

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Северо-Кавказский филиал
ордена Трудового Красного Знамени федерального государственного бюджетного образовательного учреждения высшего образования
"Московский технический университет связи и информатики"



Методические указания
к лабораторным работам

ТЕХНОЛОГИИ БАЗ ДАННЫХ

Запросы. Формы. Отчеты

Направление подготовки:

09.03.01 Информатика и вычислительная техника

Профиль:

Вычислительные машины, комплексы, системы и сети
Программное обеспечение и интеллектуальные системы

Ростов-на-Дону
2019

УДК 681.3.06 (076)
ББК 32.07

Чикалов А.Н. Технологии баз данных. Запросы. Формы. Отчеты. Методические указания к лабораторным работам. Ростов-на-Дону: Северо-Кавказский филиал МТУСИ, 2019.- 76 с.

В пособии изложены методические рекомендации, содержательные материалы и контрольные задания для проведения лабораторных и практических занятий по формированию простых и сложных запросов, форм и отчетов реляционных баз данных. В качестве инструмента использована среда разработки MS Access. Пособие содержит необходимые справочные материалы.

Методические указания предназначены для студентов, обучающихся по направлению подготовки 09.03.01 Информатика и вычислительная техника, профилей Вычислительные машины, комплексы, системы и сети, Программное обеспечение и интеллектуальные системы.

Пособие предназначено для использования при изучении дисциплин Технологии баз данных, а также может быть использовано преподавателями и студентами при изучении родственных дисциплин и в процессе самостоятельной работы.

Учебное пособие обсуждено и одобрено на заседании кафедры ИВТ
Протокол №1 от 26.08.2019

Рецензент Зав. кафедрой ИВТ д.т.н. профессор Соколов С.В.

СОДЕРЖАНИЕ

1. Создание простых запросов	4
1.1. Конструирование простых запросов	4
1.2. Создание запросов с параметром	13
2. Создание сложных запросов	15
2.1. Запросы с вычисляемыми полями	15
2.2. Групповые операции в запросах	23
2.3. Запросы к связанным таблицам БД	26
3. Разработка форм	35
3.1. Создание формы с помощью мастера	35
3.2. Создание форм с помощью конструктора	46
3.3. Создание и удаление элементов управления	53
4. Разработка отчетов	67
4.1. Создание отчета	67
4.2. Модификация отчета	76

1. СОЗДАНИЕ ПРОСТЫХ ЗАПРОСОВ

Цель

1. Получить представление об этапах и принципах проектирования баз данных от постановки задачи до использования СУБД;
2. Приобрести навыки разработки и использования запросов к базе данных.

Учебные вопросы

- 1.1. Конструирование простых запросов;
- 1.2. Создание запросов с параметром.

Литература для подготовки к занятию

1. Иллюстрированный самоучитель по Access.
<http://www.selfteachers.ru>
2. Вейскас Д. Эффективная работа с Microsoft Access/ - СПб.: Питер Ком, 1999. – 976с. стр.322-354.
3. Работа в Microsoft Access XP. <http://www.intuit.ru>
4. Уроки программирования. Базы данных. <http://www.life-prog.ru>
5. Иллюстрированный самоучитель по Access 2002.
http://news.claw.ru/Office/Access_2002

Содержание отчета

1. Название работы;
2. Для каждого из заданий : название задания и материалы в объеме, указанном в задании.

Задание 1.1. Конструирование простых запросов

Постройте запросы, осуществляющие выборку следующих данных.

Запросы с простыми условиями отбора:

105. Сотрудники, проживающие в Ростове-на-Дону.
106. Сотрудники из г.Шахты с окладом более 10000 руб.
107. Сотрудники из г.Шахты с окладом более 10 т.руб, не являющиеся директорами магазинов.
108. Сотрудники, проживающие в г.Шахты или г.Ростове-на-Дону.
109. Список товаров стоимостью более 10 т.руб.
110. Список товаров стоимостью более 10 т.руб, не являющиеся компьютерами.
112. Выведите самый дешевый товар.
113. Выведите самый дорогой товар.
114. Выведите самый дешевый компьютер, имеющийся на складе.
115. Выведите алфавитный список товаров стоимостью в диапазоне от 5 до 10 т.руб.

132. Выведите алфавитный список товаров стоимостью в диапазоне между 25 и 30 т.руб. Используйте оператор BETWEEN.

133. Выберите сотрудников с поощрительным коэффициентом между 2 и 5. Используйте оператор BETWEEN.

134. Выберите список товароведов. Используйте оператор LIKE.

135. Выберите список сотрудников, по задаваемому фрагменту фамилии.

136. Выберите список товаров фирмы Aser. Используйте оператор LIKE.

Простые запросы

Пользоваться сортировкой и фильтрацией очень удобно, но при работе с несколькими таблицами их возможностей недостаточно. В таких случаях применяются запросы. Самый распространенный тип запросов отображает записи, удовлетворяющие определенным условиям, но существуют и другие типы, приведенные ниже.

Запрос на выборку извлекает данные из одной или нескольких таблиц и представляет их в табличном виде. Этот тип запроса можно использовать для группировки записей, вычисления сумм, средних величин и других итоговых значений. Работая с результатами запроса, можно одновременно редактировать данные из нескольких таблиц.

Параметрический запрос запрашивает ввод параметров (например, начальную и конечную дату). Этот тип запросов часто используется для получения отчетов за определенный период времени.

Перекрестный запрос выполняет расчеты и группирует данные для анализа информации. Для элементов, расположенных в левом столбце и в верхней строке результатов запроса, могут вычисляться итоговые значения (сумма, количество или средняя величина). Ячейки на пересечении строк и столбцов также содержат вычисляемые значения.

Запрос на действие вносит множественные изменения за одну операцию. Собственно, это запрос на выборку, который выполняет определенные действия над результатами отбора. Возможны четыре типа действий: обновление, удаление и добавление записей и создание таблицы. В двух последних случаях результаты запроса на выборку либо добавляются в существующую таблицу, либо для них создается новая таблица.

Access включает также запросы SQL, но в этом пособии они не рассматриваются.

Основные различия между применением фильтрации и сортировки и использованием запросов состоят в следующем:

- фильтрация и сортировка выполняются быстрее, чем запросы;
- фильтры и условия сортировки хранятся временно. Запрос можно сохранить на диске и использовать в дальнейшем;

- фильтры и сортировка применяются только к активной таблице или форме. Запрос может базироваться на нескольких таблицах и других запросах, причем необязательно, чтобы они были открыты.

Запрос можно создать вручную или с помощью мастера. При этом создается выражение, описывающее критерий, которому должны соответствовать данные в указанных таблицах. В результате выполнения запроса записи, отвечающие заданным условиям, отобразятся в табличном виде.

Создание запроса в режиме конструктора

Создание запроса возможно:

1. В режиме конструктора (наиболее гибкий и правильный способ);
2. С помощью мастера.

Рассмотрим подробнее способ создания запроса в режиме конструктора, показывающего курсантов, обучающихся в 1111 уч. группе.

Для выполнения этой задачи необходимо:

1. В контейнере БД на панели **Объекты** выбрать объект **Запросы**. При этом в правом окне появятся возможные варианты создания запроса. Выбрать пункт **Создание запроса в режиме конструктора**.

2. В открывшемся окне **Добавление таблицы** необходимо добавить таблицы, которые будут участвовать в запросе (рис.1.1).

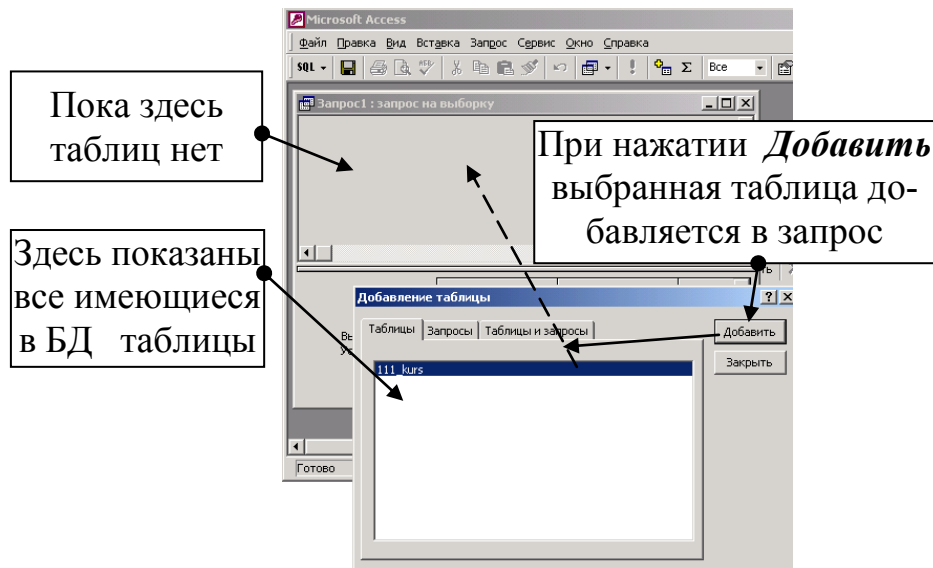


Рис.1.1. Добавление таблиц в запрос

3. Отметить таблицу **111_kurs** и нажать **Добавить**. В окно конструктора запроса добавится выбранная таблица, содержащая списки полей. Эта верхняя область для размещения участвующих в запросе таблиц называется **Область таблиц**. Вверху каждого списка полей имеется звездочка, пред-

ставляющая собой все поля таблицы. Ключевое поле отображается полужирным шрифтом.

При наличии других таблиц, если это необходимо, можно добавить и их. После этого нажать кнопку **Заккрыть**. Внешний вид конструктора после добавления таблиц представлен на рис. 1.2.

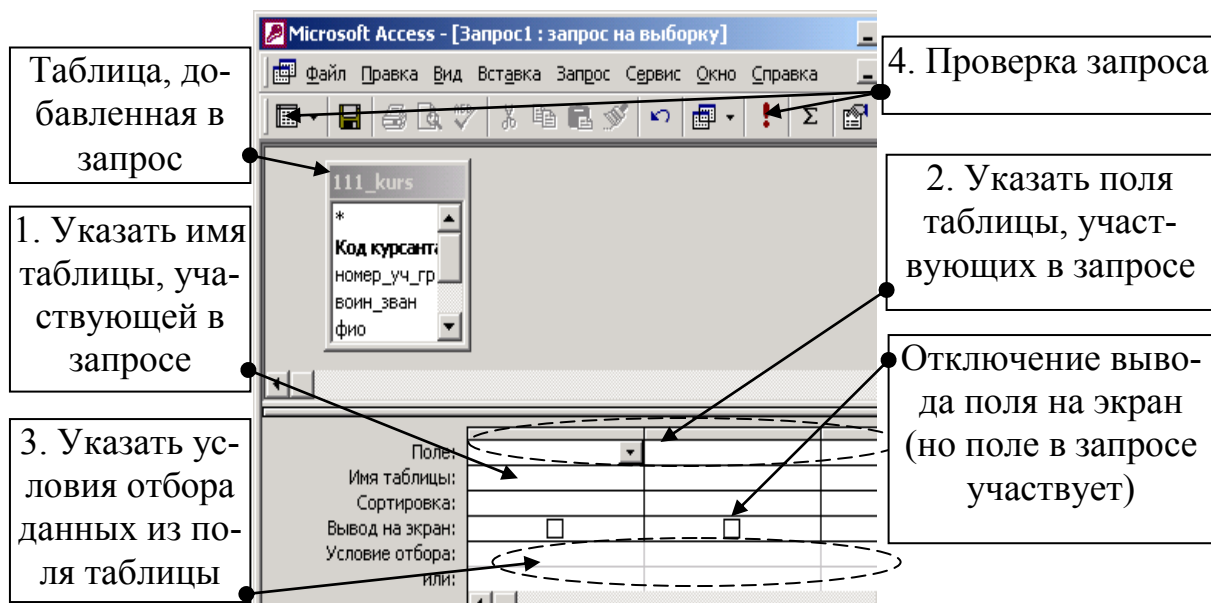



Рис.1.2. Вид запроса в Конструкторе после добавления таблицы

4. Нижняя часть окна запроса занята **Бланком запроса**, предназначенным для построения условий отбора. Условия отбора строятся по отношению к конкретным полям таблицы (или таблиц). Чтобы включить поля в запрос, нужно перетащить их из списка таблицы вверху окна в последовательные столбцы бланка запроса. Перетащите следующие поля: **Номер_уч_гр, Фео**.

Включить поля в бланк запроса возможно также следующими способами:

- щелкнуть дважды на нужном поле в таблице;
- установить курсор на чистом поле **Бланка запроса** в строке **Поле**, открыть выпадающее меню и выбрать нужное поле;
- чтобы скопировать сразу все поля таблицы, выделите нужный список (щелкнув дважды на имени таблицы), а затем перетащите выделенный объект на бланк запроса. Когда отпустите кнопку мыши, все поля разместятся в последовательных столбцах бланка.

5. В строке **Условие отбора** (рис.1.3) в поле **111_kurs** указать условие **1111**. Окно запроса должно выглядеть, как показано на левой стороне рис. 5.3.

6. Щелкните на кнопке **Запуск** (Run) , чтобы выполнить запрос и отобразить результаты в виде таблицы (правая сторона рис.1.3).

7. Сохранить запрос под логически понятным именем.

номер_уч_гр	фами
1111	Сергей И.П.
1111	Иванов А.А.
1111	Василенко А.Д.

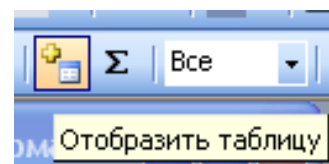
Для редактирования запроса щелкните на кнопке **Вид** (View), чтобы вернуться в режим конструктора.

Рис.1.3. Бланк запроса и результат выполнения запроса

Добавление таблиц в запрос

В процессе редактирования запроса может возникнуть необходимость удаления и добавления таблиц в запрос. Для удаления таблицы достаточно выбрать таблицу в области таблиц и нажать клавишу **Del**.

Чтобы добавить в запрос дополнительные таблицы, откройте диалоговое окно **Добавление таблицы**. Для этого щелкните правой кнопкой мыши в верхней части окна запроса и воспользуйтесь командой **Добавить таблицу** (Show Table) в контекстном меню или щелкните на кнопке **Отобразить таблицу** (Show Table) на панели инструментов.



Установка свойств полей в запросе

Обычно поля, выводимые в наборе записей запроса, наследуют свойства для полей из соответствующих таблиц. Однако в ряде случаев имеет смысл изменить свойства или установить их, если поля формируются как вычисляемые и для них просто не существует родительских свойств.

Чтобы задать свойства поля, щелкните на любой ячейке соответствующего столбца бланка запроса. Далее можно поступить одним из способов:

- нажать кнопку **Свойства** на панели инструментов;
- выбрать команду **Вид-Свойства** из меню;
- выбрать команду **Свойства** из контекстного меню, извлекаемого правой кнопкой.



В появившемся окне **Свойства поля** (рис.1.4) измените необходимые свойства. Следует иметь в виду, что наследуемые свойства поля в этом окне не отображаются, поэтому их можно ввести заново.

Рис.1.4. Окно задания свойств поля в запросе

Задание условий отбора в запросе

Если отобрать необходимо поля с конкретным значением ("жесткое" условие), то в строке *Условие отбора* просто вводится это значение. Так было сделано в предыдущем примере: нас интересуют курсанты 1111 учебной группы, поэтому условие формулируется просто **1111**. Записи отбираются при совпадении значения поля с условием.

Текстовое значение должно быть заключено в кавычки, но Access сам добавляет кавычки при идентификации условия как текстового значения. Если это не очевидно (условие состоит из текстовых символов цифр), то об этом должен позаботиться пользователь.

Запрос: *Показать курсантов, имеющих оценки "хорошо" и "отлично" и обучающихся в 1111 уч.гр.*

В этом запросе не одно, а два условия, которые выполняются одновременно.

В случае, если необходимо найти записи, содержащие значения поля, находящиеся в некотором диапазоне (например "от 2 до 4", "больше или равно 18", обучающиеся в 1112 и 1114 учебных группах), то используются логические операторы, либо знаки сравнения. Некоторые типовые операторы представлены в таблице.

Логические операторы		
OR	(1990) OR (1999)	ИЛИ. Вывод записей, в которых значение этого поля =1990 или =1999.
AND	(>3) AND (<10)	И. Вывод записей, в которых значение этого поля больше 3 и меньше 10, т.е. лежит в диапазоне от 3 до 10, причём 3 и 10 не входят в диапазон.
NOT	NOT 100	Отрицание. Вывод записей, в которых значение этого поля не равно 100. <i>NOT NULL</i> - Вывод записей, в которых значение этого поля не пустое (равно чему-либо).
Операторы сравнения		
=	=240 ="сержант"	Вывод записей, в которых значение этого поля равно 240; ="сержант".
<	<31 <30.12.2006	Вывод записей, в которых значение этого поля меньше 31; вывод записей, в которых значение даты, содержащееся в этом поле раньше 30.12.2006
<=	<=56	Вывод записей, в которых значение этого поля меньше или равно 56
>	>44	Вывод записей, в которых значение этого поля больше 44
>=	>=3.08.2000	Вывод записей, в которых значение даты в этом поле больше или равно 3.08.2000

<	< 0	Вывод записей, в которых значение в этом поле не равно 0
---	-----	---

Строится запрос аналогично. Внешний вид бланка запроса и результат его выполнения представлен на рис.1.5.

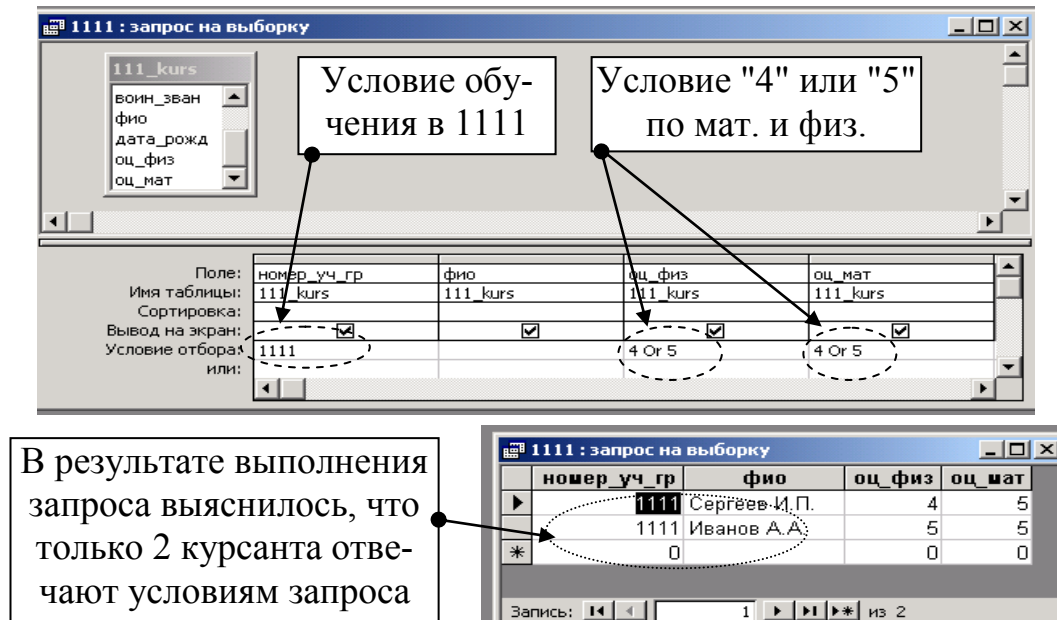


Рис.1.5. Бланк запроса и результат работы запроса с логическими условиями

Условия отбора оценок созданы с использованием логического оператора **ИЛИ (OR)**. Логические операторы выполняются над частными условиями отбора, находящимися в одной строке: 4 ИЛИ 5 для каждой дисциплины. Однако в одной строке запроса находятся логические операторы для отбора оценок по физике, математике и условию отбора по номеру учебной группы. Поэтому все эти три записи выполняются вместе по логическому оператору **И**. Иными словами, реализуется по выбранным полям сложное логическое условие отбора по формуле: **1111 AND (4 OR 5) AND (4 OR 5)**.

Если отбор осуществляется по отдельному полю и по горизонтальной строке **Условие отбора** не образуется сложных связей, то оператор **ИЛИ** можно реализовать, если для этого поля одно условие внести в строке **Условие отбора**, а последующие в строке **или**. Такое расположение оценок эквивалентно записи **4 OR 5**.

BETWEEN, IN и LIKE

Кроме обычных операторов сравнения Access предоставляет три специальных оператора, полезных для отбора данных, выводимых в наборе записей запроса:

- BETWEEN** Определяет диапазон значений. *Between 10 And 20* означает то же самое, что и $\geq 10 \text{ And } \leq 20$.
- IN** Задаёт используемый для сравнения список значений. Выражение *IN ("WA", "CA", "ID")* означает то же самое, что и *"WA" Or "CA" Or "ID"*
- LIKE** Оператор полезен при поиске образцов в текстовых полях. В образце поиска можно включить символы шаблона:
- ? - заменяет любой символ в данной позиции,
 - * - обозначает любое количество (включая нулевое) символов в данной позиции;
 - # - указывает, что в данной позиции должна стоять цифра. Допустимый диапазон символов для этой позиции заключается в квадратные скобки;
 - ! - используется для указания исключений.
- Диапазон [0-9] указывает, что в этой позиции должна находиться цифра, [a-z] — буква, а [!0-9] — любой символ, кроме цифр от 0 до 9. Например, с помощью выражения *Like "?[a-k]d[0-9]*"* проверяется наличие произвольного символа в первой позиции, буквы от *a* до *k* во второй позиции, буквы *d* в третьей позиции, любого символа от 0 до 9 в четвертой позиции и произвольного числа символов в конце строки.

Условия отбора для дат и времени

Microsoft Access хранит значения дат и времени как числа с плавающей точкой двойной точности. Значение слева от десятичной точки соответствует дате, а дробная часть числа представляет время суток. К счастью, вам не нужно заботиться о преобразовании внутреннего формата хранения при задании условий отбора для дат, поскольку Access обрабатывает значения дат и времени в различных форматах.

Чтобы сообщить Access о том, что вы вводите дату и время, заключайте значение в символы числа (#). Для указания конкретной даты используйте нотацию, которая вам кажется наиболее удобной. Например, #15 Апрель 1997#, #15/04/97\$ и = 15-апр-1997# определяют одну и ту же дату. Точно так же #5:30 PM# и = 17:30# определяют время «семнадцать часов тридцать минут».

Access предоставляет несколько функций, которые могут оказаться полезными при задании условий отбора для дат и времени.

- Day(дата)** Возвращает значение дня месяца в диапазоне от 1 до 31. Если вам нужны записи со значениями в поле Date, превышающими десятое число любого месяца, задайте вычисляемое поле *Day([Date])* и определите для него условие отбора > 10

Month (дата)	Возвращает значение месяца года в диапазоне от 1 до 12. Если вы хотите найти все записи, в которых значение месяца в поле Date равно июню, задайте вычисляемое поле <i>Month([Date])</i> и определите для него условие отбора 6
Year(дата)	Возвращает значение года в диапазоне от 100 до 9999. Если вы хотите найти записи, в которых значение года в поле Date равно 1997, задайте вычисляемое поле <i>Year([Date])</i> и определите для него условие отбора 1997
Weekday(дата)	По умолчанию возвращает целое число от 1 (воскресенье) до 7 (суббота), соответствующее дню недели. Для поиска записей для рабочих дней задайте вычисляемое поле <i>Weekday([Date])</i> и определите для него условие отбора <i>Between 2 And 6</i>
Hour(дата)	Возвращает целое число от 0 до 23, представляющее значение часа. Чтобы найти представления в клубах, начинающиеся до полудня, задайте вычисляемое поле <i>Hour([Date])</i> и определите для него условие отбора <12
Datepart (интервал, дата)	Возвращает номер квартала или номер недели в зависимости от заданного кода интервала. Полезные коды интервала — «q» для определения квартала (значение от 1 до 4) и «ww» — для определения порядкового номера недели в году (значение от 1 до 53). Например, чтобы отобрать даты во втором квартале, задайте вычисляемое поле <i>Datepart(«q», [Date])</i> и определите для него условие 2
Date()	Возвращает текущую системную дату. Чтобы найти записи с начальной датой более чем на 30 дней ранее текущей, введите < <i>Date()-30</i> в качестве условия отбора
Month-Name(Nº)	Возвращает название месяца. В качестве аргумента функции № выступает номер месяца. Его можно получить по функции Month
Weekday-Name(Nº)	Возвращает название дня недели. Аргументом функции № выступает номер дня недели

Сортировка в запросе

Access выводит строки в порядке их ввода в таблицы. Порядок вывода можно изменить.

Строка **Сортировка** (Sort) (третья на бланке) позволяет указать поле и принцип сортировки (по возрастанию или убыванию) записей. Щелкните в ячейке **Сортировка** (Sort) в столбце **Фино**, щелкните на стрелке и щелкните на команде **По возрастанию** (Ascending). При выполнении запроса фамилии курсантов будут выведены по алфавиту.

Сортировка по нескольким полям выполняется в порядке расположения полей в бланке запроса. слева направо. Если, например, сортировать поля с фамилией и именем, то будут выведены подряд курсанты с одинаковыми

фамилиями, а имена расположены по алфавиту в пределах одинаковых фамилий.

Количество записей при выборке

Пример: Требуется найти фамилию курсанта, сдавшего информатику с самым низким балом.

Для решения этой задачи необходимо в запросе отсортировать записи по возрастанию оценки по информатике и вывести одну запись. Это и будет фамилия курсанта, сдавшего информатику с самым низким балом.

В строке **Сортировка** конструктора запросов необходимо выбрать **По возрастанию**, затем в окне панели инструментов **Набор значений** установить количество записей 1.

Внешний вид запроса в конструкторе и результат его выполнения представлен на рис.1.6.

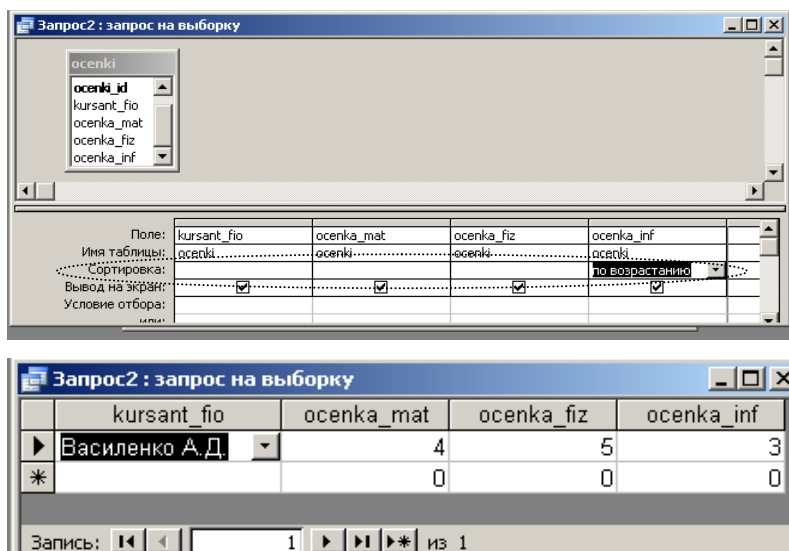
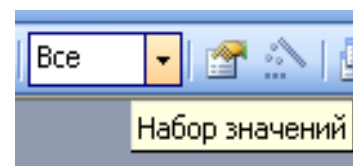


Рис.1.6. Запрос с сортировкой

Задание 1.2. Создание запросов с параметром

Постройте следующие запросы:

216. Выберите товары задаваемого типа стоимостью не выше задаваемой.
217. Найдите характеристики сотрудника с известной фамилией.
218. Найдите магазины уже открытые к указанному времени.
219. Найдите магазины, работающие в указанное время.
220. Найдите список магазинов по задаваемой категории в задаваемом городе.

Запросы с параметром

Для расширения возможностей в запросе вводят параметр, который запрашивается при выполнении запроса. Чтобы определить параметр, введите в строку **Условие отбора** вместо конкретного значения имя или фразу, заключенную в квадратные скобки. То, что заключено внутри квадратных скобок, будет рассматриваться как имя параметра. Оно выводится в окне диалога при выполнении запроса. Поэтому запись внутри квадратных скобок должна быть содержательной. Параметров в запросе может быть несколько, но их имена должны быть различны.

Например, для отбора сотрудников по задаваемой фамилии запрос должен быть сформирован так, как показано на рис.1.7. На этом же рисунке показано окно задания параметра для этого запроса.

По умолчанию при вводе параметра данные считаются текстовыми. Если требуется изменить тип данных, то необходимо выполнить команду **Запрос-Параметры**. В окне Параметры запроса следует ввести имя каждого параметра и тип данных для него из раскрывающегося списка. Это позволит кроме всего осуществлять проверку введенных значений на соответствие типов.

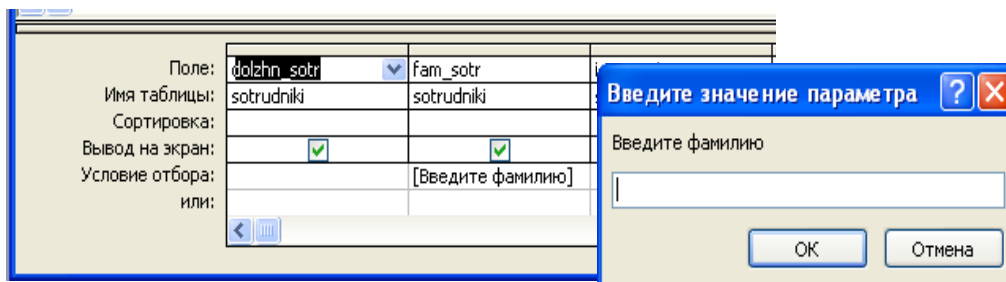


Рис.1.7. Запрос с параметром и окно диалога для ввода его значения

Вопросы для самопроверки

1. Каким образом выполняются запросы в реляционных базах данных?
2. Для каких целей используются результаты запросов?
3. Сформулируйте технологию формирования запроса к БД.
4. Как создаются запросы с параметром? Каковы преимущества такого запроса?
5. Как выполняется сортировка в запросе?
6. Выведите алфавитный список товаров стоимостью в диапазоне между от... до... .
7. В чем суть запросов с групповыми операциями?
8. Выведите самый дорогой товар.
9. Выберите список товароведов. Используйте оператор LIKE.
10. Найдите магазины уже открытые к указанному времени.
11. Покажите максимальную стоимость аренды в каждом городе.

2. СОЗДАНИЕ СЛОЖНЫХ ЗАПРОСОВ

Цель

1. Получить представление об этапах и принципах проектирования баз данных;
2. Приобрести навыки разработки и использования сложных запросов к базе данных.

Учебные вопросы

- 2.1. Запросы с вычисляемыми полями;
- 2.2. Групповые операции в запросах;
- 2.3. Запросы к связанным таблицам БД.

Литература для подготовки к занятию

6. Иллюстрированный самоучитель по Access.
<http://www.selfteachers.ru>
7. Вейскас Д. Эффективная работа с Microsoft Access/ - СПб.: Питер Ком, 1999. – 976с. стр.322-354.
8. Работа в Microsoft Access XP. <http://www.intuit.ru>
9. Уроки программирования. Базы данных. <http://www.life-prog.ru>
10. Иллюстрированный самоучитель по Access 2002.
http://news.claw.ru/Office/Access_2002

Содержание отчета

1. Название работы;
2. Для каждого из заданий : название задания и материалы в объеме, указанном в задании.

Задание 2.1. Запросы с вычисляемыми полями;

Постройте запросы, осуществляющие выборку данных. Для каждого запроса представить бланки запросов:

421. Вычислите совокупную стоимость содержания магазинов в каждом городе.
424. Выведите сумму начисленной зарплаты и сумму подоходного налога каждого сотрудника.
425. Вычислите суммарные расходы на зарплату всех сотрудников и суммарный подоходный налог.
426. Выведите стоимость аренды, коммунальные платежи и суммарные расходы на содержание каждого магазина.
427. Посчитайте суммарные расходы на содержание всех магазинов.
428. Выведите суммарные затраты на зарплату и содержание всех магазинов и суммарный объем расходов.

429. Выведите одной строкой должность, имя и фамилию всех сотрудников. Используйте операцию сцепления строк.

430. Сформируйте данные для ведомости на выдачу зарплаты с полями: должность_имя_фамилия, начислено, налог, к_выдаче. Используйте операцию сцепления строк.

431. Подготовьте данные для рекламы магазинов с полями: название_магазина_адрес, время_работы_с_по, аренда_коммунальные_итого_. Используйте операцию сцепления строк.

437. Выведите алфавитный список сотрудников с указанием возраста.

440. Выведите продолжительность работы каждого магазина.

Вычисляемые поля

Для выполнения вычислений с любыми полями базовой таблицы можно создать выражение и определить его в качестве нового поля запроса. При создании вычисляемого поля можно использовать арифметические операции и любые встроенные функции Access (некоторые из них, связанные с обработкой дат и времени, приведены выше). Например, запись о стоимости покупки содержит поля цены отдельного товара и количество товаров, но общая стоимость покупки (количество товаров, умноженное на цену отдельного товара) в ней не хранится. Чтобы включить общую стоимость покупки в набор записей запроса, нужно ввести выражение $\text{Количество} * \text{Стимость}$ в пустую ячейку строки **Поле** бланка запроса.

Вычисляемое поле может также содержать результат конкатенации (объединения) значений текстовых полей, строковых констант или числовых данных. Чтобы создать строковую константу, заключите текст в двойные или одинарные кавычки. В качестве оператора конкатенации используется символ "&".

В выражениях используются следующие операторы:

+	Складывает два числовых выражения
-	Вычитает из первого числового выражения второе
*	Перемножает два числовых выражения
/	Делит первое числовое выражение на второе
\	Округляет два числовых выражения до целых значений и делит первое на второе. Результат округляется до целого
^	Возводит первое числовое выражение в степень, задаваемую вторым числовым выражением
MOD	Округляет оба числовых выражения до целых значений, делит первое на второе и возвращает остаток
&	Создает новую текстовую строку, присоединяя вторую строку к концу первой. Если один из операндов является числом, Access перед проведением конкатенации преобразует его в строку символов

Например, для конкатенации двух полей в вычисляемом поле необходимо выполнить следующее:

1. Установить курсор в чистое окно строки **Поле**.
2. Набрать имена присоединяемых полей, установив между ними знак конкатенации: **dolzhn_sotr&fam_sotr**. После щелчка курсором на соседнем поле перед набранным выражением будет установлено слово **Выражение** с очередным порядковым номером и двоеточием, а имена полей будут взяты в квадратные скобки.

3. Выполнить запрос. В указанном поле появится строка из наименования должности и фамилии сотрудника.



Рис.2.1. Окно ввода выражения

Для удобства ввода можно открыть окно **Область ввода**, нажав **Shift+F2** (рис.2.1).

Более удобный способ конструирования выражений для вычисляемого поля основан на использовании построителя выражений.

Использование построителя выражений

Построитель выражений предназначен для:

- конструирования сложных условий отбора, включающих вложенные функции и условия (выражение вставляется в строку запроса **Условие отбора**).
- создания вычисляемых полей в запросе (выражение вставляется в строку запроса **Поле**).

Для рассмотрения этого вопроса добавим в таблицу **kurs** три поля:

1. kursant_ves – вес курсанта (числовое поле)
2. kursant_rost – рост курсанта (числовое поле)
3. kursant_data – дата рождения курсанта (дата/время, краткий формат даты)

Для этого необходимо открыть таблицу **kurs** в режиме конструктора и

kursant_id	kursant_fio	kursant_ves	kursant_rost	kursant_data
2	Иванов А.А.	85	180	03.08.1988
3	Сухарев И.Д.	90	192	05.04.1988
4	Андронов Н.Т.	79	184	23.03.1987
5	Межинский Р.О.	92	175	12.12.1987
6	Василенко А.Д.	68	170	10.11.1989
(Счетчик)		0	0	

Рис.2.2. Внешний вид исходной таблицы

добавить перечисленные поля. Заполнить новые поля соответствующими данными и закрыть таблицу.

Внешний вид заполненной таблицы представлен на рис.2.2.

Конструирование сложных условий отбора

Пусть требуется вывести фамилии курсантов, родившихся в 1988 году.

В начале требуется уяснить, что означает "родившиеся в 1988 году". Резонно будет заметить, что дата рождения этих курсантов должна лежать в пределах от 1.01.1988 до 31.12.1988 включительно. Поэтому диапазон дат, которые нас интересуют: $\geq 1.01.1988$ и $\leq 31.12.1988$.

Для построения такого, достаточно сложного условия отбора воспользуемся построителем выражений. На рис.2.3 изображён общий подход использования построителя выражений для решения поставленной задачи.

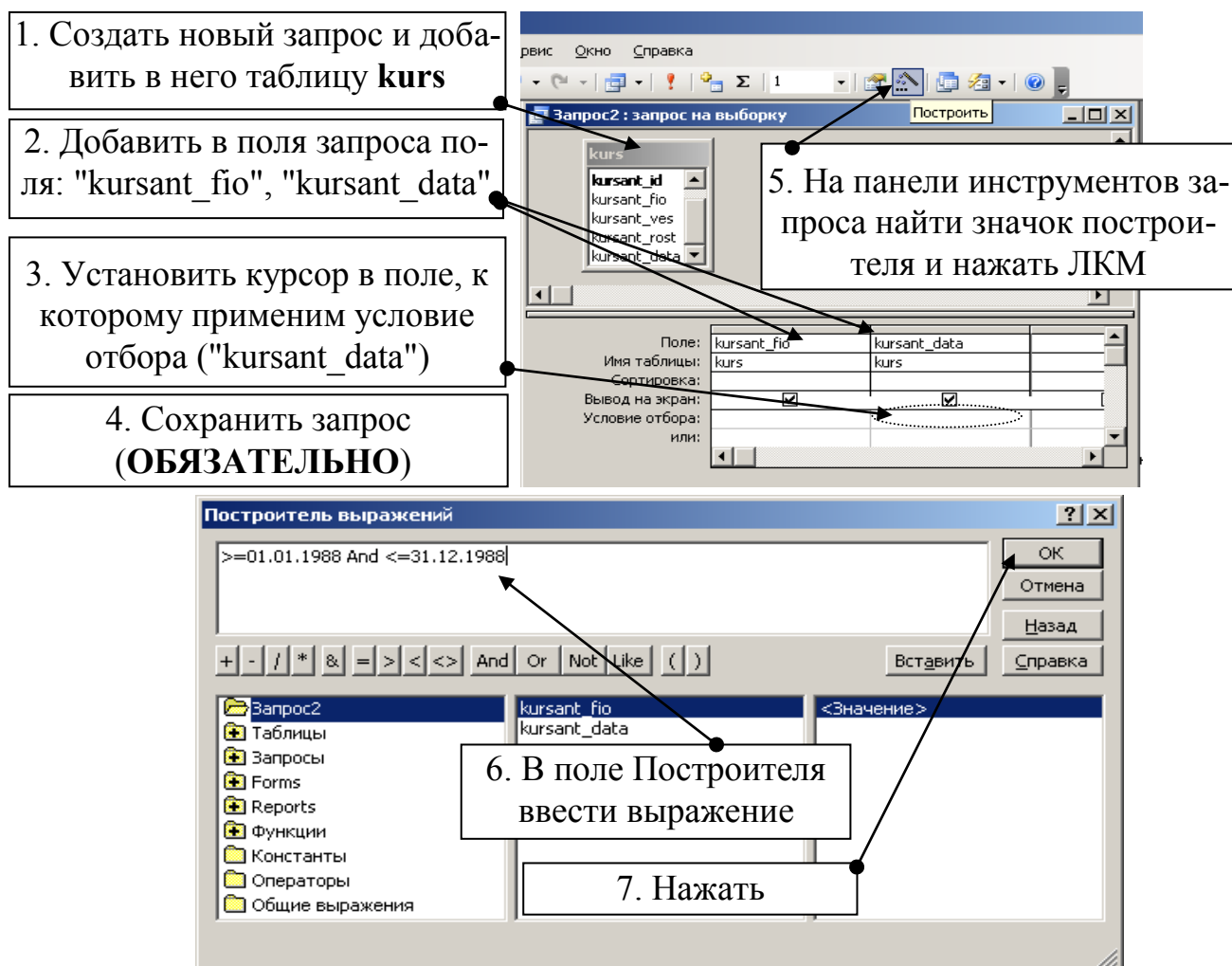


Рис.2.3. Построитель выражений

Условие отбора в конструкторе запросов представлено на рис.2.4, а результат работы запроса – на рис. 2.5.

Создание вычисляемых полей в запросе

Создание вычисляемых полей – ещё одна важная задача, которая может быть решена с помощью построителя выражений.

Отличительной чертой вычислимого поля является то, что на самом деле такого поля ни в одной таблице БД может не существовать. Оно создаётся в момент выполнения запроса и уничтожается после его закрытия.

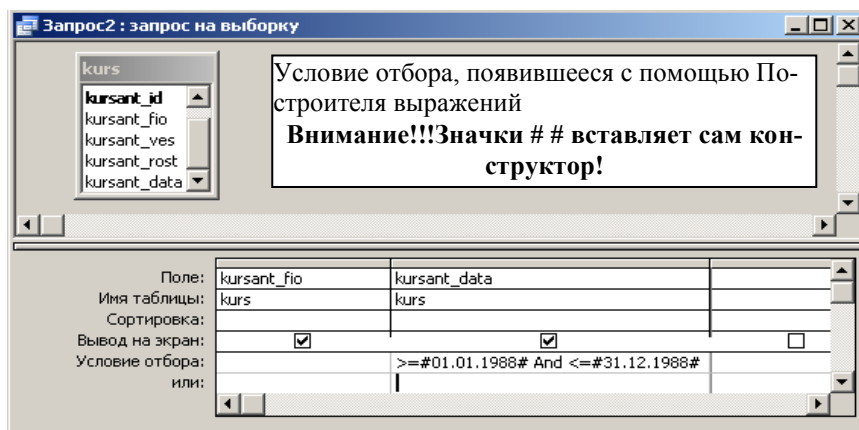
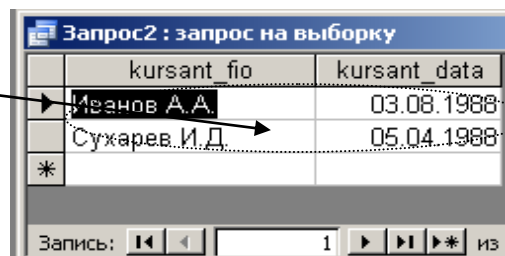


Рис.2.4. Условие отбора

Условию отбора
удовлетворяют
только две
записи

Рис.2.5. Результат
выполнения запроса

Вычисляемые поля создаются для того, чтобы на основе данных, хранящихся в полях таблицы, или запроса получить новые данные.

Единственное правило, которое должен выполнить пользователь следующее: ОПЕРАНДЫ (ПОЛЯ), УЧАСТВУЮЩИЕ В ВЫРАЖЕНИИ ДЛЯ ВЫЧИСЛИМОГО ПОЛЯ ДОЛЖНЫ БЫТЬ, КАК ПРАВИЛО, ОДНОГО ТИПА.

Иными словами, нельзя умножить данные, хранящиеся в поле текстового типа, на данные в поле числового формата и т.п.!

Пример. На основе данных о росте и весе курсанта создать запрос, который выведет фамилии курсантов, у которых разность между ростом и весов превышает 100 (признак избыточного веса).

Для решения этой задачи необходимо создать новый запрос, добавить в него таблицу **kurs** и в поля запроса добавить поля, которые в нём участвуют (рис.2.6).

В открывшемся окне Построителя выражений набрать выражение и нажать **ОК** (рис.2.7).

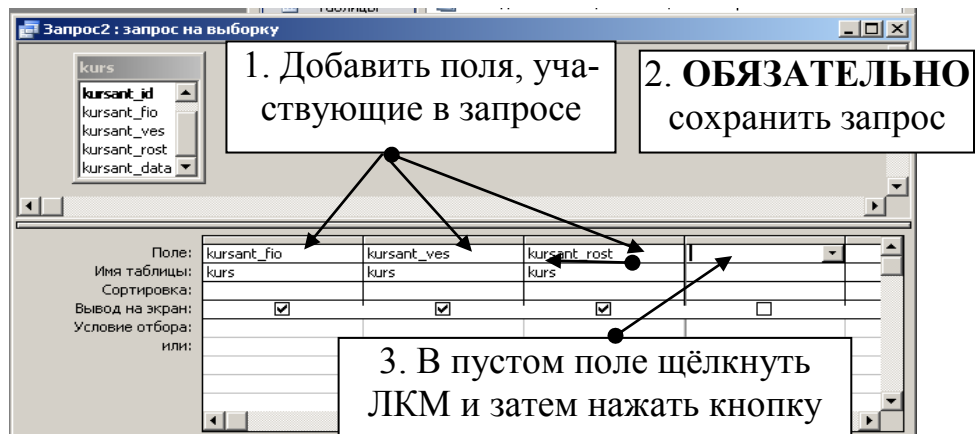


Рис.2.6. Задание вычисляемого поля

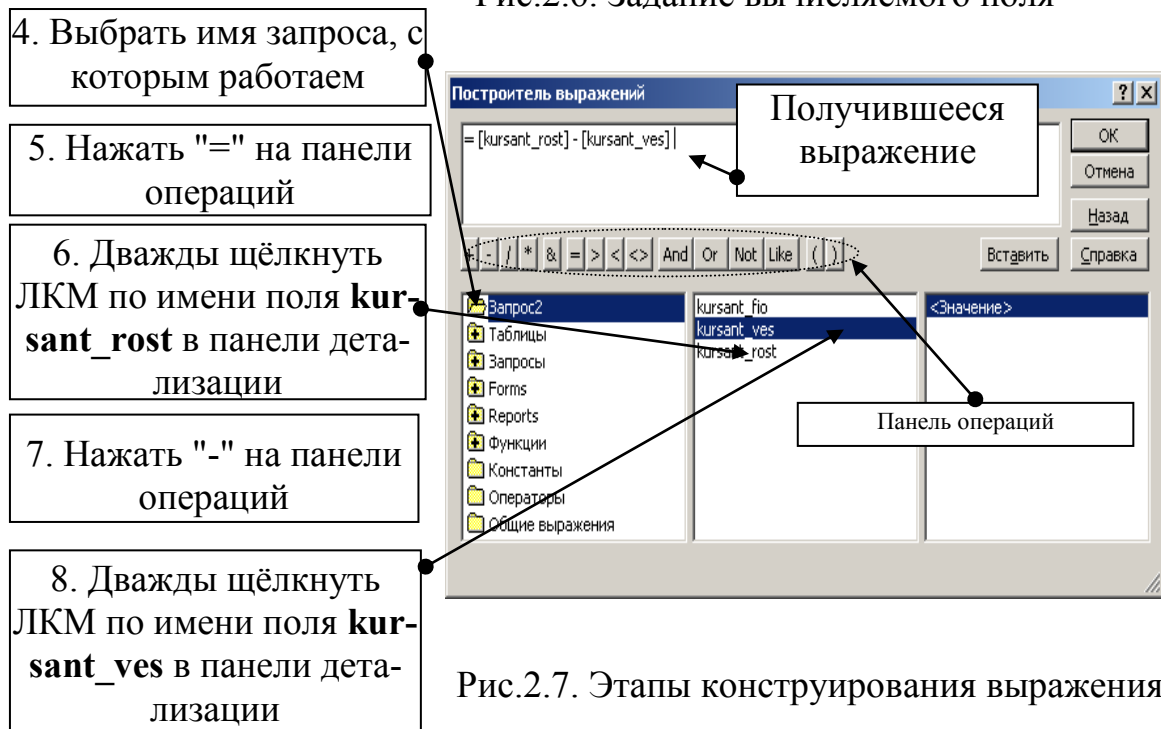


Рис.2.7. Этапы конструирования выражения

В результате в конструкторе запросов создается новое вычисляемое поле, в котором будет находиться результат вычитания веса из роста курсанта.

Так как в задаче необходимо вывести не все записи таблицы, а только те, в которых найденная разность больше 100, то в условии отбора полученного вычисляемого поля необходимо записать условие ">100". В этом случае запрос выведет только записи, соответствующие условию задачи.

Внешний вид запроса в конструкторе и результат его работы представлен на рис.2.8.

Использование функций в вычисляемых полях

Помимо простых операций с полями БД предоставляет достаточно широкий спектр функций для работы с данными.

Все функции делятся на группы, объединённые по принципу области применения и используемых аргументов. Построение функции идёт по иерархическому принципу: практически любая функция может в качестве аргументов использовать значения, возвращаемые вложенными в неё другими функциями, в качестве аргументов которых, в свою очередь, могут использоваться значения, возвращаемые вложенными уже в них функциями и т.д.

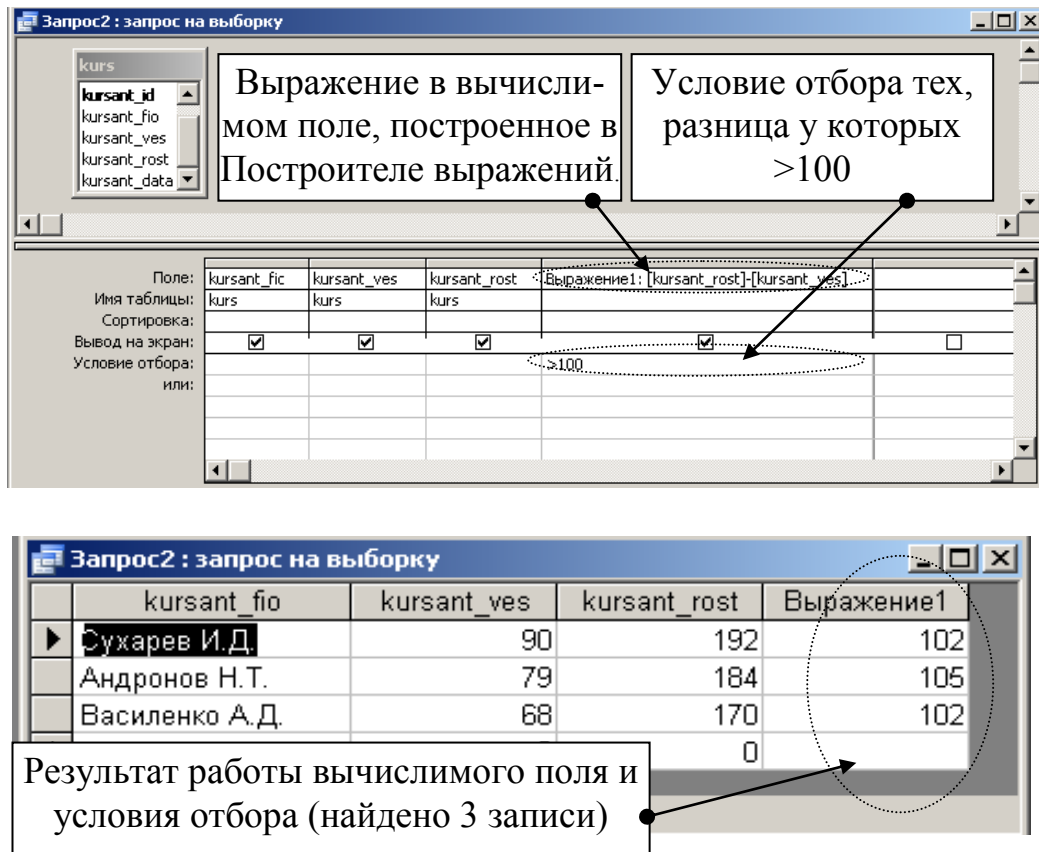


Рис.2.8. Вид запроса в Конструкторе и результаты его работы

Например, если требуется вычислить значение выражения

$$y = \ln(\sin(x)),$$

то очевидно, что здесь имеет место функция $\ln(a)$, где $a=\sin(x)$. Здесь $\sin(x)$ – вложенная функция, а $\ln(a)$ – внешняя по отношению к ней функция.

Подобный принцип "вложенности" используется и при построении сложных формул для создания вычисляемых полей в запросе.

Структура стандартных функций в Построителе выражений представлена на рис.2.9.

Пример.

Используя стандартные функции Построителя выражений и данные, находящиеся в таблице **kurs**, создать запрос, вычисляемые поля которого содержали бы название месяца и дня недели даты рождения курсанта.

Для решения поставленной задачи необходимо определить какие функции будут использоваться для извлечения из даты рождения названия месяца и дня недели, соответствующих этой дате.

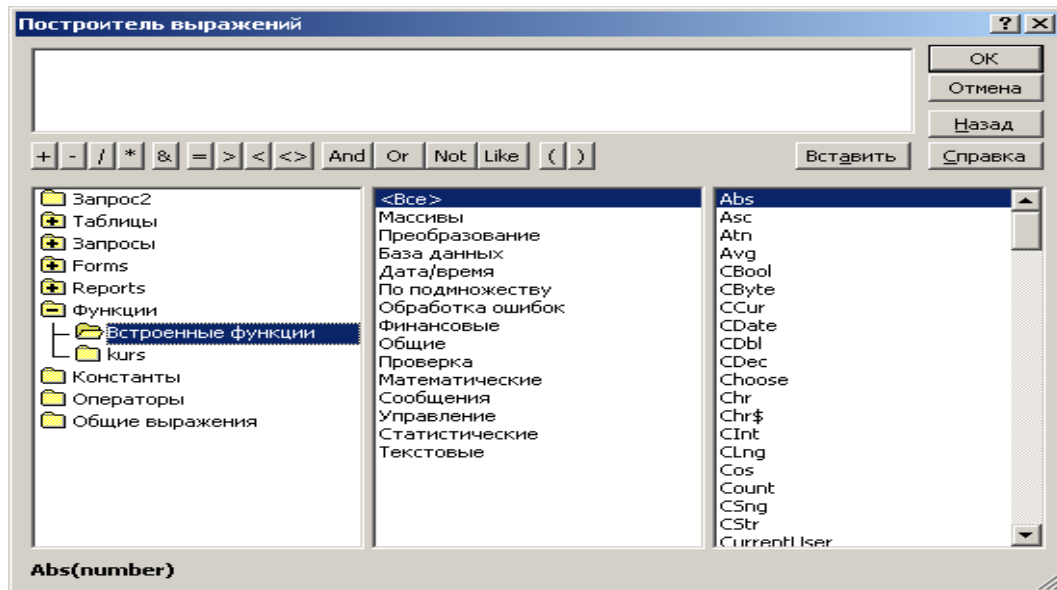


Рис.2.9. Структура встроенных функций Построителя выражений

Для извлечения названия месяца понадобятся две функции из группы функций *Дата/время*:

1. **Month(аргумент)** – возвращает номер месяца конкретной даты, которая выступает в качестве аргумента этой функции. В нашем случае аргументом будет выступать поле *kursant_data* таблицы **kurs**;

2. **MonthName(аргумент)** – возвращает название месяца. В качестве аргумента выступает номер месяца.

Итого, полный вид функции, вычисляющей название месяца по любой известной дате: **Month(MonthName(аргумент))**, где аргумент – поле *kursant_data* таблицы **kurs**.

Для извлечения названия дня недели понадобятся две функции из группы функций *Дата/время*:

1. **Weekday(аргумент)** – возвращает номер дня недели конкретной даты, которая выступает в качестве аргумента этой функции. В нашем случае аргументом будет выступать поле *kursant_data* таблицы **kurs**;

2. **WeekdayName (аргумент)** – возвращает название дня недели. В качестве аргумента выступает номер дня недели.

Полный формат функции **WeekdayName** предполагает определение пользователем номера дня недели, с которого начинается неделя (установим - 0).

Итого, полный вид функции, вычисляющей название дня недели по любой известной дате: **WeekdayName(Weekday (аргумент))**, где аргумент – поле *kursant_data* таблицы **kurs**.

Внешний вид запроса со структурой определённых выше вычисляемых полей и результат его работы представлен на рис.2.10.

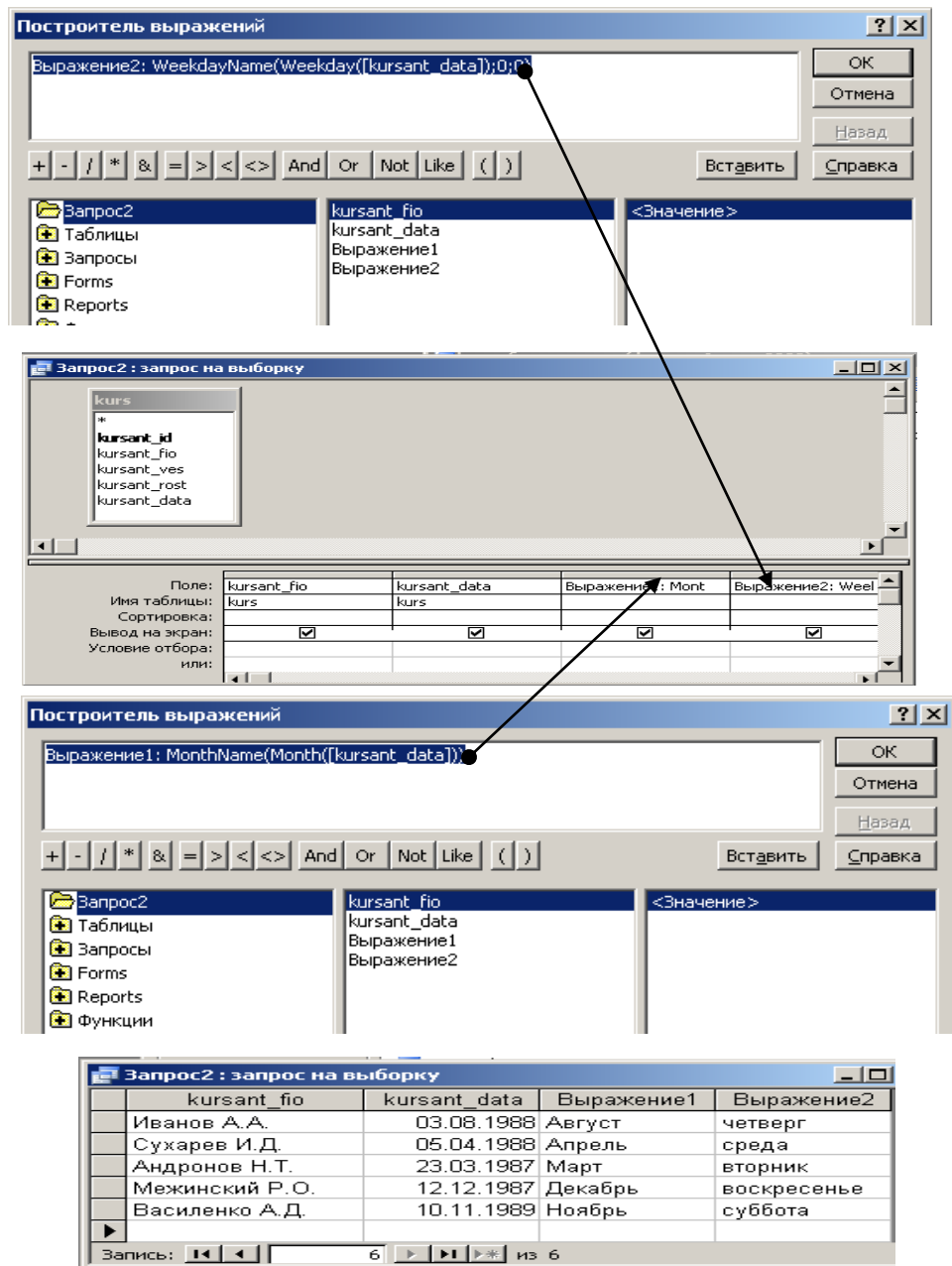


Рис.2.10. Структура запроса с месяцем и днем недели и его результаты

Задание 2.2. Групповые операции в запросах

Постройте запросы, осуществляющие выборку данных. Запишите бланки выполненных запросов:

301. Посчитайте количество магазинов в каждом городе.
302. Посчитайте количество сотрудников в каждом городе.
303. Покажите максимальную стоимость аренды в каждом городе.
304. Покажите минимальную стоимость аренды в каждом городе.
305. Покажите количество товаров и число наименований в каждой группе на складе.

311. Выведите количество наименований товаров в каждой группе товаров.

Групповые операции в запросах

Помимо обработки данных, связанных с получением сведений непосредственно находящихся в полях таблиц БД часто необходимо получать некоторые обобщённые данные, такие как сумма значений, среднее арифметическое, сортировать данные по какому-либо полю, выводить определённое количество записей на экран и другие.

Для решения этих задач используются **Групповые операции** в запросах. При нажатии этой кнопки в конструкторе запроса появляется строка групповых операций (рис.2.11).

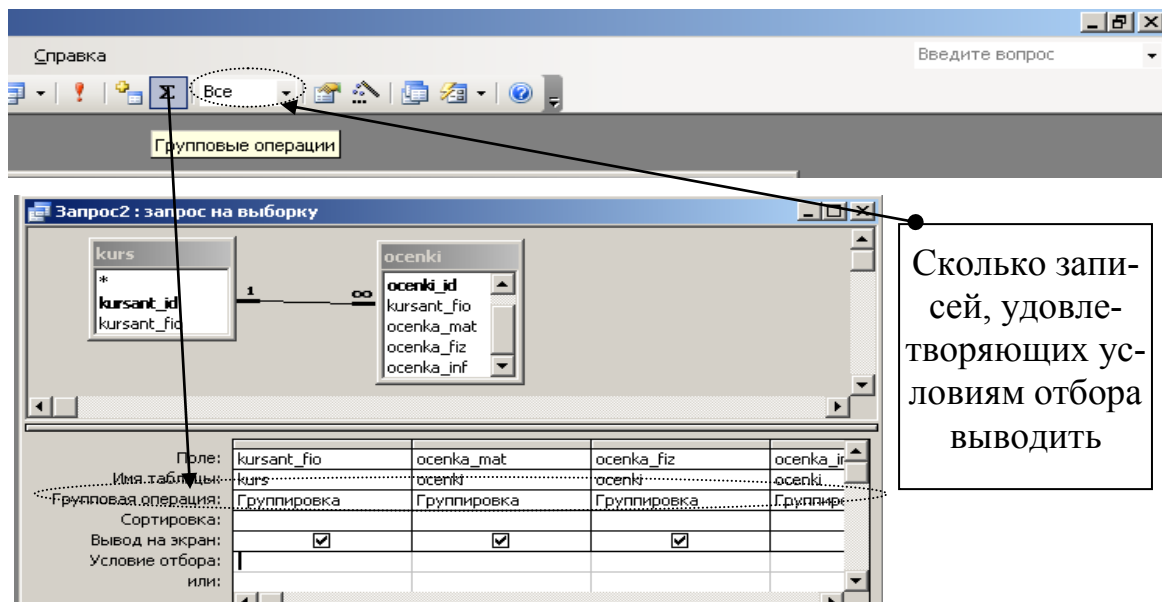


Рис.2.11. Включение групповых операций в запросе

Введение групповых операций обеспечивает объединение всех записей, имеющих в поле одинаковые данные, в одну единую группу, с которой допускается проводить целый ряд вычислений. Следует иметь в виду, что при использовании нескольких полей в запросе, группы, выделенные по признакам первого слева поля, будут разделены на подгруппы по признакам равенства следующего поля в запросе и т.д. Поэтому имена используемых полей и их количество в запросах с групповыми операциями должны быть строго обоснованы.

Для каждого поля могут быть назначены соответствующие групповые операции (функции) из выпадающего списка. Они выполняются для записей каждой группы и носят характер итоговых функций (обобщающих) для этой группы. Список таких функций приведен в таблице.

Функции строки *Групповая операция*

Функция	Назначение
SUM	Вычисляет сумму значений, содержащихся в группе этого поле
AVG	Вычисляет среднее арифметическое для всех значений группы
COUNT	Определяет число значений поля, не считая пустых (Null) значений
MIN	Находит наименьшее значение в группе
MAX	Находит наибольшее значение в группе
StDev	Определяет среднеквадратичное отклонение от среднего значения
VAR	Вычисляет дисперсию значений группы
FIRST	Выводит первое значение в группе
LAST	Выводит последнее значение в группе

Пример: Требуется найти средний балл, с которым группа курсантов сдала физику, математику, информатику. Так как необходимо найти обобщённые данные то поле, в котором отображается фамилия включать в запрос не нужно. Внешний вид запроса в конструкторе и результат его выполнения представлен на рис.2.12.

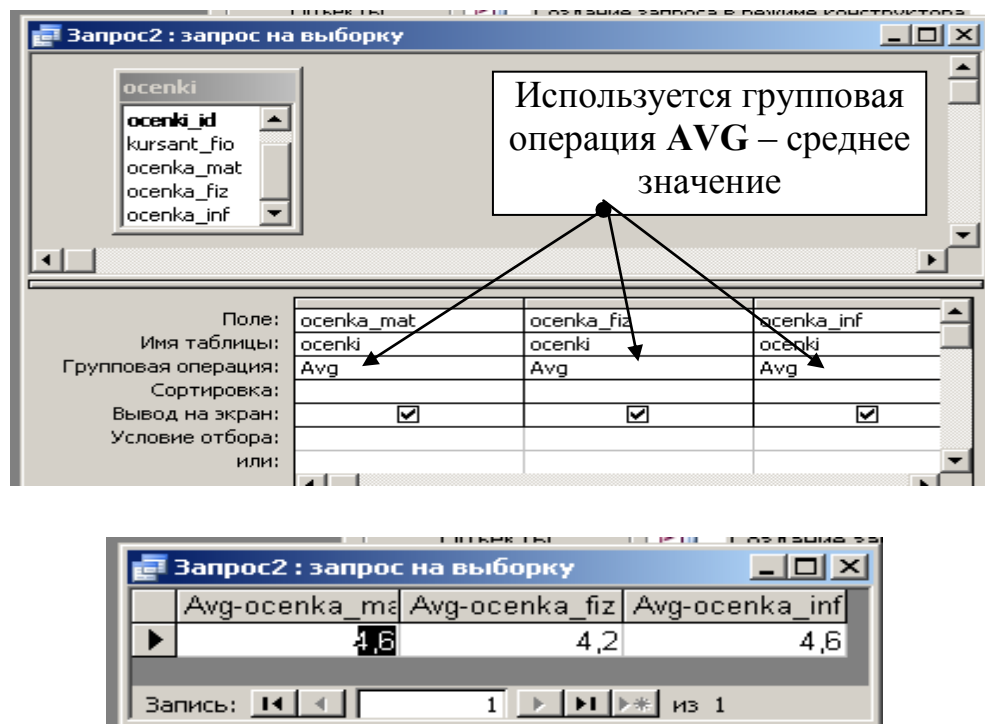


Рис.2.12. Вычисление среднего балла по всему списку

Задание 2.3. Запросы к связанным таблицам БД

Постройте запросы, осуществляющие выборку данных. Запишите бланки выполненных запросов:

- 501. Величина всех расходов на содержание магазинов.
- 502. Возможный доход от продажи товаров в магазине.
- 503. Баланс доходов и расходов.
- 504. Все магазины компьютерного профиля.
- 505. Кто из сотрудников ездит на работу в другой город?
- 506. Сколько магазинов продают фототовары?
- 507. Где работают сотрудники Горин и Конкина?
- 509. Какова ожидаемая прибыль от продажи в каждом магазине?
- 510. Сколько товаров каждого наименования осталось на складе?
- 511. Распечатать прайс-лист магазинов г.Шахты.
- 512. Расходы на каждый магазин.
- 514. Количество товаров во всех магазинах.
- 515. Список сотрудников, работающих в Новочеркасске.
- 516. Список товаров во всех компьютерных магазинах.

Необходимость связывания

До сих пор рассматривалась БД, в которую входила только одна таблица. В действительности, такие базы данных практически никогда не встречаются.

Как правило, в БД входит несколько таблиц, которые связаны друг с другом посредством ключевых полей. Связывание позволяет с одной стороны обеспечить непротиворечивость данных, а с другой – целостность БД. В качестве примера, поясняющего сказанное, рассмотрим две таблицы: *kurs* и *ocenki* (рис.2.13).

<i>kurs</i>		<i>ocenki</i>			
kursant_id	kursant_fio	kursant_id	oc_fiz	oc_mat	oc_inf
1012	Петров И.Т.				
1013	Королёв С.П.				

Рис.2.13. Исходные таблицы для связывания

Анализ таблиц показывает, что для работы с этими двумя таблицами целесообразно решить три задачи:

1. Заполнение поля *kursant_id* таблицы *ocenki* должно происходить с участием поля *kursant_id* таблицы *kurs*, иначе рано или поздно пользователь может ввести в поле *kursant_id* таблицы *ocenki* фамилию несуществующего

курсанта, либо ошибиться при вводе. В этом случае БД расценит эту запись как появление нового курсанта.

2. При удалении курсанта из таблицы *kurs* данные о нём должны автоматически удалиться из таблицы *ocenki*.

3. При изменении данных в поле *kursant_id* таблицы *kurs* данные в поле *kursant_id* таблицы *ocenki* должны адекватно измениться, иначе пользователь будет вынужден, например, при изменении фамилии курсанта в таблице *kurs* изменить вручную данные в поле *kursant_id* таблицы *ocenki* во всех записях, касающихся этого курсанта.

Таким образом, решив поставленные задачи, мы подойдём вплотную к созданию так называемой реляционной базы данных. В реляционной базе данных предполагается, что таблицы делятся на две категории: ведущие (master) и детализирующие (detailed). Это означает, что одной записи в ведущей таблице может соответствовать одна, или множество записей в детализирующей таблице. В нашем примере каждому курсанту таблицы *kurs* может соответствовать много записей об оценках (полученных в различных семестрах) таблицы *ocenki*. Т.е. уникальный номер курсанта в поле *kursant_id* таблицы *ocenki* будет присутствовать многократно. Это соответствует отношению "один ко многим". Кроме того, решение задачи 2 означает обеспечение каскадного удаления записей, а решение задачи 3 – обеспечение каскадного обновления записей. Перечисленные выше три задачи обеспечивают целостность данных.

Пример заполненных таблиц, связанных так как описано выше, представлен на рис.2.14.

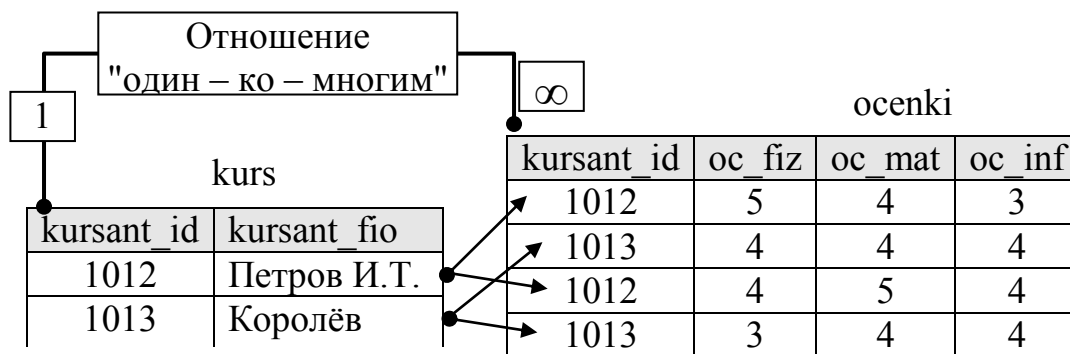
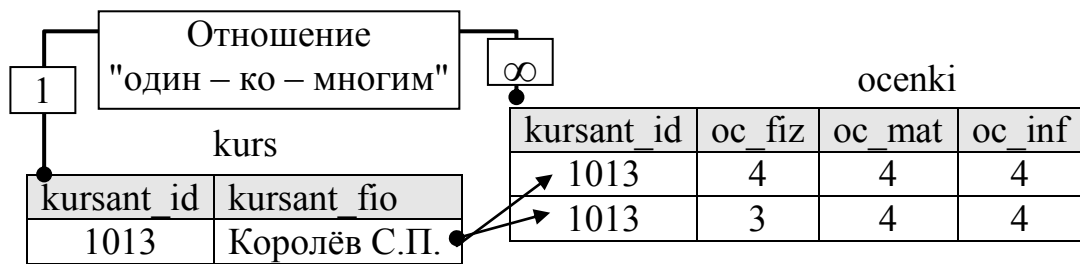


Рис.2.14. Связанные таблицы

При удалении из таблицы *kurs* записи №1 (курсант Петров И.Т.) из таблицы № 2 удалятся все записи, относящиеся к удаляемой записи. Результат удаления отображён на рис.2.15.

Рис.2.15. Удаление из таблицы **Kurs** записи с №1012

Таким образом, преимущества хранения данных в связанных таблицах следующие:

- экономия времени, т.к. одни и те же данные не вводятся многократно;
- объем базы данных сокращается, поэтому экономится место на диске и облегчается перенос БД;
- уменьшается число ошибок, т.к. ввод выполняется только один раз. В остальные таблицы вводится только короткий код - идентификатора информации для всех повторяющихся записей.
- обновление данных делается только в одной таблице.

Технология связывания таблиц базы данных

Наиболее целесообразным и понятным способом является следующий: в процессе создания таблицы **ocenki** с помощью Мастера подстановки установить связь с полем **kursant_id** таблицы **kurs**. Далее открыть схему данных и настроить вид связи.

Для связывания таблиц БД в обеих таблицах должны быть ключевые поля, данные в которых не должны повторяться (должны быть уникальны). Все таблицы БД должны быть закрыты.

Этап 1. Создание ведущей таблицы **kurs** (рис.2.16)

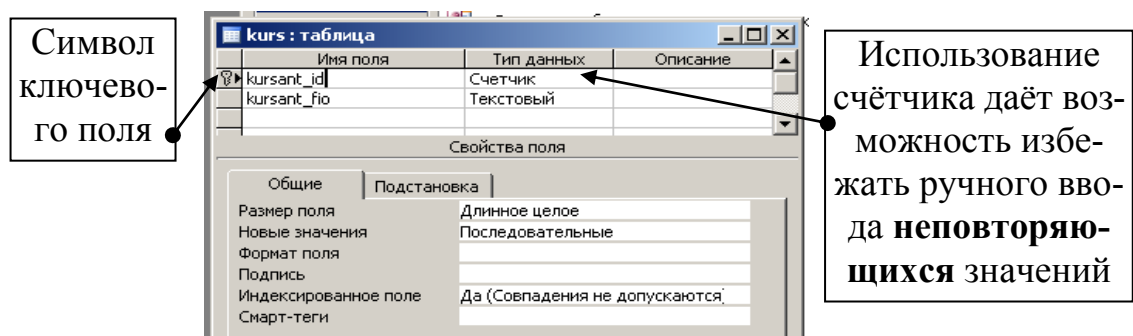


Рис.2.16. Создание ведущей таблицы

Этот этап ничем не отличается от обычной технологии создания таблицы, но ключевое поле должно быть обязательно.

Этап 2. Создание детализирующей таблицы **ocenki** (рис.2.17).

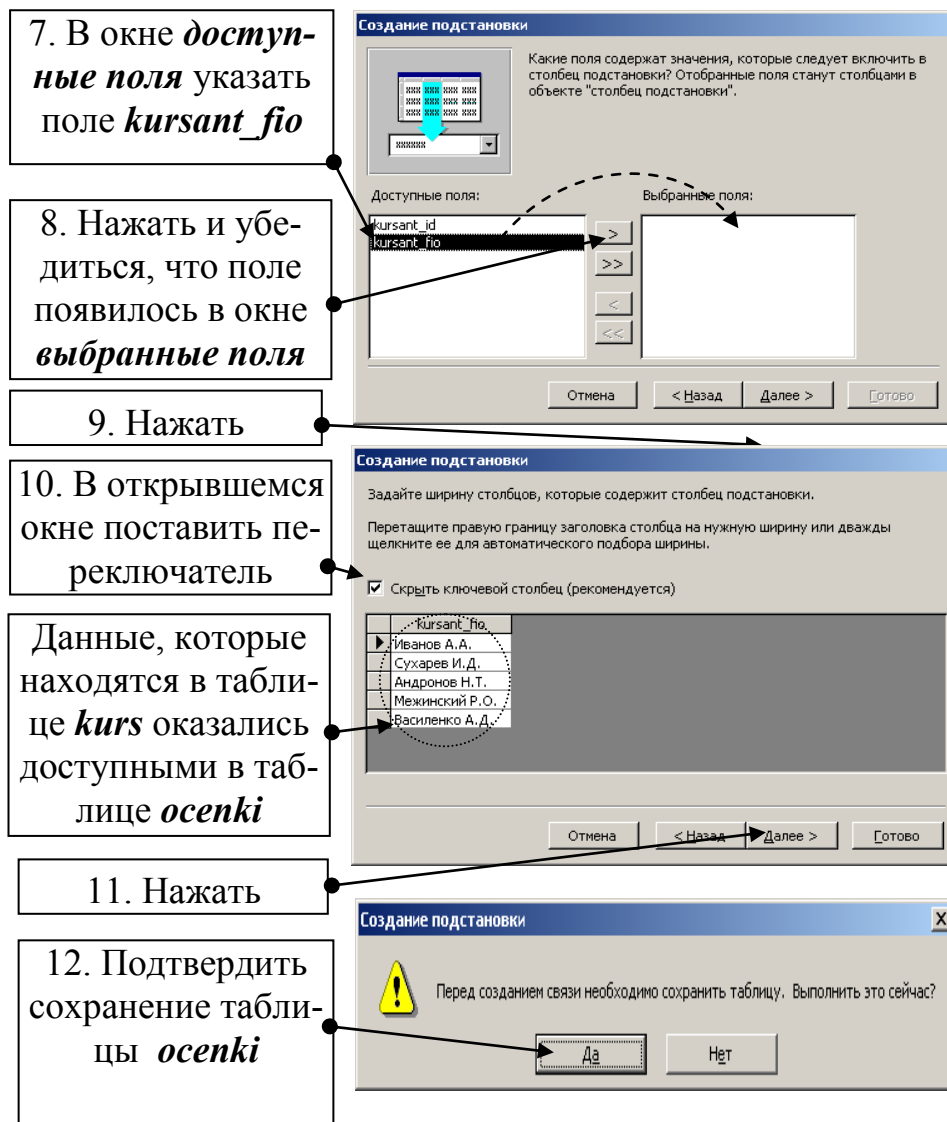


Рис.2.17 (продолжение)

Несмотря на то, что мы указываем связь к назначенному смысловому полю, БД связывает таблицу *ocenki* с полем *kursant_id* таблицы *kurs*, т.к. именно оно является ключевым, и именно оно вносится в таблицу *ocenki*. (Связь реализуется только по ключевым полям, но пользователю показывается именно то поле, которое он выбрал).

Для реализации связи и подстановки выполните шаги 1-12 рис.2.17.

В результате две таблицы окажутся связанными друг с другом, причём таблица *kurs* станет ведущей, а таблица *ocenki* – детализирующей.

Этап 3. Настройка свойств связи.

Созданная на предыдущем этапе связь между таблицами *kurs* и *ocenki* позволяет использовать данные, находящиеся в поле *kursant_fio* таблицы *kurs* в поле *kursant_id* таблицы *ocenki*.

Внешнее проявление этой связи проявляется в появлении нового элемента в таблице *ocenki* в виде выпадающего списка (рис.2.18). Естественно предположить, что так как данные для выпадающего списка берутся из таблицы

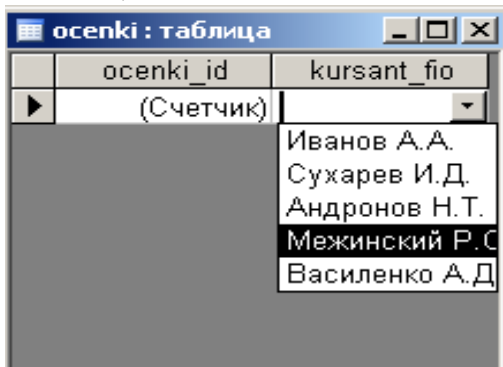


Рис.2.18. Список после подстановки

kurs, то при любых изменениях в поле *kursant fio* таблицы *kurs*, эти изменения немедленно отобразятся в данных из выпадающего списка таблицы *ocenki*.

Вместе с тем созданная связь ещё не обеспечивает целостность данных, т.е. каскадное удаление и каскадное обновление связей.

Для этого необходимо в главном меню БД выбрать пункт **Сервис – Схема данных** и в открывшемся окне изменить связь (рис.2.19).

Если всё сделано верно, то схема данных изменится и будет выглядеть так, как изображено на рис.2.20.

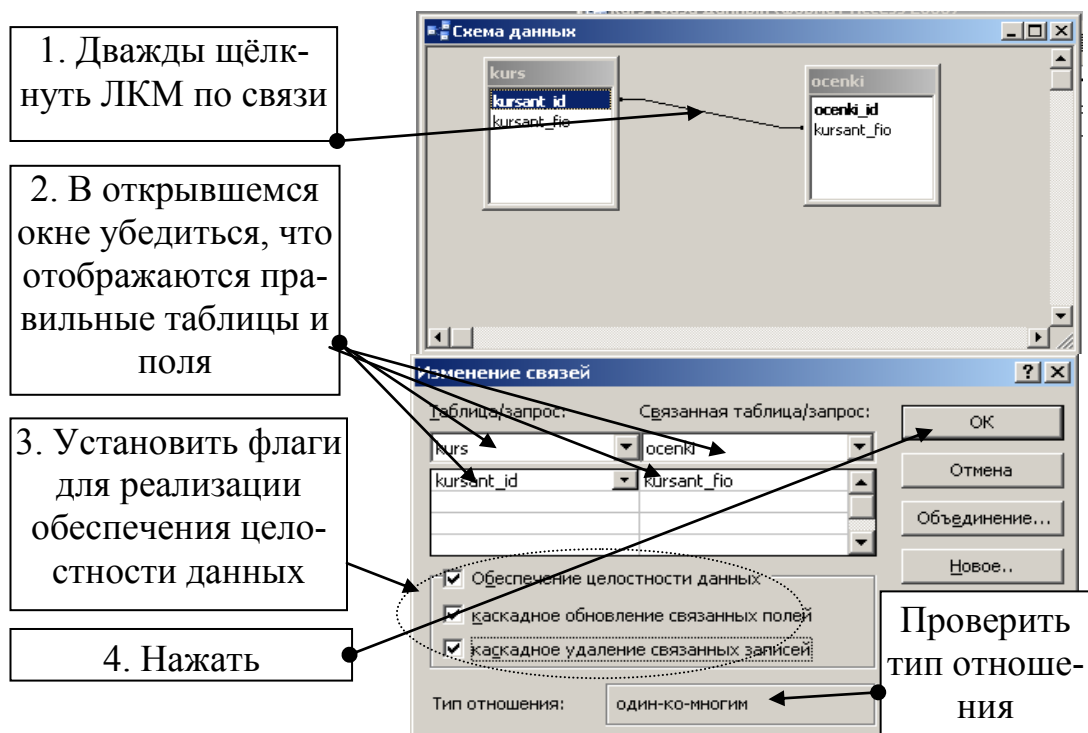


Рис.2.19. Процесс изменение связи

Результатом подобной настройки вида связей стала возможность автоматически изменять содержимое поля *kursant fio* таблицы *ocenki* при изменении содержимого поля *kursant fio* таблицы *kurs*.

Кроме того, при удалении записи о каком-либо курсанте из таблицы *kurs*, из таблицы *ocenki* автоматически удалятся все записи, касающиеся этого курсанта.

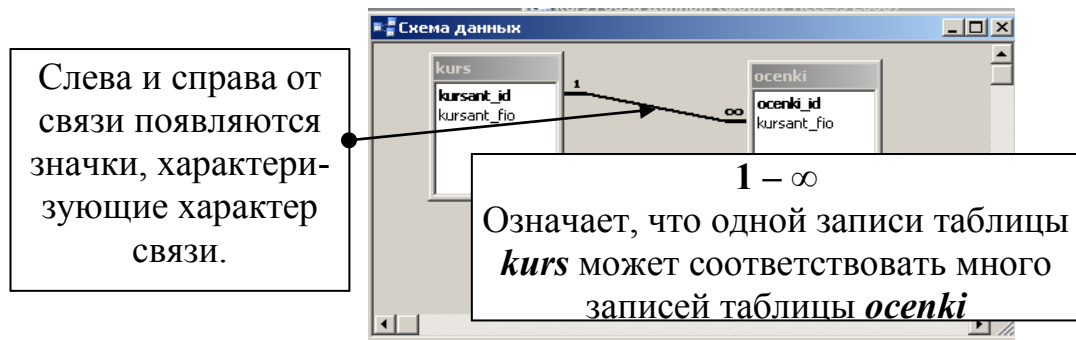


Рис.2.20. Вид связи после задания свойств связи

Создание запросов к связанным таблицам

Создание запросов к связанным таблицам несколько отличается от запроса к обычной таблице. Отличие заключается в основном в том, что в запросе используются не одна, а несколько таблиц.

Правило, в соответствии с которым определяется состав таблиц в запросе следующее.

Если в запросе требуется использовать поля, присутствующие в детализированной таблице в виде связей к ведущим таблицам, и при этом условия отбора не затрагивают эти поля, то достаточно в запрос добавить только детализированную таблицу (Пример 1).

Если в условиях отбора присутствует поле детализированной таблицы, данные в котором берутся из одной из ведущих таблиц, то эта ведущая таблица должна быть в запросе и условие отбора должно быть применено к полю ведущей таблицы (Пример 2).

Пример 1: Вывести фамилии отличников по всем предметам.

Структура запроса и результат его выполнения на рис.2.21.

Так как все оценки курсантов находятся в детализированной таблице *ocenki*, и условие применяется именно к ним, то для решения задачи достаточно использовать только эту таблицу и её поля. В поле *Kursant_fio* расположены коды из ведущей таблицы, но отображаются в этом поле фамилии курсантов за счет работы мастера подстановки.

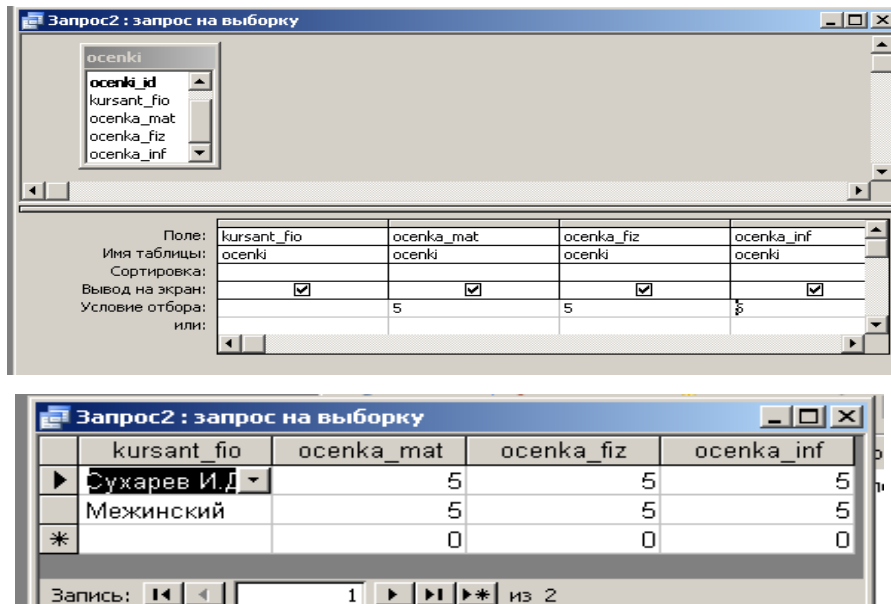


Рис.2.21. Поля для связи в отборе не участвуют

Пример 2: Вывести оценки курсанта **Василенко А.Д.** по всем предметам.

Для добавления таблицы в конструктор запросов необходимо щёлкнуть ПКМ в поле таблиц и в открывшемся окне выбрать необходимую таблицу, нажать **Добавить** (рис.2.22).

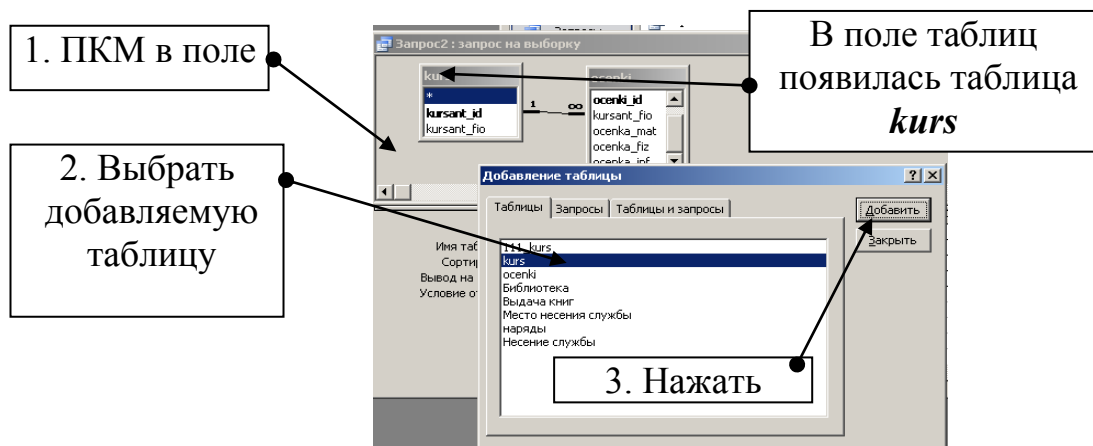


Рис.2.22. Добавление таблицы в запрос

Структура запроса и результат его выполнения показаны на рис.2.23.

В примере 2 условие отбора (фамилия) применяется к полю, которое находится в ведущей таблице **kurs**. Поэтому необходимо добавить эту таблицу в конструктор запросов, а в список полей, выводимых на экран, добавить поле **kursant_fio** из этой таблицы. Условие отбора **Василенко А.Д.** также следует применить к этому полю, а не к полю **Kursant_fio** таблицы **Ozenki**. В этом случае даже целесообразно в таблице **Ozenki** сохранить название поля **Kursant_id** как в основной таблице. Это лишний раз будет напоминать о типе данных, хранящихся в этом поле.

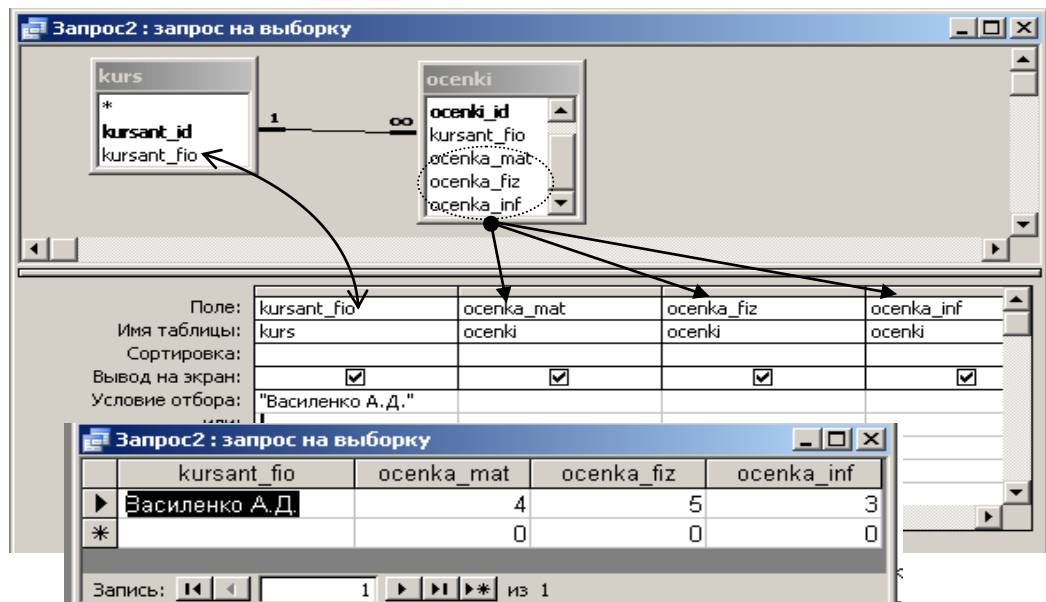


Рис.2.23. Запрос к связанным таблицам

В первом примере фамилию в результатах запроса можно увидеть только за счет подстановки: в этом поле находится код для связи таблиц и фамилии в поле фактически нет. Но для второго запроса отбор осуществляется по строке фамилии, и отбор можно осуществить только при наличии данных с фамилиями. Поэтому необходимо привлекать вторую таблицу, в которой фамилия и размещена и может выполняться сравнение строк с условием отбора для формирования результата запроса.

Вопросы для самопроверки

1. Что такое вычисляемые поля?
2. Для каких целей формируются запросы с вычисляемыми полями?
3. Какова технология создания запроса с вычисляемыми полями?
4. Какова технология создания запроса к связанным таблицам?
5. Какие преимущества дают запросы к связанным таблицам?
6. Выведите суммарные затраты на зарплату и содержание всех магазинов и суммарный объем расходов.
7. Выведите количество товаров во всех магазинах.
8. Список товаров во всех компьютерных магазинах.
9. Величина всех расходов на содержание магазинов.
10. Вычислите совокупную стоимость содержания магазинов в каждом городе.

3. РАЗРАБОТКА ФОРМ

Цель

1. Выработать первичные навыки создания и использования Системы управления базами данных (СУБД);
2. Приобрести навыки разработки и использования форм для ввода данных, а также приемов модификации форм..

Учебные вопросы

- 3.1. Создание формы с помощью мастера;
- 3.2. Создание форм с помощью конструктора;
- 3.3. Создание и удаление элементов управления

Литература

Литература аналогична той, которая указана в разделе 1.

Задание 3.1. Создание формы с помощью мастера

Задание выполняется в соответствии с выданным индивидуальным вариантом.

Особенности форм

Форма — это объект базы данных, так же как таблицы и запросы. Форма - это средство, упрощающее ввод, редактирование и отображение информации, хранящейся в таблицах базы данных. Она представляет собой настраиваемое диалоговое окно с набором *элементов управления*. Элемент управления — это визуальный объект, используемый для отображения информации, ввода и изменения данных, выполнения определенных действий или просто для украшения формы. Элементами управления являются текстовые поля для ввода и правки данных, кнопки, флажки, переключатели, списки, надписи, линии, а также рамки объектов для отображения графики и объектов OLE.

Форма сама по себе не хранит информацию, она просто обеспечивает удобный способ доступа к информации, хранящейся в одной или нескольких таблицах. Как правило, один элемент управления обеспечивает доступ к одному полю таблицы (хотя некоторые элементы управления вообще не связаны с полями). Формы по сравнению с обработкой данных в режиме таблицы обладают следующими преимуществами:

- форма позволяет в каждый момент времени наблюдать все поля отдельной записи целиком. В отличие от формы таблицы отображают сразу несколько записей, так, что приходится прокручивать окно, чтобы просмотреть содержимое всех полей;

- элементы управления на форме можно перегруппировать логичным образом, облегчающим чтение и работу с данными;
- отдельные элементы управления могут облегчить ввод и изменение некоторых данных, позволяют предотвратить ошибки при вводе, проводить вычисления, обеспечивают доступ к данным в связанных таблицах с помощью подчиненных форм;
- такие объекты как рисунки, анимации, звуки и видеоклипы, могут отображаться или воспроизводиться только в режиме формы, но не в режиме таблицы.

Способы создания форм

Создать форму можно несколькими способами, но для большинства задач лучше выбрать объект **Форма** на панели **Объекты** в окне базы данных, а затем щелкнуть на кнопке **Создать**. При этом откроется окно **Новая форма** (рис.3.1).

Из списка в этом диалоговом окне можно выбрать один из способов создания новой формы.

Конструктор. Способ используется для самостоятельного создания формы путем постепенного добавления элементов управления в режиме конструктора с последующей настройкой необходимых свойств.

Мастер форм. Способ использует Мастер форм для создания формы с помощью СУБД, но в соответствии с вашими требованиями. Этот параметр даст вам возможность выбрать определенные поля для включения в форму, которые могут принадлежать одной или нескольким таблицам или запросам.

Автоформа При выборе одного из трех параметров Автоформа, запускается мастер форм, опирающийся на таблицу или запрос по вашему выбору. Вместо того чтобы запрашивать у вас подробные указания (как делает мастер форм), этот способ быстро создаст и откроет определенный тип формы — в **столбец**, **ленточный** или **табличный** — с помощью параметров по умолчанию. Форма, созданная посредством автоформы, всегда включает элементы управления для всех полей одной таблицы или запроса.

Диаграмма. Этот способ получает информацию в режиме таблицы или запроса, и создает диаграмму с помощью Microsoft Graph. Диаграмма включается в новую форму.

Сводная таблица. Предполагается получение информации из таблицы, создание сводной таблицы, как описано для Microsoft Excel.

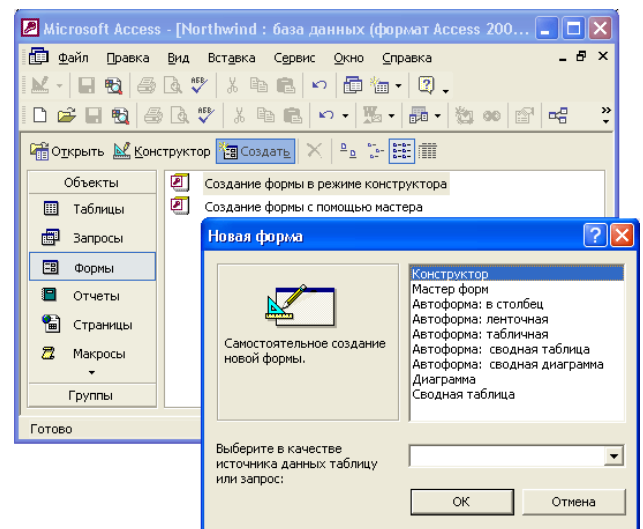


Рис.3.1. Окно **Новая форма**

В общем случае легче всего создать грубый эскиз вашей формы с помощью мастера форм или автоформы. Затем можно настраивать форму с помощью технических приемов в режиме конструктора.

Автоматическое создания форм

Это самый простой способ создания формы: автоматическое создание формы на основе таблицы или запроса. Такая форма включает все поля выбранного источника данных.

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

- автоформа, организованная "в столбец" (columnar). В такой форме поля каждой записи отображаются в виде набора элементов управления, расположенных в один или несколько столбцов. Это компактное и в этом смысле самое удачное представление для быстрого создания формы;

- табличная (datasheet). Форма будет выглядеть так же, как обычная таблица Access;

- ленточная (tabular). В такой форме поля каждой записи располагаются в отдельной строке. Это очень 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 по вашему вкусу и для которого можно определить функции обработки событий, т. е. "оживить" поле с помощью программирования.

Чтобы создать форму с помощью средства автоматического создания форм необходимо:

1. Щелкнуть по ярлыку **Формы** (forms) в окне база данных (database) и нажать кнопку **Создать** (new). Появится диалоговое окно **Новая форма** (new form).

2. В списке диалогового окна **Новая форма** выделите один из вариантов автоформы, например: **Автоформа: в столбец** (autoform: columnar).

3. В поле со списком, находящимся в нижней части диалогового окна **Новая форма**, содержатся имена всех таблиц и запросов базы данных, которые могут быть использованы в качестве источника данных для формы.

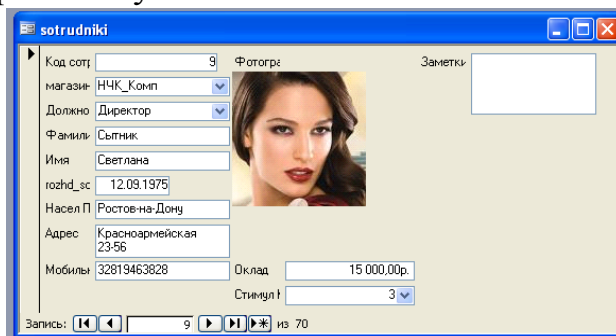


Рис.3.2. Форма в столбец

Щелкните левой кнопкой мыши по кнопке со стрелкой, чтобы раскрыть список, и выберите в нем нужный элемент. Нажмите кнопку **Ок**.

В результате будет автоматически создана и открыта форма выбранного вида (рис.3.2). Чтобы созданную форму можно было использовать в дальнейшем, ее необходимо сохранить. Для сохранения формы

выберите команду **Файл, Сохранить** (file, save) или нажмите на кнопку **Сохранить** (save) на панели инструментов. В поле **Имя формы** (form name) появившегося диалогового окна **Сохранение** (save as) введите нужное название и нажмите кнопку **Ок** (рис.3.3).



Рис.3.3. Окно сохранения форм

Создание формы с помощью мастера

Простым и быстрым способом создания формы является использование мастера форм. С помощью мастера можно создавать формы на основе одной таблицы и более сложные формы на основе нескольких таблиц и запросов, имеющие подчиненные формы. Намного проще и быстрее создавать формы с помощью мастера, а затем усовершенствовать их в режиме конструктора.

Мастер форм разбивает процесс создания формы на несколько этапов. На каждом этапе требуется установить определенные параметры в одном из диалоговых окон мастера, каждое из которых определяет один шаг создания формы. Если на каком-нибудь шаге была допущена ошибка или возникла необходимость изменения каких-либо установленных параметров, для возвращения к предыдущему шагу нажмите кнопку **Назад** (back). Кроме того, в любой момент можно нажать кнопку **Отмена** (cancel) для отказа от создания формы и возвращения к окну базы данных.

Процедура создания формы для ввода и редактирования информации для источника в виде типовой таблицы предполагает следующие шаги:

Щелкните по ярлыку **Формы** (forms) в окне базы данных.

1. Нажмите кнопку **Создать** (new) на панели инструментов окна базы данных. В списке вариантов в появившемся диалоговом окне **Новая форма** (new form) (см. рис.3.1) выделите элемент **Мастер форм** (form wizard) и нажмите кнопку **Ок**. То же самое можно сделать, дважды щелкнув по ярлыку **Создание формы с помощью мастера** (create form by using wizard), находящемуся перед списком существующих форм в базе данных.

2. Появится первое диалоговое окно мастера форм (рис.3.4). В поле со списком **Таблицы и запросы** (tables/queries), как и в раскрывающемся списке в окне **Новая форма**, будут отображены имена всех таблиц и запросов базы данных, которые могут использоваться в качестве источника данных для формы. Раскройте этот список и выберите имя таблицы или запроса. В нашем примере, например, это таблица **Товар**.

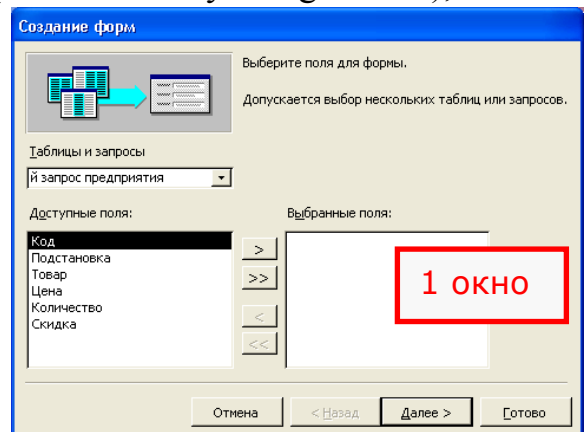


Рис.3.4. 1-й этап работы Мастера форм

3. В списке *Доступные поля* (available fields) этого диалогового окна (см. рис.3.4) отображаются все поля выбранной таблицы или запроса. Добавление полей в форму позволит просматривать и редактировать данные выбранной таблицы. Чтобы добавить в создаваемую форму только некоторые поля, выделите каждое из этих полей и нажмите кнопку ">". Выделенное поле будет перемещено из списка *Доступные поля* в список *Выбранные поля* (selected fields). Чтобы добавить в создаваемую форму сразу все поля из выбранной таблицы или запроса, нажмите кнопку ">>". Нажмите кнопку *Далее* (next) для отображения второго диалогового окна мастера форм.

4. Во втором диалоговом окне мастера можно определить вид формы. Чтобы задать внешний вид формы, выберите один из переключателей: *В один столбец* (columnar), *Ленточный* (tabular), *Табличный* (datasheet), *Выровненный* (justified), *Сводная таблица* (pivottable) или *Сводная диаграмма* (pivotchart). Нажмите кнопку *Далее* (рис.3.5).

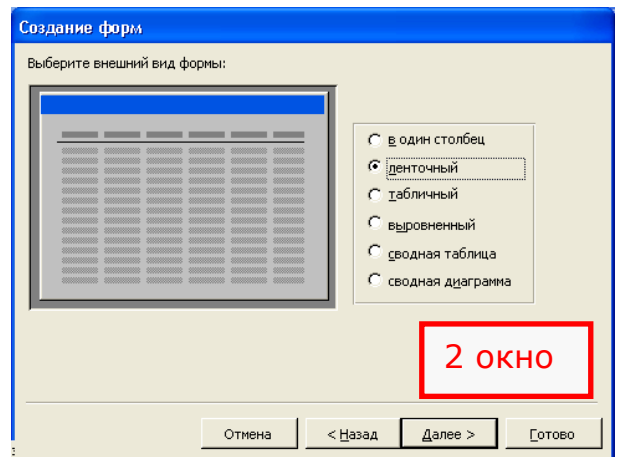


Рис.3.5. 2-й этап работы Мастера форм

5. Третье диалоговое окно мастера форм (рис.3.6) предназначено для выбора стиля оформления новой формы. Мастер предлагает несколько стандартных стилей оформления. Можно определить собственные стили оформления форм с помощью диалогового окна Автоформат (autoformat). Тогда эти стили будут отображаться вместе со стандартными в этом диалоговом окне мастера форм. Выберите один из предлагаемых стилей и нажмите кнопку *Далее*.

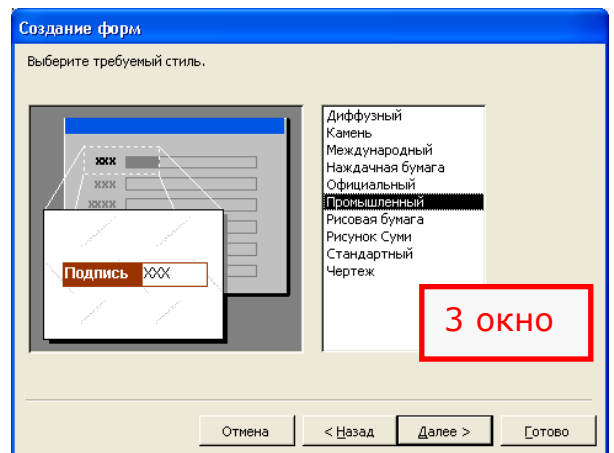


Рис.3.6. 3-й этап работы Мастера форм

6. В последнем четвертом диалоговом окне мастера форм (рис.3.7) требуется указать название формы. В поле ввода этого диалогового окна введите название формы: *Товар*. Чтобы отобразить созданную мастером форму в режиме формы, выберите переключатель *Открыть форму для просмотра и ввода данных* (open the form to view or enter information). А если после автоматического создания формы с помощью мастера требуется внести собственные изменения, выберите переключатель *Изменить макет формы* (modify the form's design), тогда созданная форма будет открыта в режиме

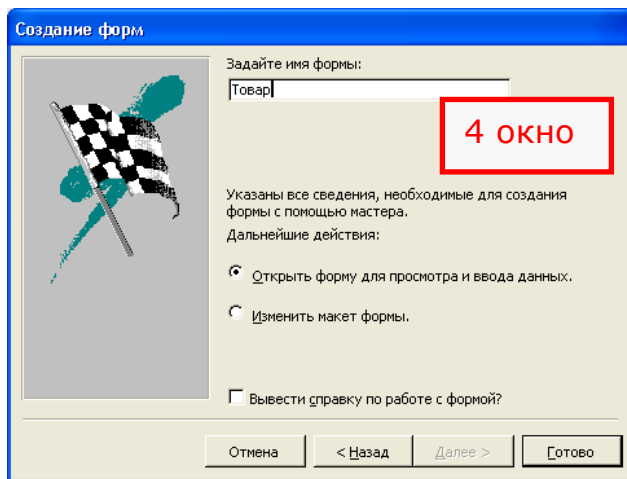


Рис.3.7. 4-й этап работы Мастера форм

В результате мастером будет создана форма в соответствии с выбранными параметрами и сохранена с указанным именем, затем эта форма будет открыта в заданном режиме. На рис.3.8. представлена созданная форма, открытая в режиме формы.

Созданная с помощью мастера форма "Товар" готова к использованию. Однако можно улучшить ее внешний вид, изменив некоторые параметры оформления. Рационально это делать в режиме конструктора.

конструктора. Если необходима справка о работе с созданной формой, установите флажок **Вывести справку по работе с формой?** (display help on working with the form?), после чего нажмите кнопку **Готово** (finish).

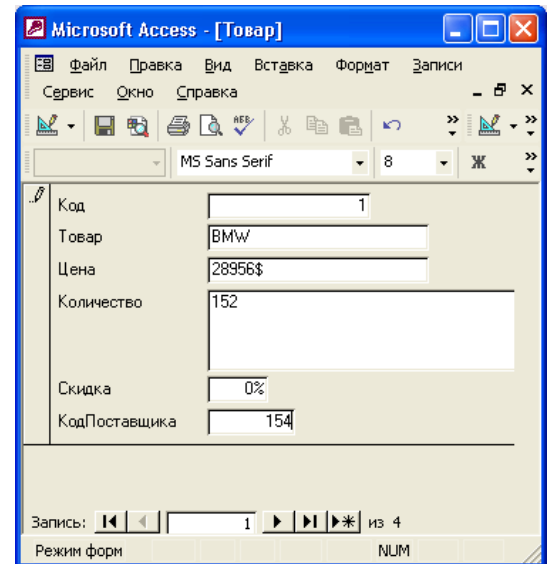
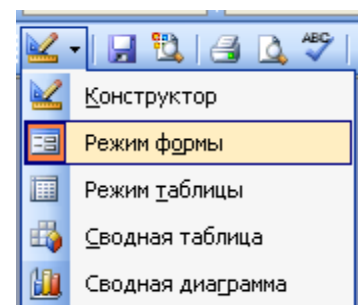


Рис.3.8. Форма ленточная

Режимы работы с формами

Работа с формами может происходить в пяти режимах:

- в режиме формы,
- в режиме таблицы,
- в режиме конструктора,
- в режиме сводной таблицы и
- в режиме сводной диаграммы.



Выбрать режим работы можно либо с помощью кнопки **Вид** (view) на панели инструментов текущего режима работы с формой, либо с помощью соответствующей команды меню **Вид** (view).

Режим формы является "рабочим" для пользователя базы данных. В этом режиме осуществляются просмотр и редактирование записей, удаление записей или добавление новых. Таким способом пользователь работает с

таблицей или запросом, являющимися источниками данных для формы. Чаще всего пользователи приложений Access работают с формами именно в этом режиме (см. рис.3.2). В этом же режиме по умолчанию открывается форма из окна базы данных.

Режим таблицы для пользователя может оказаться удобным. В этом режиме, как и в режиме формы, можно просматривать и редактировать, добавлять и удалять записи в таблице или запросе, являющемся источником данных для формы. Однако в этом режиме не применяются параметры форматирования элементов управления.

Режим конструктора позволяет настроить различные свойства формы, изменить ее внешний вид, структуру и функциональность. Этим режимом обычно пользуются разработчики.

На рис.3.9 показан внешний вид формы в режиме конструктора и таблицы. Они соответствуют форме, показанной на рис.3.2.

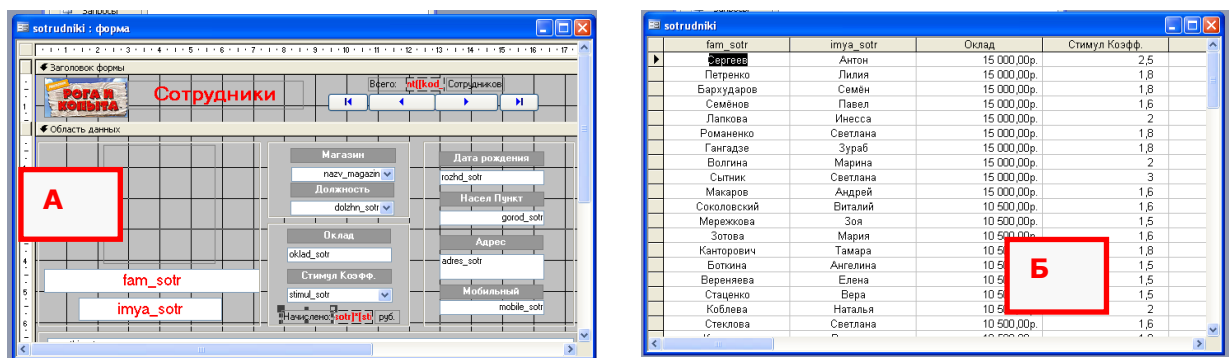


Рис.3.9. Форма в режиме конструктора (А) и таблицы (Б)

Режим сводной таблицы

Режим сводной диаграммы

Работа с записями

Основной целью простых форм является добавление новых записей, удаление записей или редактирование данных в одной или более таблицах, являющихся источником данных для формы.

Добавление, удаление и редактирование записей с помощью формы происходит при работе с ней в режиме формы или в режиме таблицы. Работа с формой в режиме таблицы ничем не отличается от работы с обычными таблицами Access. Однако режим формы более функционален по сравнению с режимом таблицы. Простота и удобство при использовании формы существенно зависит от степени учета конкретных потребностей пользователя.

Перемещение по записям

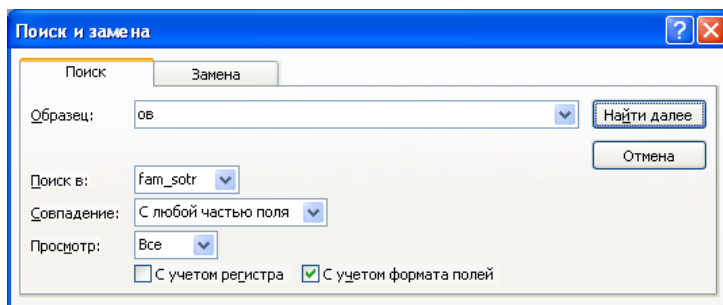
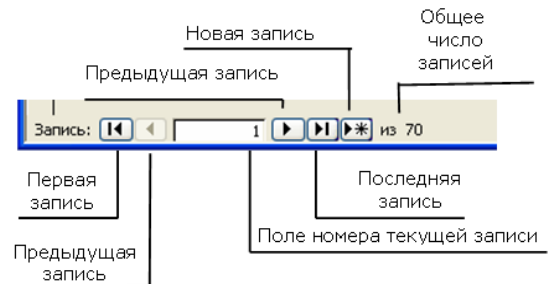
Основная часть элементов управления на стандартной форме непосредственно работает с полями, принадлежащими источнику записей (хотя элементы могут использоваться и для других целей). В левом нижнем углу формы находится небольшая панель инструментов, предназначенная для перемещения по существующим записям и добавления новых. Назначение кнопок показано на рисунке.

Для выполнения тех же операций с помощью клавиатуры возможно использовать следующие комбинации клавиш:

- **PageDown** – перемещение на одну запись вперед;
- **PageUp** – перемещение на одну запись назад;
- **Ctrl+Home** – переход на начало первой записи
- **Ctrl+End** – переход на конец последней записи.

Для просмотра конкретной записи, можно щелкнуть на **Поле номера текущей записи**, ввести ее номер и нажать **Enter**.

Чтобы найти конкретную запись, содержащую определенный текст в одном из полей, необходимо выполнить команду **Найти** из меню **Правка**, либо нажать кнопку Найти на панели инструментов, либо **Ctrl+F**. В окне **Поиск и замена** следует уточнить поле, в котором осуществляется поиск и вариант в окне **Совпадение**. Поиск осуществляется до



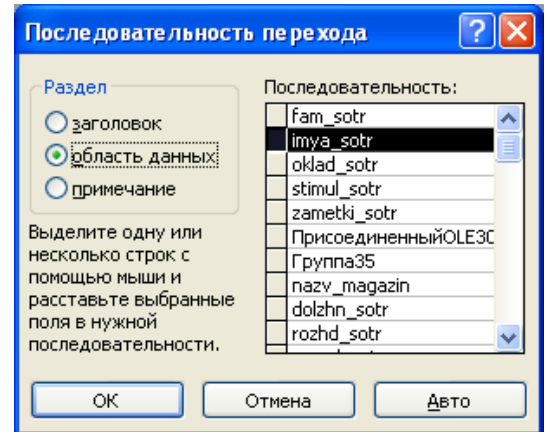
обнаружения первого совпадения. Все знаки, замещающие символы и их сочетания (звездочка и вопросительный знак) в этом режиме также допустимы. Для перехода к следующему управляющему элементу формы надо нажать клавише **Tab** или **Enter**. Порядок переходов можно настроить желаемым образом. Чтобы вернуться к предыдущему элементу управления, щелкните по нему или нажмите комбинацию клавиш **Shift+Tab**. С помощью мыши можно сразу перейти к любому элементу управления в форме.

Задание последовательности перехода для элементов управления в форме

С помощью клавиатуры переход от одного элемента управления к другому осуществляется только в заданной последовательности. По умолчанию

последовательность перехода определяется порядком создания элементов управления в режиме конструктора. Изменить эту последовательность позволяет диалоговое окно **Последовательность перехода** (tab order).

Чтобы отобразить это диалоговое окно в режиме конструктора формы, выберите команду **Вид, Последовательность перехода** (view, tab order) или щелкните правой кнопкой мыши по форме и выберите в контекстном меню команду **Последовательность перехода** (tab order). В этом диалоговом окне отображается список всех элементов управления в форме. Чтобы изменить последовательность перехода, нужно изменить порядок следования элементов в списке, перетаскивая элементы списка с помощью мыши. Перетаскивать нужно, щелкнув по серому квадратику слева от названия поля и удерживая нажатой левую кнопку мыши.



Добавление записи

В формах, так же как и в таблицах, предусмотрена пустая запись, которая предназначена для добавления новых записей в таблицу. Эта пустая запись, как и в таблице, отображается в форме после всех заполненных записей. К этой записи можно перейти, пролистав в форме все записи с помощью кнопок перехода по записям или с помощью специальной кнопки перехода на новую запись. После ввода данных в эту запись и ее сохранения данные автоматически попадают в таблицу.

При добавлении новой записи в таблицу с помощью формы необходимо, переместившись на пустую запись, заполнить поля этой формы в соответствии с правилами, определенными разработчиком формы. Для сохранения введенных значений нажмите комбинацию клавиш **Shift+Enter**.

Чтобы можно было добавлять новые записи в таблицу в режиме формы, свойство **Разрешить добавление** (allow additions) должно иметь значение **Да** (yes). Значение этого свойства можно изменить, используя окно свойств формы в режиме конструктора. Вызвать окно после открытия формы в режиме конструктора можно нажав кнопку **Свойства** на панели инструментов или выполнив нажатие правой кнопкой и выбрав **Свойства** в контекстном меню. Свойство устанавливается на вкладке **Данные** (рис.310).

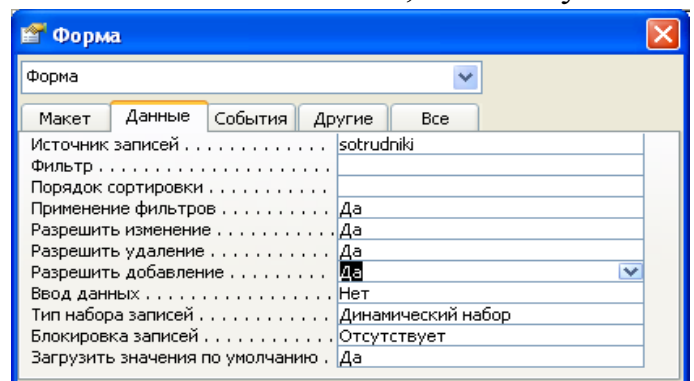


Рис.3.10. Окно свойств формы

Чтобы добавить новую за-

пись в таблицу с помощью формы:

1. Откройте форму в режиме формы. Редактирование полей для формы разрешено до тех пор, пока это не будет запрещено в свойстве формы **Разрешить изменение** (allow edits);



2. Перейдите к пустой записи, предназначенной для ввода новых данных. Для этого нажмите кнопку **Новая запись** (new record) на панели инструментов **Режим формы** (form view) либо маленькую кнопку **Новая запись** (new record), расположенную в нижней части окна формы. Чтобы отменить создание новой записи после нажатия кнопки **Новая запись** на панели инструментов, просто нажмите кнопку перехода к предыдущей записи.



3. При переходе на новую запись все текстовые поля и другие элементы управления в форме станут пустыми, кроме тех, значение которых определено по умолчанию;

4. В поля формы введите требуемые значения, используя стандартные методы редактирования. В текстовые поля введите нужные значения, а в раскрывающихся списках или полях со списками с помощью мыши выберите необходимые элементы;

5. После завершения ввода данных Access сохранит их в памяти, но не добавит новую запись в таблицу. Для добавления записи в таблицу нажмите комбинацию клавиш **Shift+Enter** или выберите команду **Записи, Сохранить запись** (records, save record).

Для запрета добавления новых записей необходимо установить значение **Нет** в свойство **Разрешить добавление**. Теперь кнопка **Новая запись** (new records), как на панели инструментов **Режим формы** (form view), так и внизу окна формы, будет заблокирована, и добавление новых записей в таблицу с помощью формы будет невозможно.

Копирование записи

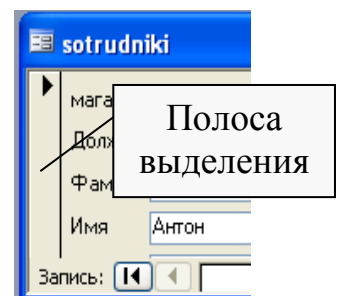
Повторяющиеся записи удобно вводить, используя операцию копирования. Похожую запись копируют, а затем производят необходимые изменения. Такая технология значительно эффективней прямого набора новой записи.

Для копирования выполнить следующие действия:

1. Отобразить в окне формы источник информации, и выделить запись, щелкнув в левой части окна формы. Эта полоса эквивалентна заголовку строки в режиме таблицы и выполняет те же функции.


2. Выполнить команды **Правка, Вставить** или **Ctrl+C**.

3. Отобразить в форме запись, куда будет производиться копирование (уже существующую или новую запись).



4. Выполнить команды **Правка**, **Вставить** или **Ctrl+V**. Значение каждого элемента управления исходной записи будет скопировано в соответствующий элемент записи-приемника, за исключением ключевых элементов.

Изменение записи

Редактирование существующих записей аналогично добавлению новых записей. Однако прежде чем приступить к редактированию записи, ее нужно найти. Для этих целей можно воспользоваться кнопками перехода по записям  (кнопками навигации), расположенными в нижней части окна формы. При неизвестном номере записи воспользуйтесь кнопкой **Найти**. Для упорядочения записей по конкретному полю, необходимо выделить поле, а затем нажать кнопку сортировки по возрастанию или убыванию на панели инструментов.

Чтобы **запретить редактирование записи** таблицы с помощью формы в окне свойств формы раскройте вкладку **Данные** (data) и в списке свойства **Разрешить изменение** (allow edits) выберите элемент **Нет** (no). Теперь редактирование записей, содержащихся в таблице, с помощью формы будет невозможно.

Чтобы **запретить редактирование какого-либо конкретного поля** записи:

- откройте форму в режиме конструктора и выделите нужное поле;
- на панели инструментов **Конструктор форм** (form design) нажмите кнопку **Свойства** (properties). То же самое получится при выборе из контекстного меню принадлежности правой кнопки на выделенном поле;
- на вкладке **Данные** (data) в раскрывающемся списке свойства **Блокировка** (locked) выберите элемент **Да** (yes). Сохраните форму.

Чтобы какое-либо поле формы имело заранее заданное значение, которое нельзя было бы изменить, укажите для этого поля формы нужное значение в свойстве **Значение по умолчанию** (default value) и установите значение свойства **Блокировка** (locked) равным **Да** (yes).

Отмена изменения записи

Access не внесет в таблицу изменения до тех пор, пока не будет выделена другая запись или не будет сохранена отредактированная новая запись, например с помощью команды **Записи**, **Сохранить запись** (records, save records) или комбинации клавиш **Shift+Enter**. Последнее сохранение записи можно отменить, если сразу же после сохранения записи в таблице нажать кнопку **Отменить ввод** (undo typing) на панели инструментов **Режим формы** (form view) или выбрать команду **Правка**, **Отменить ввод** (edit, undo typing).



Удаление записи

Для удаления текущей записи в форме перейдите на нее с помощью кнопок навигации и нажмите кнопку **Удалить запись** (delete record) на панели инструментов **Режим формы** (form view) или выберите команду **Правка, Удалить запись** (edit, delete record) либо щелкните по области выделения записи, находящейся слева от области данных формы, а затем нажмите клавишу **Del**. Если текущая запись не имеет подчиненных записей в связанных таблицах или разрешено каскадное удаление записей, ее удаление возможно. В этом случае появится диалоговое окно, запрашивающее подтверждение удаления записи. Подтвердите удаление записи, нажав кнопку **Да** (yes).



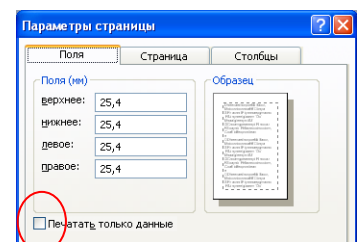
Печать формы

Форма, так же как и таблица, запрос или отчет базы данных, может служить документом для печати. Напечатать форму можно, находясь в любом режиме работы с ней. При этом внешний вид печатаемой формы будет соответствовать текущему режиму. Команды печати и способ настройки параметров печати точно такие же, как в других офисных приложениях.

Чтобы увидеть, как форма будет выглядеть при печати, следует перейти в режим предварительного просмотра. Для этого выберите команду **Файл, Предварительный просмотр** (file, print preview) или нажмите кнопку **Предварительный просмотр** (print preview) на панели инструментов. Чтобы вернуться в прежний режим работы с формой, нажмите кнопку **Заккрыть** (close) на панели инструментов.



Особый интерес представляет флажок **Печатать только данные** на вкладке **Поля** диалогового окна **Параметры страницы**. Этот флажок позволяет напечатать данные экранной формы прямо на готовый бумажный бланк, без вывода текстовых меток и линий, которые образуют собственно форму. Это удобно при оформлении стандартных текстовых документов, заполнять которые можно непосредственно из базы данных. Для этого достаточно сконструировать экранную форму, расположение полей на которой совпадает с оригиналом бланка.



Задание 3.2. Создание форм с помощью конструктора

Задание выполняется в соответствии с выданным индивидуальным вариантом.

Режим конструктора

Модифицировать структуру формы после того, как она будет создана мастером или автоматически, можно только в режиме конструктора. В этом режиме можно изменить источник данных для формы, количество отображаемых полей, внешний вид формы и элементов управления, добавить или удалить элементы управления, настроить их свойства. В этом же режиме можно и создавать форму, но требуется весьма высокая квалификация пользователя.

К форме можно применить один из предлагаемых стилей оформления для быстрой и качественной настройки ее внешнего вида, либо создавать свои собственные стили оформления и применять их к другим формам.

Чтобы создать пустую форму, не пользуясь автоматическим созданием форм и мастером форм, в окне базы данных щелкните по ярлыку **Формы** (forms), нажмите кнопку **Создать** (new) на панели в окне базы данных. В появившемся диалоговом окне **Новая форма** (new form) выделите элемент **Конструктор** (design view) и нажмите кнопку **Ок**. То же самое можно сделать, просто дважды щелкнув по ярлыку **Создание формы в режиме конструктора** (create form in design view), находящемуся перед списком существующих форм в базе данных.

Окно приложения в режиме конструктора

При работе в режиме конструктора форм используются следующие элементы окна приложения.

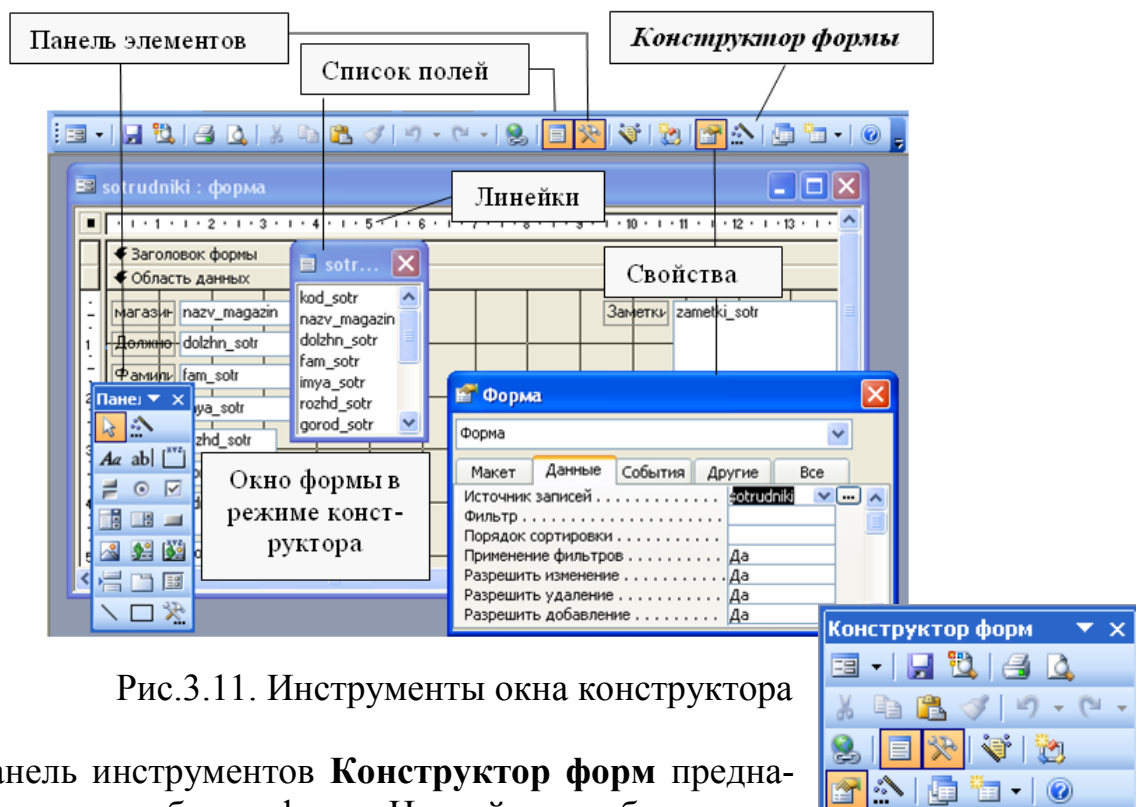


Рис.3.11. Инструменты окна конструктора

Панель инструментов **Конструктор форм** предназначена для разработки форм. На ней продублированы

основные команды, меню, используемые в режиме конструктора форм.

Большинство инструментов панели уже использовались неоднократно и смысл их понятен, назначение показывается на всплывающих подсказках при наведении указателя на кнопку. Следует только учесть, что их действие в данном случае распространяется на выделенные объекты активной формы. Новыми кнопками являются:

- **Список полей** - вывод списка полей, содержащихся в базовом источнике записей (запросе или таблице) основной формы (аналог – команды **Вид, Список полей**);

- **Автоформат** - выбор заранее определенных форматов для форм, включая такие характеристики, как фоновый рисунок для форм и шрифты для элементов управления (аналог – команды **Формат, Автоформат**);

- **Программа** - открытие модуля процедур обработки событий формы в окне редактора VBA (**Вид, Программа**);

- **Свойства** - открытие окон свойств выделенных объектов: формы, раздела (например, верхнего или нижнего колонтитула), а также элементов управления (аналог – команды **Вид, Свойства**).

Панель инструментов **Панель элементов** (toolbox), отображаемая в режиме конструктора, позволяет создавать элементы управления, при этом некоторые элементы управления можно создавать с помощью мастера элементов, осуществляющего пошаговое руководство этим процессом.



Панель элементов появляется на экране всякий раз при переходе в режим конструктора. Если закрыть эту панель инструментов, в следующий раз она не появится. Тогда, чтобы отобразить ее, нажмите кнопку **Панель элементов** на панели инструментов конструктора форм или выберите команду **Вид, Панели инструментов** (view, toolbars), а затем установите флажок **Панель элементов** (toolbox). Аналогичным образом вызываются и все другие панели инструментов.

Вертикальная и горизонтальная линейки предназначены для разметки формы по заданным размерам и выделения нескольких соседних элементов управления, лежащих в одном вертикальном или горизонтальном сегменте макета формы от одной отметки на линейке до другой.

Правая и нижняя границы макета формы позволяет задать ширину и высоту формы. Чтобы изменить размеры формы, перетащите границу макета формы с помощью мыши.

Структура формы

Макет формы состоит из разделов. Любая форма может включать следующие разделы (рис.3.12):

Рис.3.12. Структура формы

лонтитул страницы, и отображается только когда форма открыта в режиме предварительного просмотра. При печати многостраничной формы верхний колонтитул отображается вверху каждой страницы;

- **область данных** (detail) определяет основную часть формы, содержащую данные, полученные из источника. Данный раздел может содержать элементы управления, отображающие данные из таблиц и запросов, а также неизменяемые данные, например надписи. При печати многостраничной формы этот раздел отображается на каждой странице;

- **нижний колонтитул** (page footer) определяет нижний колонтитул страницы при печати формы. Этот раздел добавляется в форму вместе с разделом, определяющим верхний колонтитул страницы. Он отображается только когда форма открыта в режиме предварительного просмотра. При печати многостраничной формы нижний колонтитул отображается внизу каждой страницы;

- **примечание формы** (form footer) определяет нижнюю часть формы. Этот раздел добавляется в форму вместе с разделом заголовка формы. При печати многостраничной формы примечание формы будет отображено только внизу последней страницы.

Из всех перечисленных разделов обязательным является только раздел области данных. Высоту области любого раздела формы можно изменить с помощью мыши, перетаскивая нижнюю часть раздела вверх или вниз.

Разделы заголовка и примечания формы могут не отображаться в режиме формы, поскольку оба раздела имеют нулевую высоту. При создании новой пустой формы при помощи конструктора форм разделы заголовка и примечания не добавляются.

Для добавления или удаления разделов заголовка и примечания формы выберите команду **Вид, Заголовок/примечание формы** (view, form header/footer). Если форма уже содержит эти разделы, то команда **Заголо-**

- **заголовок формы** (form header) определяет верхнюю часть формы. Этот раздел добавляется в форму вместе с разделом примечания формы. В область заголовка формы можно поместить текст, графику и другие элементы управления. При печати многостраничной формы раздел заголовка отображается только на первой странице;

- **верхний колонтитул** (page header) определяет верхний колонтитул страницы при печати формы. Этот раздел добавляется в форму вместе с разделом, определяющим нижний ко-

вок/примечание формы будет отмечена флажком. Для удаления разделов заголовка и примечания формы сбросьте этот флажок.






Для добавления или удаления разделов верхнего и нижнего колонтитулов страницы выберите команду **Вид, Колонтитулы** (view, page header/footer). Если форма уже содержит эти разделы, то команда **Колонтитулы** будет отмечена флажком. Для удаления разделов верхнего и нижнего колонтитула страницы сбросьте этот флажок.
















Значения свойств всей формы, ее разделов и каждого элемента управления в форме задаются в окнах свойств соответствующих элементов. Для отображения на экране окна свойств формы, раздела или элемента управления выделите его и нажмите кнопку **Свойства** (properties) на панели инструментов **Конструктор форм** или выберите команду **Свойства** в контекстном меню.

Текстовые поля, флажки, переключатели и кнопки имеют связанные с ними метки. При выделении таких элементов управления и метка и элемент управления выделяются одновременно.

Панель элементов

Режим конструктора позволяет добавлять, удалять и изменять элементы управления на форме. Выбор типа элемента осуществляется на панели **Панель элементов**. Назначение кнопок панели показаны в таблице.

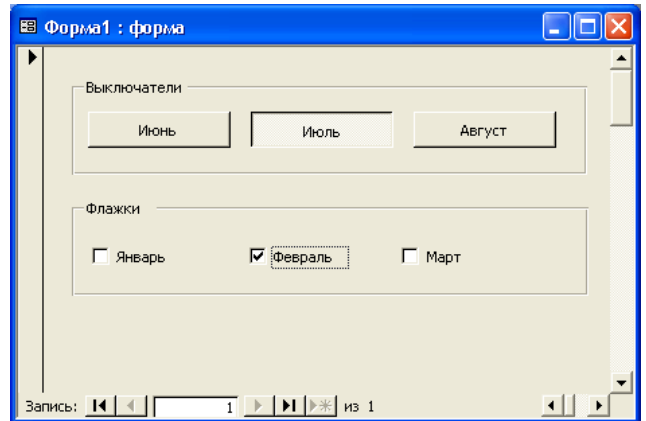
Кнопка	Название	Описание
	Выбор объектов	Позволяет выделить определенный элемент управления, щелкнув на нем, или группу элементов, поочередно щелкая на них при нажатой клавише Shift . При нажатой кнопке можно выделить элементы управления перетаскиванием прямоугольника выделения
	Надпись	Позволяет ввести неизменяемый описательный текст: заголовки, подписи, инструкции, чтобы сделать форму более понятной и легко читаемой. Большинству элементов управления надписи придаются автоматически
	Группа переключателей	Создается ряд элементов управления, среди которых можно производить выбор и которые присваивают полю числовое значение
	Переключатель	Создается круглая кнопка, которая обычно используется в группе взаимоисключающих переключателей (например, мужской или женский). Выделенная кнопка обозначена черным кружком
	Поле со списком	Создается элемент управления, состоящий из текстового поля со списком под ним. Вы можете или ввести текст в текстовое поле, или выбрать пункт из списка

	Кнопка	Добавляется прямоугольная кнопка, на которой можно щелкнуть для выполнения действия (таких как Ок или Заккрыть)
	Свободная рамка объекта	Отображаются объекты OLE, такие как рабочий лист Excel. Объект является постоянным, он не меняется с каждой записью
	Разрыв страницы	Форма разделяется на несколько экранов (или несколько страниц при печати)
	Подчиненная форма/отчет	В форму добавляется информация из дополнительной таблицы, так что вы можете просматривать или изменять ее данные
	Прямоугольник	На форме рисуется прямоугольник, чтобы сгруппировать или выделить ряд элементов управления
	Мастера	Если эта кнопка нажата, то при размещении нового элемента управления в форме процессом его настройки будет руководить Мастер
	Поле	Добавляется поле для отображения, ввода или изменения текстовых данных. Текстовые поля позволяют вводить произвольные значения (хотя можно ограничить множество допустимых значений для поля, задав с помощью окна свойств текстового поля, например, маску ввода или правило проверки введенного значения).
	Выключатель	Добавляется прямоугольная кнопка, которую можно использовать для включения и выключения параметра. При включенном параметре кнопка нажата
	Флажок	Добавляется маленькое квадратное окошко, которое можно использовать для включения и выключения параметра. Когда параметр включен, окошко содержит контрольную метку
	Список	Добавляется элемент управления, отображающий раскрывающийся список, из которого можно выбрать пункт
	Рисунок	В форму добавляется постоянный рисунок (он не меняется при изменении записей)
	Присоединенная рамка объекта	Отображаются объекты OLE, хранящиеся в записях таблицы, такие как фотографии служащих. При просмотре различных записей объект будет меняться
	Набор вкладок	Позволяет разделить форму на отдельные вкладки подобно тому, что наблюдается в диалоговых окнах
	Линия	Дает возможность нарисовать на форме отдельную прямую линию для зрительного разделения или выделения элементов управления
	Другие элементы	Добавляются из списка дополнительных элементов управления, предлагаемого приложением Office. Эти

		элементы отображены в меню, появляющемся при щелчке на этой кнопке
--	--	--

Чтобы предоставить пользователю выбор из определенного набора значений, вместо текстовых полей применяют такие элементы управления, как флажки, переключатели, выключатели, объединенные в группы однотипных элементов, а также списки.

С каждым элементом управления в группе сопоставляется уникальное число, однозначно определяющее данный элемент в группе, и это число сохраняется в качестве значения **Свойства значение** (option value) элемента управления. Отдельно стоящий элемент управления типа флажки (check box), переключатели (option button) или выключатели (toggle button), не включенный в группу подобных элементов, определяет значение логического типа:



Да (yes) или **Нет** (no). В данном случае вместо значения **Да** допускается использовать значение **Истина** (true) или любое положительное число, а вместо значения **Нет** — значение **Ложь** (false) или **ноль**. Элементы управления Флажки (check box), Переключатели (option button), Выключатели (toggle button) и Группа переключателей (option group) имеют свойство **Значение по умолчанию** (default value), позволяющее задать первоначальное состояние элемента управления или группы. Для отдельного элемента управления в качестве значения этого свойства используется значение логического типа, а для группы — число, сопоставленное с одним из элементов группы (с тем элементом, который будет выбран в группе по умолчанию).

Кроме групп элементов, для предоставления выбора значения какого-либо параметра из заданного набора можно использовать элементы управления Список (list box) или Поле со списком (combo box). Ввести значение в поле со списком можно двумя способами: ввести значение в поле или выбрать значение в раскрывающемся списке.

Когда требуется выбрать одно из нескольких возможных значений, пользователю намного удобнее работать со списком или группой элементов, чем с текстовым полем. Во-первых, чтобы сделать выбор, пользователю не нужно вводить значение полностью, а достаточно лишь щелкнуть кнопкой мыши, чтобы выделить нужный элемент. Во-вторых, нет необходимости помнить все допустимые при выборе значения, поскольку они явно перечислены в элементах группы или списка. Поэтому, чтобы сделать интерфейс приложения более удобным для пользователя, используйте в формах группы или списки вместо текстовых полей всегда, когда нужно обеспечить выбор

одного из нескольких значений. А также используйте отдельные флажки, выключатели или переключатели вместо текстовых полей, когда требуется предоставить пользователю выбор между альтернативами "да" — "нет" для заданных параметров.

Задание 3.3. . Создание и удаление элементов управления

Задание выполняется в соответствии с выданным индивидуальным вариантом.

Создание и удаление элементов управления

Access предоставляет два простых метода создания элементов управления: с помощью панели инструментов **Панель элементов** (toolbox) и с помощью кнопки **Список полей** (field list). Кроме того, при создании элементов управления с помощью панели элементов можно воспользоваться мастером элементов, позволяющим пройти шаг за шагом процесс создания более сложных элементов управления, например таких как списки, поля со списком и группы элементов управления.

Создание элементов управления с помощью панели элементов

Надпись — самый простой для использования тип элементов управления панели элементов. Для добавления надписи в какой-либо раздел формы:

1. На панели элементов нажмите кнопку **Надпись** (label). Когда указатель мыши попадет в активную область формы, он примет вид крестика со значком элемента управления Надпись. Центр крестика определяет позицию верхнего левого угла элемента управления надписью;

2. Поместите указатель мыши в виде крестика в область нужного раздела формы. Нажмите левую кнопку мыши и, удерживая ее, перетащите указатель мыши в нижний правый угол надписи, затем отпустите кнопку (рис.3.13).

3. Внутри контура надписи появится текстовый курсор. Введите нужный текст. Если, создав надпись, не ввести в нее хотя бы одного символа, то после следующего щелчка кнопкой мыши этот элемент управления исчезнет.

После добавления элемента управления для его перемещения и изменения размеров используются угловой маркер пере-

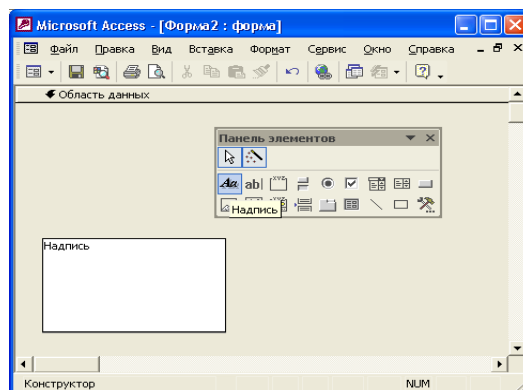


Рис3.13. Добавление элемента **Надпись**

мещения и маркеры изменения размеров. Положение углового маркера перемещения определяет значения свойств элемента управления **От левого края** (left) и **От верхнего края** (top). Маркеры изменения размеров устанавливают значения свойств элемента управления **Ширина** (width) и **Высота** (height).

Для добавления большинства других типов элементов управления (Выключателей, Переключателей, Флажков) в форму используется процесс, описанный выше. Некоторые кнопки панели элементов, например Список (list box) или Поле со списком (combo box), при создании элемента управления вызывают мастер элементов, если кнопка мастера (control wizards) нажата.

Чтобы создать элемент управления **Список** с помощью мастера:

1. На панели элементов нажмите кнопку **Мастера** (control wizards), если она еще не нажата.

2. На панели элементов нажмите кнопку **Список** (list box).

3. Поместите указатель мыши в виде крестика в область нужного раздела формы. Нажмите левую кнопку мыши и, удерживая ее, перетащите указатель мыши в нижний правый угол списка, затем отпустите. Вместе с перемещением указателя мыши будет изменяться и контур списка. Число строк и количество символов текущего типа шрифта, которые может отобразить список, выводятся в **Строке состояния**.

4. Появится первое диалоговое окно мастера списков. В этом диалоговом окне выберите переключатель объект **Список. будет использовать значения из таблицы или запроса** (i want the list box to look up the values in a table or query). Нажмите кнопку **Далее** (next). Появится второе диалоговое окно мастера списков (рис.3.14).

Список может также содержать набор фиксированных значений, который мастер предоставит возможность перечислить. Для этого необходимо выбрать **Будет введен фиксированный набор значений**.

Довольно часто при разработке формы для ускорения процесса перехода по большому количеству записей возникает потребность в создании специального поля со списком, при выборе элемента которого в форме будет отображаться соответствующая запись из источника данных. В Access с помощью мастера можно создать список или поле со списком специально для поиска нужной записи в форме. Для этого следует выбрать соответствующий переключатель в первом диалоговом окне мастера такого элемента: **Поиск записи в форме на основе значения, которое содержит список** (find a record on my form based on the value i selected in my list box). Этот переключатель отображается в случае, если для формы задан источник записей: таблица или запрос. В режиме формы выде-

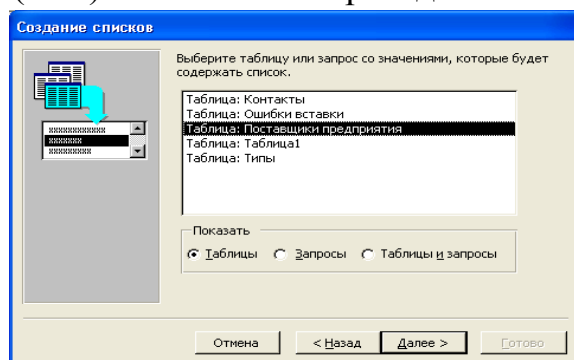


Рис.3.14. Второе диалоговое окно

ление элемента в этом списке будет приводить к переходу на соответствующую запись в форме. Переход реализуется с помощью процедуры обработки события *После обновления* (after update), созданной мастером.

5. Во втором диалоговом окне мастера списков нужно выделить таблицу или запрос, значения из которого должны появляться в списке. Перейдите к третьему диалоговому окну мастера (рис.3.15).

6. В третьем окне нужно определить поля, которые будут отображаться в списке. Чаще всего в списке должно отображаться одно поле, значение из которого будет выбираться пользователем. Но следует помнить, что выбор осуществляется из таблицы или запроса и значением данного элемента управления будет значение ключевого поля той записи, которая отображена по признакам из списка. Перейдите к следующему окну диалога.

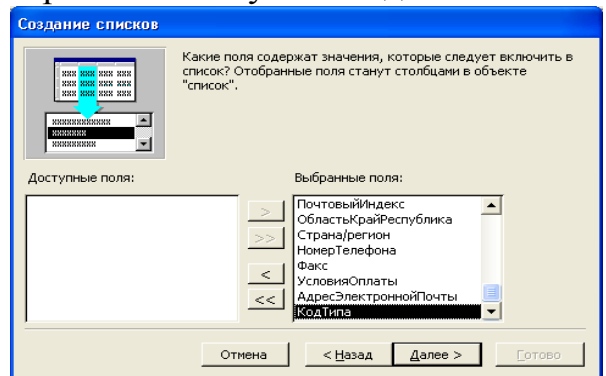


Рис.3.15. Третье диалоговое окно

7. В четвертом диалоговом окне (рис.3.16) отображаются выбранные поля, причем поле *Код_типа*, которое показывать пользователю не нужно,

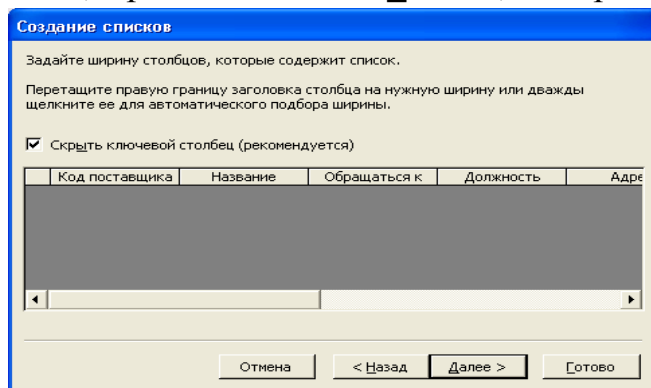


Рис.3.16. Четвертое диалоговое окно мастера списков

скрыто (установлен флажок *Скрыть ключевой столбец*). Перемещая с помощью мыши правую границу столбцов, можно задать ширину столбцов. Чтобы настроить ширину столбца по ширине самого длинного значения, достаточно подвести указатель мыши к правой границе столбца и дважды щелкнуть левой кнопкой. Нажмите кнопку *Далее*.

8. Если для формы задан источник записей: таблица или запрос, появится пятое диалоговое окно мастера списков, представленное на рис.3.17. В этом диалоговом окне нужно указать

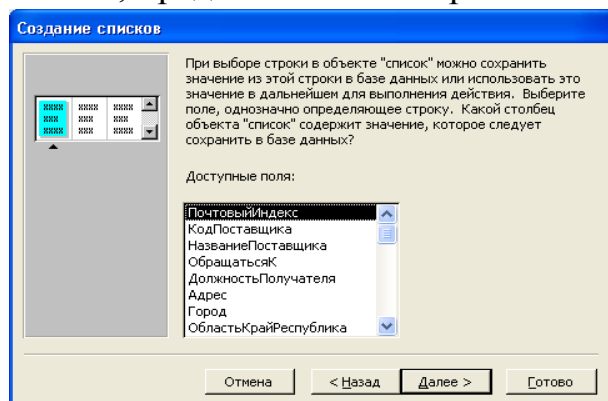


Рис.3.17. Пятое окно диалога

то поле из списка, которое следует сохранить в поле источника данных формы. Перейдите к последнему диалоговому окну мастера списков.

9. В последнем диалоговом окне можно указать название, которое будет использоваться в качестве метки для создаваемого списка. Чтобы после создания списка получить справку по возможностям его настройки, установите флажок, распо-

ложенный внизу диалогового окна.

10. Для завершения процесса создания списка нажмите кнопку **Готово** (finish).

При создании таких элементов управления без помощи Мастера элементов свойства создаваемых элементов нужно настраивать вручную с помощью окна свойств элемента управления. Чтобы открыть окно свойств какого-либо элемента управления формы, выделите его и нажмите кнопку **Свойства** (properties) на панели инструментов **Конструктор форм** (form design) либо дважды щелкните по элементу управления. На рис.3.18 изображено окно свойств списка.

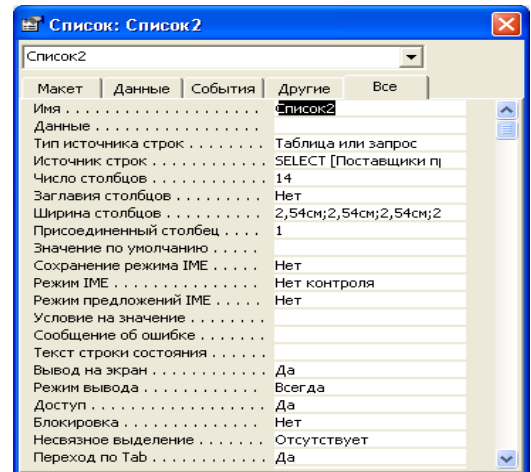



Рис.3.18. Окно свойств списка

Создание элементов управления с помощью кнопки список полей

Еще одним достаточно простым и удобным способом создания элементов управления является использование списка полей.

Чтобы создать элемент управления, связанный с каким-либо полем источника данных разрабатываемой формы:

1. На панели инструментов **Конструктор форм** (form design) нажмите кнопку **Список полей** (field list). Появится всплывающее окно, 

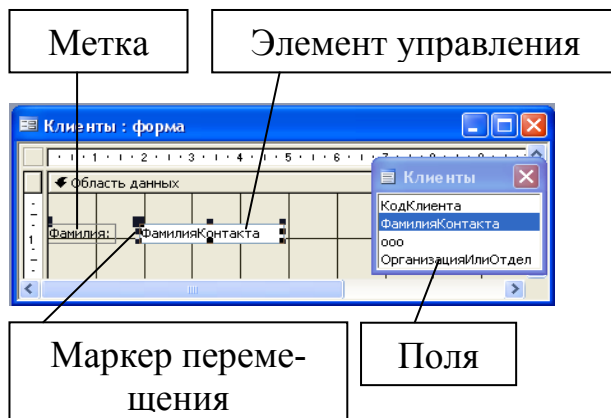


Рис.3.19. Поля источника формы

содержащее поля из источника данных формы (рис.3.19).

2. В открывшемся окне выделите нужное поле и с помощью мыши перетащите его в область одного из разделов формы. В форме появится выбранное поле с меткой, содержащей название поля таблицы или запроса. Теперь его можно перемещать и форматировать стандартными методами, используемыми для модификации элементов управления.

Удаление элементов формы

Чтобы удалить из формы какой-либо элемент управления:

1. Выделите элемент управления или несколько элементов сразу;
2. Нажмите клавишу **Del** или выберите команду **Правка, Удалить**.

Копирование и вставка элементов управления

Все стандартные команды работы с буфером обмена Windows применимы и для элементов управления. Выделенный элемент управления или группу элементов управления можно удалить или скопировать в буфер обмена, а затем вставить в форму и надлежащим образом разместить. В Access для работы с буфером обмена используются также стандартные сочетания клавиш Windows:

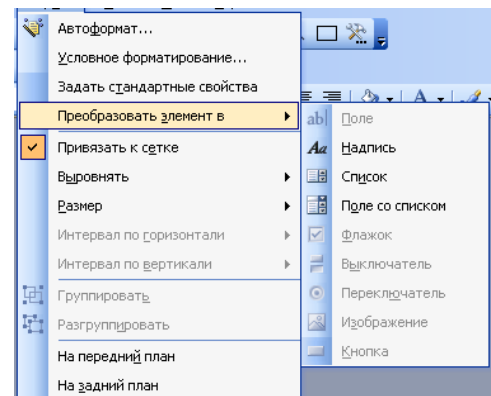
- **Ctrl+X** — для перемещения выделенных элементов управления в буфер обмена с удалением их из формы;
- **Ctrl+C** — для копирования выделенных элементов управления в буфер обмена;
- **Ctrl+V** — для вставки содержимого буфера обмена.

Изменение типа элемента управления

Если при разработке формы возникла необходимость заменить тип элемента управления, то для любого из девяти наиболее часто применяющихся типов элементов управления это легко сделать с помощью команды меню **Формат, Преобразовать элемент в** для выделенного элемента. Например, текстовое поле — в поле со списком.

Девяти типам элементов соответствуют девять команд этого меню.

Конечно, для элемента данного типа возможно преобразование в элемент не любого из названных девяти типов, а только в элемент похожего типа. Очевидно, что текстовое поле не преобразовать в переключатель, зато его можно преобразовать в надпись, список или поле со списком. В меню будут доступны соответствующие команды. При преобразовании сохраняются значения общих свойств элементов обоих типов.



Перемещение и изменение размеров элементов управления

При разработке формы, даже созданной с помощью мастера, наверняка потребуется изменить расположение и размеры элементов управления. Можно изменить размеры и местоположение отдельного элемента, а можно настроить сразу несколько элементов так, чтобы они были одинакового размера и чтобы были расположены на форме ровно и на равном расстоянии друг от друга. От этого форма сразу приобретет привлекательный вид, и работать с ней будет намного удобнее.

Перемещение элемента управления:

1. Выделите элемент управления.
2. Поместите указатель мыши на границу элемента управления, но не на маркер изменения размеров. Когда указатель мыши превратится в изображение руки, нажмите левую кнопку мыши и, не отпуская ее, перетащите элемент управления на новое место.

Отдельно элемент управления или его метку можно перетаскивать также за маркер перемещения, находящийся в левом верхнем углу элемента управления. Указатель мыши, помещенный над маркером перемещения, примет вид руки с вытянутым вверх указательным пальцем. Щелкните по маркеру перемещения и, удерживая нажатой левую кнопку мыши, перетащите элемент управления на новое место, а затем отпустите кнопку мыши.

Изменение размеров элемента управления.

Чтобы изменить ширину и высоту элемента управления, поместите указатель мыши над одним из трех угловых маркеров изменения размера выделенного элемента управления так, чтобы он превратился в наклонную двустороннюю стрелку. Нажмите левую кнопку мыши и, удерживая ее нажатой, перетащите границу элемента управления, а затем отпустите кнопку мыши.

Чтобы изменить только высоту (или только ширину) выделенного элемента управления, выполните аналогичное перетаскивание границ элемента управления, используя верхний или нижний (левый или правый) маркер изменения размера.

Еще один способ изменения размеров выделенного элемента управления: с помощью команд меню **Формат, Размер, По размеру данных** (format, size, to fit). Это позволяет автоматически изменить размеры элемента в соответствии с размером отображаемых в нем данных. К примеру, для текстового поля меняется размер по вертикали в зависимости от заданного размера шрифта для текста.

Формат, Размер, По узлам сетки (format, size, to grid) - команда выполняет автоматическое изменение размеров и положения элемента так, чтобы его границы лежали на ближайших линиях, соединяющих узлы сетки. Действие этой команды заметно, если элементы имеют такие размеры, что их границы лежат между узлами сетки, например, когда первоначальные размеры элементов устанавливались при отключенном режиме выравнивания элементов по сетке (о выравнивании по сетке см. ниже).

Выделение нескольких элементов управления (например, обводкой указателем мыши или удерживая клавишу **Shift**) позволяет задавать для них общие свойства, перемещать их и изменять их размеры одновременно. Отображение окна свойств для нескольких выделенных элементов производится так же, как и для одного выделенного элемента.

Выравнивание элемента управления по сетке

Чтобы отобразить сетку, выберите команду **Вид, Сетка** (view, grid). Соответствующая команда меню будет отмечена галочкой. По умолчанию сетка отображается во всех новых формах. Чтобы отменить отображение сетки, снова повторите те же действия. В окне конструктора форм будет отображена сетка, состоящая из линий толщиной в один пиксел, которые проводятся через каждое деление горизонтальной и вертикальной линеек окна конструктора форм (то есть через сантиметр). С ее помощью можно выровнять элементы управления по горизонтали или по вертикали.

Даже если сетка не видна, элементы управления можно "привязать к сетке", выбрав команду **Формат, Привязать к сетке** (format, snap to grid). Если перемещать элемент управления при включенной привязке к сетке, то левый верхний угол объекта будет двигаться дискретно по узлам сетки. Аналогично изменение размеров элемента управления в этом режиме будет происходить дискретно по узлам сетки.

Если команда привязки элементов формы к сетке включена, то для перемещения или изменения размеров элемента управления без привязки к сетке нажмите и удерживайте клавишу **Ctrl**.

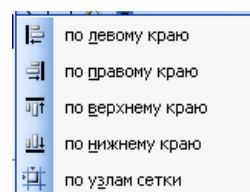
Еще один способ выравнивания отдельного элемента управления по сетке: выделите его и выберите команду **Формат, Выровнять, По узлам сетки** (format, align, to grid). В этом случае элемент управления автоматически переместится так, чтобы его левый верхний угол находился в ближайшем узле сетки.

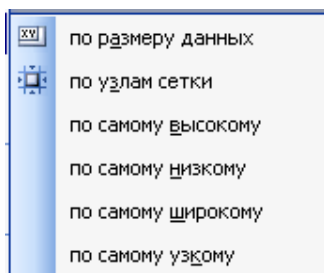
Расстояние между узлами сетки определяется густотой сетки по горизонтали (координата x) и по вертикали (координата y). Густота узлов сетки задается числом, которое означает количество равных частей, на которые будет разделен каждый сантиметр сетки в данном направлении (по вертикали или по горизонтали). допустимой является густота узлов величиной от 1 до 64 делений на сантиметр. в случае, если густота узлов сетки превышает 10 узлов на сантиметр, узлы сетки не отображаются на экране, т. к. они слишком мало отстоят друг от друга. Чтобы изменить густоту узлов сетки:

1. Выберите команду Правка, Выделить форму (edit, select form);
2. На панели инструментов Конструктор форм (form design) нажмите кнопку **Свойства** (properties). Появится окно свойств формы;
3. В окне свойств формы раскройте вкладку **Макет** (format) и при помощи полосы прокрутки найдите свойства **Число делений по x** (grid x) и **Число делений по y** (grid y);
4. Измените значение свойств **Число делений по x** и **Число делений по y**.

Выравнивание нескольких элементов управления

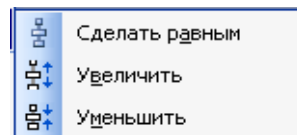
Чтобы выровнять один или несколько элементов управления по сетке или относительно друг друга, выделите выравниваемые элементы, выберите команду **Формат, Выровнять** (format, align). Затем выбрать один из вариантов выравнивания.





Для выравнивания размеров нескольких элементов выделите их любым описанным ранее способом, а затем выберите одну из команд меню **Формат, Размер** (format, size).

Чтобы настроить расстояния между несколькими вертикально расположенными (в один "столбец") элементами управления, выделите их и выберите одну из команд меню **Формат, Интервал по вертикали** (format, vertical spacing). Аналогично настраиваются расстояния между горизонтально расположенными элементами (**Формат, Расстояния по горизонтали**).



По умолчанию порядок перекрывания одного элемента другим определяется порядком создания элементов. Можно отодвинуть верхний элемент или переместить его на задний план. Для этого выделите этот элемент, а затем выберите команду **Формат, На задний план** (format, send to back). Чтобы переместить выделенный элемент на передний план, выберите команду **Формат, На передний план** (format, bring to front).

В режиме конструктора можно объединить несколько элементов в группу, выделив их и выбрав команду **Формат, Группировать** (format, group). Тогда можно будет оперировать всей группой как единым элементом. Чтобы отменить группировку элементов, выделите группу и выберите команду **Формат, Разгруппировать** (format, ungroup).

Оформление формы

Чтобы сделать форму более привлекательной, выделить на общем фоне какие-то важные элементы, вы можете изменить цвет фона формы и цвета ее элементов, выбрать оформление рамок для элементов управления, задать подходящий шрифт, его цвет и размер для текста надписей и других элементов. Параметры форматирования задаются с помощью панели инструментов **Формат (форма/отчет)** (formatting (form/report)) (Рис.____) и с помощью окна свойств формы или ее элементов.

Для печати лучше всего выбрать белый фон для формы и ее элементов, не очень жирные рамки для элементов и шрифт обычной толщины для текста. Если же форма имеет хорошее цветовое оформление, которое важно сохранить для интерактивной работы, печатайте ее в режиме таблицы.

Применение автоформата

Автоформат — это набор параметров оформления формы и ее элементов, который можно применить для форматирования формы в автоматическом режиме. Наряду с имеющимися вариантами есть возможность создать собственный вариант (рис.3.20).

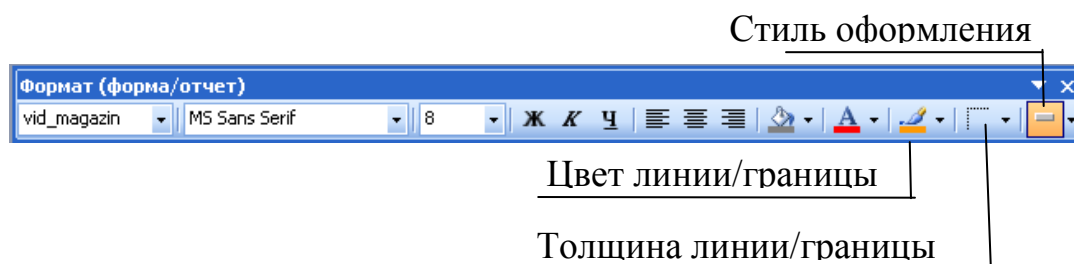


Рис.3.20. Панель инструментов **Формат (форма/отчет)**

Откройте форму в режиме конструктора и выполните:

1. Выделите всю форму. Для применения автоформата только для отдельных элементов формы - выделите их, а не всю форму.
2. Выберите команду **Формат, Автоформат** (format, autoformat) или нажмите кнопку Автоформат (autoformat) на панели инструментов **Конструктор форм** (form design). Появится диалоговое окно **Автоформат** (autoformat) (рис.3.21).

3. Выделите в списке автоформатов тот, который нужно применить к форме. В правой половине диалогового окна отображается эскиз выбранного в списке автоформата. Нажмите кнопку **Ок**.

4. Если нужно применить к форме только часть автоформата, исключив применение таких параметров автоформата, как шрифт, цвет и оформление границ элементов управления, то прежде чем нажимать кнопку **Ок** в диалоговом окне **Автоформат**, нажмите кнопку **Параметры** (options). В раскрывшейся нижней части диалогового окна сбросьте флажки для тех параметров формата, которые вы хотели бы оставить в текущей форме прежними.

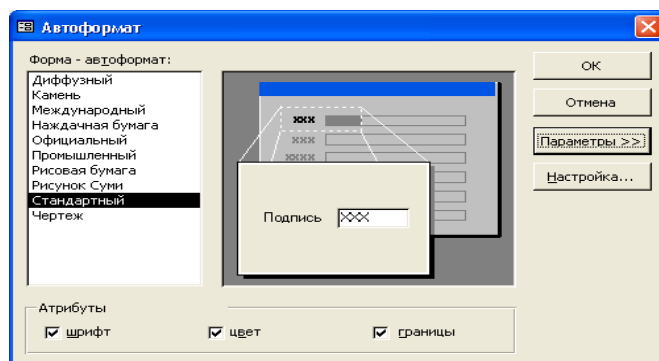


Рис.3.21. Диалоговое окно **Автоформат**

Создание собственного автоформата

Чтобы создать собственный автоформат, необходимо оформить в требуемом стиле любую форму, которая послужит образцом для создания автоформата:

1. Откройте форму образец в режиме конструктора;
2. Выберите команду **Формат, Автоформат** и в открывшемся диалоговом окне нажмите кнопку **Настройка** (customize). Появится диалоговое окно **Настройка автоформата** (customize autoformat);
3. Выберите переключатель **Создание нового стиля на основе стиля объекта: "имя текущей формы"** и нажмите кнопку **Ок**;
4. В появившемся диалоговом окне введите имя нового автоформата, нажмите кнопку **Ок**.

5. Нажмите кнопку **Ок**, чтобы применить только что созданный автоформат к текущей форме, или кнопку **Заккрыть** (close), чтобы просто закрыть диалоговое окно Автоформат.

Этот автоформат появится в списке рядом с другими автоформатами Access.

Чтобы удалить ставший ненужным автоформат, выделите его в списке в диалоговом окне Автоформат и нажмите кнопку **Настройка**. Выберите переключатель **Удаление стиля "имя автоформата"** (delete "имя автоформата"). Если же требуется всего лишь обновить параметры выбранного автоформата на основе формата текущей формы, выберите в этом диалоговом окне переключатель **Обновление стиля "имя автоформата" с атрибутами объекта: "имя текущей формы"** и нажмите кнопку **Ок**, а затем — кнопку **Заккрыть**, чтобы закрыть диалоговое окно Автоформат.

Изменение цвета фона формы и ее элементов

Основным параметром, определяющим внешний вид формы, является цвет ее фона. Поэтому при выборе цвета фона необходимо руководствоваться тем, для каких целей предназначена разрабатываемая форма и как в дальнейшем планируется ее использовать.

Цвет фона раздела формы или ее элемента определяется значением свойства **Цвет фона** (back color). В форме, созданной с помощью мастера форм, фон всех ее разделов зависит от стиля оформления, выбранного при ее создании. Например, автоформат стандартный (standard) задает светло-серый цвет фона.

Чтобы изменить цвет фона раздела формы или ее элемента:

1. Выделите раздел или элемент формы;
2. На панели инструментов **Формат (форма/отчет)** (formatting (form/report)) нажмите стрелку справа от кнопки **Цвет заливки/фона** (fill/back color) для вызова палитры цветов. Выберите в палитре нужный цвет.

Чтобы поменять цвет всех разделов формы, нужно задать новый цвет для каждого раздела отдельно.

Цвет фона раздела формы не влияет на цвет фона элементов управления, находящихся в нем. для каждого элемента управления также можно выбрать свой цвет фона и цвет шрифта. для меток элементов управления обычно задан прозрачный цвет фона, т. е. в качестве фона метки используется та часть фона раздела формы, которую она занимает. по умолчанию для текстовых полей используется белый цвет фона, поэтому цвет текстовых полей (и содержащихся в них данных) контрастен цвету фона формы.

Изменение фонового рисунка формы

В качестве фона формы можно задавать не только гладкие цвета, но и растровые рисунки, хранящиеся в разнообразных графических форматах, таких как wmf и bmp.

Чтобы задать фоновый рисунок для формы:

1. В режиме конструктора выделите всю форму;
2. На панели инструментов **Конструктор форм** (form design) нажмите кнопку **Свойства** (properties). Появится окно свойств формы;
3. Раскройте вкладку **Макет** (format) в окне свойств формы и с помощью полосы прокрутки переместитесь в конец списка свойств, где находится свойство **Рисунок** (picture).
4. В поле свойства **Рисунок** введите путь и имя файла рисунка либо воспользуйтесь построителем: поместите точку ввода в поле **Рисунок**, затем нажмите кнопку **Построить** (build), которая появится справа от поля в виде маленького квадрата с изображением многоточия на нем, после чего откроется диалоговое окно **Выбор рисунка** (insert picture), с помощью которого можно найти и указать необходимый файл рисунка. После того как файл рисунка будет найден, выделите его название, а затем нажмите кнопку **Ок**. Путь и имя файла будут отображены в поле свойства **Рисунок**.
5. Закройте окно свойств формы. В отличие от цветов фона, фоновый рисунок распространяется на все разделы формы. Чтобы удалить фоновый рисунок, просто удалите содержимое текстового поля свойства **Рисунок**.

С помощью свойств, сгруппированных на вкладке **Макет** окна **Свойств формы**, можно настроить отображение фонового рисунка:

- свойство **Тип рисунка** (picture type) определяет метод, используемый для присоединения рисунка к форме: **внедренный** (что делает форму автономной) или **связанный** (что экономит объем памяти, но создает ограничения на манипулирование объектами);

- свойство **Масштабы рисунка** (picture size mode) можно определить размеры фонового рисунка: элемент **Фрагмент** (clip) выводит рисунок в исходных размерах; **Вписать в рамку** (stretch) - рисунок изменяется до размеров формы, возможно с изменением пропорций; **По размеру рамки** (zoom) - приведет к растяжению или сжатию рисунка с сохранением пропорций до максимально возможных размеров, при которых не происходит обрезка рисунка;

- свойство **Выравнивание рисунка** (picture alignment) - определяет положение фонового рисунка относительно формы. Чтобы фоновый рисунок центрировался относительно формы, а не окна формы, установите значение **По центру формы**, а не **По центру**.

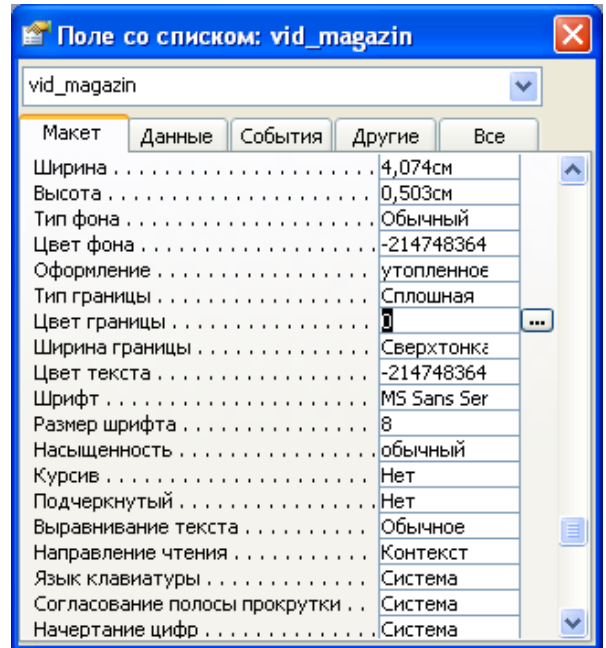
После изменения значения свойства результат этого изменения немедленно отображается в форме.

Форматирование элементов формы

Изменить внешний вид элемента можно с помощью окна свойств элемента управления на вкладке **Макет**.


Найдите свойства, отвечающие за оформление элемента: высота, ширина, цвет фона и шрифта, размер шрифта, ширина границ и т.д. Выбор значений для большей части свойств осуществляется при помощи раскрывающихся списков. Для определения значения свойства, определяющего цвет какого-либо параметра, необходимо ввести число или выбрать из палитры, открывающейся при выборе свойства и нажатии справа на прямоугольник с точками.

Более удобным для форматирования элементов управления считается использование панели инструментов **Формат (форма/отчет)**.



Копирование формата элемента управления

Отработанный формат элемента управления можно перенести на другие элементы. Выполните следующую процедуру:

1. Выделите уже отформатированный элемент управления;
2. Нажмите кнопку **Формат по образцу** (format painter) на панели инструментов **Конструктор форм** (form design). 
3. Поместите указатель мыши над элементом управления, который нужно отформатировать по имеющемуся образцу. Указатель мыши примет вид инструмента копирования формата (изображение малярной кисти). Щелкните левой кнопкой мыши.

Будут скопированы только те параметры форматирования, применение которых возможно для данного типа элемента управления.

Чтобы отформатировать по образцу сразу несколько элементов управления, следует после выделения элемента, имеющего необходимый формат, щелкнуть дважды по кнопке **Формат по образцу**. Включится режим копирования формата. После этого нужно подряд щелкать левой кнопкой мыши по всем элементам управления, которые требуется отформатировать. Чтобы закончить данную операцию, нажмите на эту кнопку (формат по образцу) еще раз или нажмите клавишу **Esc**.

Чтобы все новые элементы некоторого типа, создаваемые в текущей форме, автоматически приобретали необходимые параметры форматирования, выделите элемент управления этого типа, который послужит образцом формата, и выберите команду **Формат, Здать стандартные свойства** (format, set control defaults).

Условное форматирование элементов формы

Допустимо динамическое изменение форматирования элементов управления при работе с формой в режиме формы или режиме таблицы в зависимости от выполнения некоторых условий. Этот инструмент называется формат по условию. Такой способ удобен, например, для контроля выхода параметра за заданные границы.

Для применения условного форматирования откройте форму в режиме конструктора, выделите необходимый элемент управления и выберите команду **Формат, Условное форматирование** (format, conditional formatting). Появится диалоговое окно **Условное форматирование** (conditional formatting) (рис.3.22). Это диалоговое окно позволяет задать от одного до трех условий, от выполнения которых зависит формат элемента управления. Условия могут быть трех типов:

- значение элемента управления сравнивается с константой или со значением выражения;
- задается логическое выражение, значение которого (истина (true)) соответствует выполнению условия;
- условие выполняется при наведении курсора (событие получения фокуса) на данный элемент управления.

Тип условия задается выбором соответствующего элемента в самом левом раскрывающемся списке в группе параметров условия в диалоговом окне. Таких элементов списка, соответственно, три.

Операндами выражений здесь могут быть ссылки на другие поля формы в том виде, как они обычно указываются в выражениях Access, и даже вызовы функций VBA. Например, для контроля закупок не более пяти экземпляров продукта при известной его цене выражение следует сконструировать следующим образом. Для поля **Стоимость заказа** вызвать окно условного форматирования и набрать: **Значение поля – Больше – $5 * [Цена]$** – цвет шрифта **Красный**. Если заданное условие выполнено, то элемент форматировается определенным для этого условия образом. Если ни одно из заданных условий не выполнилось, элемент форматировается в соответствии с форматом по умолчанию, который устанавливается в самой верхней части диалогового окна.

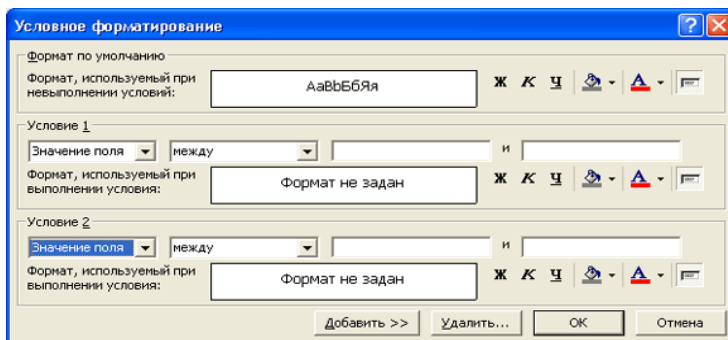
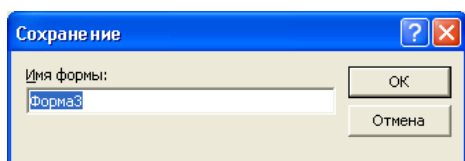


Рис.3.22. Диалоговое окно
Условное форматирование

Сохранение формы

Для сохранения только что созданной или отредактированной формы, открытой в режиме конструктора, выберите команду **Файл, Сохранить** (file,

save) или нажмите кнопку *Сохранить* (save) на панели инструментов Конструктор форм (form design). Если форма была создана и сохраняется в первый раз, появится диалоговое окно **Сохранение**



(save as) с полем для ввода имени новой формы **Имя формы** (form name). Введите имя новой формы и нажмите кнопку **Ок**.

Контрольные вопросы

1. Что такое форма, и для каких целей она создается? В чем ее преимущества по сравнению с таблицами?
2. Какова технология создания формы с помощью мастера?
3. Какие существуют типы форм?
4. Какие элементы образуют конструкцию формы?
5. Какие режимы работы с формами используются в СУБД?
6. Какие существуют возможности по оформлению формы?
7. Что такое элементы управления формы и как они создаются?
8. Какова технология создания, редактирования и удаления элементов управления?
9. Какие элементы оформления формы используются в СУБД?
10. В чем заключается механизм условного форматирования элементов формы?

4. РАЗРАБОТКА ОТЧЕТОВ

Цель

Приобрести навыки разработки, модификации и использования отчетов.

Учебные вопросы

4.1. Создание отчета;

4.2. Модификация отчета.

Литература

Литература аналогична той, которая указана в разделе 1.

Задание 4.1. Создание отчета

Задание выполняется в соответствии с выданным индивидуальным вариантом.

Документирование БД

Отчёт является средством документирования БД. Он позволяет выбрать из базы данных требуемую пользователем информацию и оформить ее в виде документов, которые можно просмотреть и напечатать. Данные для отчетов извлекаются из таблицы или запроса. Кроме этого, в отчете могут отображаться вычисленные по исходным данным значения, например итоговые суммы.

Отчеты и формы имеют много общего. Однако, в отличие от форм, отчеты не предназначены для ввода и правки данных в таблицах. Они позволяют лишь просматривать и печатать данные. В отчете невозможно изменить исходные данные с помощью элементов управления, как это можно сделать с помощью форм. Хотя в отчетах можно использовать такие же элементы управления для указания состояния переключателей, флажков и списков.

Отчёт – достаточно гибкий инструмент. Он позволяет не только определить состав полей, выводимых на печать, но и изменять структуру вывода, форматировать документ и добавлять различную служебную информацию, также выводимую на печать.

Автоматическое создание отчета

Существует несколько способов создания отчетов. На панели объектов окна **База данных** щелкните по ярлыку **Отчеты** (Reports) и нажмите кнопку **Создать** (New). Появится диалоговое окно **Новый отчет** (New Report), в котором можно выбрать один из способов (рис.4.1).

Наиболее простым способом является использование средств автоматического создания отчета (автоотчета). Автоотчет, созданный с использованием любого из двух форматов, будет включать все поля и записи, имеющие-

ся в выбранном источнике данных. Источник данных выбирается в том же окне в поле с выпадающим списком.

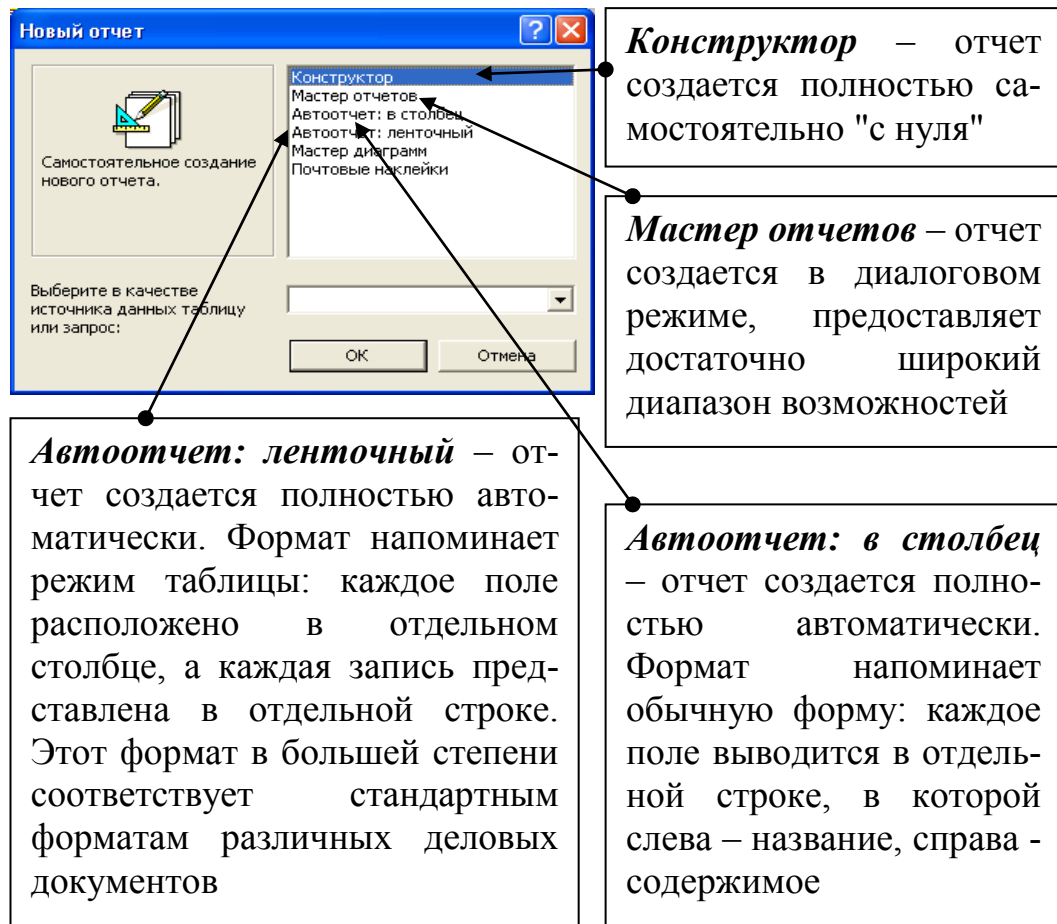


Рис. 4.1. Способы создания отчетов

Чтобы созданный отчет можно было использовать в дальнейшем, его необходимо сохранить. Эта операция выполняется стандартным образом для всех документов офисных пакетов программ.

Создание отчёта с помощью Мастера

Для создания отчёта необходимо:

1. В контейнере БД выбрать элемент БД **Отчёты-Создание отчёта с помощью мастера**. Или **Отчеты - Создать-Мастер отчетов**. Откроется окно (рис.4.2 - 4.4).

Если на каком-либо шаге создания отчета при помощи Мастера отчетов была допущена ошибка или возникла необходимость изменения каких-либо параметров, то для возвращения к предыдущему шагу нажмите кнопку **Назад** (Back). Кроме того, в любой момент можно нажать кнопку **Отмена** (Cancel) для отказа от создания отчета и возвращения к окну базы данных.

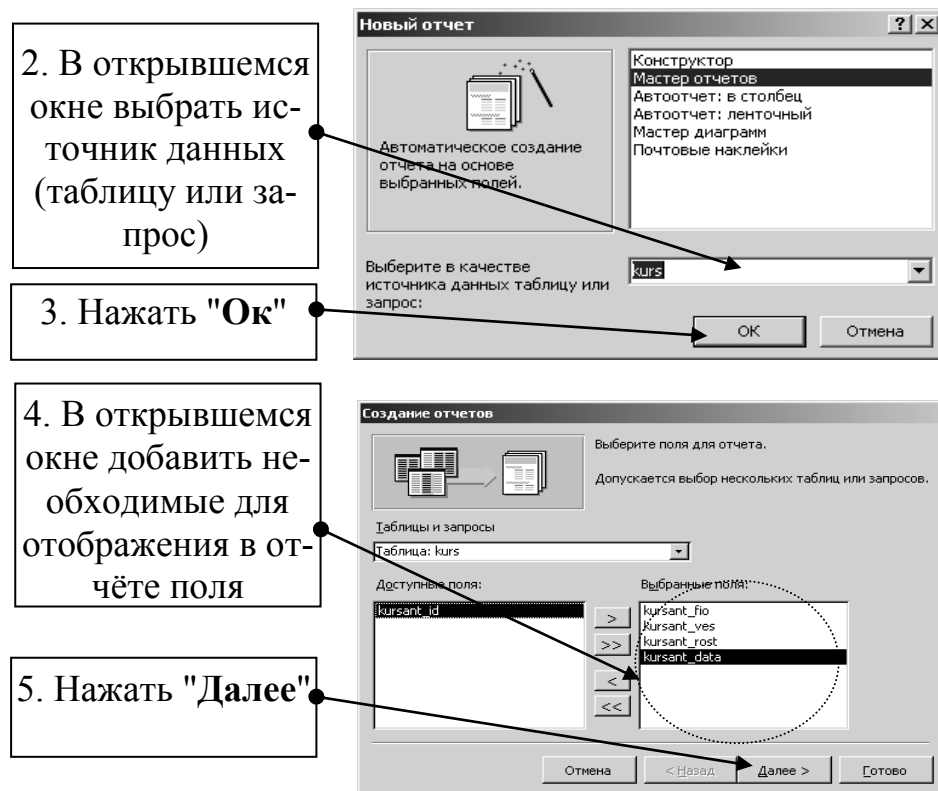


Рис.4.2. Этапы 2 - 5 создания отчета

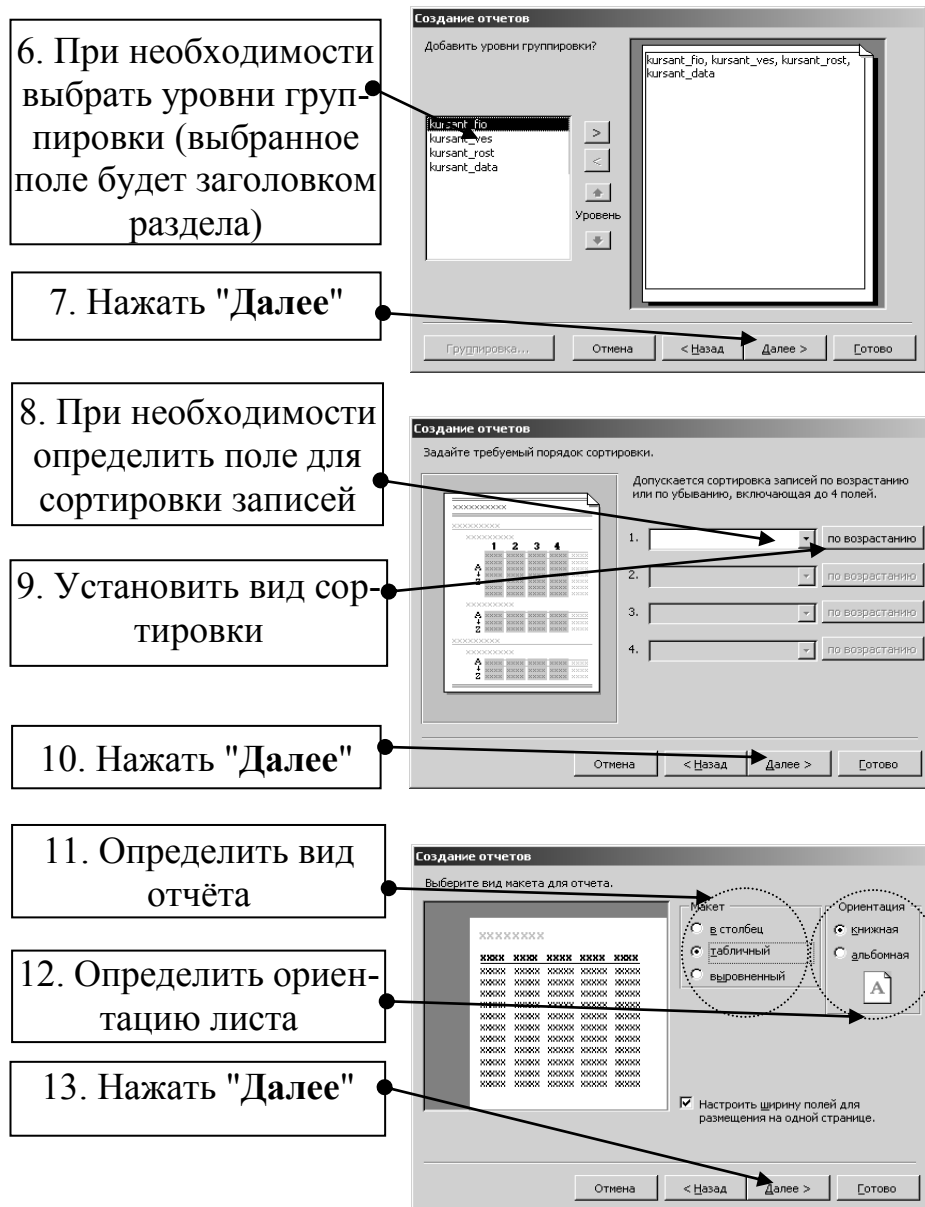


Рис.4.3. Этапы 6 - 13 создания отчета

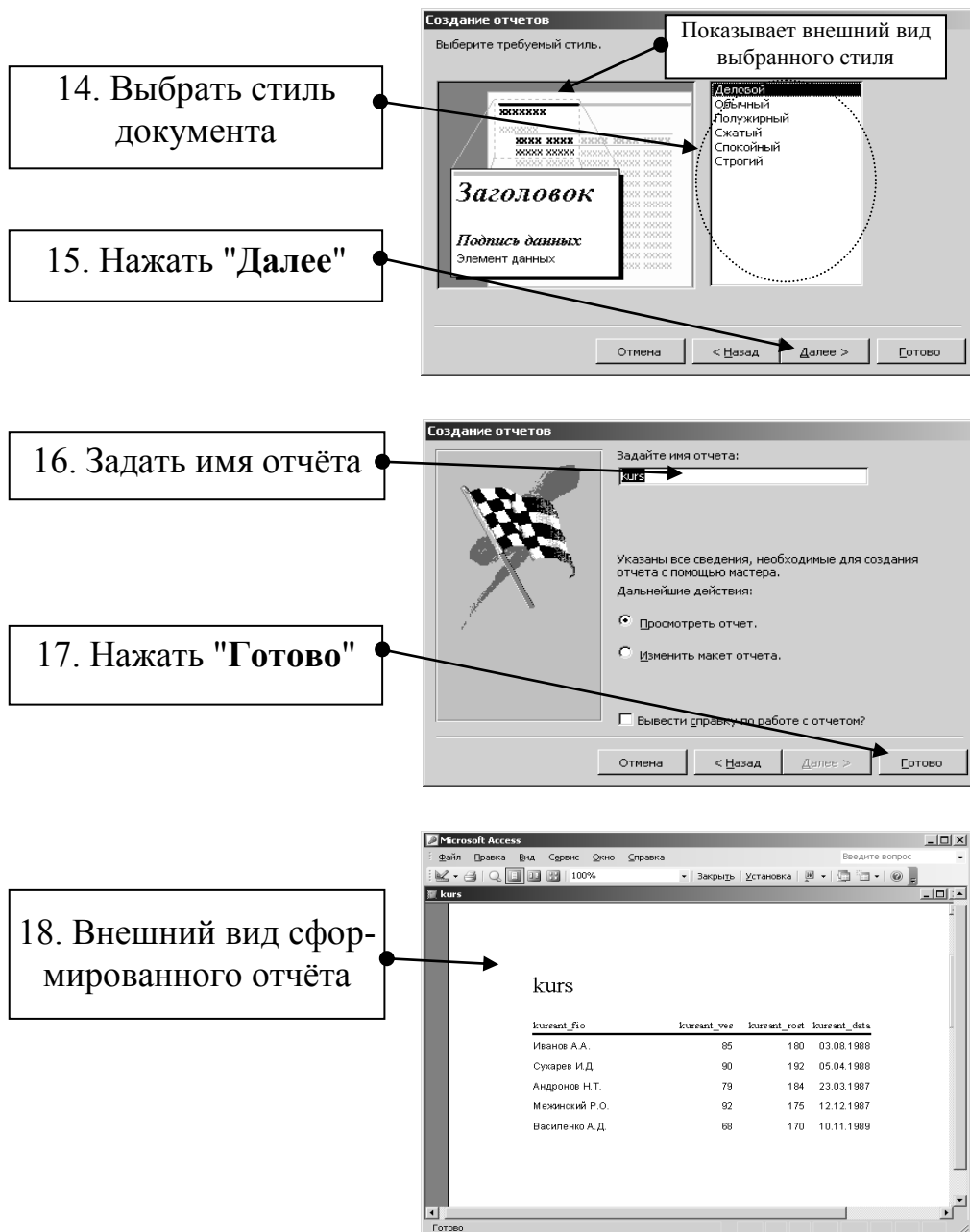


Рис.4.4. Этапы 14 - 18 создания отчета

Экспорт отчетов

Подготовленный отчет можно просмотреть лишь в базе данных, в которой он сохранен. Этот вариант неприменим для электронной почте или при использовании в других документах.

Преобразование отчета в формат документов Word:

1. Откройте отчет в любом режиме или просто выделите его в окне базы данных.
2. Выберите команду *Сервис, Связи с Office, Публикация в Microsoft Word*.

С помощью соседней команды меню *Сервис, Связи с Office, Анализ в Microsoft Excel* можно преобразовать отчет в электронную таблицу Excel. При выборе этой команды автоматически создается новая рабочая книга Excel, в которую помещается рабочий лист с представлением отчета в виде электронной таблицы. В качестве имени файла XLS используется имя отчета.

Задание 4.2 . Модификация отчета

Задание выполняется в соответствии с выданным индивидуальным вариантом.

Модификация отчёта

Сформированный с помощью Мастера Отчётов документ не всегда отвечает требованиям к оформлению документов в организации. Для приведения полученного документа к требуемому виду необходимо модифицировать полученный отчёт.

Для начала модификации отчёта необходимо перевести отчёт в режим конструктора (рис.4.5), выполнить модификацию, просмотреть результат работы и сохранить модифицированный отчёт.

Рассмотрим некоторые варианты модификации.

Форматирование документа

В режиме конструктора документ выглядит несколько непривычно, но на самом деле именно в этом режиме его удобно модифицировать. Все элементы отчёта являются визуальными компонентами, подготовленными для изменения их свойств пользователем.

На рис.4.6 представлен вид созданного отчёта в режиме конструктора.

Не все разделы отчёта выводятся одинаково. Так, все элементы, расположенные в разделе Область данных выводятся на экран столько раз, сколько записей выводится в источнике данных (таблице, или запросе), с которым связан отчёт.

Например, если в этот раздел поместить рисунок, или линию, то эти элементы будут выведены столько раз, сколько записей содержится в источнике данных.

Для изменения шрифта, которым выводятся данные, необходимо:

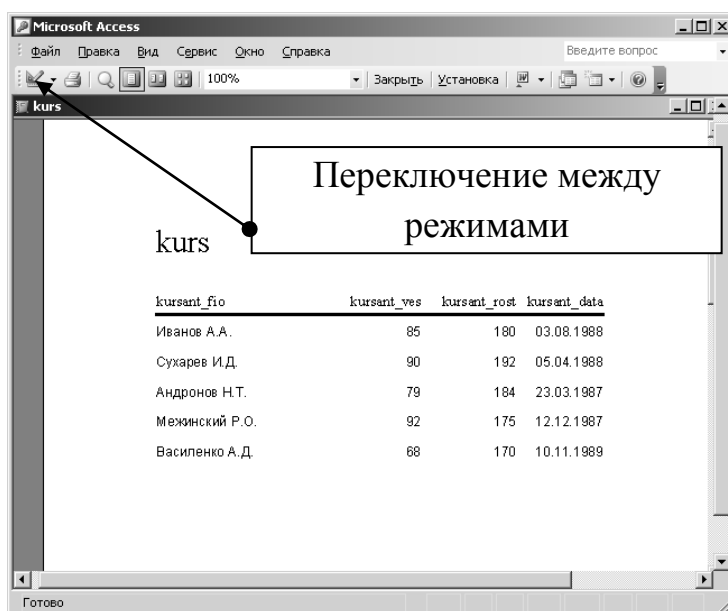


Рис.4.5. Переходы между режимом **Конструктора** и **Просмотра**

1. Выделить элемент отчёта (например, поле **kursant_fio** в области данных).
2. В панели средств форматирования текста выбрать требуемые: шрифт, размер шрифта, цвет и т.д.

В заголовке по умолчанию расположено имя таблицы, на основе которой создан отчёт (может быть изменено пользователем без нарушения связей)

Средства форматирования текста (похожи на аналогичные средства в текстовом процессоре)

Панель элементов (содержит элементы, связанные с БД и другие элементы)

Верхний колонтитул
По умолчанию расположены подписи полей, выводимых в отчёт. Так как это **не имена** полей, то вместо подписей можно вписать правильные заголовки (вместо **kursant_fio** – **ФИО курсанта**)

Область данных
Здесь расположены **ИМЕНА ПОЛЕЙ**, поэтому **изменять их нельзя**. При изменении имени поля БД не найдёт такого поля в таблице и выдаст ошибку. Разрешено только изменять параметры текста

Нижний колонтитул
Как правило, содержит вспомогательную информацию (можно изменять, добавлять, удалять)

Рис.4.6. Отчет в режиме конструктора

Добавление элементов оформления

Отчёт может выглядеть более читабельным, если внутри него расположить некоторые элементы, структурирующие выводимые данные: линии, рисунки, символы и др.

Для вставки перечисленных элементов используется **Панель Элементов** (рис.4.7). Работа с элементами интуитивно понятна. Чтобы вставить тот или иной элемент в отчёт необходимо:

1. Нажать кнопку с пиктограммой элементов на панели.
2. Щёлкнуть ЛКМ в том разделе отчёта, в который он должен быть вставлен.

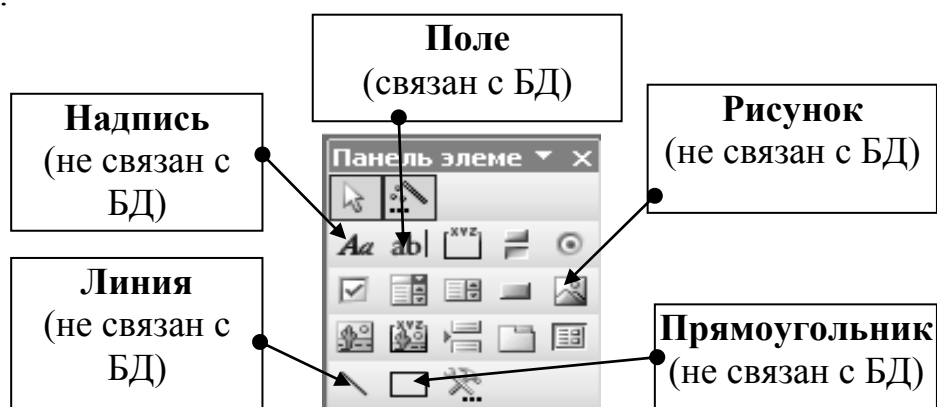


Рис.4.7. Панель элементов

Надпись – позволяет вставить любую надпись в отчёт в любом месте.

Линия – позволяет вставить линию любого размера и цвета в отчёт в любом месте.

Поле – для вывода в отчёте каких либо полей источника данных.

Рисунок – вставляет графические объекты в отчёт (например, угловой штамп организации, эмблемы, логотипы). Пользователь в диалоговом режиме должен указать файл с рисунком, который и помещается в отчёт.

Прямоугольник – аналогичен элементу **линия**.

На рис.4.8, 4.9 показан внешний вид отчёта, в который вставлен элемент **Линия** в различные разделы отчёта.

Продемонстрированные примеры показывают, каким образом оформить отчёт с помощью элементов, расположенных на Панели элементов.

"Тонкая настройка" каждого элемента производится с помощью панели свойств элементов. Для того, чтобы открыть эту панель, необходимо:

1. Выделить в конструкторе отчётов требуемый элемент (например, линию, установленную в отчёт);
2. Щёлкнуть ПКМ и выбрать **Свойства**;
3. В открывшейся **Панели свойств** произвести настройки.

Внешний вид отчёта, созданного на основе данных в таблице kurs и оформленного с использованием рассмотренных элементов представлен на рис.4.10.

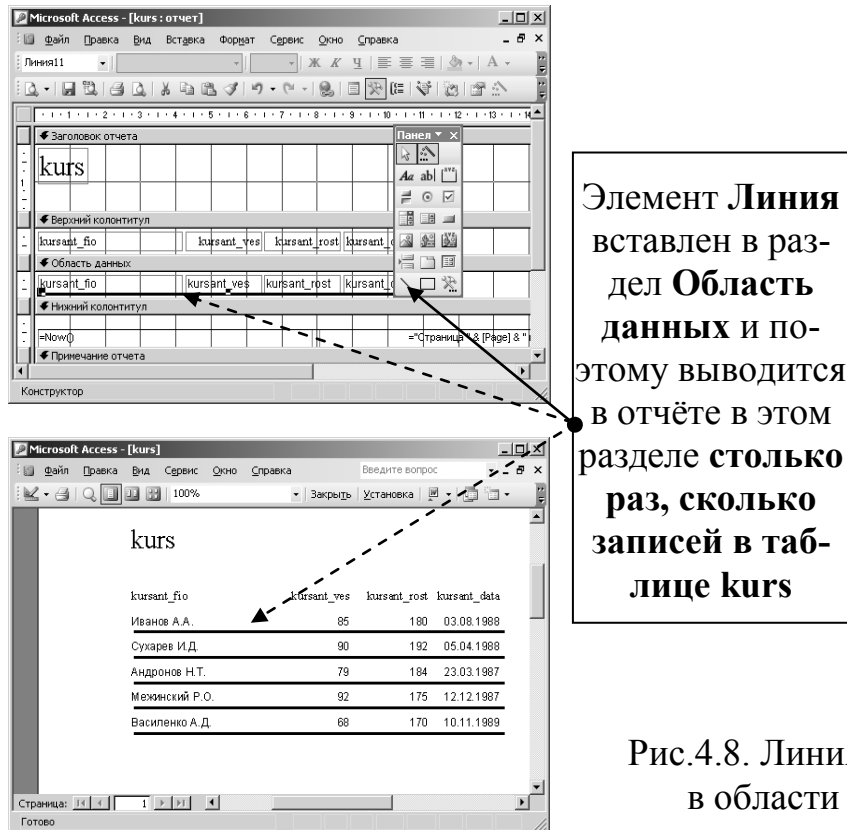


Рис.4.8. Линия вставлена в области данных

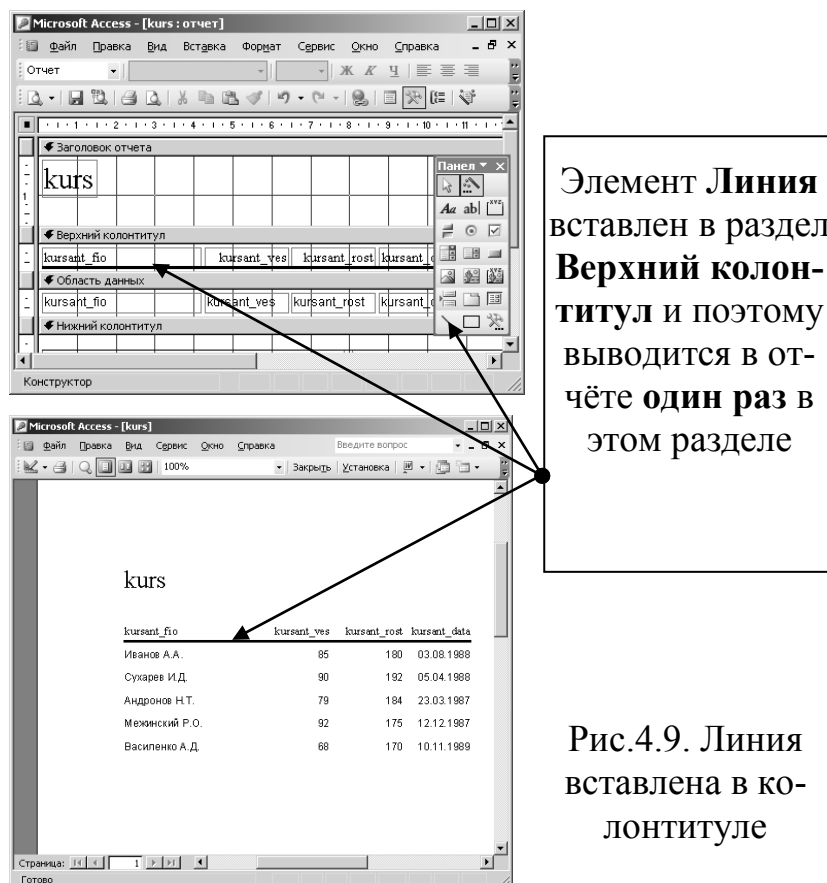


Рис.4.9. Линия вставлена в колонтитуле

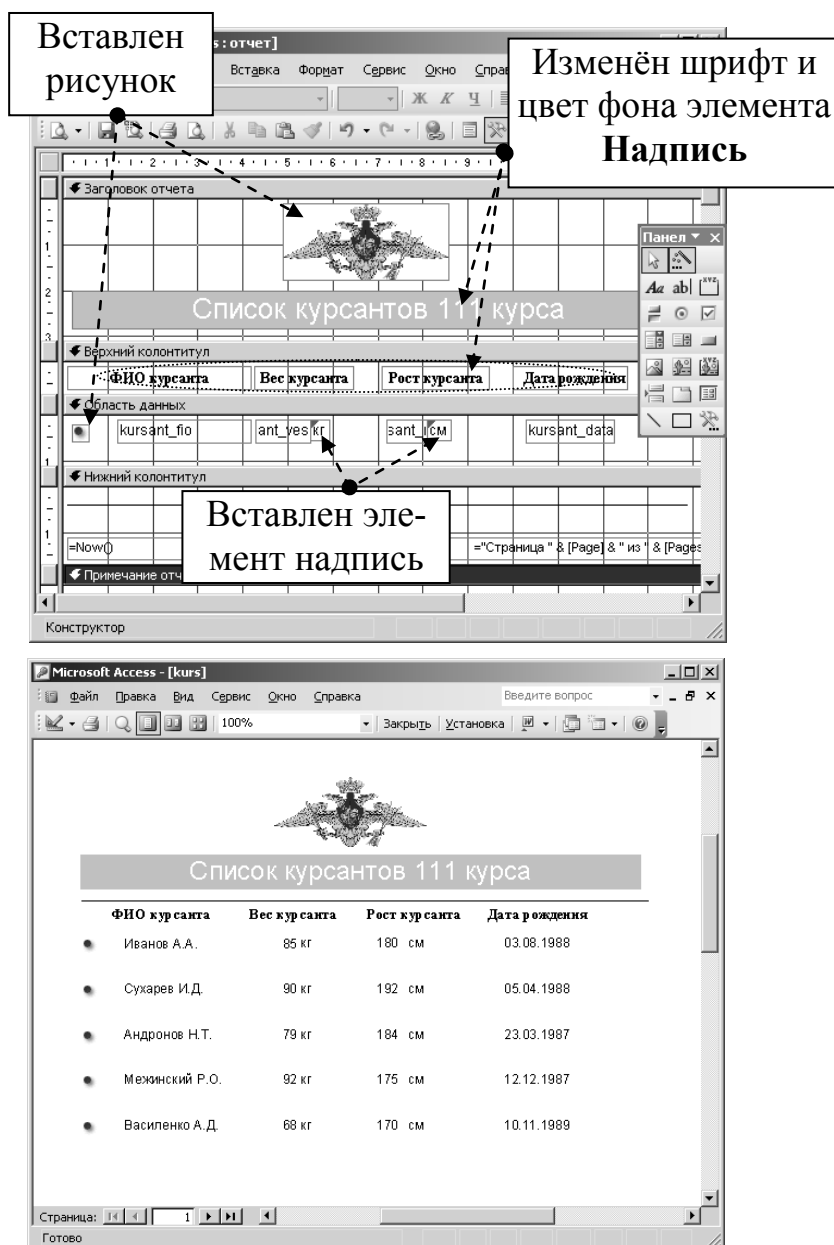


Рис.4.10. Пример оформления отчета

Вопросы для самопроверки

1. Каково назначение отчета?
2. Чем отчеты отличаются от форм?
3. Что является основой для получения данных для отчетов?
4. Что собой представляет автоотчет?
5. Какова технология разработки отчета с помощью мастера?
6. С какой целью используется экспорт отчета в другие форматы?
7. Какова технология экспорта в другие форматы?
8. Какие существуют возможности по форматированию документа отчета?
9. Какие дополнительные элементы могут быть использованы для повышения читабельности документа?
10. В каком случае создаются составные и многостраничные отчеты?