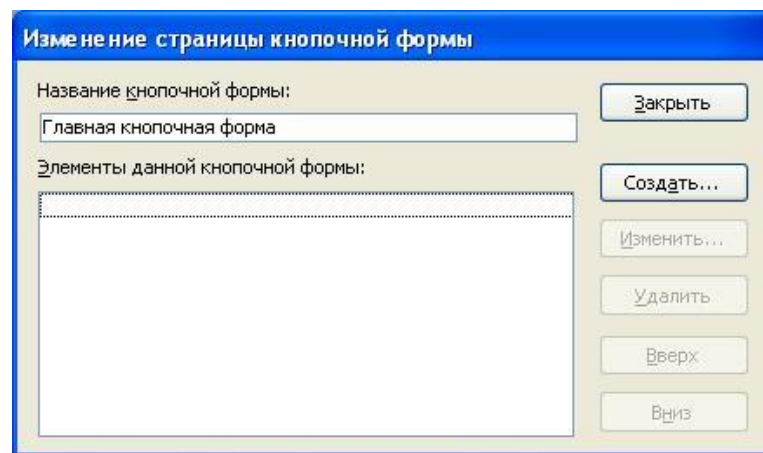
на кнопке *Создать* и в появившемся окне ввести имя новой страницы *Формы ввода данных*.

Аналогичным образом надо создать еще две страницы, в итоге получим четыре страницы кнопочных форм, которые отображаются в окне *Диспетчер кнопочных форм*.

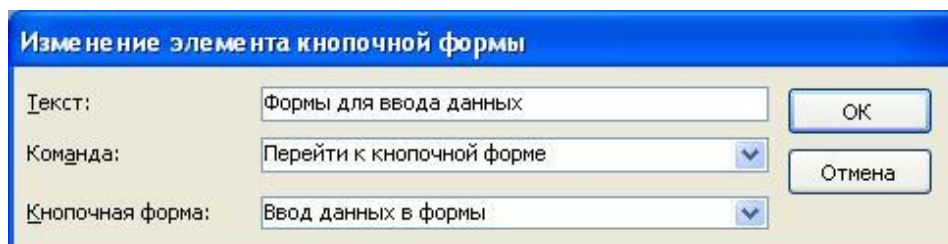


После этого создаем элементы *Главной кнопочной формы*.

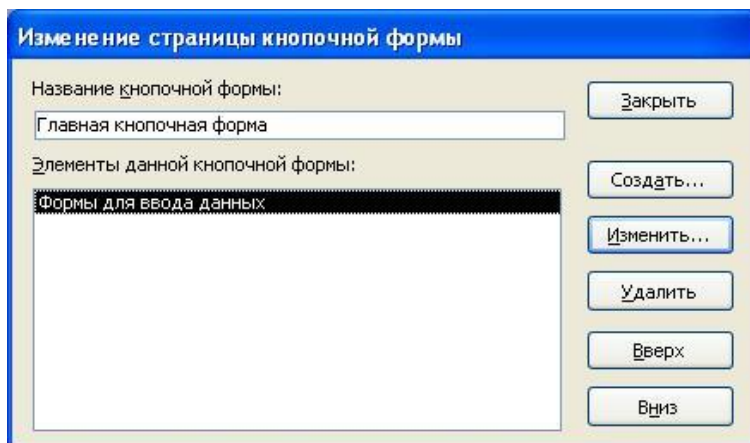
Для этого в *Окне диспетчер кнопочных форм* выделяем страницу *Главная кнопочная форма / Изменить / Изменение страниц кнопочной формы*.



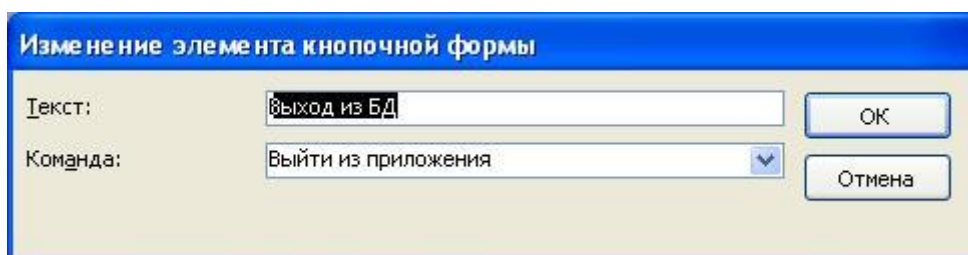
В этом окне щелкаем *Создать* и в окне *Изменение элемента кнопочной формы* выполняем следующие действия:



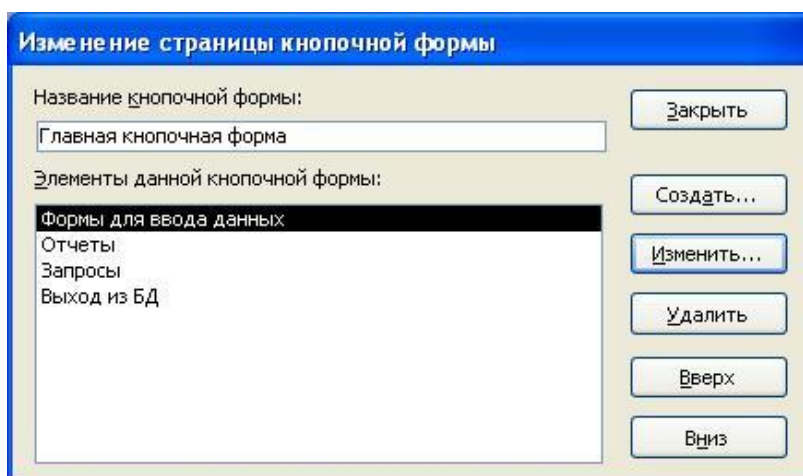
- вводим название *Элемента кнопочной формы* (например, *Открытие формы для ввода данных*);
- выбираем команду (например *Перейти к кнопочной форме*);
- выбираем название открываемой кнопочной формы (например, *Ввод данных в формы*).



Аналогичным методом можно создать элементы *Запросы* и *Отчеты*, а затем элемент (кнопку) *Выход из БД*.

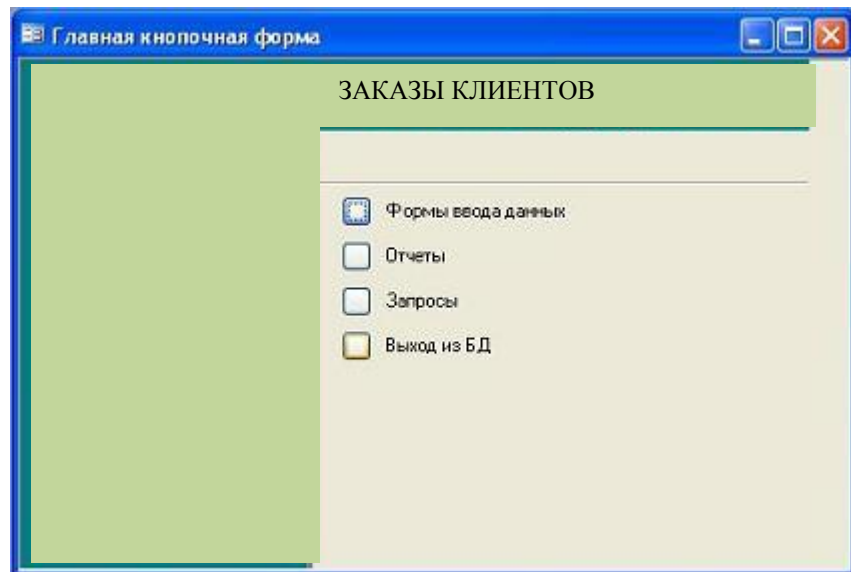


В результате в окне *Изменение страницы кнопочной формы* будут отображаться все элементы главной кнопочной формы.

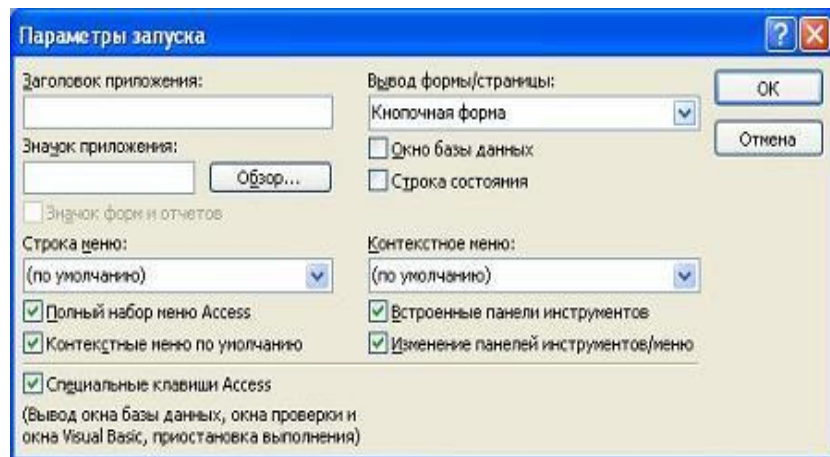


Аналогично создаются элементы кнопочных форм *Формы ввода данных*, *Отчеты*, *Запросы*.

Кнопочная форма появится в списке в области окна базы данных на вкладке *Формы* на панели *Объекты*, а на вкладке *Таблицы* в списках появится таблица *Switchboard Items*.



Для того чтобы *Главная кнопочная форма* автоматически запускалась при запуске базы данных, необходимо в окне *Сер-*



*вис/Программы запуска* выбрать *Кнопочная форма* из раскрывающегося списка и снять флажки *Окно базы данных* и *Строка состояния*.

При запуске базы данных при нажатой клавише *Shift* база данных будет открываться без автоматического запуска *Главной кнопочной формы*.

### Задания:

1. Перейти во вкладку **Формы** окна базы данных и вызвать **Диспетчер кнопочных форм** (пункт меню **Сервис / Служебные программы / Диспетчер кнопочных форм**).



2. Создать кнопочные формы с именами страниц кнопочной формы: **Исходные данные, Запросы, Отчеты** (кнопка **Создать** в окне **Диспетчера кнопочных форм**).

3. Создать элементы страницы кнопочной форм **Исходные данные** (кнопка **Изменить** окна **Диспетчера кнопочных форм** / кнопка **Создать** окна **Изменить страницу кнопочной формы**):

а) элемент **Клиенты** (Текст – *Клиенты*, Команда – *Открытие формы в режиме редактирования*, Форма – *Клиенты*);

б) элемент **Поставщики** (Текст – *Поставщики*, Команда – *Открытие формы в режиме редактирования*, Форма – *Поставщики*);

с) элемент **Товары** (Текст – *Товары*, Команда – *Открытие формы в режиме редактирования*, Форма – *Товары*);

д) элемент **Заказы** (Текст – *Заказы*, Команда – *Открытие формы в режиме редактирования*, Форма – *Заказы*);

е) элемент **Выход** (Текст – *Выход*, Команда – *Переход к кнопочной форме*, Форма – *Главная кнопочная форма*)

4. Для кнопочной формы с именем страницы **Отчеты** создать аналогичные элементы кнопочных форм, вызывающие созданные ранее отчеты.

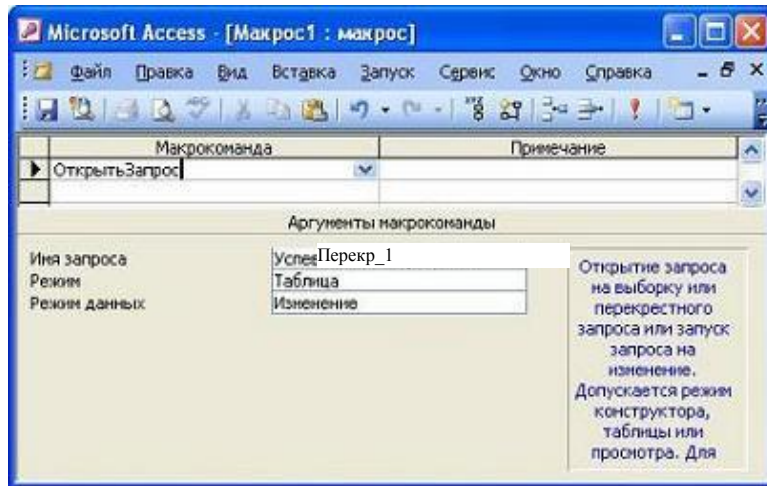
5. Для создания кнопочных форм с именем страницы кнопочной формы **Запросы**:

а) в окне **Диспетчера кнопочных форм** создать кнопочные формы с именами страниц кнопочной формы: **Многотабличные запросы, Итоговые запросы, Параметрические запросы, Перекрестные запросы**;

б) для кнопочной формы **Запросы** создать элементы кнопочных форм, вызывающих данные кнопочные формы и кнопку возврата в главную форму;



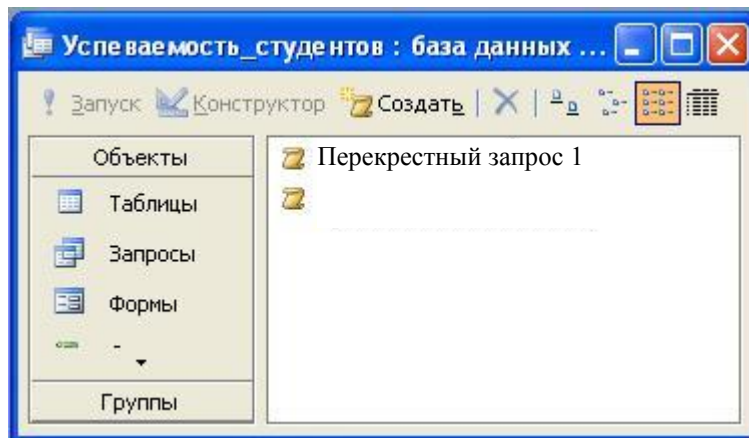
Поскольку в *Главной кнопочной форме* не предусмотрена команда отображения *Запросов*, для их добавления в кнопочную форму необходимо создать макросы. Для этого используется вкладка *Макросы* команды *Создать / Макрос*. В открывшемся окне выбрать макрокоманду *Открыть Запрос* и из раскрывающегося списка выберите имя запроса – например *Перекр\_1*:



Затем щелкнуть *Закреть*. В появившемся окне *Сохранить изменение макета или структуры объекта Макрос1?* ответить *Да* и дать название Макросу – например *Перекрестный запрос 1*:



Имя макроса сохранится в списке макросов в окне базы данных.



с) Для вызова *Параметрических* и *Перекрестных* запросов создать *Макросы*, позволяющие открыть данные запросы в режиме таблицы. Для этого:

- закрыть окно **Диспетчера кнопочных форм**;
- перейти во вкладку **Макросы** окна базы данных;
- последовательно создать макросы для выполнения **Макрокоманды ОткрытьЗапрос** с аргументами, определяющими имена созданных Вами *параметрических* и *перекрестных* запросов в поле **Имя запроса** (кнопка **Создать** вкладки **Макросы**);

- вернуться в **Диспетчер кнопочных форм** (вкладка **Формы** окна базы данных / **Кнопочная форма** / **Сервис** / **Надстройки** / **Диспетчер кнопочных форм**);

- для форм **Параметрические запросы** и **Перекрестные запросы** создать элементы кнопочных форм, выполняющие команды запуска созданных Вами макросов (кнопка *Изменить* окна *Диспетчера кнопочных форм* / кнопка *Создать* окна *Изменить страницу кнопочной формы*);

d) для вызова *итоговых* и *многотабличных* запросов:

- закрыть окно **Диспетчера кнопочных форм**;
- создать формы для каждого из созданных Вами ранее запросов (кнопка *Автоформы* на панели инструментов);

- вернувшись в **Диспетчер кнопочных форм**, создать элементы кнопочных форм для форм: **Многотабличные запросы** (вызывающие формы многотабличных запросов), и **Итоговые запросы** (вызывающие формы итоговых запросов);

e) переименовать страницу кнопочной формы **Главная кнопочная форма** в **Главное меню** (кнопка **Изменить**) и создать элементы данной кнопочной формы (кнопка **Создать**), выполняющие команды перехода к кнопочным формам **Исходные данные**, **Запросы**, **Отчеты**, а также

команду выхода из приложения, позволяющую закрыть базу данных **ЗАКАЗЫ\_КЛИЕНТОВ**.

6. Определить *Параметры запуска* базы данных **ЗАКАЗЫ\_КЛИЕНТОВ**, задав режим автоматической загрузки при открытии файла **Кнопочной формы**, а также отключить загрузку окна базы данных и строки состояния (пункт меню **Сервис / Параметры запуска**).

7. Войти в *Конструктор Кнопочной формы* и добавить рисунок (кнопка *Рисунок* на Панели элементов), установив размеры в окне **Свойства** "по размеру рамки" (**Свойства / Макет / Установка размеров**).

8. Закрыть базу данных, а затем загрузить повторно (с и без использования клавиши Shift).

9. Ответить на контрольные вопросы.



#### ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И КОНТРОЛЯ ЗНАНИЙ

1. Определите преимущества и недостатки создания иерархических форм с использованием *Мастера* форм.

2. Как создать *подчиненную* форму? Для каких типов отношений между таблицами возможно созданием подчиненной формы?

3. Определите понятие *Отчета* в Access. Перечислите его основные отличия от *Формы*.

4. Перечислите основные этапы создания *Отчетов* с помощью *Мастера* в Access.

5. Перечислите основные *Области отчета* и их назначение.

6. Что такое *Группировка* в отчете? Приведите примеры ее применения.

7. Как вывести на экран области *Заголовка / Примечания группы*?

8. Как установить *Сортировку* записей в *Отчете*?

9. Чем отличаются в Access *Табличные отчеты* от отчетов в *Свободной форме*?

10. Основное назначение *Главной кнопочной формы*.
11. Как установить автоматический запуск *Главной кнопочной формы* при открытии базы данных? Как осуществить запуск базы данных без автоматического запуска *Главной кнопочной формы*?
12. Дайте определение *Макроса*. Приведите пример создания *Макроса* в Access.

Навчальне видання

## ІНФОРМАТИКА

### РОЗДІЛ «СИСТЕМИ УПРАВЛІННЯ БАЗАМИ ДАНИХ»

Лабораторний практикум

(російською мовою)

Комп'ютерна верстка О.М. Гришкіної

---

Підписано до друку 5.02.2014. Формат 60×84/16.

Ум. друк. арк. 5,35. Тираж 100 пр. Зам. № .

---

ПВНЗ «Дніпропетровський університет імені Альфреда Нобеля».

49000, м. Дніпропетровськ, вул. Набережна В.І. Леніна, 18.

Тел. (056) 778-58-66, e-mail: gio@duer.edu

Свідоцтво ДК № 4611 від 05.09.2013 р.

Віддруковано у ТОВ «Роял Принт».

49052, м. Дніпропетровськ, вул. В. Ларіонова, 145.

Тел. (056) 794-61-05, 04

Свідоцтво ДК № 4121 від 27.07.2011 р.