

УДК 681.3.06 (072)  
6Н1

Методические указания к выполнению лабораторной работы по дисциплине "Компьютерная подготовка" с применением СУБД ACCESS. Выпуск 1. / Сост.: Т.В. Бугреева, И.А. Ильина, А.Е. Макаренко, Е.В. Лемешко, М.В. Седун; ГУУ. – М., 2002. – 32 с.

Составители  
старший преподаватель  
Т.В. БУГРЕЕВА  
старший преподаватель  
И.А. ИЛЬИНА  
кандидат технических наук, доцент  
А.Е. МАКАРЕНКО  
кандидат экономических наук, доцент  
Е.В. ЛЕМЕШКО  
старший преподаватель  
М.В. СЕДУН

Ответственный редактор  
заведующий кафедрой компьютерных технологий,  
кандидат экономических наук, доцент  
В.А. МАШУРЦЕВ

Рецензент  
доцент кафедры прикладной математики ГУУ  
И.С. КАРАНДАЕВ

© Колл. авторов под ред. В.А. Машурцева, 2002  
© Государственный университет управления, 2002

## ВВЕДЕНИЕ

Настоящие методические указания посвящены вопросам создания баз данных (Тема 5 программы учебной дисциплины « Компьютерная подготовка») и программированию на языке запросов (Тема 15 этой программы). По этим темам разработана лабораторная работа, соответствующая работе 10 раздела 1 той же программы. Работа состоит из нескольких частей и рассчитана на 12 учебных часов. Она позволяет студентам освоить работу в системе управления базами Access.

Система управления базами данных **Access** входит в стандартный набор прикладных программ пакета **Microsoft Office** и является одним из самых удобных и надежных программных продуктов.

**База данных** – это набор записей и файлов, которые организованы специальным образом. Базой данных могут быть документы, созданные с помощью текстовых редакторов и сгруппированные по темам, или файлы с электронными таблицами, объединенные в группы по способу их использования. База данных может состоять из файлов, имеющих постоянную для данного файла структуру записи.

**Запись** – это информация об одном объекте. В файле всегда много записей, количество которых заранее неизвестно. Запись состоит из полей.

**Поле** – это составляющая записи, которая описывает один атрибут объекта. В записи может быть много полей (не более 255).

**Система управления базой данных** – это совокупность программных средств, позволяющих создавать, модифицировать и контролировать базу данных, осуществлять различные выборки информации, готовить их к просмотру или печати.

Знакомство с Access следует начать с изучения его основных объектов, которыми являются **таблицы и запросы**.

Таким образом, методические указания позволяют студентам постепенно освоить работу с СУБД Access.

## ЛАБОРАТОРНАЯ РАБОТА

**Часть 1. Основные задачи части 1:**

- познакомиться с основными понятиями базы данных;
- научиться создавать таблицу;
- освоить режимы работы с таблицей;
- освоить основные приемы редактирования таблиц;
- освоить простую сортировку таблиц;
- научиться создавать копии таблиц или удалять таблицы.

### Основные понятия

**Таблица** – это объект базы данных, который содержит информацию определенной структуры, имеющую общую логику ее обработки. Таблица – это один из файлов базы данных. База данных может содержать несколько таблиц. Таблица состоит из строк и столбцов. Строка таблицы – это **запись**, столбец – **поле** записи. Записи в таблице должны быть уникальными и описывать объекты одного типа.

### Создание базы данных

Осуществляется открытием окна **MS Access**, выбором переключателя Новая база данных, вводом в поле Имя базы данных любого соответствующего содержанию базы данных значения, например, "Фирма". Если мы уже находимся в системе Access то: **Файл – Создать базу данных – Имя базы данных**.

Считается, что база данных существует, если определено ее имя.

### Создание таблицы

Таблица имеет два режима: режим **конструктора** и режим **таблицы**. Переход из режима конструктора в режим таблицы можно осуществить с помощью кнопки **Вид** на панели инструментов или из окна базы данных, выбрав режим **Открыть**. В

режиме конструктора описываются поля таблицы, в режиме таблицы осуществляется заполнение полей таблицы конкретными значениями. Таблица считается созданной, если описаны ее поля и задано ее имя.

Для того чтобы создать таблицу нужно в окне базы данных выбрать вкладку **Таблица – Создать – Конструктор** – далее следует описать поля.

**Описать поле** – это значит определить его имя и тип. Кроме этого для каждого поля можно задать ряд характеристик, влияющих на размер памяти (**размер поля**), контроль вводимых значений (**условие на значение**), сделать содержательными заголовки столбцов (**подпись**). Использование подпери подписи позволяет имена полей делать короткими и определять латинскими буквами. Для текстовых полей или дат можно задать формат ввода (**маска ввода**), можно определить свое сообщение при вводе неверных данных (**сообщение об ошибке**). Эти характеристики полей называются свойствами, список которых выводится для каждого поля в нижней части конструктора таблицы.

**Маску ввода** можно создать с помощью мастера. Чтобы вызвать мастера масок таблица должна находиться в режиме конструктора. Щелчком мыши выделяется нужное поле, на вкладке **Общие** области **Свойства поля**, расположенной в нижней части окна конструктора, щелкнуть в строке **Маска ввода**, справа от нее появится кнопка с тремя точками – **построитель**: щелкните на этой кнопке и, прокрутив список доступных масок, выберете нужную.

Для полей, содержимое которых в пределах одной задачи ограничено, можно задать список, значения которого выбираются при заполнении таблицы. В нашей задаче такими полями могут быть: «firm», «izd» и «kod», так как таблица содержит информацию об определенных фирмах и товарах. Для создания полей со списком в графе **Тип данных** выбираем значение **Мастер подстановок** и, пройдясь по окнам мастера, вводим фиксированный набор значений для вышеперечисленных полей. Если мы уже создали поле со списком «izd» (название изделия), то для заполнения поля «kod» (код изделия) создадим список с двумя столбцами: код и название, но в поле таблицы будет записываться только одно – код.

Создадим таблицу следующей структуры: (табл. 1.1).

Таблица 1.1

Фирма	Изделие	Код	Цены по кварталам				Годовой выпуск
			1 кв	2 кв	3 кв	4 кв	

Для этого в окне базы данных выберем объект **Таблица** и режим **Конструктор**, в котором опишем следующие поля (табл. 1.2).

Таблица 1.2

Имя поля	Тип данных	Примечания	Свойства поля
1	2	3	4
Firm	Мастер подстановок	Название фирмы	<i>Размер: 20 байт</i> <i>Подпись: Фирма</i> <i>Подстановка</i> <i>Тип эл-та: поле со списком</i> <i>Источник строк: «Меридиан», «Техно», «Стройдвор», «Агат», «Топаз»</i>
Izd	Мастер подстановок	Название изделия	<i>Размер: 20 байт</i> <i>Подпись: Изделие</i> <i>Подстановка</i> <i>Тип эл-та: поле со списком</i> <i>Источник строк: «Телевизор»; «Видеомагнитофон»; «Обогреватель»; «Электрочайник»; «Магнитофон»; «Пылесос»; «Вентилятор»</i>

Продолжение табл. 1.2

1	2	3	4
Kod	Мастер подстановок	Код изделия	<i>Размер: длинное целое</i> <i>Подпись: Код изделия</i> <i>Подстановка</i> <i>Тип эл-та: поле со списком</i> <i>Источник строк: 123; «Телевизор»; 345; «Видеомагнитофон»; 234; «Обогреватель»; 567; «Электрочайник»; 456; «Магнитофон»; 333; «Пылесос»; 222; «Вентилятор»</i>
C1	Числовой	Цена в 1 квартале	<i>Размер: длинное целое</i> <i>Подпись: 1 кв</i>
C2	Числовой	Цена во 2 квартале	<i>Размер: длинное целое</i> <i>Подпись: 2 кв</i>
C3	Числовой	Цена в 3 квартале	<i>Размер: длинное целое</i> <i>Подпись: 3 кв</i>
C4	Числовой	Цена в 4 квартале	<i>Размер: длинное целое</i> <i>Подпись: 4 кв</i>
God	Числовой	Годовой выпуск	<i>Размер: длинное целое</i> <i>Подпись: Годовой выпуск</i>

По окончании описания полей таблицу необходимо сохранить: **Файл – Сохранить** или на панели инструментов щелкнуть по кнопке **«дискета»**.

Если вы забыли сохранить таблицу, то при переходе в режим таблицы, появится диалоговое окно, которое запросит подтверждение сохранения таблицы. При этом будет предложено ввести имя таблицы или оно будет дано автоматически как Таблица 1. Для нашей таблицы задано имя **«Фирма»**.

**Заполнение таблицы** осуществляется в режиме таблицы. При заполнении можно попробовать вводить ошибочные значения для полей с указанными условиями на значение и увидеть сообщения о "своих" ошибках. По окончании ввода (не менее 10 записей) данные следует сохранить. Сохранение следует проводить после любого изменения в любом объекте базы данных.

После заполнения таблица «Фирма» примет вид (рис. 1.1)

Фирма	изделие	код	1 кв	2 кв	3 кв	4 кв	год вып
Меридиан	телевизор	123	300	300	300	300	6 000,00р.
Техно	пылесос	234	100	150	150	150	4 000,00р.
Техно	телевизор	123	400	400	400	400	7 000,00р.
Стройдвор	магнитофон	456	200	200	200	230	4 000,00р.
Техно	электрочайник	567	100	100	110	110	5 000,00р.
Техно	магнитофон	456	900	900	950	1000	8 000,00р.
Меридиан	магнитофон	456	150	160	165	165	4 000,00р.
Стройдвор	телевизор	123	350	400	400	400	6 000,00р.
Стройдвор	электрочайник	567	500	500	650	650	8 000,00р.
Агат	телевизор	123	320	320	330	330	7 000,00р.
Меридиан	пылесос	333	200	230	220	210	5 000,00р.
Топаз	пылесос	333	200	195	210	200	5 000,00р.
Топаз	магнитофон	456	120	126	100	110	5 000,00р.

Рис. 1.1.

### Редактирование таблицы

Редактирование таблицы включает как изменение внешнего вида таблицы, так и ее содержания.

**Изменение ширины столбца** осуществляется путем растягивания мышью правой границы столбца или с помощью меню: **Формат – ширина столбца** – по ширине данных, стандартная или ввести вручную.

**Изменение высоты строк** осуществляется путем растягивания мышью или с помощью меню **Формат – Высота строки** – стандартная или ввести вручную.

**Изменение шрифта** можно осуществить с помощью меню **Формат – Шрифт** – выбрать название шрифта, начертание, размер, атрибуты и набор символов.

Для **изменения цвета ячейки** её нужно выделить, затем **Меню – Формат – Ячейки – Цвет фона**.

Чтобы **изменить порядок столбцов** нужно выделить столбец, который хотим переместить, и с помощью мыши «перетащить» его в нужное место.

Для **переименования полей** следует выполнить следующую последовательность действий: **Меню – Формат – Переименовать столбец** – написать новое имя или с помощью двойного щелчка мышью на названии столбца удалить старое и написать новое, или в режиме конструктора изменить свойство подпись для изменяемого поля.

Если при просмотре таблицы некоторые поля не нужны, то их можно **скрыть** с помощью меню **Формат – Скрыть столбец**. Чтобы **Отобразить столбцы**, нужно поставить "v" напротив того поля, которое нужно отобразить, или перетащить правую границу столбца влево до его исчезновения.

**Вставить поле** можно с помощью меню **Вставка – столбец**. При добавлении поля необходимо ввести его имя или в режиме конструктора описать новое поле.

**Удалить столбец** можно с помощью меню, выполнив следующие действия: **Правка – Удалить столбец**, предварительно его выделив, или в режиме конструктора выделить строку и удалить, либо с помощью контекстного меню – **Вырезать**, предварительно выделив столбец.

**Вставить запись** можно с помощью меню, выполнив следующие действия: **Вставка – Новая запись**.

**Добавить запись** можно просто заполнив пустую последнюю строку.

**Удалить запись** можно с помощью меню **Правка – Удалить запись** или на панели инструментов нажать кнопку "Ножницы", предварительно запись следует выделить.

### Сортировка данных

В режиме таблицы можно сортировать данные по какому-либо столбцу. Для этого нужно выделить столбец и в меню выбрать режим **Записи – Сортировка – по возрастанию/убыванию**. Более легким способом является использование панели инструментов – кнопка «**Сортировка по возрастанию/убыванию**»

### Копирование таблицы

В режиме таблицы можно копировать любые значения столбцов или строк, используя в меню команды **Правка – Копировать**, а затем **Правка – Вставить**.

Копировать можно целиком таблицы, но для того чтобы производить операции над таблицами необходимо вернуться в окно базы данных.

Для создания копии таблицы требуется выделить имя таблицы, которую вам необходимо копировать, затем в меню выполнить следующие действия **Правка – Копировать** и **Правка – Вставить**, а затем ввести имя новой таблицы.

Для **удаления таблицы** необходимо выделить в Окне базы данных имя таблицы, а затем с помощью меню выполнить следующие действия **Правка – Удалить**.

## Часть 2. Основные задачи части 2.

### Использование фильтров:

поиск по образцу  
поиск по выделенному  
простой фильтр  
расширенный фильтр

### Создание запросов:

запрос с условиями (Like, Between, In)  
запрос на выборку всех, кроме  
запрос с параметром

### Основные понятия

**Фильтр** – это режим, который позволяет осуществлять выбор записей из таблицы по различным параметрам, задаваемым с помощью выделения конкретного значения какого-либо поля или задания условий и значений нескольких полей таблицы. В результате работы фильтра мы получаем динамический набор записей, соответствующий условиям, но само определение фильтра и условия поиска не сохраняются и, если выбор необходимо повторить, то фильтр организуется заново. Фильтры работают с таблицей, открытой в режиме **Таблица**.

**Запрос** – это объект базы данных, который позволяет выбирать информацию из таблицы или из других запросов, проводить вычисления и создавать новые поля в динамических наборах записи. Запросы являются объектом, так как сохраняются в базе данных и, в отличие от фильтров, могут выполняться многократно. Запросы имеют три режима: режим **конструктора**, режим **SQL**, режим **таблицы**.

В режиме конструктора запрос формируется в виде окна запроса, состоящего из двух частей. В верхней части выводятся поля таблиц, используемых в запросе, или других запросов. В нижней части перечисляются поля и выражения, выводимые в динамический набор записей. Для каждого поля можно указать условия отбора, вывод на экран этого поля, использование его в сортировке.

### Использование фильтров

**Задача 2.1.** Используя **поиск по образцу** найти информацию об изделии «телевизор».

Для этого необходимо выделить нужное поле в таблице, затем в главном меню выбрать команду **Правка – Найти**, либо нажать кнопку **Найти (бинокль)** на панели инструментов. В появившемся окне в поле **Образец** ввести значение **телевизор**, затем нажать кнопку **Найти**. В результате в таблице будет найдена первая запись с указанным значением в выделенном поле. Вместо полного названия можно ввести строку, состоящую из шаблонов: "\*", "?". При этом "\*" используется для замены строки символов произвольной длины, а "?" для замены одного символа. Например, можно в поле **Образец** ввести "т\*" вместо полного значения телевизор. Нажав кнопку **Поиск далее**, можно найти следующую запись с тем же условием. Найденные поля выделяются в исходной таблице.

### Поиск по выделенному.

Предыдущую задачу можно решить, используя **Фильтр по выделенному**. Для этого в таблице нужно выделить значение **телевизор**. Затем в главном меню выбрать команду **Записи – Фильтр по выделенному**.

В обоих случаях мы получим следующий результат, представленный на рис. 2.1.

Фирма	изделие	код	1кв	2кв	3кв	4кв	год вып
Меридиан	телевизор	123	300	300	300	300	6 000,00р.
Техно	телевизор	123	400	400	400	400	7 000,00р.
Стройдвор	телевизор	123	350	400	400	400	6 000,00р.
Агат	телевизор	123	320	320	330	330	7 000,00р.

Рис. 2.1

### Простой фильтр

Для поиска записей по значениям нескольких полей нужно использовать простой фильтр. Включение режима простого фильтра осуществляется командами главного меню **Запись – Фильтр – Изменить фильтр**. На экране появится таблица с одной пустой строкой, в каждый столбец которой можно ввести условие отбора для соответствующего поля.

**Задача 2.2.** Вывести информацию об изделиях фирмы "Техно" с годовым выпуском более 4000 руб.

В фильтре введем соответствующие условия в столбцы **Фирма** и **год** (рис. 2.2).

Фирма	изделие	код	1кв	2кв	3кв	4кв	год вып
Техно							>4000

Рис. 2.2

Для включения фильтра необходимо нажать кнопку **Применить фильтр** на панели инструментов. Результат для нашей таблицы будет следующим (рис. 2.3).

Фирма	изделие	код	1кв	2кв	3кв	4кв	год вып
Техно	телевизор	123	400	400	400	400	7 000,00р.
Техно	электрочайник	567	100	100	110	110	5 000,00р.
Техно	магнитофон	456	900	900	950	1000	8 000,00р.

Рис. 2.3

Для отключения фильтра надо еще раз нажать кнопку **Применить фильтр**, которая превратилась в кнопку **Удалить фильтр**.

### Расширенный фильтр.

Позволяет задавать более сложные условия, включая сортировку.

В фильтр можно включать параметр, который позволяет вводить любое, из имеющихся в таблице значение для отбора записей в процессе выполнения фильтра.

**Задача 2.3.** Для произвольно заданного изделия вывести список предприятий в алфавитном порядке, которые не повышали в течение года цену на это изделие.

В данной задаче параметром будет любое, вводимое с клавиатуры, название изделия. Для использования **расширенного фильтра** в главном меню надо выбрать команду **Записи – Фильтр – Расширенный фильтр**.

В результате появится окно конструктора расширенного фильтра. Нижнюю часть окна фильтра заполним следующим образом (рис. 2.4).

Поле	Фирма	изделие	код	1кв	2кв	3кв	4кв	год вып
Сортировка:								
Условие отбора:								

Рис. 2.4

Чтобы можно было выводить информацию по любому изделию, для поля **изд** в строке условия отбора введем параметр – подсказку в квадратных скобках.

После нажатия кнопки **Применить фильтр** появится диалоговое окно для ввода значения параметра с текстом: «Введи изделие над окном ввода». Параметр позволяет ввести любое имя изделия, имеющееся в таблице, и будет выведена соответствующая информация. Если введем значение "телевизор", то получим следующий результат (рис. 2.5).

Фирма	изделие	код	1кв	2кв	3кв	4кв	год вып
Меридиан	телевизор	123	300	300	300	300	6 000,00р.
Техно	телевизор	123	400	400	400	400	7 000,00р.

Рис. 2.5

Для выключения действия фильтра надо нажать кнопку **Удалить фильтр**. Расширенный фильтр может быть сохранен для многократного использования. Для этого в главном меню выполнить команду **Файл -Сохранить как запрос**, а затем присвоить имя запросу. Чтобы использовать сохраненный фильтр, надо открыть окно **Фильтр** и выполнить команду **Файл – Загрузить из запроса**.

### Запросы по условиям

**Задача 2.3.** Выдать на экран список изделий, начинающихся на букву «В». Здесь используется запрос с оператором **Like**.

Для создания запроса необходимо в окне базы данных выбрать вкладку **Запросы – Создать – Конструктор – Добавить таблицу**. Ввести имя таблицы, по которой создается запрос. На экране получим окно конструктора, в нижней части которого необходимо заполнить строки, представленные ниже.

**Строка поле:** в каждом столбце этой строки мы должны указать имена полей таблицы, которые мы собираемся выводить в результат запроса. Если выводятся все поля, то указывается **имя таблицы** со знаком \* (звездочка). Если какое-либо поле участвует в условии отбора, но не выводится на экран, то символ  (галочка) должен быть снят. Каждое выводимое поле описывается в отдельном столбце. Для нашего случая нижняя часть окна конструктора запроса будет выглядеть следующим образом (рис. 2.6).

Поле:	Фирма *	изд		
Имя таблицы:	Фирма	Фирма		
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Условие отбора:		Like "h"		

Рис. 2.6

Для выполнения запроса нужно нажать на кнопку **выполнить (!)**. Результат запроса представлен на рис. 2.7.

Фирма	изделие	код	1кв	2кв	3кв	4кв	год вып
Техно	пылесос	234	100	150	150	150	4 000,00р.
Меридиан	пылесос	333	200	230	220	210	5 000,00р.
Топаз	пылесос	333	200	195	210	200	5 000,00р.
			0	0	0	0	0,00р.

Рис. 2.7

Конструктору запроса соответствует оператор запроса на языке **SQL**. Оператор **SQL** запроса строится автоматически по конструктору, но его можно написать самому в окне **SQL**. Переход в это окно производится из окна конструктора

кнопкой **Конструктор**, который осуществляет переходы в различные режимы запроса, или кнопкой **SQL**.

Синтаксис простейшего оператора запроса следующий:

**SELECT** <список полей>

**FROM** <список таблиц>

**WHERE** <условия отбора>;

В инструкции **SELECT** можно указывать выражения, функции, имена полей таблиц. Но если нужно в запрос включить все поля таблицы, то можно указать имя таблицы и затем поставить символ "\*" или просто "\*" Например **SELECT** Фирма\* или **SELECT** \* указывает, что в запрос включаются все поля таблицы **Фирма** или активной таблицы. Инструкция **FROM** определяет имена таблиц или запросов, которые служат источником данных для запроса.

В инструкции **WHERE** записываются логические выражения, истинность которых определяет набор записей, помещаемых в логическую таблицу запроса. В логических выражениях могут использоваться операции отношений "<, >, =, <=, >=", "<=", " связывающие однотипные данные,

логические операции **NOT, OR, AND, IN**,

и операции **LIKE, BETWEEN**.

Операция **BETWEEN** используется в логическом выражении для сравнения поля или выражения с заданным диапазоном значений.

Синтаксис операции: a **BETWEEN** b **AND** c, где

a – сравниваемое поле или выражение,

b – начальное значение диапазона,

c – конечное значение диапазона.

Операция **IN** определяет, равно ли некоторое поле или выражение одному из значений, перечисленных в списке значений этой операции. Синтаксис операции a **IN** (v1;v2;...), где

a – сравниваемое поле или выражение, v1, v2.. – символьные константы, заключенные в кавычки и разделяемые символом ";".

Операция **LIKE** используется для сравнения поля или выражения с образцом. Образец записывается в кавычках и определяет, какие символы должны входить в поле сравнения. В образце используются символы шаблона:

? – заменяет одиночный символ,

\* – заменяет произвольное число символов,

# – заменяет одиночную цифру,

[A-Z] – заменяет позицию любой буквы в заданном диапазоне, [1-9] – заменяет позицию любой цифры в заданном диапазоне.

Например, наш запрос можно записать следующим оператором:

**SELECT** Фирма.\*

**FROM** Фирма

**WHERE** (izd **Like** "п\*");

После выполнения запроса получим тот же результат, что и на рис. 2.6.

**Задача 2.4.** Вывести сведения об изделиях, не содержащих сочетание «еле» (телевизор). Для решения этой задачи используется оператор **Not Like** (рис. 2.8).

Поле:	Фирма*	Изд	
Имя таблицы:	Фирма	Фирма	
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Условие отбора:		Not Like "еле"	
Имя:			

Рис. 2.8

Оператором **SQL** этот запрос можно записать следующим образом:

**SELECT** Фирма.\*

**FROM** Фирма

**WHERE** (izd **Not Like** "еле");

После выполнения этого запроса получим следующую таблицу (рис. 2.9):

Фирма	Изделие	код	1кв	2кв	3кв	4кв	год вып.
Техно	пылесос	234	100	150	150	150	4 000,00р.
Стройдвор	магнитофон	456	200	200	200	230	4 000,00р.
Техно	электрочайник	567	100	100	110	110	5 000,00р.
Техно	магнитофон	456	900	900	950	1000	8 000,00р.
Меридиан	магнитофон	456	150	160	165	165	4 000,00р.
Стройдвор	электрочайник	567	500	500	650	650	8 000,00р.
Меридиан	пылесос	333	200	230	220	210	5 000,00р.
Топаз	пылесос	333	200	195	210	200	5 000,00р.
Топаз	магнитофон	456	120	126	100	110	5 000,00р.

Рис. 2.9

**Задача 2.5.** Вывести данные по тем изделиям, цена которых в первом квартале не менее 150 рублей, но и не более 310 руб. Для создания этого запроса используется оператор **Between**.

В **SQL** такой запрос можно записать следующим образом:

**SELECT** Фирма.\*

**FROM** Фирма

**WHERE** ((c1) **Between** 150 **And** 310);

Поле таблицы нужно заключить в квадратные скобки, если оно состоит из нескольких слов, совпадает с ключевым словом или начинается с цифры, но скобки можно ставить всегда.

После выполнения запроса получим следующую таблицу (рис. 2.10):

Фирма	Изделие	код	1кв	2кв	3кв	4кв	год вып.
Меридиан	телевизор	123	300	300	300	300	6 000,00р.
Стройдвор	магнитофон	456	200	200	200	230	4 000,00р.
Меридиан	магнитофон	456	150	160	165	165	4 000,00р.
Меридиан	пылесос	333	200	230	220	210	5 000,00р.
Топаз	пылесос	333	200	195	210	200	5 000,00р.

Рис. 2.10

**Задача 2.6.** Вывести данные об изделиях, цена которых в первом квартале была менее 150 и более 310 руб. (всех, кроме тех, у кого 150 > [c1] < 310). Для решения этой задачи воспользуемся предыдущим запросом, поменяв условие **Between** на **Not Between**.

В режиме **SQL** такой запрос можно записать следующим оператором:

**SELECT** Фирма.\*

**FROM** Фирма

**WHERE** ((c1) **Not Between** 150 **And** 310);

После выполнения запроса получим следующий результат (рис. 2.11):

Фирма	изделие	код	1кв	2кв	3кв	4кв	год вып
Техно	пылесос	234	100	150	150	150	4 000,00р.
Техно	телевизор	123	400	400	400	400	7 000,00р.
Техно	электрочайник	567	100	100	110	110	5 000,00р.
Техно	магнитофон	456	900	900	950	1000	8 000,00р.
Стройдвор	телевизор	123	350	400	400	400	6 000,00р.
Стройдвор	электрочайник	567	500	500	650	650	8 000,00р.

Рис. 2.11

**Задача 2.7.** Вывести все данные о предприятиях "Меридиан", "Агат" и "Топаз". Для реализации этой задачи используется оператор **In**. Вид конструктора запросов представлен на рис. 2.12.

Поле:	Фирма	Firm	
Имя таблицы:	Фирма	Фирма	
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:	In ("Меридиан"; "Агат"; "Топаз")		
или:			

Рис. 2.12

В режиме SQL такой запрос можно записать следующим оператором:

```
SELECT Фирма.*
FROM Фирма
WHERE ([firm] In ("Меридиан"; "Агат"; "Топаз"));
```

После выполнения запроса получим результат (рис. 2.13).

Фирма	изделие	код	1кв	2кв	3кв	4кв	год вып
Меридиан	телевизор	123	300	300	300	300	6 000,00р.
Меридиан	магнитофон	456	150	160	165	165	4 000,00р.
Агат	телевизор	123	320	320	330	330	7 000,00р.
Меридиан	пылесос	333	200	230	220	210	5 000,00р.
Топаз	пылесос	333	200	195	210	200	5 000,00р.
Топаз	магнитофон	456	120	126	100	110	5 000,00р.

Рис. 2.13

**Задача 2.8.** Вывести все данные о всех предприятиях, кроме "Меридиан", "Топаз" и "Агат". Для решения этой задачи используем предыдущий запрос, но условие вывода **IN** заменим на **Not In**.

В режиме SQL такой запрос можно записать следующим оператором:

```
SELECT Фирма.*
FROM Фирма
WHERE ([firm] Not In ("Меридиан"; "Агат"; "Топаз"));
```

**Задача 2.9.** Вывести данные по фирмам, названия которых начинаются на букву «Т» и годовым выпуском больше 300. Здесь используется комбинация условий. Вид конструктора представлен на рис. 2.14.

Поле:	Фирма	Firm	год
Имя таблицы:	Фирма	Фирма	Фирма
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	Like "Т*"		> 3000
или:			

Рис. 2.14

В режиме SQL такой запрос можно записать следующим оператором:

```
SELECT Фирма.*
FROM Фирма
WHERE (([firm] Like «Т*») And ([god] >3000)).
```

Результат выполнения запроса представлен на рис. 2.15.

Фирма	изделие	код	1кв	2кв	3кв	4кв	год вып
Техно	пылесос	234	100	150	150	150	4 000,00р.
Техно	телевизор	123	400	400	400	400	7 000,00р.
Техно	электрочайник	567	100	100	110	110	5 000,00р.
Техно	магнитофон	456	900	900	950	1000	8 000,00р.
Топаз	пылесос	333	200	195	210	200	5 000,00р.
Топаз	магнитофон	456	120	126	100	110	5 000,00р.

Рис. 2.15

### Запрос с параметром

**Задача 2.10.** Вывести всю информацию по двум произвольно задаваемым изделиям.

Решение можно представить следующей программой на SQL.

```
SELECT *
FROM Фирма
WHERE izd In([введите изделие1],[введите изделие2]);
```

При выполнении запроса дважды будут выводиться диалоговые окна, в которые мы будем вводить названия задаваемых изделий. Символ «\*» означает, что в динамический набор записей этого запроса требуется включить все поля таблицы Фирма.

### Часть 3. Основные задачи части 3.

Создание более сложных запросов:

- запросы с вычисляемым полем,
- запросы с подведением итогов по таблице и группе записей.

Запросы на модификацию таблиц:

- создание таблицы по существующей;
- добавление записей из одной таблицы в другую,
- обновление записей,
- удаление записей.

В качестве исходной взята таблица Фирма (рис. 1.1)

### Создание вычисляемого поля в запросах

В запросе можно создавать новое поле, значение которого вычисляется в процессе выполнения запроса. Для построения выражения используются **операции над полем и встроенные функции**. Эти выражения записываются в строке запроса **поле** или используется **Построитель выражений**. Создаваемому полю Access присваивает имя (типа Выражение1). Это имя можно изменить, введя собственное, более содержательное имя, или использовать свойство **подпись** в окне конструктора. Если в вычисляемом выражении осуществляется деление, то результат может содержать до 9 знаков в дробной части. Чтобы получить читаемый результат, можно воспользоваться функцией Cint() для округления результата до нужного количества знаков после десятичной точки. Например, выражение Cint((v1+v2+v3)/3\*100)/100 даст 2 знака после точки. Такой же результат будет, если в свойстве **формат** этого поля выбрать значение **фиксированный**.

**Задача 3.1.** По каждой записи исходной таблицы подсчитать среднюю цену за год. Для решения этой задачи нужно создать запрос с вычисляемым полем. Для этого из окна Базы данных выбирается вкладка **Запросы – Создать – Конструктор** – **Добавить («Фирма») – Заккрыть.**

Окно конструктора запросов (рис. 3.1).

Поле:	Фирма*	средняя цена: $([c1]+[c2]+[c3]+[c4])/4$	
Имя таблицы:	Фирма		
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:			
или:			

Рис. 3.1

**Средняя цена** – имя поля, которым заменили стандартное имя **Выражение1**.

Запрос можно сохранить, задав ему имя.

В режиме **SQL** этот запрос записан следующим образом:

```
SELECT Фирма.*, ([c1]+[2c]+[3c]+[4c])/4 AS [сред цена]
```

```
FROM Фирма;
```

Результат запроса представлен в таблице (рис. 3.2).

Фирма	изделие	код	1кв	2кв	3кв	4кв	год вып.	средняя
Меридиан	телевизор	123	300	300	300	300	6 000,00р.	300
Техно	пылесос	234	100	150	150	150	4 000,00р.	137,5
Техно	телевизор	123	400	400	400	400	7 000,00р.	400
Стройдвор	магнитофон	456	200	200	200	230	4 000,00р.	207,5
Техно	электрочайник	567	100	100	110	110	5 000,00р.	105
Техно	магнитофон	456	900	900	950	1000	8 000,00р.	937,5
Меридиан	магнитофон	456	150	160	165	165	4 000,00р.	160
Стройдвор	телевизор	123	350	400	400	400	6 000,00р.	387,5
Стройдвор	электрочайник	567	500	500	650	650	8 000,00р.	575
Агат	телевизор	123	320	320	330	330	7 000,00р.	325
Меридиан	пылесос	333	200	230	220	210	5 000,00р.	215
Топаз	пылесос	333	200	195	210	200	5 000,00р.	201,25
Топаз	магнитофон	456	120	126	100	110	5 000,00р.	114

Рис. 3.2

**Задача 3.2.** Вычислить изменение цен за год в процентах по отношению к началу года. Результат вывести в порядке уменьшения процента и только для фирм, содержащих в своем названии букву **р**.

Создадим запрос с вычисляемым полем (рис. 3.3).

Поле:	Фирма*	Фирм	Процент: $Abs(Int((100*[4c]/[1c]-100))$	
Имя таблицы:	Фирма	Фирма		
Сортировка:	Фирма		по убыванию	
Вывод на экран:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:				
или:		Like "р"		

Рис. 3.3

В режиме **SQL**:

```
SELECT Фирма.*, Abs(Int((100*[4c]/[1c]-100)) as [процент]
```

```
FROM Фирма
```

```
WHERE (Firm Like "р")
```

```
ORDER BY Abs(Int((100*[4c]/[2]-100) DESC;
```

Предложение **ORDER BY** определяет, по какому столбцу происходит упорядочивание и порядок сортировки: **ASC** – по возрастанию, **DESC** – по убыванию. Результат запроса в таблице (рис. 3.4).

Фирма	изделие	код	1кв	2кв	3кв	4кв	год вып.	Процент
Стройдвор	электрочайник	567	500	500	650	650	8 000,00р.	30
Стройдвор	магнитофон	456	200	200	200	230	4 000,00р.	15
Стройдвор	телевизор	123	350	400	400	400	6 000,00р.	14
Меридиан	магнитофон	456	150	160	165	165	4 000,00р.	10
Меридиан	пылесос	333	200	230	220	210	5 000,00р.	5
Меридиан	телевизор	123	300	300	300	300	6 000,00р.	0

Рис. 3.4

**Запрос с итоговыми вычислениями**

Итоговые вычисления позволяют получать информацию по группе записей, относящихся к одному и тому же объекту. **ACCESS** предоставляет специальные функции для групповых операций. Вот некоторые из них.

**SUM(поле)**-вычисляет суммы всех значений заданного поля или выражения в группе.

**AVG(поле)**-вычисляет среднее арифметическое заданного поля или выражения для каждой группы.

**MIN(поле), MAX(поле)** – находят наименьшее и наибольшее значения заданного поля или выражения в каждой группе.

**COUNT(поле)**-находит число записей в каждой группе.

**FIRST(поле), LAST(поле)**- находят первое и последнее значения заданного поля или выражения в каждой группе.

Для создания итогового запроса в конструкторе надо добавить строку **Групповая операция**.

**Задача 3.3.** По каждому виду изделия найти среднее значение цены в первом квартале и максимальный годовой выпуск.

Для решения этой задачи нужно создать запрос со строкой **Групповая операция**. Это можно сделать, щёлкнув по кнопке  $\Sigma$  или в меню, выбрав команду. Группировка устанавливается для того поля, одинаковое значение которого определяет группу. В нашей задаче таким полем является изделие с именем **lzd**. Для полей **цена** в 1 квартале (**c1**) и **годовой выпуск(God)** в этой строке выберем из выпадающего списка функции **Avg**, для получения средней цены и **Max**, для максимального годового выпуска. Вид конструктора (рис. 3.5).

Поле:	lzd	Сред цена c1	Макс выпуск: God	
Имя таблицы:	Фирма	Фирма	Фирма	
Сортировка:	Фирма			
Группировка:	Фирма			
Сортировка:		Avg		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:				
или:				

Рис. 3.5

В режиме **SQL** для формирования групп с одинаковым значением заданного столбца используется предложение **GROUP BY**, после которого указывается имя столбца. В предложении **SELECT** список должен состоять только из столбцов, указанных в предложении **GROUP BY**, или из итоговых функций. Другими словами, каждый столбец, включенный в итоговый набор записей, должен определять группу или представлять статистическую величину, вычисляемую для каждой группы.

```
SELECT Izd, Avg[1c] AS [Сред цена], Max[God] AS [Макс выпуск]
FROM Фирма
GROUP BY Izd;
```

После выполнения запроса получим таблицу (рис 3.6).

изделие	Сред цен	Макс выпт
магнитофон	342,5	8000
пылесос	166,6667	5000
телевизор	342,5	7000
электрочайник	300	8000

Рис. 3.6

Итоговые записи могут быть получены по разным группам.

Рассмотрим другую задачу, в которой группировка идет по другому полю (по названию фирмы).

**Задача 3.4.** По каждой фирме найти изделия с максимальным и минимальным годовым выпуском.

Максимальный и минимальный выпуски можно вычислить с помощью соответствующих функций. Но найти соответствующие этим значениям изделия можно только в упорядоченной по годовому выпуску таблице. При этом следует учесть, что группировка (предложение **GROUP BY**) автоматически устанавливает сортировку по возрастанию поля группировки, а функции **LAST** и **FIRST** привязаны к расположению аргумента в исходной таблице.

Для этой задачи создадим копию таблицы **Фирма**, в которой все записи будут упорядочены по возрастанию значений поля **God** (годовой выпуск продукции) и сохраним ее с именем **Фирма1**.

Используя итоговые функции **First** и **Last**, которые возвращают значение полей из первой и последней записи в каждой группе результирующего набора записей, можно получить информацию о максимальных (минимальных) выпусках и соответствующих им названиях изделий. Конструктор запроса должен быть заполнен следующим образом (рис. 3.7).

Поле	Имя	God	Izd	God	Izd
Имя таблицы	Фирма1	Фирма1	Фирма1	Фирма1	Фирма1
Результат операции	Группировка	Last	Last	First	First
Сортировка					
Вывод на экран	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора					
Имя					

Рис. 3.7

В режиме **SQL** этот запрос выглядит следующим образом.

```
SELECT firm, Last((God)), Last((Izd)), First((God)), First((Izd))
FROM Фирма1
GROUP BY Firm;
```

После выполнения запроса получаем следующий результат (рис. 3.8):

Фирма	Last_God	Last_Izd	First_God	First_Izd
Агат	7000	телевизор	7000	телевизор
Меридиан	5000	пылесос	6000	телевизор
Стройдвор	8000	электрочайник	4000	магнитофон
Техно	8000	магнитофон	4000	пылесос
Топаз	5000	магнитофон	5000	пылесос

Рис. 3.8

### Запрос на создание таблицы

В предыдущей задаче нам понадобилась таблица, полностью совпадающая с существующей, но имеющая другой порядок следования записей. Такую таблицу можно создать и с помощью запроса. Использование реальной таблицы вместо запроса ускоряет доступ к данным. Для этого надо создать **Запрос-создание** таблицы из главного меню, используя команду меню **вид запроса**. Вид конструктора представлен на рис. 3.9.

Поле	Фирма	God	
Имя таблицы	Фирма	Фирма	
Сортировка		по возрастанию	
Вывод на экран	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Условие отбора			
Имя			

Рис. 3.9

В процессе выполнения запроса система затребуется ввести имя создаваемой таблицы. Введем имя **Фирма1**.

В режиме **SQL** для создания таблицы на основе записей другой таблицы или запроса используется предложение **SELECT INTO**.

```
Синтаксис этого предложения следующий:
SELECT <список полей> INTO <имя новой таблицы>
FROM <имя таблицы-источника>
WHERE <условие отбора>
ORDER BY <поля сортировки>
```

Для создания таблиц можно использовать итоговые запросы, запросы с предложением **GROUP BY**.

Программа создания упорядоченной по возрастанию значения поля **God** новой таблицы с именем **Фирма1**, содержащей все поля таблицы источника **Фирма**, представлена следующим предложением.

```
SELECT * INTO Фирма1
FROM Фирма
ORDER BY God;
```

После выполнения запроса в окне базы данных на вкладке **Таблицы** увидим это имя и откроем таблицу (рис. 3.10).

*Запрос → Создание таблицы*



Фирма	Изделие	код	1 кв	2 кв	3 кв	4 кв	год вып
Стройдвор	магнитофон	456	200	200	200	230	4 000,00р.
Техно	пылесос	234	100	150	150	150	4 000,00р.
Топаз	магнитофон	456	120	126	100	110	5 000,00р.
Топаз	пылесос	333	200	195	210	200	5 000,00р.
Меридиан	пылесос	333	200	230	220	210	5 000,00р.
Техно	электрочайник	567	100	100	110	110	5 000,00р.
Стройдвор	телевизор	123	350	400	400	400	6 000,00р.
Меридиан	телевизор	123	300	300	300	300	6 000,00р.
Агат	телевизор	123	320	320	330	330	7 000,00р.
Техно	телевизор	123	400	400	400	400	7 000,00р.
Стройдвор	электрочайник	567	500	500	650	650	8 000,00р.
Техно	магнитофон	456	900	900	950	1000	8 000,00р.

Рис. 3.10

В создаваемую с помощью запроса таблицу можно выбрать не все записи, а соответствующие определенному условию.

**Задача 3.5.** Создать таблицу, содержащую все записи с изделием телевизор. С помощью конструктора таблица создаётся в два этапа.

Сначала создаётся запрос на выборку: **Запросы – Создать – Конструктор**  
Вид конструктора этого запроса на рис. 3.11.

Поле:	Фирма*	Izd	
Имя таблицы:	Фирма	Фирма	
Сортировка:			
Вывод записей:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Условие отбора:		Like "т*"	
или:			

Рис. 3.11

На втором этапе запрос на выборку изменяется на запрос-создание: **Тип запроса – Создание таблицы** Имя новой таблицы (**Изделие**).

В режиме **SQL** может быть следующий набор операторов:

**SELECT \* INTO Изделие**

**FROM Фирма**

**WHERE ((Izd)= Like "т\*");**

После выполнения запроса создается таблица **Изделие**. Чтобы просмотреть созданную таблицу нужно выбрать вкладку **Таблицы**, найти имя созданной таблицы и открыть (рис 3.12).

Фирма	Изделие	код	1 кв	2 кв	3 кв	4 кв	год вып
Меридиан	телевизор	123	300	300	300	300	6 000,00р.
Техно	телевизор	123	400	400	400	400	7 000,00р.
Стройдвор	телевизор	123	350	400	400	400	6 000,00р.
Агат	телевизор	123	320	320	330	330	7 000,00р.

Рис. 3.12

Любой запрос на выборку можно преобразовать в запрос на создание. Например, итоговый запрос с изделиями с максимальным и минимальным годовым выпус-

ком по каждой фирме преобразуем в запрос на создание новой таблицы и назовем ее **Итор**.

**SELECT firm, Last([God]),Last([Izd]), First([God]), First([Izd]) INTO Итор  
FROM Фирма1  
GROUP BY Firm;**

Таблица будет иметь такие же поля, как и запрос (рис. 3.13).

Фирма	Last_God	Last_Izd	First_God	First_Izd
Агат	7000 телевизор		7000 телевизор	
Меридиан	5000 пылесос		6000 телевизор	
Стройдвор	8000 электрочайник		4000 магнитофон	
Техно	8000 магнитофон		4000 пылесос	
Топаз	5000 магнитофон		5000 пылесос	

Рис. 3.13

### Запрос на удаление

Удалить группу записей из базы данных можно с помощью запроса на удаление. Обычно такой запрос используется для удаления устаревших записей. Запрос может содержать параметр, чтобы во время выполнения запроса пользователь мог управлять удалением записей. Сначала создается запрос на **выборку** и после проверки его работы он изменяется на **запрос на удаление** с помощью команд **Запрос/Удаление** главного меню или соответствующей кнопки.

**Задача 3.6.** Удалить из таблицы **Изделие** все записи, относящиеся к фирме **"Техно"**.

**Запрос – Создать – Конструктор – Тип запроса – Удаление**  
Вид конструктора на рис. 3.14.

Поле:	Изделие*	Имя	
Имя таблицы:	Изделие	Изделие	
Удаление:	Из	Условие	
Условие отбора:		Техно	
или:			

Рис. 3.14

В режиме **SQL** для удаления строк из таблиц используется инструкция **DELETE**. Синтаксис этого оператора следующий.

**DELETE <таблица.>**

**FROM <таблица-источник>**

**WHERE <условие отбора>;**

Опцию **таблица.\*** в предложении **DELETE** не обязательно применять в запросах на удаление, которые относятся к одной таблице. Всегда удаляется одна запись. В нашем случае этот запрос может быть записан следующим оператором:

**DELETE**

**FROM Изделие**

**WHERE (((firm)="Техно"));**

В операторе **DELETE** мы не указали ничего, так как удаляются целиком все записи, удовлетворяющие указанному условию.

После выполнения этого запроса таблица **Изделие** примет вид (рис 3.15).

Firm	Izd	Kod	c1	c2	c3	c4	God
Меридиан	телевизор	123	300	300	300	300	6000
Стройдвор	телевизор	123	350	400	400	400	6000
Агат	телевизор	123	320	320	330	330	7000

Рис. 3.15

**Запрос на добавление**

В существующую таблицу можно добавлять записи из других таблиц. Для этого сначала надо создать **Запрос-выборку**, затем с помощью кнопки **Тип запроса** изменить его на **Добавление**.

**Задача 3.7.** Добавить в таблицу **Изделие** записи, относящиеся к фирме **Техно**.

Для решения этой задачи выполняется следующая последовательность действий: **Запрос – Создать – Конструктор** – Имя таблицы из которой добавляем записи (**Фирма**). В конструкторе необходимо задать условия отбора записей, затем **Тип запроса – Добавление** (в конструкторе появляется новая строка – **Добавление**). В нее вводим имя таблицы, в которую добавляем записи (**Изделие**). Вид конструктора (рис. 3.16).

Поле:	Фирма	Фирма	
Имя таблицы:	Фирма	Фирма	
Сортировка:			
Добавление:	Изделие*		
Условие отбора:	Имя "Техно"		

Рис. 3.16

Программа на языке запроса **SQL** для копирования строк из одной таблицы или запроса в другую должна иметь следующую структуру:

```
INSERT INTO <таблица-назначения>
```

```
SELECT –инструкция;
```

Инструкция **INSERT INTO** может содержать любую допустимую инструкцию **SELECT**, включая инструкцию с группировкой записей.

Вид запроса в **SQL** для решения поставленной выше задачи следующий:

```
INSERT INTO Изделие
```

```
SELECT Фирма.*
```

```
FROM Фирма
```

```
WHERE ([firm]="Техно");
```

После выполнения запроса таблица **Изделие** примет вид (рис. 3.17).

Firm	Izd	Kod	c1	c2	c3	c4	God
Меридиан	телевизор	123	300	300	300	300	6000
Стройдвор	телевизор	123	350	400	400	400	6000
Агат	телевизор	123	320	320	330	330	7000
Техно	пылесос	234	100	150	150	150	4000
Техно	телевизор	123	400	400	400	400	7000
Техно	электрочайн	567	100	100	110	110	5000
Техно	магнитофон	456	900	900	950	1000	8000

Рис. 3.17

Вторая форма инструкции **INSERT INTO** служит для добавления в таблицу и заполнения значениями одной строки. Синтаксис этой инструкции имеет следующий вид:

```
INSERT INTO <таблица-назначение> [столбец1,столбец2,...]
VALUES(значение1,значение2,...);
```

Имена столбцов могут быть опущены, если значения столбцов в предложении **VALUES** соответствуют описанию их в таблице. Если имена столбцов все же используются, то можно опустить некоторые столбцы или задавать их значения в любом порядке.

```
INSERT INTO Изделие firm, God, Izd
```

```
VALUES( «Агат»,6500, «пылесос»);
```

**Запрос на обновление записей**

С помощью запроса можно произвести одинаковые изменения во множестве записей. Сначала необходимо создать запрос на выборку тех записей, в которые вносятся одинаковые изменения, а затем преобразовать его в запрос –обновление командами **Запрос – Обновление** или соответствующей кнопкой.

**Задача 3.8.** В таблице **Изделие** заменить название изделия **телевизор** на название **TV SONI**.

**Запрос – Создать – Конструктор** и в главном меню выберем команду **Тип Запрос – Обновление**. Появится строка **Обновление**, в которой надо указать новое название, а в строке **Условие отбора** – старое (телевизор). В языке запросов для изменения значения в одном или нескольких столбцах таблицы используется инструкция **UPDATE**. Ее синтаксис:

```
UPDATE <таблица или запрос>
```

```
SET <столбец1=выражение1, столбец2=выражение2,...>
```

```
[WHERE <условие отбора>];
```

Вид конструктора для решения поставленной задачи следующий (рис. 3.18).

Поле:	ИЗ		
Имя таблицы:	Изделие		
Обновление:	"TV SONI"		
Условие отбора:	"телевизор"		

Рис. 3.18

В режиме **SQL** запрос будет таким:

```
UPDATE Изделие
```

```
SET Izd="TV SONI"
```

```
WHERE ([Izd]="телевизор");
```

После выполнения запроса таблица примет следующий вид (рис. 3.19).

Firm	Izd	Kod	c1	c2	c3	c4	God
Меридиан	TV SONI	123	300	300	300	300	6000
Стройдвор	TV SONI	123	350	400	400	400	6000
Агат	TV SONI	123	320	320	330	330	7000
Техно	пылесос	234	100	150	150	150	4000
Техно	TV SONI	123	400	400	400	400	7000
Техно	электрочайн	567	100	100	110	110	5000
Техно	магнитофон	456	900	900	950	1000	8000

Рис. 3.19

Запросы на создание, добавление, удаление и обновление являются запросами-действиями, они осуществляют модификацию таблиц. Чтобы обезопасить себя от каких-либо случайностей, необходимо перед выполнением запросов сделать резервную копию таблицы.

#### Часть 4. Основные задачи части 4.

Работа со статистическими функциями по подмножеству.

Создание запроса с функцией DMAX.

Создание запроса с функцией DAVG.

Создание запроса с функцией DMIN с условием.

#### Основные понятия

Статистические функции по подмножеству являются аналогами итоговых функций, входящих в групповые операции. Но в отличие от итоговых функций эти функции применимы к детальным записям и работают без группировки. Функции могут содержать условия для отбора обрабатываемых ими записей. В качестве условия отбора используются те же логические операторы, что и в запросах.

ACCESS располагает следующими функциями:

DAvg – подсчет среднего арифметического значения,

DCount – подсчет количества,

DFirst – нахождение первого значения из группы,

DLast – нахождение последнего значения из группы,

DMax – определение максимального значения,

DMin – определение минимального значения,

DSum – подсчет суммы значений, и другие.

Синтаксис операторов следующий.

Имя\_функции ("аргумент1";"аргумент2 ";"аргумент3")

Функции имеют два обязательных аргумента и один необязательный. Все аргументы строкового типа.

**Аргумент1** – является обязательным и определяет поле или выражение, над значениями которого будет выполняться функция. Этот аргумент должен быть строкой символов, то есть должен заключаться в кавычки и содержать имя одного поля или выражения над полями. Например, "Firm", "([c1]+[c2]+[c3]+[c4])/4".

**Аргумент2** – также является обязательным и определяет таблицу или запрос (источник записей), по которому вычисляется функция.

**Аргумент3** – «критерий отбора», необязательный. Если он присутствует, то определяет, для каких записей вычисляется функция, если отсутствует, то функция вычисляется по всем записям.

Функции могут использоваться в предложении **SELECT** и определять вычисляемый столбец, выводимый в динамический набор записей. Функции также могут использоваться в условиях отбора и определять набор записей, используемый для вычисления.

**Задача 4.1.** Вывести информацию о фирмах и изделиях с максимальным годовым выпуском.

В конструкторе в качестве поля для вычисления максимума используется поле «God». Оно записывается как строковое выражение. Второе строковое выражение «Фирма» определяет таблицу, являющуюся источником данных. Функция используется в качестве критерия отбора для поля «God». Условие отбора имеет вид: DMax("God"; "Фирма"). Поиск будет идти по всем записям таблицы, т.е. результатом функции является максимальное по всей таблице значение годового выпуска.

Вид конструктора запроса представлен на рис. 4.1.

Поле:	Firm	Izd	Kod	God
Имя таблицы:	Фирма	Фирма	Фирма	Фирма
Сортировка:				
Выход на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				DMax("God";"Фирма")

Рис. 4.1

Этот же запрос в режиме SQL будет следующим:

```
SELECT Firm, Izd, Kod, God
```

```
FROM Фирма
```

```
WHERE God=Dmax("God","Фирма");
```

После выполнения запроса получим результат (рис. 4.2).

Фирма	изделие	код	год.вып
Техно	магнитофон	456	8 000,00р.
Стройдвор	электрочайник	567	8 000,00р.
		0	0,00р.

Рис. 4.2

**Задача 4.2.** Вывести сведения о годовом выпуске изделий произвольно заданной фирмы, а также сам средний выпуск по всей таблице, только по тем фирмам, где годовой выпуск не больше среднего годового по всей таблице.

На бланке формирования запроса заданы два условия: условие на значение поля Firm – равенство произвольно вводимому значению [Введите фирму], и условие на годовой выпуск: <=DAvg("God";"Фирма").

Функция DAVg вычисляет средний годовой выпуск среди всех записей таблицы **Фирма** и результат выводится в строке «поле» конструктора запроса. В процессе выполнения запроса требуется вводить параметр «Введите фирму», например Меридиан.

Вид конструктора представлен на рис. 4.3.

Поле:	Firm	Izd	Kod	God	Средн. (Int(DAvg("God";"Фирма")))
Имя таблицы:	Фирма	Фирма	Фирма	Фирма	
Сортировка:					
Выход на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	[вв Фирму]			<=DAvg("God";"Фирма")	

Рис. 4.3

В режиме SQL такой запрос будет следующим:

```
SELECT Firm, Izd, Kod, God, DAVg("God","Фирма") AS [Средний]
```

```
FROM Фирма
```

```
WHERE Firm=[Введите фирму] And God<=Int(DAvg("God","Фирма"));
```

Результат запроса смотри ниже (рис. 4.4).

Фирма	изделие	код	год вып	Средн
Меридиан	магнитофон	456	4 000,00р	5 692,00р
Меридиан	пылесос	333	5 000,00р	5 692,00р
		0	0,00р	

Рис. 4.4.

В приведенном выше SQL запросе функция DAVg используется как вычисляемое выражение в списке вывода предложения SELECT, ее результат преобразуется в целое значение функцией Int.

**Задача 4.3.** Вывести сведения о фирмах, годовой выпуск на которых для изделий с кодом 456 равен минимальному среди этих изделий по всей таблице.

В функции DMin используется третий параметр «условие», годовой выпуск равен минимальному только среди изделий с кодом, равным 456 («Kod=456»). В режиме SQL такой запрос будет следующим:

```
SELECT Firm, Izd, Kod, God
FROM Фирма
WHERE (Kod=456) And God=DMin("God", "Фирма", "Kod=456");
```

Запрос содержит два условия отбора, которые должны выполняться одновременно. Первое требует, чтобы значение поля "Kod" было равно 456. Второе условие накладывается на значение поля для годового выпуска. Это значение должно быть равно минимальному среди подмножества записей таблицы, определяемое условием: [Kod]=456. Поэтому функция Dmin имеет три аргумента:

DMin ("God"; "Фирма"; "Kod=456")

Третий аргумент определяет, что действие функции распространяется не на все записи таблицы, а только на группу записей.

Результат запроса представлен на рис. 4.5.

Фирма	изделие	код	год вып
Стройдвор	магнитофон	456	4000
Меридиан	магнитофон	456	4000
		0	0

Рис. 4.5

Если в условии отбора записей используется текстовая переменная, то все выражение заключается в апострофы. Например для нахождения минимальной средней цены у изделия «магнитофон» и вывода информации об этих изделиях можно записать следующий запрос:

```
SELECT *
Dmin("c1+c2+c3+c4)/4", "Фирма", 'Izd Like "м*' ) as [миним.цена]
FROM
WHERE (Izd Like "м*")
And (c1+c2+c3+c4)/4=Dmin("c1+c2+c3+c4)/4", "Фирма", 'Izd Like "м*' );
```

## Часть 5. Основные задачи части 5.

Создание запросов на основе других запросов и запросов с подзапросами:

создание запроса со средним по всей таблице

создание запроса со средним по группам и средним по всей таблице

создание запроса с максимумом по всей таблице

создание запроса с максимумом по группам и максимумом

и средним по всей таблице.

создание перекрестного запроса.

Для решения этих задач будем использовать данные таблицы **Фирма** (рис. 1.1).

### Основные понятия.

При создании запросов в качестве источника записей могут быть не только таблицы, но и запросы. Можно так же создать запрос, в котором в качестве Условия отбора в свою очередь используется SQL –запрос. Такие запросы называются запросами с подзапросами. SQL –запрос может записываться в строке **Поле** конструктора запросов так же, как вычисляемое выражение. Итоговые функции могут вычисляться по всей таблице без применения группировки, а результат такого вычисления – одно значение.

Например, запрос

```
SELECT Max(God)
```

```
FROM Фирма;
```

вычислит одно значение, которое можно использовать в других запросах.

### Создание запроса со средним по всей таблице

**Задача 5.1.** Вывести из таблицы все записи с годовым выпуском большим общего среднего значения годового выпуска по всей таблице.

Задачу можно решить двумя способами: с помощью двух запросов и с помощью запроса с подзапросом.

**Первый способ.** Создадим два запроса: Запрос нахождения среднего значения поля God по всей таблице (**Зап1**) и, используя записи таблицы и этого запроса, как источника данных, создадим второй запрос (**Зап2**). Результат **Зап1** – число, равное среднему арифметическому из всех значений поля God, получим в следующем виде (рис. 5.1).

Общий сре
5692,31

Рис. 5.1

В режиме SQL этот запрос выглядит следующим образом:

```
SELECT Avg(God) as [Avg-God]
```

```
FROM Фирма;
```

Второй запрос (Рис. 5.2) выглядит следующим образом:

Поле:	Фирма	God	Avg-God
Имя таблицы:	Фирма	Фирма	зап1
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Условие отбора или:	>=[зап1][Avg-God]		

Рис. 5.2

Объектами этого запроса являются таблица **Фирма** и запрос **Зап1**.

В запросе предусматривается вывод всех записей, для которых значение поля **God** больше или равно полю **Avg-God**, вычисленному в запросе **Зап1**.

Этот же запрос в режиме SQL будет следующим:

```
SELECT Фирма.*
FROM Фирма, Зап1
WHERE God>=[Зап1].[Avg-God];
```

Здесь в операторе FROM указаны имя таблицы и имя запроса, а в операторе WHERE в условном выражении указано уточненное имя из запроса ([Зап1].[Avg-God]).

Результат запроса представлен на рис. 5.3.

Фирма	название	код	1 кв	2 кв	3 кв	4 кв	год вып
Меридиан	телевизор	123	300	300	300	300	6 000,00р
Техно	телевизор	123	400	400	400	400	7 000,00р
Техно	магнитофон	456	900	900	950	1000	8 000,00р
Стройдвор	телевизор	123	350	400	400	400	6 000,00р
Стройдвор	электрочайник	567	500	500	650	650	8 000,00р
Агат	телевизор	123	320	320	330	330	7 000,00р

Рис. 5.3.

Другим способом решения этой задачи является создание запроса с подзапросом.

Конструктор запроса выглядит следующим образом (рис. 5.4).

Рис. 5.4

Результат этого запроса такой же как и предыдущего (рис. 5.3).

На языке запросов эта задача решается следующим образом.

```
SELECT Фирма.*, (Select AVG(GOD) From Фирма) AS [Общий средний ]
FROM Фирма
WHERE (GOD >= ( Select AVG(GOD) Form Фирма));
```

**Создание запроса со средним по группам и средним по всей таблице**

В этих запросах условие отбора применяется не к каждой записи, а к группе записей, т.е. к итоговым записям. Эти условия записываются в строке **Условие отбора** в режиме конструктора или в предложении **HAVING** режима SQL, и могут использовать только те столбцы, которые указаны в предложении **GROUP BY** (группировка), а также итоговые функции и выражения, содержащие итоговые функции. Предложение HAVING ставится после предложения GROUP BY.

**Задача 5.2.** По каждому предприятию вычислить средний выпуск и вывести информацию о тех предприятиях, у которых средний выпуск меньше общего среднего по таблице.

Эту задачу можно решить двумя способами:

- создав запрос, который вычисляет средний годовой выпуск по каждому предприятию **Зап2**, и используя запрос **Зап1** из предыдущей задачи (Средний годовой выпуск по всей таблице), объединив эти запросы в один **Зап3** –запрос на выборку.

Конструктор запроса **Зап2** имеет следующий вид (рис. 5.5).

Рис. 5.5

В свойствах поля **God** в свойстве **Подпись** запишем название столбца **Средний выпуск**, а для свойства **Формат** выберем значение **фиксированный**.

Конструктор запроса **Зап3** использует в качестве источника записей запросы **Зап1** и **Зап2** и имеет следующий вид (рис. 5.6):

Рис. 5.6

В свойствах поля **Avg-God** из **Зап1** в свойстве **Подпись** запишем названия столбца **Общий средний** и в свойстве **Формат** выберем значение **фиксированный**.

Запрос **Зап2** в режиме SQL будет следующим

```
SELECT Firm, Avg((god)) AS [Средний выпуск]
FROM Фирма
GROUP BY Firm;
```

Запрос **Зап3** в режиме SQL будет следующим

```
SELECT Зап2.*, Зап1.[Avg-God] AS [Общий средний ]
FROM Зап1,Зап2
WHERE ((Зап2).[Avg-God]<=[Зап1].[Avg-God]);
```

Результат запроса **Зап3** представлен на рис. 5.7.

Фирма	Средний вы	Общий сред
Меридиан	5000.00	5692.31
Топаз	5000.00	5692.31

Рис. 5.7

Такой же результат можно получить с помощью одного запроса, используя SQL запрос в качестве условия отбора.

Конструктор этого запроса на рис. 5.8.

Рис. 5.8

Этот запрос на языке запросов будет следующим:

```
SELECT Firm, Avg(God) AS [Средний вып.],(Select Avg(God) From Фирма) AS
[Общий средний ]
FROM Фирма
GROUP BY Firm
```

**HAVING** (Avg(God) <= (Select Avg(God) From Фирма));

Сохраним запрос с именем **Зап4**.

Результат запроса такой же, как на рис. 5.7.

#### Создание запроса с максимумом по всей таблице

**Задача 5.3.** Вывести информацию об изделиях и предприятиях с максимальным годовым выпуском.

Задачу можно решить двумя способами. С помощью двух запросов и с помощью запроса с подзапросом.

Первый запрос находит максимальный годовое выпуск по всей таблице (**Зап1**).

В режиме **SQL** этот запрос выглядит следующим образом:

```
SELECT Max(God) as [Max-God]
FROM Фирма;
```

Результат такого запроса — число, равное наибольшему из всех значений поля **God** получим в виде таблицы (рис. 5.9).

Max_God
8000

Рис. 5.9

Второй запрос выбирает все записи из таблицы, в которых годовое выпуск равен максимальному по таблице. Конструктор этого запроса на рис. 5.10.

Поле:	Фирма*	God	
Имя таблицы:	Фирма	Фирма	
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:		[Зап1].[Max_God]	
или:			

Рис. 5.10

Объектами этого запроса являются таблица **Фирма** и запрос **Зап1**. В запросе предусматривается вывод всех записей, для которых значение поля **God** равно полю **Max-God**, вычисленному в запросе **Зап1**.

```
SELECT *
FROM Фирма, Зап1
WHERE God.=[Зап1].[Max-God];
```

Здесь в операторе **FROM** указаны имя таблицы и имя запроса, а в операторе **WHERE** в условном выражении указано уточненное имя из запроса (**[Зап1].[Max-God]**).

Результат запроса на рисунке 5.11

Фирма	изделие	код	1кв	2кв	3кв	4кв	год вып.
Троицк	электрочайник	567	500	500	650	650	8 000.00р
Техно	магнитофон	456	900	900	950	1000	8 000.00р

Рис. 5.11.

Другим способом решения этой задачи является создание запроса с подзапросом.

Конструктор запроса представлен на рис. 5.12.

Поле:	Фирма*	God	
Имя таблицы:	Фирма	Фирма	
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:		(Select Max(God) From Фирма)	
или:			

Рис. 5.12

Источником данных для этого запроса является только таблица **Фирма**. Здесь в качестве условия отбора записей используется **SQL** запрос, в котором вычисляется максимальный годовое выпуск по всей таблице.

```
SELECT *
FROM Фирма
WHERE ((God)=(Select Max(God) From Фирма));
```

#### Создание запроса с максимумом по группам и максимумом и средним по всей таблице

**Задача 5.4.** По каждому изделию вычислить максимальный годовое выпуск и вывести информацию о тех изделиях, у которых максимальный выпуск больше среднего по всей таблице.

Для этого сначала создается запрос **Зап1**, вычисляющий средний по всей таблице (Задача 5.2) и запрос, который вычисляет максимальный годовое выпуск по каждому предприятию **Зап2**, затем эти запросы объединяются в один — **Зап3**. Конструктор запроса **Зап2** имеет вид (рис. 5.13).

Поле:	Фирма	God	
Имя таблицы:	Фирма	Фирма	
Сортировка:	Группировка	Мак	
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:			
или:			

Рис. 5.13

В свойствах поля **God** в строке **Подпись** запишем названия столбца **Макс выпуск**.

Конструктор запроса **Зап3** использует в качестве источника записей **Зап1** и **Зап2** и имеет следующий вид (рис. 5.14).

Поле:	Max_God		
Имя таблицы:	Зап2	Зап2	
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:		>=[Зап1].[Avg_God]	
или:			

Рис. 5.14

Этот же запрос в режиме **SQL** имеет следующий вид.

```
SELECT lzd, Max(God) AS [Макс вып.]
FROM Фирма
GROUP BY lzd
HAVING (Max(God) >= (Select Avg(God) From Фирма));
```

В запросе используется предложение **HAVING** которое применяется только после группировки (**GROUP BY**) и определяет условие отбора для уже сформированных итоговых записей. Только те группы, которые удовлетворяют указанным условиям, выводятся в динамический набор записей. В то время, как максимум или среднее вычисляется по всей таблице. Напомним, что предложение **WHERE** в групповом запросе, отбирает записи для вычисления.

Результат запроса на рис. 5.15

изделие	Средний выпуск
магнитофон	8000,00
телевизор	7000,00
электрочайник	8000,00

Рис. 5.15.

Аналогично можно найти изделия, у которых максимальный годовой выпуск меньше среднего по таблице.

```
SELECT lzd, Max(God) AS [Макс выпуск]
FROM Фирма
GROUP BY lzd
HAVING (Max(God) <= (Select Avg(God) From Фирма));
```

**Задача 5.5.:** Вывести для каждого изделия разброс выпусков.

Для того, чтобы решить эту задачу необходимо сначала создать запрос, который будет считать минимальный и максимальный выпуск для каждого изделия **Запрос1** (рис. 5.16).

Поле:	God	God	God
Имя таблицы:	Фирма	Фирма	Фирма
Групповая операция:	Группировка	Min	Max
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			
Имя:			

Рис. 5.16

Теперь создадим запрос **Запрос2** на основе **Запрос1** (рис. 5.17).

Поле:	lzd	Разброс: (Max God)-(Min God)
Имя таблицы:	Запрос1	
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		
Имя:		

Рис. 5.17

Результат выполнения этого запроса на рис. 5.18.

изделие	Разброс
магнитофон	4000
пылесос	1000
телевизор	1000

Рис. 5.18

На языке запросов будет записана следующая программа.

```
SELECT lzd as [изделие], Max(God)-Min(God) as [Разброс]
FROM Фирма
GROUP BY lzd;
```

**Создание перекрестного запроса**

Перекрестный запрос – это специальный тип итогового запроса, позволяющий увидеть вычисляемые значения в виде перекрестной таблицы. При выборе запроса типа **Перекрестный** Access добавляет в таблицу конструктора строку **Перекрестная таблица**.

В этой строке для каждого поля нужно выбрать одно из четырех значений: **Заголовки строк**, **Заголовки столбцов**, **Значение** (выводимое с сетке перекрестного запроса). Поля, являющиеся заголовком строк и столбцов, должны иметь в строке **Групповая операция** значение **Группировка**, а для поля, имеющего установку, **Значение**, нужно выбрать одну из итоговых функций.

Для создания перекрестного запроса на языке **SQL** используется инструкция **TRANSFORM**. Она имеет следующий синтаксис:

**TRANSFORM** <итоговая функция>

**SELECT**-инструкция

**PIVOT** <заголовки столбца>;

Итоговая функция в предложении **TRANSFORM** определяет значения в каждой ячейке результирующего набора записей. **SELECT**-инструкция представляет собой инструкцию, содержащую предложение **GROUP BY**, и имеющую хотя бы два столбца в этом предложении и хотя бы один столбец в предложении **SELECT**, который будет заголовком строк. В предложении **PIVOT** указывается имена полей, которые должны стать заголовками столбцов.

**Задача 5.6.** Создать перекрестный запрос, который будет отражать годовой выпуск каждого изделия, выпускаемого каждым предприятием, а так же количество наименований товаров и и суммарный годовой выпуск каждым предприятием.

Конструктор этого запроса представлен на рис. 5.19.

Поле:	Firm	lzd	Сумма God	Сумма God	Кол-во God
Имя таблицы:	Фирма	Фирма	Фирма	Фирма	Фирма
Групповая операция:	Группировка	Группировка	Sum	Sum	Count
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:					
Имя:					

Рис. 5.19

Результат запроса на рис. 5.20.

Фирма	Сумма God	Кол-во изд.	магнитофон	пылесос	телевизор	электрочайник
Аксел	7000	1			7000	
Меридиан	15000	3	4000	5000	6000	
Стройдвор	18000	3	4000		6000	8000
Техно	24000	4	8000	4000	7000	5000
Тораз	10000	2	5000	5000		

Рис. 5.20

В **SQL** перекрестном запросе программа на языке запросов будет следующей **TRANSFORM**. Sum(God):

```
SELECT Firm, Sum(God) AS [Сумма God], Count(*) AS [Кол-во изд],
```

```
FROM
```

```
GROUP BY Firm
```

```
PIVOT lzd;
```

В приведенном примере результатом столбцов является сумма годового выпуска на каждой фирме по каждому изделию (предложение **TRANSFORM**). Заголовками столбцов являются содержимое поля **lzd** (**PIVOT lzd**), заголовками строк будет все, что перечислено в предложении **SELECT**. Это могут быть только поля группировки и итоговые функции. В нашем примере это значения поля **Firm**, функции

Sum(сумма годового выпуска в целом по фирме) и функции Count ( количество видов изделий выпускаемых каждой фирмой).

Результат такого запроса такой же как на рис. 5.20.

#### ЛИТЕРАТУРА

1. В. Пасько. Access 97. Киев: «Ирина», ВНУ, 2000.
2. С. Робинсон. Microsoft Access 2000, СПб.
3. К. Гетц, П. Литвин, М. Гилберт. Access 2000: Руководство разработчика. Киев: «Ирина», ВНУ, 2000.
4. А. Васильев, А. Андреев. VBA в Office 2000 – СПб., 2001.
5. П. Нортон, В. Андерсон. Разработка предложений в Access 97 – СПб., 1998.

**X**

#### СОДЕРЖАНИЕ

Введение.....	3
Лабораторная работа.....	3
Часть 1.....	3
Часть 2.....	7
Часть 3.....	13
Часть 4.....	22
Часть 5.....	25
Литература.....	32

*MAX, MIN, Avg, Count  
First Last*

*условие (Having)*

Министерство образования Российской Федерации

Государственное образовательное учреждение  
высшего профессионального образования  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ УПРАВЛЕНИЯ

Институт информационных систем управления  
Кафедра компьютерных технологий

Утверждено  
первым проректором ГУУ  
проф. Ю.Л. Старостиным  
10 июня 2002 г.

### МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторной работы

по дисциплине

**«КОМПЬЮТЕРНАЯ ПОДГОТОВКА»**  
С ПРИМЕНЕНИЕМ СУБД ACCESS

Выпуск 1

для студентов специальности  
«Менеджмент организации» – 061100