

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

---

Казанский государственный  
энергетический университет

Н. К. ПЕТРОВА

## **ОСНОВЫ РАБОТЫ В СУБД MICROSOFT ACCESS**

Методическое пособие к лабораторным работам, практическим занятиям, расчетному заданию и самостоятельной работе студентов по дисциплинам «Информатика», «Компьютерные технологии в науке и образовании», «Программные средства информатики», «Новые информационные технологии»

Казань 2004

УДК 516.62 (07)

ББК 32.81

П29

**Петрова Н.К.**

Основы работы в СУБД MICROSOFT ACCESS. Методическое пособие к лабораторным работам, практическим занятиям, расчетному заданию и самостоятельной работе студентов. Казань: Казан. гос. энерг. ун-т, 2004.

Рассматриваются основные вопросы, касающиеся работы пользователя с системой управления реляционными базами данных Microsoft Access. Содержатся как теоретические сведения, так и рекомендации и практические задания для освоения основных приёмов конструирования и функционирования базы данных. Предназначается студентам всех специальностей, осваивающим Access в рамках дисциплин «Информатика», «Компьютерные технологии в науке и образовании», «Программные средства информатики», «Новые информационные технологии», а также других информационно-ориентированных дисциплин. Может быть использовано при выполнении лабораторных, практических работ, расчетного задания и для самостоятельной работы студентов.

---

## ПРЕДИСЛОВИЕ

Данное пособие предназначено для изучения – самостоятельно или под руководством преподавателя – одной из популярных СУБД – Microsoft Access. Основная *цель* пособия – дать знания и укрепить навыки, достаточные для самостоятельного изучения СУБД. Последовательно переходя от темы к теме и применяя полученные знания на компактном примере разработки базы данных *Факультет*, можно овладеть возможностями, предоставляемыми программой Access.

Для начала работы необходимо знать и уметь:

- ✓ основные приемы работы в операционной системе Windows: работать с горизонтальным и контекстным меню, буфером обмена;
- ✓ текстовый редактор MS-Word и табличный процессор MS-Excel (краткое знакомство);
- ✓ основные приемы работы с графическими объектами MS Office;
- ✓ уверенно использовать манипулятор «мышь»: методы протягивания и перемещения мышью, щелчок, двойной щелчок и т.п.;

В тексте пособия приняты такие соглашения:

1. Элементы пользовательского интерфейса (а это все текстовые элементы, которые появляются на экране компьютера, например, команды, кнопки, опции, имена объектов Access) выделены таким стилем.

Если для выполнения какой-либо команды необходимо последовательно открыть несколько меню (например, в инструкции «Выполните команду **Формат>Интервал по вертикали>Сделать равным**»), то следует сначала щелкнуть на меню **Формат**, а затем поместить указатель на элемент **Интервал по вертикали** и, наконец, щелкнуть на команде **Сделать равным** открывающегося дополнительного меню.

2. *Курсивом* выделены новые термины, имена объектов базы данных и их полей, иногда – описание реакции программы на ваши действия.

3. Функции и операторы в выражениях выделены **таким шрифтом – Avg(1\_e1)**.

4. Значения данных, вводимых в ячейки таблиц, в поля контекстных меню, в выражения, выделяются **полужирным** шрифтом. Этим же шрифтом, иногда с подчеркиванием, выделяются наиболее важные элементы текста.

5. Названия и комбинации клавиш заключены в угловые скобки, например <Delete>.

## ОБЩИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

В самом общем смысле **база данных** – это один или несколько файлов, содержащие связанную (по смыслу) между собой информацию. Допустим, в одном файле хранятся все данные о поступивших в ВУЗ студентах (ФИО, № группы, Дата рождения и т.д.), в другом – результаты их успеваемости, в третьем – данные о каждой группе (кол-во студентов, староста, куратор и т.д.). Но что делать, если какие-то данные должны использовать несколько приложений или людей, и при этом нельзя допустить чтобы одни и те же данные корректировали одновременно? Когда возникают такие проблемы, то для разрешения их можно воспользоваться **системой управления базами данных** (СУБД).

Access – это **система управления базами данных (СУБД)**. Под системой управления понимается комплекс программ, который позволяет не только хранить большие массивы данных в определенном формате, но и обрабатывать их, представляя в удобном для пользователей виде. Access дает возможность также автоматизировать часто выполняемые операции (например: ввод и оценка результатов сессии, расчет стипендии, составление экзаменационных ведомостей и т.п.).

Access – это **реляционная СУБД**. Название «реляционная» связано с тем, что каждая запись содержит информацию, относящуюся только к одному объекту. В таких базах данные не дублируются, а связываются по определенным полям. Например, при указании куратора группы нецелесообразно упоминать все данные о кафедре, где он работает, – эти данные хранятся в отдельной таблице. Достаточно связать эти таблицы по имени кафедры. В реляционной базе данных сведения из каждого источника сохраняются в отдельной таблице, а затем между таблицами устанавливаются связи, что позволяет совместно работать с данными из нескольких таблиц.

### Основные элементы интерфейса Access. Объекты Access

Работа с объектами баз данных (БД) производится в окне БД, имеющем стандартный интерфейс Windows: горизонтальное меню, служебные кнопки и панели инструментов. В программе Access имеет несколько типов окон, каждый из которых соответствует объектам Access, например *окно базы данных* (рис.1). Структурно это окно разделено на две части: панель с *кнопками объектов* БД: таблиц, запросов, форм и т.д. После щелчка на такой кнопке открывается вкладка соответствующего объекта, например, на рисунке открыта вкладка *Таблицы* и выведен весь список имеющихся в БД таблиц, па-

нель инструментов, которая позволяет просматривать данные объекта и выбирать режим работы с объектом.

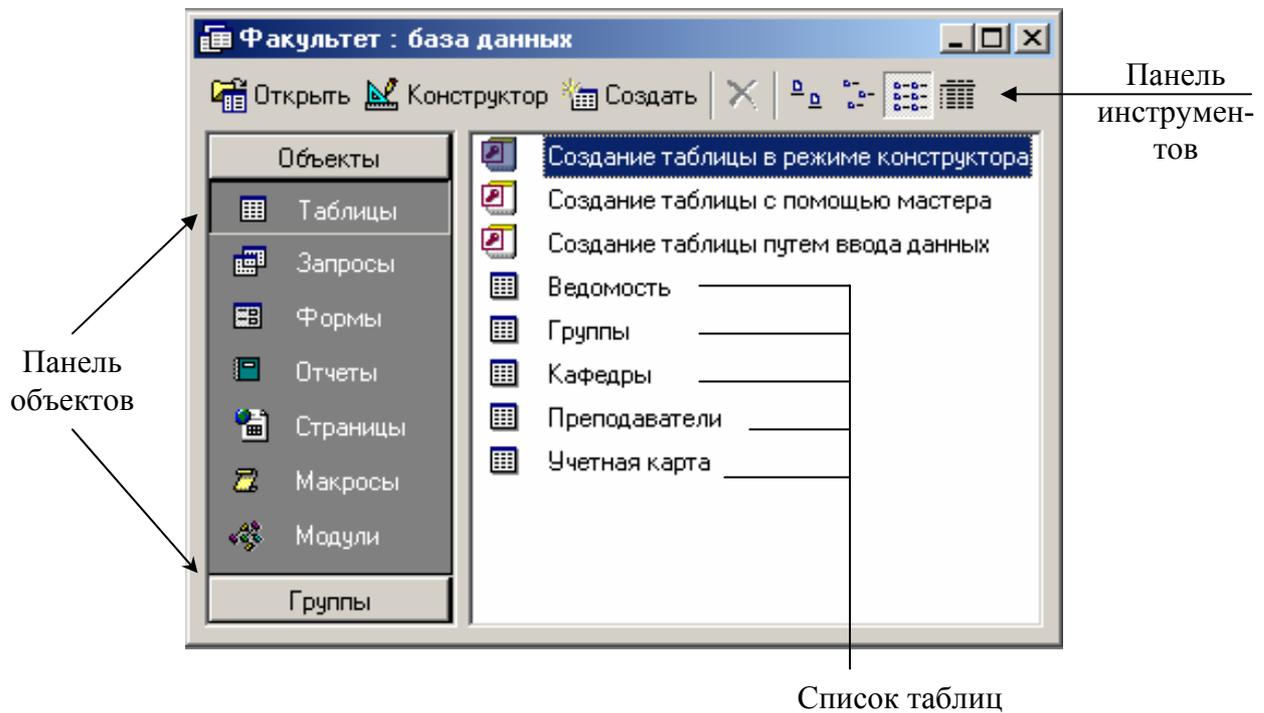


Рис.1. Окно базы данных

Создаваемые объекты базы данных – *формы, запросы, отчеты* – позволяют быстро и эффективно обновлять данные, получать ответы на вопросы, печатать отчеты, диаграммы и т.д.

*Запросы* позволяют проанализировать данные с помощью вычислений групповых операций и отбора данных, удовлетворяющих некоторым условиям.

Для добавления, просмотра и изменения данных одной или нескольких записей применяются *формы*. Они позволяют работать с данными сразу из нескольких таблиц.

*Отчеты* позволяют вывести на печать данные из полей таблиц, надписи, результаты групповых операций, диаграммы, рисунки и другие объекты, включая отчеты и постовые наклейки.

*Макросы* – это макрокоманды. Если какие-то операции с БД производятся очень часто, имеет смысл сгруппировать несколько команд в один макрос и назначить его выделенной комбинации клавиш.

*Модули* – программные процедуры, написанные на VBA. Если не хватает стандартных средств Access, можно расширить возможности системы, написав для этого необходимые модули.

## Основные свойства и параметры таблиц базы данных

База данных (БД) – это совокупность информации по определенной теме. Реляционная БД состоит из набора связанных между собой таблиц. Данные в таблицах должны быть организованы так, чтобы обеспечить объединение разнородной информации, исключить ее дублирование.

Все составляющие базы данных – таблицы, отчеты, запросы, формы и объекты – в Access хранятся в едином дисковом файле. Основным структурным компонентом базы данных является *таблица*.

Таблиц в базе данных может быть несколько. Каждая запись таблицы содержит всю необходимую информацию об отдельном элементе базы данных. Например, запись о студенте может содержать фамилию, имя, отчество, дату рождения, адрес и т.п. Как правило, каждая таблица посвящена определенной «теме», а точнее – определенному классу объектов, о которых хранится информация в БД. Так, например, если БД обслуживает учебный процесс ВУЗа, такими классами могут быть студенты, преподаватели, аудиторный фонд и т.п.

**Поле** составляет отдельный столбец в таблице. При разработке структуры таблицы, прежде всего, необходимо определить *названия полей*, из которых она должна состоять, *типы полей* и их *размеры*. Каждому полю таблицы присваивается уникальное (в пределах таблицы) имя, которое не может содержать более 64 символов. Далее надо решить, данные какого типа будут содержаться в каждом поле: текстовые, числовые и т.д. В каждое поле вносится информация об объекте, которая называется *значением поля*. Допустимое количество полей – 255.

**Запись**. Полные сведения об одном объекте (например, конкретном студенте или преподавателе) содержатся в одной *записи* таблицы. Запись – это совокупность значений всех полей для одного объекта. Количество записей в таблице соответствует количеству объектов: если в ВУЗе 100 студентов, то в таблице, содержащей список студентов, будет 100 записей. Общее количество записей ограничивается емкостью жесткого диска. Всем записям таблицы соответствует одно и то же множество полей, хотя в определенных полях любая запись может содержать пустые (Null) значения.

**Ключ**. Каждое поле в таблице уникально. В реляционных БД для каждой записи вводится уникальный идентификатор – *первичный ключ* – это уникальная характеристика для каждой записи в пределах таблицы. В качестве такого ключа выбирается одно или несколько полей записи. В качестве примера простого первичного ключа можно использовать поле ФИО таблиц, содержащих списки студентов или преподавателей.

**Понятие связей между таблицами**. Такой механизм, как первичный ключ таблицы, помимо однозначной идентификации записей, позволяет реа-

лизовать и связи между таблицами. Благодаря связям информация из одной таблицы становится доступной для другой.

Связь устанавливается за счет того, что в разных таблицах присутствуют поля с одинаковыми значениями. При этом необязательно, чтобы эти поля имели одинаковые имена, достаточно соблюдать однозначное соответствие значений. Для одной из таблиц таким полем может являться первичный ключ, а поле второй таблицы в этом случае играет роль внешнего ключа.

### Типы данных и их представление

Чтобы создать таблицу необходимо определить ее поля, типы данных этих полей и, иногда, некоторые дополнительные свойства этих полей. Всего поддерживается 10 основных типов данных. В таблице 1 перечислены все типы полей, определенные в Microsoft Access, и дается описание их размеров и сохраняемых в них значений.

Таблица 1. Типы данных и их представление

Тип данных. Содержимое поля	Размер
1	2
<p><b>Текстовый.</b> Текст или комбинация текстовых и числовых значений, например, адреса. Кроме того, в такие поля записывают числовые значения, для которых не предполагается выполнение расчетов, такие как телефонные или инвентарные номера или почтовые индексы.</p>	<p>До 255 символов. Максимальное число символов, которые можно ввести в поле, зависит от свойства «Размер поля»</p>
<p><b>Поле МЕМО.</b> Длинный текст, например, примечания или описания.</p>	<p>До 65536 знаков</p>
<p><b>Числовой.</b> Числовые данные, допускающие использование в математических вычислениях за исключением денежных расчетов (для последних определен тип "Денежный"). Конкретный числовой тип определяется значением свойства "Размер поля".</p>	<p>1, 2, 4 или 8 байт</p>
<p><b>Дата/время.</b> Значения даты или времени.</p>	<p>8 байт.</p>

1	2
<p><b>Денежный.</b> Денежные значения. Тип "Денежный" позволяет проводить вычисления без округления значений. Максимальная точность составляет 15 знаков слева от десятичной запятой и 4 знака справа от запятой.</p>	8 байт.
<p><b>Счетчик.</b> Уникальные последовательные (с шагом 1) или случайные номера, автоматически вставляемые при вставке записи.</p>	4 байт
<p><b>Логический.</b> Поля, которые могут иметь только одно значение из пары значений, таких как Да/Нет, Истина/Ложь или Вкл/Выкл.</p>	1 бит.
<p><b>Мастер подстановок.</b> Создает поле, позволяющее выбрать с помощью раскрывающегося списка значение из другой таблицы или из списка значений. При выборе данного значения в списке типов данных вызывается Мастер, заполняющий список подстановок.</p>	Размер, равный размеру ключевого поля, используемого для выполнения подстановок, обычно, 4 байт.
<p><b>Поле объекта OLE.</b> Объекты (например, документы Word, электронные таблицы Excel, рисунки и т.д. и т.п.), созданные в других приложениях, поддерживающих протокол OLE (Object Linking and Embedding), которые могут быть связаны или внедрены в таблицу Microsoft Access. Для вывода содержимого поля объекта OLE в форме или отчете необходимо создать <i>связанную рамку</i> объекта.</p>	До 1 Гбайт (ограничивается объемом диска).
<p><b>Гиперссылка (Для Access 2000 и выше).</b> Обеспечивает связь с WEB-страницей, расположенной в интернет, внутренней сети или локальном компьютере. Позволяет переходить из текущего поля к информации в другом файле. При выборе этого поля Access автоматически запускает Web-браузер и отображает указанную страницу.</p>	

Для числовых полей в Access предусмотрено несколько подтипов числовых данных. При выборе подтипа числового поля следует учитывать размер предполагаемых данных и, по возможности, использовать минимальное количество байтов. Варианты доступных подтипов перечислены в таблице 2.

Таблица 2. Подтипы полей для типа Числовой

Размер поля	Диапазон значений	Десятичные знаки	Размер памяти в байтах
Байт	От 0 до 255	Нет	1 байт
Целое	От – 32 768 до 32 767	Нет	2 байт
Длинное целое	От – 2 147 483 648 до 2 147 483 647	Нет	4 байт
С плавающей точкой (4 байт)	От – 3,402823E38 до 3,402823E38	7	4 байт
С плавающей точкой (8 байт)	от – 1,7976931348623E308 до 1,7976931348623E308	15	8 байт
Код репликации	Глобальный уникальный идентификатор	Нет	16 байт

### Общее проектирование и цель создания БД

В качестве наглядного примера проектирования базы данных будем использовать БД *Факультет*, работа с которой отражает упрощенную модель учебного процесса ВУЗа.

Во - первых, допускаем, что ВУЗ имеет 3 кафедры: **КФ1, КФ2 и КФ3**.

Во - вторых, на каждом из трех курсов обучается по две группы. Названия группам будем давать по принципу, принятому во многих ВУЗах: начало отражает принадлежность к определенной структуре ВУЗа и является, как правило, текстовой составляющей, последние символы в названии, как правило, – цифры, соответствующие году поступления в ВУЗ. Поэтому в нашей модели будем использовать такие названия групп, как А-*N* и Б-*N*, где *N* – последняя цифра года поступления, например, для 2003/4 учебного года А-3 и Б-3 – группы I курса, А-2 и Б-2 – II курса, А-1 и Б-1 – I курса. Номера студенческих билетов комбинируются из года поступления и порядкового номера студента на данном курсе, например, 1 - 10 (10-й студент 2001 года поступления).

В - третьих, для того, чтобы не занимать много времени на занесение однотипной информации в таблицы, принимаем, что количество студентов в группе не больше 5 – 6 человек.

В-четвертых, принимаем, что учебный план не меняется в течении всех лет обучения, т.е. форма и количество зачетов и экзаменов фиксировано: на каждом курсе сдается один зачет и два экзамена. Каждый учебный год – это один учебный семестр.

В - пятых, номера аудиторий комбинируются из имени корпуса ВУЗа (текстовая составляющая) и номера аудитории внутри данного корпуса – во многих ВУЗах принята именно такая система нумерации.

Принятые ограничения не изменяют качественной сути модели, а затрагивают лишь ее количественную сторону.

Итак, база *Факультет* может содержать следующие таблицы: *Учетная карта* (табл. 3), *Ведомость* (табл. 4), *Группы* (табл. 5), *Преподаватели* (табл. 6), *Кафедры* (табл. 7).

Таблица *Учетная карта* аналогична учетной карте Отдела кадров, которая заполняется как при поступлении студента в ВУЗ, так и во время его обучения (поощрения, награды, выговоры и т.д.). Для упрощения ввода в поле ФИО будем задавать только Фамилию и инициалы. Номер билета будет иметь текстовый тип, хотя можно определить его и как числовой.

Таблица 3. Предполагаемые поля таблицы *Учетная карта*

Имя поля	Тип данных	Размер	Описание
ФИО	Текстовый	25	Фамилия И.О. студента
Нбил	Текстовый	4	№ студ. билета (например: 1-01)
Нгр	Текстовый	3	№ группы
Пол	Текстовый	1	Женщина – Ж, Мужчина – М
ФормаОплаты	Текстовый	1	Бюджетная – Б, Коммерческая – К
Льготы	Логический	1	Есть льготы (сирота, инвалид и т.д.) – Да, в противном случае – Нет
ДатаРожд	Дата/время		Дата рождения
Телефон	Числовой	Длинное целое	Контактный телефон
Фото	OLE		Фотография студента
Примечание	МЕМО		Особые отметки: характеристика, награды, поощрения, выговоры и т.п.

Таблица *Ведомость* содержит информацию об успеваемости каждого студента. Для простоты будем считать, что учебный план фиксирован: количество предметов, по которым сдаются экзамены и зачеты не меняется в течение всех лет обучения. В эту учебную таблицу введем лишь несколько полей с формальными названиями предметов и выпускной квалификационной работы.

Таблица 4. Предполагаемые поля таблицы *Ведомость*

Имя поля	Тип данных	Размер	Описание
1	2	3	4
ФИО	Текстовый	25	Фамилия И.О. студента
1_зач	Текстовый	1	Отметка: зачтено – з, не зачтено – н
1_Э1	Числовой	Целое	Оценка дисциплины 1-го экзамена I курса
1_Э2	Числовой	Целое	Оценка дисциплины 2-го экзамена I курса
2_зач	Текстовый	1	Отметка: зачтено – з, не зачтено – н
2_Э1	Числовой	Целое	То же, что и 1_Э1, но для II курса
2_Э2	Числовой	Целое	То же, что и 1_Э2, но для II курса
3_зач	Текстовый	1	Отметка: зачтено – з, не зачтено – н
3_Э1	Числовой	Целое	То же, что и 1_Э1, но для III курса
3_Э2	Числовой	Целое	То же, что и 1_Э2, но для III курса
Диплом	Числовой	Целое	Оценка за выпускную работу (диплом, госэкзамен)

Конечно, можно вместо формальных названий предметов включать конкретные предметы из вашего учебного процесса. Отметим, что поле *ФИО* – это и есть первичный ключ таблиц *Учетная карта* и *Ведомость*.

Таблица 5. Информация об общих характеристиках группы помещается в таблице *Группы*

Имя поля	Тип данных	Размер	Описание
1	2	3	4
№гр	Текстовый	3	№ группы – идентификатор группы

1	2	3	4
Куратор	Текстовый	20	Фамилия И.О. куратора
Староста	Текстовый	20	ФИО старосты
КолСтуд	Числовой	Целое	Количество студентов в группе
Курс	Числовое	Целое	Курс, на котором обучается группа, вычисляется по номеру группы

Следует отметить, что последние два поля таблицы *Группы* можно не включать на этапе построения таблицы, т.к. их значения являются производными от значений других полей: *КолСтуд* зависит от количества записей в таблице *Учетная карта* с данным *№г*, а номер курса – поле *Курс* – может быть вычислено, например, по последней цифре группы и текущего года. Поэтому эти поля мы отнесем к разряду *вычисляемых полей* и включим их с помощью специального запроса. Поле *№г* является первичным ключом таблицы *Группы*.

Таблица 6. Информацию о кафедрах факультета помещаем в таблицу *Кафедры*

Имя поля	Тип данных	Размер	Описание
Кафедра	Текстовый	3	Название кафедры
ЗавКаф	Текстовый	25	Фамилия И.О. заведующего кафедрой
Науд	Текстовый	5	№ аудитории
Т/ф	Числовой	Длинное целое	Номер телефона кафедры

Таблица 7. Информацию о профессорско - преподавательском составе помещаем в таблицу *Преподаватели*

Имя поля	Тип данных	Размер	Описание
ФИО	Текстовый	60	Фамилия, имя, отчество преподавателя
Кафедра	Текстовый	3	Кафедра, на которой работает преподаватель
Должность	Текстовый	10	Должность
Телефон	Числовой	Длинное целое	Контактный телефон преподавателя

## Определение связей между таблицами

Существует несколько типов связей:

связь "один-к-одному";

связь "один-ко-многим";

связь "многие-ко-многим".

Связь "**один-к-одному**" представляет собой простейший вид связи данных, когда первичный ключ таблицы является в то же время внешним ключом, ссылающимся на первичный ключ другой таблицы. Такую связь бывает удобно устанавливать тогда, когда невыгодно держать разные по размеру (или по другим критериям) данные в одной таблице, или для того чтобы не занимать оперативную память, если эти данные используются сравнительно редко. Так в БД *ФАКУЛЬТЕТ* данные о студентах разбиты на две таблицы: в одной из них хранятся результаты экзаменационных сессий (*Ведомость*) – достаточно часто используемая таблица в учебном процессе, в другой (*Учетная карта*) – подробная информация о каждом студенте, которая требуется для обработки в течение года относительно редко. В этом случае устанавливается связь "**один-к-одному**" по ключевому полю ФИО.

Связь с отношением "**один-ко-многим**" является наиболее часто используемым типом связи между таблицами. В такой связи каждой записи в таблице "А" могут соответствовать несколько записей в таблице "В", а запись в таблице "В" не может иметь более одной соответствующей ей записи в таблице "А". Например, между таблицами *Группы* и *Учетная карта* существует отношение "**один-ко-многим**": в каждой группе много студентов, но один студент может учиться только в одной группе. Связь "**один-ко-многим**" реализуется парой "*внешний ключ – первичный ключ*", т.е. когда определен внешний ключ, ссылающийся на первичный ключ другой таблицы.

В таблицах, связанных с отношением "**многие-ко-многим**", одной записи в каждой таблице могут соответствовать несколько записей в другой таблице. Для установления отношения "**многие-ко-многим**" необходимо создать третью (связующую) таблицу и включить в нее ключевые поля из обеих таблиц. В нашем примере отсутствуют связи такого типа. Продемонстрировать такую связь можно на примере работы библиотеки. В одной таблице содержатся данные о книгах (каждая книга может быть в нескольких экземплярах), в другой – данные о читателях. Каждый читатель может иметь на руках несколько книг, книга одного и того же названия может быть у не-

скольких читателей. Поэтому создается промежуточная таблица – абонентская карта (рис 2).

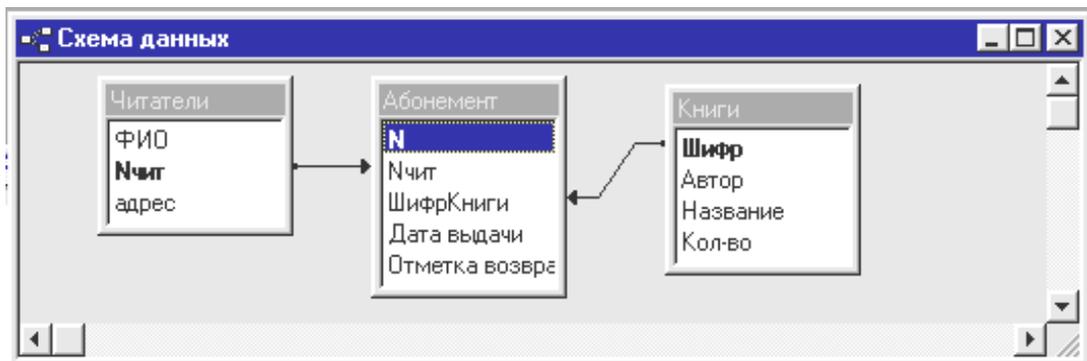


Рис 2. Реализация связи "многие-ко-многим"

*Итак, мы рассмотрели основные этапы проектирования базы данных. Факультет, ориентированной на реализацию многих функций, поддерживающих «идеализированный» учебный процесс на факультете ВУЗа.*

### Контрольные вопросы

1. Что такое база данных? Какие достоинства имеет СУБД Access ?
2. Перечислите основные элементы окна базы данных.
3. Перечислите основные объекты СУБД Access, их назначение и взаимосвязь.
4. Для чего служит структура "таблица"?
5. Что такое поле таблицы? Перечислите три его основных параметра.
6. Какого типа данные могут содержаться в поле?
7. Что такое запись таблицы? Есть ли какие-либо ограничения на количество записей в таблице? Чем поле отличается от записи?
8. Что такое первичный ключ? Как он выбирается?
9. Как реализуются связи между таблицами? Для чего они устанавливаются?
10. Могут ли MEMO-поля и поля объекта OLE использоваться в качестве первичного ключа?
11. На примере БД *Факультет* поясните, какие поля и почему выбраны в качестве ключевых.
12. Чем отличается ключевое поле от остальных полей?
13. Как соотносятся записи таблиц при реализации связи типа а) один-к-одному; б) один-ко-многим; в) многие-ко-многим?

## Лабораторная работа № 1 «СОЗДАНИЕ НОВОЙ БАЗЫ ДАННЫХ»

### Цель работы

Данная лабораторная работа предназначена для приобретения начальных навыков работы с таблицами реляционной базы данных, а именно:

- а) создавать таблицы, задавая полям различные типы данных,
- б) вводить и редактировать данные в таблицах,
- в) организовывать связи между таблицами.

### Создание пустой базы

База данных – это файл специального формата, содержащий информацию, структурированную заданным образом. Все объекты базы данных Access хранятся в одном файле с расширением .mdb. Файл новой БД занимает около 96 Кбайт содержит скрытые системные таблицы, в которые в дальнейшем будет заноситься информация обо всех других объектах БД. По мере создания новых объектов и заполнения самой БД размер этого файла будет расти, однако на протяжении всего периода существования БД вся эта совокупность элементов Access рассматривается как один физический файл.

С точки зрения пользователя, создание пустой базы данных – это некая формальная процедура, извещающая систему о ваших намерениях в дальнейшем заполнить этот файл реальными таблицами, запросами и другими объектами.

Создание объектов можно выполнять с помощью *Мастера* – специальной программы, ускоряющей создание структуры объекта, – или вручную. Для лучшего понимания структуры базы данных мы, в основном, будем ориентироваться на работу вручную – в режиме Конструктора (рис. 3).

### ЗАДАНИЕ 1. Создание таблиц в режиме Конструктора

1. Запустите программу Access.
2. В появившемся диалоговом окне, установите флажок напротив опции новая база данных и подтвердите операцию.
3. В следующем диалоговом окне вы должны выбрать папку для базы данных и задать ее имя (*Факультет*).

*На экране появится окно с шестью вкладками, это и есть ваша база данных (она пока пустая).*

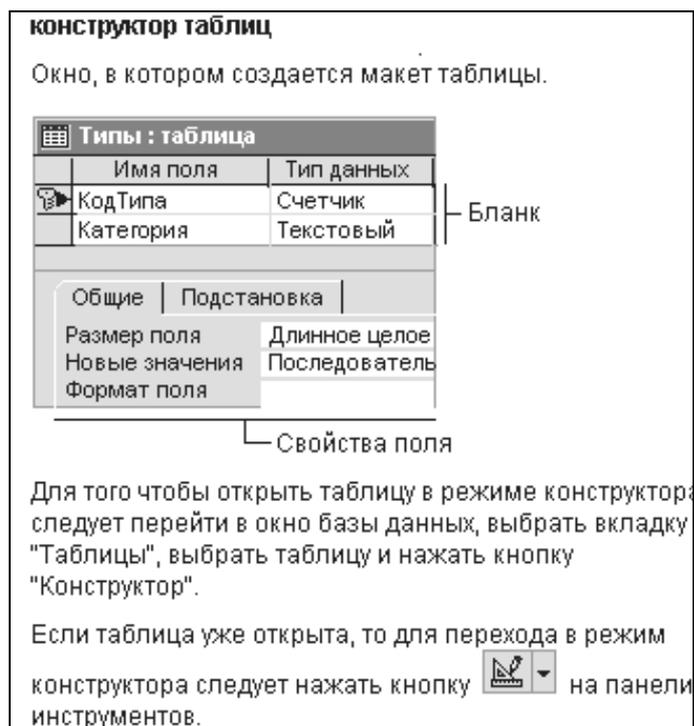


Рис. 3. Инструкция для открытия и макет таблицы в режиме Конструктора

4. Перейдите на вкладку Таблицы.

5. Щелкните мышью по кнопке Создание таблицы в режиме конструктора

Основываясь на спроектированных нами таблицах (табл. 3 - 7), будем создавать поля со свойствами, описанными в этих таблицах. Начнем с таблицы *Учетная карта*.

6. В столбец Имя поля занесите имена столбцов будущей таблицы (рис.4). Напомним, что при этом нельзя использовать некоторые символы, в том числе пробелы, точки и запятые. В столбце Тип данных выберите (используя кнопку вызова списка ) тип данных. А то, что заносится в столбец Описание затем появляется, в виде комментариев, в строке состояния.

7. Названия полей даны латинскими буквами. Чтобы из-за этого не возникало неудобств при работе с таблицей, каждому полю задайте подпись на русском языке. Для этого надо перейти на вкладку Свойства поля и задать в строке Подпись сокращенное название поля на русском языке.

Свойство Подпись – это строковое выражение длиной до 2048 символов. Оно определяет тот текст, который будет выводиться в заголовке столбца в режиме таблицы. По умолчанию в качестве заголовка выводится имя по-

ля. В подписи можно использовать любые символы, в том числе пробелы и точки.

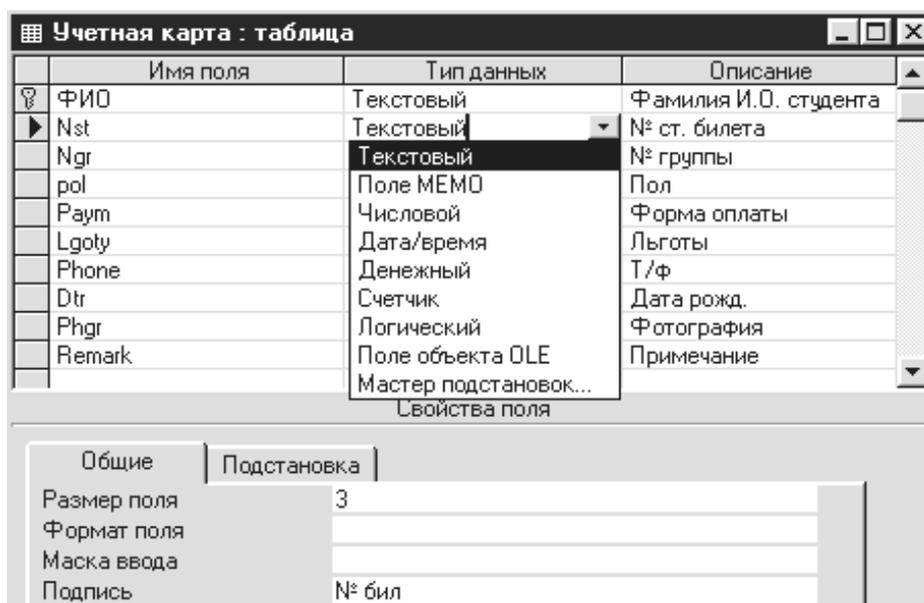


Рис. 4. Название и типы полей таблицы *Учетная карта*.

Показан разворачивающийся список типов полей по кнопке вызова

8. Здесь же, в строке *Размер поля*, на основе табл. 3 задать размер для числовых и текстовых полей.

9. После ввода полей и типов данных желательно задать ключевое поле. Для этого необходимо щелкнуть правой клавишей мыши по полю *ФИО* и, в появившемся меню, выполнить команду *Ключевое поле*.

11. После ввода всех полей, их типов, размеров и названий закройте окно Конструктора и, при запросе о сохранении, задайте имя *Учетная карта*.

12. Аналогично создайте еще четыре таблицы, структура которых показана на рис. 5. Длину полей, подписи к ним и другие свойства полей выбирайте в согласии с таблицами 4 – 7. Имена полей, их типы и назначения показаны на рисунке 5.

13. Закройте таблицы с сохранением, дав им соответствующие имена.

14. Сохраните данную базу данных на диске или дискете для последующей работы. Для этого необходимо закрыть ACCESS и скопировать файл базы одним из средств Windows.

## Изменение типа полей и реорганизация таблиц

В режиме конструктора все изменения, касающиеся структуры объектов, необходимо сохранять. Изменения, касающиеся структуры таблиц, желательно выполнять на этапе проектирования, еще до ввода данных, когда

еще не созданы формы и запросы. В противном случае изменение типов полей может повлечь за собой потерю данных, о чем Access предупреждает при попытке сохранить изменения.

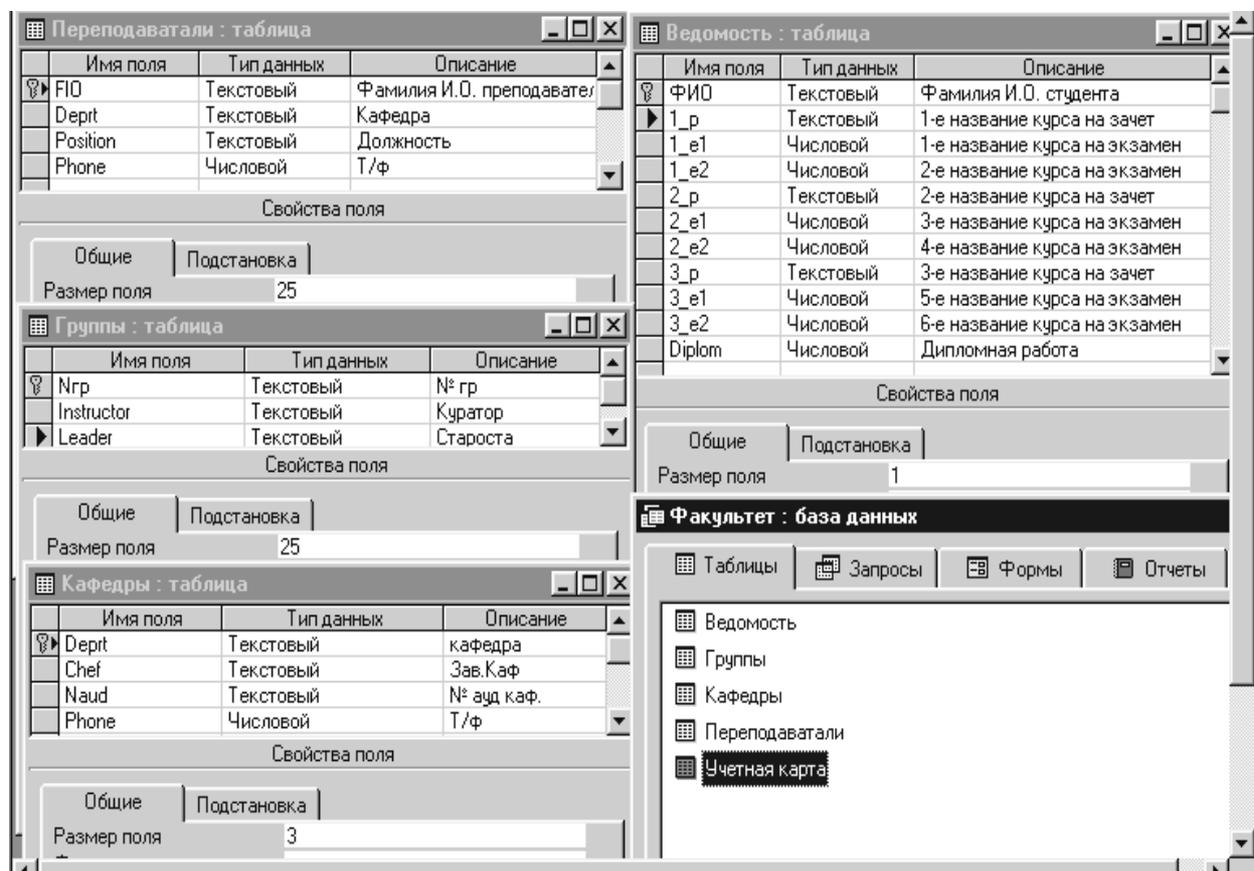


Рис. 5. База данных *Факультет* и структура ее четырех таблиц

- Чтобы *изменить имя поля* таблицы, нужно дважды щелкнуть на теге имени поля и ввести новое имя.
- Чтобы *изменить тип поля* данных для уже существующего поля, надо щелкнуть на кнопке раскрывающегося списка в столбце Тип Данных и выбрать новый тип данных.
- Для *добавления или удаления полей* можно воспользоваться контекстным меню, установив курсор на требуемой строке в списке полей, и выбрать соответствующую команду.
- При *копировании полей* копируется только их структура, а не сами значения. Для этой процедуры можно использовать средства Windows для работы с буфером обмена.
- Для *удаления поля* так же используются традиционные средства Windows (буфер обмена или клавиша <Delete> на выделенном поле).

Обратим внимание на одну особенность всех баз данных. Таблица баз данных не является самостоятельным документом. Сама база – это документ. Ей соответствует файл на диске, можно сделать его копию. Структура таблиц входит в состав общего файла базы данных наряду с запросами, формами и другими объектами. При изменении структуры таблицы СУБД всегда выдает запрос на сохранение изменений.

Но содержание таблиц – это совсем другое дело. Его нельзя сохранить принудительной командой или, наоборот, отказаться от его сохранения. Все изменения в таблицах сохраняются автоматически *в режиме реального времени*. Режим реального времени означает, что, пока мы работаем с таблицей, происходит ее непрерывное сохранение. Как только заканчивается ввод данных в одно поле и происходит переход к следующему полю, данные немедленно записываются на жесткий диск.

Экспериментируя с таблицами, надо знать, что все изменения, которые вносятся в их содержание, имеют необратимый характер. Нельзя что-то изменить, удалить, а потом отказаться от сохранения и вернуться к исходному варианту.

### **Ввод и редактирование данных в таблице**

Когда структура таблицы создана и откорректирована, можно приступать ко вводу данных. Начнем заполнение с таблицы *Учетная карта*. Чтобы добавить в таблицу первую запись, надо выполнить следующие действия.

1. Двойным щелчком (или с помощью кнопки **Открыть**) откройте таблицу – она будет представлена в режиме таблицы, а указатель будет помещен в первое поле первой записи.

2. Щелкнуть в первом поле пустой записи и вести значение для данного поля. Как только в текущую строку будут вводиться данные, в таблице появится новая пустая запись. (Если первым в таблице стоит поле с типом *Счетчик*, то оно заполняется автоматически). Для перехода в новое поле можно пользоваться клавишами <Tab>, (<Enter> или <→>).

3. Процедуру ввода повторять до тех пор, пока не будут введены данные для всех полей таблицы. Когда введено последнее поле и нажата клавиша <Tab>, (<Enter> или <→>), точка вставки переместится в первое поле новой записи.

Если понадобится внести изменения в заполненную таблицу: отредактировать данные в полях, удалить или добавить записи, то можно пользоваться всеми приемами редактирования Windows: вставка и замена символов в указанном месте поля, работа с буфером обмена и командами горизонтального и контекстного меню. Приемом *протягивание* можно выделять группу записей или несколько столбцов (для копирования или удаления), можно выделять часть таблицы и переносить выделенные фрагменты через буфер обмена в рабочие листы Microsoft Excel или таблицы Microsoft Word.

Данные для ввода в таблицу приведены на рис. 6. Для того, чтобы *уменьшить или увеличить ширину столбца*, можно использовать стандартный прием Windows – *перемещение*: установить курсор на границе столбца, зафиксированной левой кнопкой мыши переместить ее в требуемую сторону.

## **ЗАДАНИЕ 2. Ввод и редактирование данных в полях таблицы**

1. Откройте таблицу *Учетная карта*. и введите в таблицу первые две три записи (рис. 6). Обратите внимание новые записи в Access «добавляются» только внизу таблицы, в так называемую *временную новую запись*. Разместить новую запись между уже существующими или в начало таблицы – нельзя. (Чтобы визуально расположить записи в другом порядке, например, по алфавиту, можно воспользоваться функцией *Сортировка* из пункта горизонтального меню *Записи*).

2. Обратите внимание, что новая запись отмечена «звездочкой» (✳). Но когда указатель перемещается в первое поле новой записи, «звездочка» изменится на значок текущей записи (▶). Когда в поле новой записи вводятся данные, указатель меняет свой вид на пишущий карандаш (✎). Рис. 6 иллюстрирует именно этот момент.

3. Обратите внимание, что в логическое поле *Льготы* значение «Да» не надо вводить: достаточно щелкнуть в заготовленной клетке левой кнопкой мыши – появится галочка (☑) вместо «Да».

4. Обратите внимание, что для числовых полей (*T/φ*) можно вводить только цифры: при попытке ввести другие символы Access выдает предупреждение о несовпадении типа поля и данных.

Учетная карта : таблица										
	ФИО	№ бил	№ гр	Пол	Форма опл.	Льготы	Т/ф	Дата рожд.	Ф/гр	
	Андреева О.С.	1-01	А-1	ж	Б	<input type="checkbox"/>	561483	12.07.83		золотая мед
	Антонова Т.И.	1-04	Б-1	ж	К	<input type="checkbox"/>	326454	07.11.83		победитель р
	Арсентьев В.Р.	1-02	А-1	м	Б	<input checked="" type="checkbox"/>		05.10.82		многодетная
	Безногов С.Ю.	2-12	Б-2	м	К	<input type="checkbox"/>	865425682	21.09.84		
	Белов С.П.	1-03	А-1	м	К	<input type="checkbox"/>	710850	29.01.82		
	Бережной П.Р.	2-07	А-2	м	К	<input type="checkbox"/>	448532	08.01.83		
	Верхов А.В.	1-05	Б-1	м	К	<input checked="" type="checkbox"/>	357849	25.03.83		Служба в Чех
	Галкин И.Д.	2-08	А-2	м	К	<input type="checkbox"/>	438162	04.04.84		
	Денисов И.О.	1-06	Б-1	м	К	<input type="checkbox"/>	743285	20.02.82		
	Драгунов А.А.	2-09	Б-2	м	Б	<input type="checkbox"/>	824768924	24.05.84		
	Елисеева И.В.	2-10	Б-2	ж	Б	<input checked="" type="checkbox"/>	643091	08.10.83		Сирота
	Леонтьев М.В.	2-14	А-2	М	Б	<input checked="" type="checkbox"/>	557829	25.01.85		Служба в Чех
	Романова С.Л.	2-12	Б-2	ж	Б	<input checked="" type="checkbox"/>	945751	31.08.83		
	Семенов А.Л.	2-15	А-2	м	Б	<input type="checkbox"/>	570860	23.12.83		
	Торова Р.К.	2-16	Б-1	ж	Б	<input checked="" type="checkbox"/>	348064	02.03.80		многодетная
	Як.	2-11	Б-2	м	Б	<input type="checkbox"/>		25.03.84		
*						<input type="checkbox"/>				

Рис. 6. Таблица *Учетная карта* открыта в режиме таблицы для ввода записей. При вводе новой записи значок изменяется на изображение пишущего карандаша, а внизу таблицы появляется новая временная запись

5. Для поля *Дата рожд.* по умолчанию предусмотрен краткий формат даты **дд.мм.гг.** Если в качестве разделителя использовать запятую, то Access автоматически преобразует ее в точку.

6. Для поля типа OLE (*Ф/гр*) оставьте пустое место. Просмотр полей типа OLE возможен только в режиме *Формы*. При изучении работы с *Формами* рассмотрим, как внедрять в таблицу поля такого типа.

7. В поле *Примечание* можно вводить любую текстовую информацию.

8. Исправьте в первой записи фамилию **Андреева О.С.** на **Андропова О.С.** Перейдите в следующее поле.

9. Отмените сделанное изменение, используя кнопку *Отменить*  на панели инструментов или клавиши **Ctrl+Z**.

10. После ввода 2 - 3 записей из таблицы рис. 6 введите в новую запись данные о себе.

11. Выделите только что введенную запись (щелкнув на кнопке выбора записи или поместив курсор в любое место записи, вызвать из горизонтального меню *Правка>Выделить* запись, затем с помощью контекстного или горизонтального меню *вырезать* запись в буфер обмена. На предупреждение Access об опасности удаления, ответьте «Да».

12. Вставьте вырезанную запись с помощью команды горизонтального меню Правка>Добавить из буфера. Обратите внимание, что обычный способ вставки из буфера с помощью кнопки Вставка  в данном случае не работает.

13. Снова выделите эту же запись и удалите ее, нажав клавишу <Delete> на клавиатуре, или выполнив команду Правка>Удалить запись.

14. Закройте таблицу, используя кнопку Закреть  или ту же команду горизонтального меню.

## Форматирование данных и контроль ввода информации

При заполнении таблиц БД очень важно обеспечить достоверность и корректность данных в таблицах, а также представить их в максимально удобном для эффективной работы виде. Ключевая роль в решении этой задачи принадлежит таким свойствам полей, как *Формат поля*, *Маска ввода* и *Условие на значение*.

### Форматирование

Форматирование данных позволяет определить способ отображения на экране вводимых данных. Изменив формат, можно представить данные в более привлекательном, информативном и доступном виде. Форматы применяются строго для определенных типов данных и доступны для любого типа, за исключением типа Поле объекта OLE.

В Access используются две разновидности форматов: *встроенные* и *специальные*. Последние создаются с помощью специальных символов. Для каждого типа полей существуют свои настройки форматирования и свои специальные символы.

Форматы для текстовых и MEMO- полей создаются с помощью четырех специальных символов (табл. 8).

Таблица 8. Символы форматирования для текстовых и MEMO-полей

Символы	Описание	Примеры форматов (вводимые данные – вид на экране)
1	2	3
@	Прототип символа. В форматированной строке вместо @ отображается введенный символ или, если символ не указан, выводится пробел.	<p>@@@-@@-@@@@ (465073799 – 465-07-3799)</p> <p>@@@@@@@@@@@ (465-07-3799 – 465-07-3799)</p>

1	2	3
<	Переводит все символы на нижний регистр.	иванов иванов ИВАНОВ иванов Иванов иванов
>	Переводит все символы на верхний регистр.	иванов ИВАНОВ ИВАНОВ ИВАНОВ Иванов ИВАНОВ

Так, например, для полей, содержащих фамилии и инициалы, во всех таблицах БД *Факультет* можно указать формат > (рис. 7), и тогда при вводе информации можно не следить за регистром: все фамилии и инициалы будут отображаться заглавными буквами.

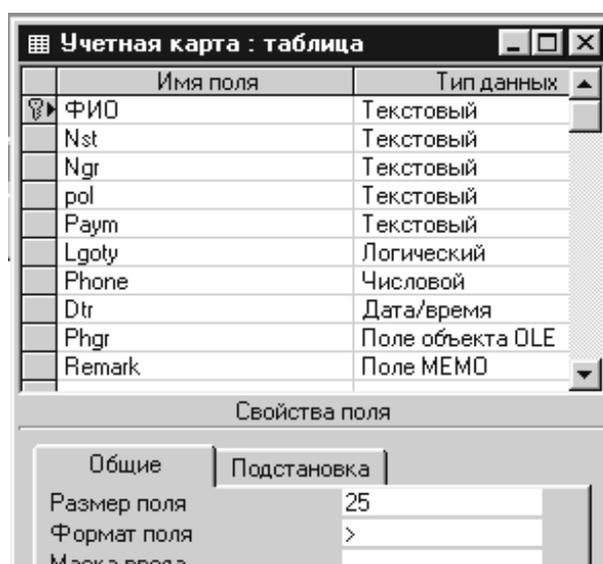


Рис. 7. Для поля ФИО указан формат >, в результате чего данные этого поля выводятся прописными буквами

Для форматирования числовых и денежных полей программа Access предлагает ряд встроенных форматов. Все они включены в список опции Формат поля (рис. 8) в виде пар: название формата и образец его использования.

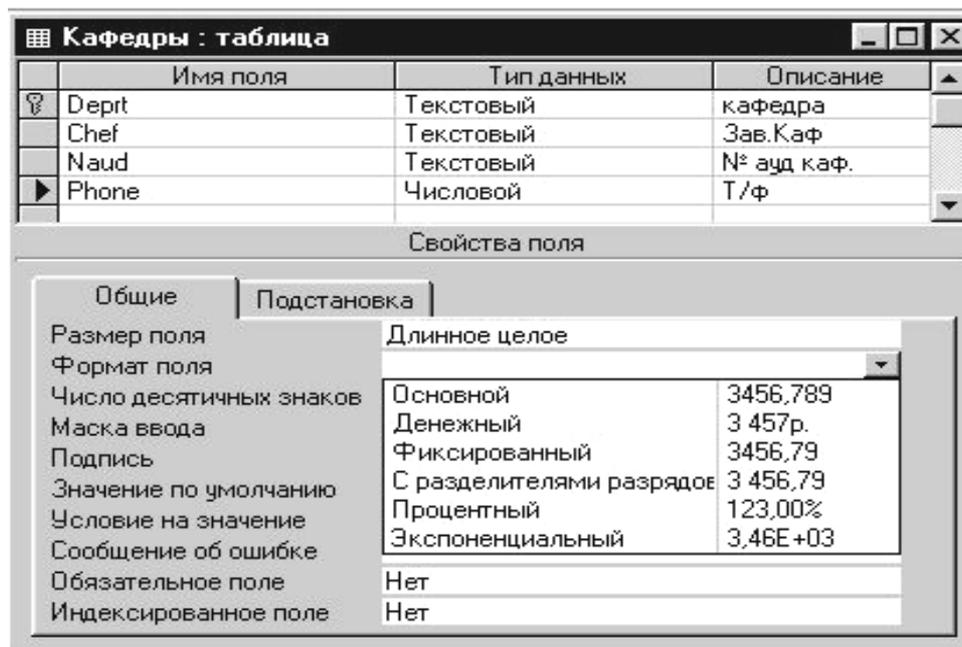


Рис. 8. Список числовых форматов, предлагаемых в окне конструктора для свойства `Формат поля`

Наиболее распространенные *специальные форматы* создаются с помощью символов, представленных в табл. 9.

Таблица 9. Символы, используемые в специальных форматах для числовых и денежных полей

Символ	Описание
.	Десятичный разделитель.
,	Разделитель тысяч.
0	Прототип разряда. Независимо от числа символов 0, целая часть числа будет отображаться полностью. Например, число 345,6 в формате 00000,00 выглядит как 00345,60
#	Прототип разряда. Независимо от числа символов #, целая часть числа будет отображаться полностью. Например, число 345,6 в формате #####.## выглядит как 345,6
%	Процентный формат. Значение умножается на 100 и выводится со знаком процентов.
E- или e-	Экспоненциальная нотация с выводом знака минус перед отрицательным показателем и без знака перед положительным показателем. Используется вместе с другими специальными символами, например, 0.00E-00.

Список встроенных форматов для полей типа *Дата/время* показан на рис. 9.

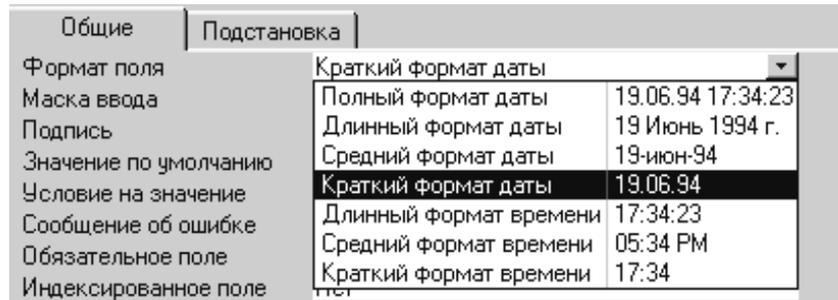


Рис. 9. Список встроенных форматов полей типа *Дата/время*

Здесь для каждого формата представлено его название и пример использования. Выбрать подходящий формат можно, щелкнув, на кнопке разворачивающегося списка.

### Использование масок ввода и проверка значений

Если форматирование помогает варьировать представление данных в полях таблицы, то такие свойства, как *Маска ввода* и *Условие на значение* предусмотрены, чтобы не допустить попадание в таблицу некорректных данных.

*Маска* позволяет при вводе информации показывать в поле готовый шаблон, в который заносятся данные. Если, например, тип поля задан как текстовый, но в него должны входить только цифры (например, номер студенческого билета), можно оговорить это условие с помощью маски.

Естественен вопрос, чему отдать предпочтение: форматированию или маске ввода? Принципиальное отличие этих опций в следующем: свойство *Формат поля* применяется уже после того, как данные введены и сохранены в поле, а свойство *Маска ввода* действует еще до размещения информации в таблице, регламентируя ввод данных в конкретном поле.

Свойство *Формат поля* не влияет на значения, хранимые в таблице. Свойство *Маска ввода* используется в дополнение к формату или вместо него. Если для поля одновременно задаются формат отображения и маска ввода, то Access использует маску ввода при вводе или редактировании данных и свойство *Формат поля* при отображении данных после сохранения записи. В таблице 10 представлены некоторые часто используемые маски ввода и соответствующие им значения.

Таблица 10. Символы, используемые для создания масок ввода

Символ	Описание
1	2
0	Цифра (от 0 до 9, ввод обязателен; символы "плюс" [+] и "минус" [-] не допускаются).
9	Цифра или пробел (ввод не обязателен; символы "плюс" и "минус" не допускаются).
#	Цифра или пробел (ввод не обязателен; пустые символы преобразуются в пробелы, допускаются символы "+" и "-").
L	Буква (A-Z или A-Я, ввод обязателен).
?	Буква (A-Z или A-Я, ввод не обязателен).
A	Буква или цифра (ввод обязателен).
&	Любой символ или пробел (ввод обязателен).
C	Любой символ или пробел (ввод не обязателен).
. , : ; - /	Десятичный разделитель и разделители тысяч, значений даты и значений времени.
</>	Указывает перевод всех следующих символов на нижний/верхний регистр.
\	Указывает ввод любого следующего символа как постоянного символа. Используется для ввода специальных символов как постоянных символов (например, \A выводится как символ "A").
Пароль	Значение "Пароль", заданное для свойства "Маска ввода" создает поле ввода пароля. Любой символ, введенный в поле, сохраняется как символ, но отображается при вводе звездочкой (*).

Порядок действий при создании маски несложен. Первый шаг – продумать, какой должна быть маска и какие функции она будет выполнять. Второй шаг – ввести подходящий вариант маски в поле «Маска ввода». Например, для ввода номера группы может быть создана следующая маска: **L**"-"**9**. Для ввода пола студента и формы оплаты за обучение допустим только один буквенный символ, поэтому используем простую маску **L**.

### **ЗАДАНИЕ 3. Задание масок и форматов ввода для таблиц БД Факультет**

1. Откройте таблицу *Учетная карта* в режиме Конструктора.

2. Перейдите к полю *Nst* (*№ бил*) и на вкладке *Общие*>*Маска* задайте маску **9\-99** : все символы *№ билета* – цифры, есть обязательный разделитель “-” (тире).

3. Для поля *Ngr* (*№ гр*) задайте маску **L"-9** : первый символ – буква, второй – обязательное тире и третий символ – цифра.

4. Для поля *Dtr* (*Дата рожд.*) задайте *Краткий формат даты*, выбрав его из раскрывающегося списка на вкладке *Общие*> *Формат поля* (рис. 9).

5. Сохраните изменения, внесенные в структуру таблицы, выполнив команду *Файл*>*Сохранить* или щелкнув на кнопке *Сохранить* , и закройте таблицу.

6. Откройте в режиме *Конструктора* таблицу *Группы*. и задайте маску **L\-9** для поля *Ngr* (*№ гр*). Закройте таблицу с сохранением.

7. Откройте в режиме *Конструктора* таблицу *Кафедры* и задайте маску **"КФ"-9** для поля *Deprt* (*Кафедра*).

8. Для поля *Naud* (*№ ауд*) задайте маску **L\-999**. Что она означает?

9. Закройте таблицу *Кафедры*.

10. Откройте таблицу *Преподаватели* и задайте для поля *Deprt* (*Кафедра*) ту же маску, что и в пункте 6. Закройте таблицу с сохранением.

### Проверка вводимых значений

Самый надежный способ обезопасить таблицу от ввода ошибочных или ложных данных – обеспечить их проверку на соответствие определенным требованиям непосредственно при вводе. Хотя часть работы по проверке допустимости значений данных программа *Access* выполняет автоматически, для обеспечения целостности данных следует предусмотреть хотя бы простейшие операции контроля.

Характер проверки задается в поле *Условие* на значение области *Свойства поля*, и если требование условия на значение не соблюдается, выдается сообщение об ошибке. Текст такого сообщения можно определить в опции *Сообщение об ошибке*.

*Условие на значение* вводится в виде *выражения* (команды для вычисления значений). В выражениях используются обычные символы арифметических действий: + (сложение), – (вычитание), \* (умножение), / (деление). Такие символы называются *операторами*. В любом выражении присутствует, как минимум, один оператор. Значения, над которыми производится действие, называются *операндами*.

Кроме математических, для сравнения числовых величин используются *операторы сравнения*: > (больше чем), < (меньше чем), а также = (равно), <> (не равно), <= (меньше или равно), >= (больше или равно). Для операций над логическими величинами предусмотрены *логические операторы*, например, And, Or, Not.

Для текстовых значений существует оператор конкатенации (сцепления) &, с помощью которого можно объединить несколько строк символов в одну строку, и оператор Like, который проверяет соответствие строкового объекта заданному шаблону.

В выражениях можно использовать различные *функции*, например, математические, статистические или получения даты и времени. До и после даты, входящей в выражения, ставятся символы #.

#### **ЗАДАНИЕ 4. Запись условий на значение в полях таблиц БД Факультет**

1. Откройте в режиме конструктора таблицу *Учетная карта*. Выберите поле *pol(пол)*
2. Перейдите на вкладку *Общие>Условие на значение* и задайте условие **"м" Or "ж"**
3. В области ввода *"Сообщение об ошибке"* укажите текст **Введите м или ж.**
4. Для поля *Dtr(Дата рожд.)* задайте условие **>=#01.01.70#**, предполагая, что возраст студента дневного отделения не может превышать 33 лет (на 2003 год поступления).
5. Перейдите к таблице *Ведомость*.
6. Значение для поля, содержащего оценку за экзамен, например, для поля *1\_e1* должно лежать в интервале от 2 до 5, поэтому задайте условие **>=2 and <=5** и соответствующее сообщение об ошибке **Введите целое число от 2 до 5.**
7. Задайте такое же условие на значение для поля *1\_e2*.
8. Для поля *2\_e1* и *2\_e2* ограничьте набор допустимых значений только пятью элементами Null (пустое поле), 2, 3, 4, 5, т.е. запишите условие **Null Or 2 Or 3 Or 4 Or 5**, и дайте соответствующий комментарий в строку *Сообщение об ошибке*.
9. Для поля *3\_e1* и *3\_e2* запишите то же условие, используя оператор **In(Null;2;3;4;5).**

10. Выберите любой способ для записи условий на поле *Diplom* (*Диплом*), а также на поля, содержащих данные о зачетах *1\_p*, *2\_p* и *3\_p*: (з–зачтено, н–не зачтено).

11. Проверьте все таблицы, содержащие фамилии – *Учетная карта*, *Преподаватели*, *Группы*, *Ведомость* – на наличие условия перевода всех символов на верхний регистр (рис. 7).

12. Обратите внимание, что можно работать с несколькими открытыми таблицами, переходя от одной таблицы к другой с помощью вкладки горизонтального меню *Окно*.

13. Можно расположить все таблицы, так, чтобы они были видны одновременно. Для этого на вкладке *Окно* выберите из разворачивающегося меню соответствующую форму расположения таблиц: *Сверху вниз*, *Каскадом* или *Слева направо* (рис. 10). В таком представлении таблиц можно перейти от одной таблицы к другой, щелкнув левой кнопкой в любом месте окна нужной таблицы.

14. Закройте все таблицы с сохранением структуры.

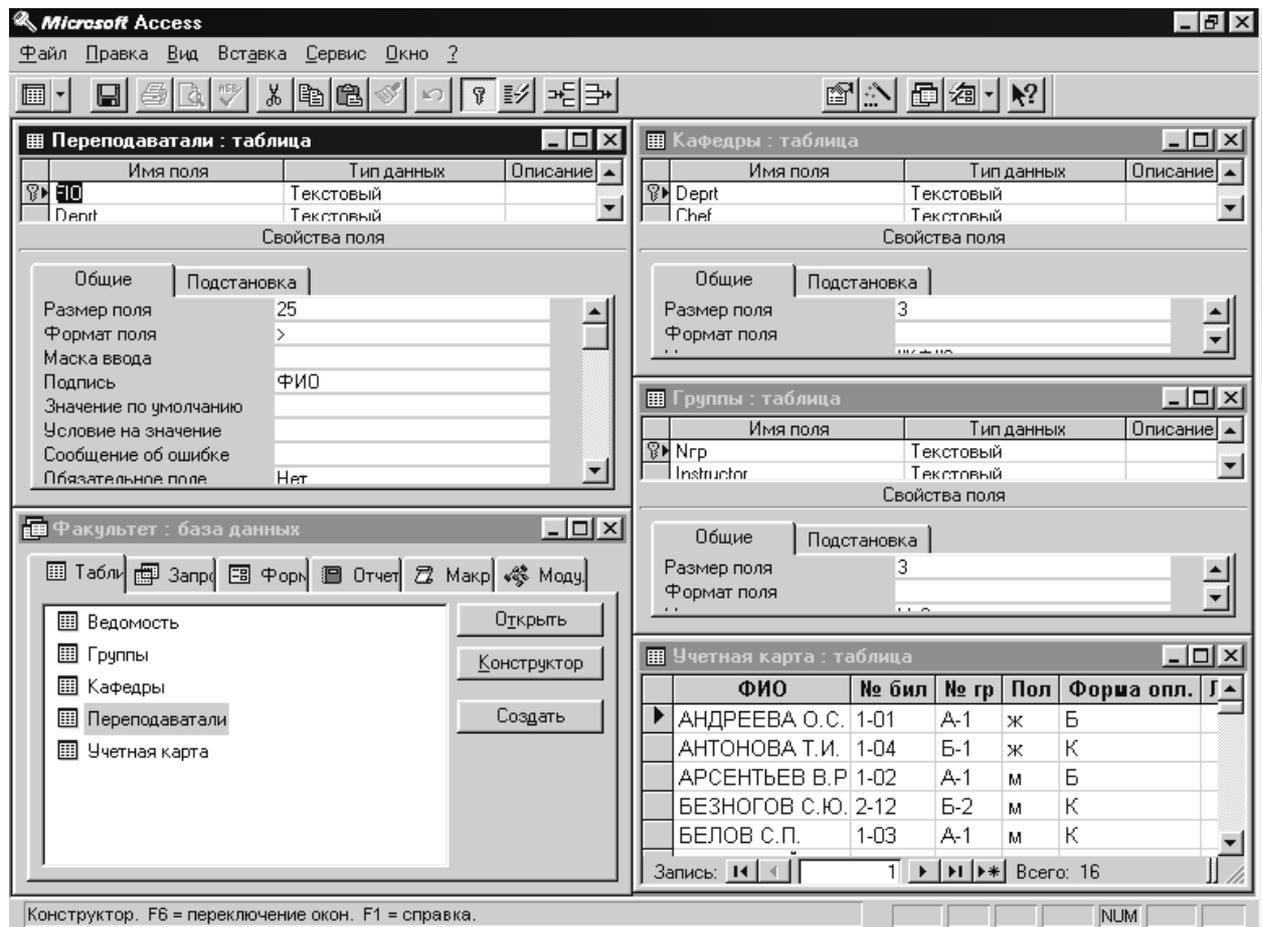


Рис.10. Расположение одновременно открытых таблиц Слева направо.

## Создание простого списка подстановки

Достаточно часто бывают случаи, когда набор вводимых значений ограничен определенным списком, например, для таблицы *Преподаватели*, требуется ввести значения в поле *Position (Должность)* пять допустимых значений: **аспирант**, **ассистент**, **доцент**, **ст.преп.** (старший преподаватель), **профессор**. Проверку условия на значение для этого поля можно выполнить, создав простой *список подстановки*.

### ЗАДАНИЕ 5. Создание простого списка подстановок для таблицы *Преподаватели*

1. Откройте таблицу *Преподаватели* в режиме Конструктора. Выделите поле *Position*, а затем щелкните на вкладке Подстановка в области Свойства поля.

2. Щелкните на поле Тип элемента управления, а затем выберите из раскрывающегося списка пункт Поле со списком или Список. Теперь на вкладке Подстановка появился новый набор опций, которые необходимо определить.

3. Щелкните на поле Тип источника строк и выберите из раскрывающегося списка пункт Список значений. Оставьте неизменным принятое для свойства поля Присоединенный столбец и Число столбцов значение 1.

4. Для свойств Ширина столбцов введите значение 1 см, а для свойства Ширина списка – Авто. Так как число опций списка равно 5, укажите значение 5 для свойства Число строк списка (по умолчанию - 8).

5. Для свойства Ограничиться списком укажите значение Да, тем самым ограничив число возможных значений для этого поля только теми, которые будут перечислены в списке подстановки.

6. Теперь введите элементы будущего списка в поле Источник строк в следующем виде: **аспирант; ассистент; доцент; ст.преп.; профессор**. Элементы ввода разделены символами точки с запятой: рис. 11.

8. *Должность* можно открыть список подстановки (рис. 12 а).

9. Откройте в режиме Конструктора таблицу *Учетная карта* и создайте список подстановки для поля *Форма Опл.* с двумя столбцами согласно рис. 12 б). Сохраните изменения.

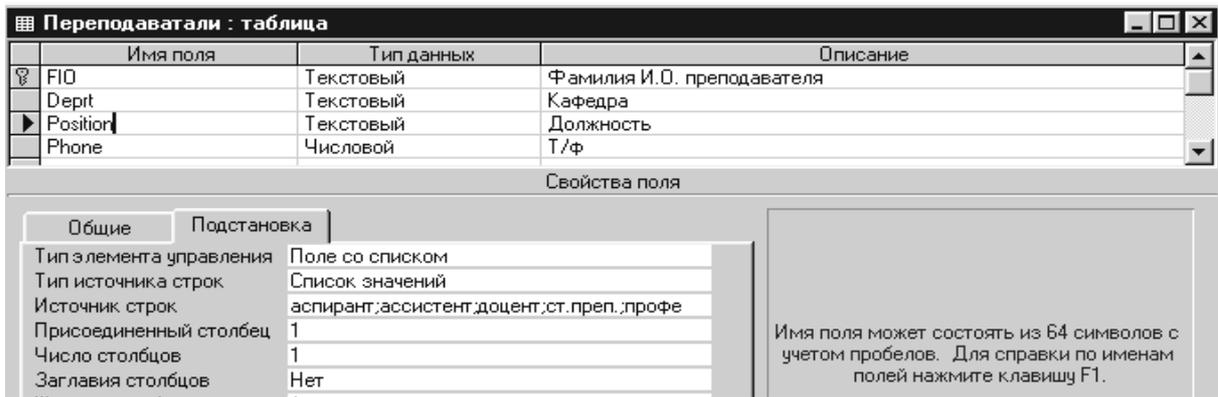


Рис. 11. Для поля *Должность* создается список подстановки.

Все значения для опций списка вводятся во вкладке  
Подстановка области Свойства поля

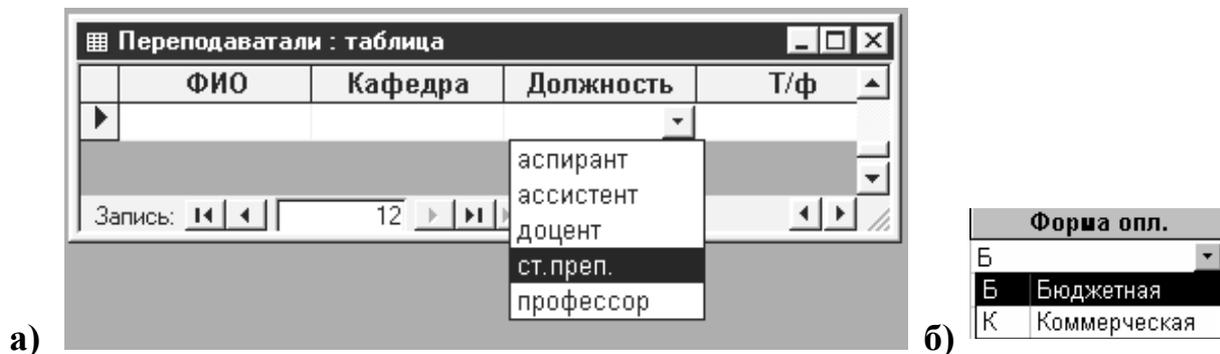


Рис. 12. а) Список подстановки для поля *Должность* включает перечень допустимых значений. б) Список подстановки для поля *Раут(Форма опл.)* таблицы *Учетная карта*

### Другие опции области Свойства поля

Прежде чем приступать к окончательному вводу данных в таблицы БД *Факультет*, рассмотрим другие свойства поля, которые активно используются при настройке структуры таблиц базы данных.

- Свойство *Значение по умолчанию*. Эта возможность позволяет упростить и ускорить ввод одинаковых значений в поле, а в случае необходимости вносить исправления уже в готовое значение, которое автоматически будет появляться в этом поле перед вводом данных. Таким значением может быть выражение, число или текст, которое соответствует типу данных поля. Для числовых и денежных данных по умолчанию устанавливается значение, равное 0.

- Свойство *Обязательное значение*. Определяет, требуется ли обязательно вводить данные в это поле. Если для свойства установлено значение «Да», при вводе или редактировании записей в таблице программа Access не «отпустит» вас, пока в этом поле не появятся данные. Пустые (Null) значения в этом поле не допускаются.

### Ввод данных в таблицы

Для того, чтобы БД *Факультет* могла функционировать нормально, необходимо в каждую таблицу ввести необходимый минимум данных. На рис. 13 – 16 представлены данные для четырех таблиц БД (данные таблицы *Учетная карта* представлены на рис. 6).

### ЗАДАНИЕ 6. Ввод данных в таблицы

1. Откройте таблицу *Учетная карта* в режиме Таблицы и продолжите набор данных согласно рис. 13. Обратите внимание, как работают маски в полях *№ бил* и *№ гр*. Для поля *Форма опл.* воспользуйтесь созданным в предыдущем **ЗАДАНИИ 5** списком подстановки.

2. Откройте таблицу *Ведомость* в режиме таблицы. В этом задании при заполнении поля *ФИО* таблицы *Ведомость* (а также и других таблиц) используйте прием *копирования данных через буфер обмена*. Этот способ не очень удобен, но на данном уровне знакомства с Access пока что является единственно доступным для ускорения ввода данных. Познакомившись с приемами связывания таблиц, создания запросов и форм, можно будет использовать более эффективные способы заполнения таблиц.

Итак, чтобы ускорить ввод данных и соблюсти однозначное соответствие записей в ключевом поле, перенесем данные в поле *ФИО* из таблицы *Учетная карта*, используя буфер обмена. Для этого выполните следующие действия.

3. Откройте таблицу *Учетная карта* и в опции *Окно* выберите расположение *Слева направо*.

4. Выделите столбец *ФИО*, щелкнув по названию поля. Выполните команду *Правка>Копировать*.

5. Перейдите к полю *ФИО* таблицы *Ведомость*. Выполните команду *Правка>Добавить из буфера*. На эти действия Access выдаст предупреждение (рис. 17).

Ведомость : таблица											
	ФИО	1_р	1_e1	1_e2	2_р	2_e1	2_e2	3_р	3_e1	3_e2	Диплом
▶	АНДРЕЕВА О.С.	з	4	3 з		3	4				
	АНТОНОВА Т.И.	з	5	4 з		4	5				
	АРСЕНТЬЕВ В.Р.	з	4	4 з		4	4				
	БЕЗНОГОВ С.Ю.	з	5	5 з		4	4				
	БЕЛОВ С.П.	з	5	3 з		5	4				
	БЕРЕЖНОЙ П.Р.	з	5	5 з		5	5				
	ВЕРХОВ А.В.	н	3	2 з		3	3				
	ГАЛКИН И.Д.	з	4	4 з		4	5				
	ДЕНИСОВ И.О.	з	4	4 з		3	5				
	ДРАГУНОВ А.А.	з	4	4 з		5	5				
	ЕЛИСЕЕВА И.В.	з	5	4 з		3	3				
	ЛЕОНТЬЕВ М.В.	з	3	4 з		3	3				
	РОМАНОВА С.Л.	з	3	4 з		3	4				
	СЕМЕНОВ А.Л.	з	4	5 з		5	4				
	ТОРОВА Р.К.	з	3	5 з		4	3				
	ЯКОВЛЕВ С.И.	з	4	3 з		5	4				
*						-1					

Запись: 1 Всего: 16

Рис.13. Таблица *Ведомость* открыта в режиме Таблицы для ввода записей

Переподаватели : таблица				
	ФИО	Кафедра	Должность	Т/ф
	АНТОНОВ В.А.	КФ-1	профессор	361456
	АРХИПОВА Е.Е.	КФ-2	доцент	710880
	БЕДРОВ Р.Г.	КФ-1	доцент	823261356
	ВЕРЕСОВ Е.А.	КФ-3	профессор	725684
	ГОЛОВАНОВА Н.И.	КФ-1	ст.преп.	642286
	ДАНИЛОВ А.И.	КФ-2	профессор	564278
	ЕРЕМИН Р.А.	КФ-2	ассистент	833225481
	ЗАЙЦЕВ А.С.	КФ-3	профессор	710153
	КОТОВ Н.М.	КФ-3	ст.преп.	551203
	ЛЕОНОВА Н.К.	КФ-1	доцент	865215680
	СОЛОВЬЕВ О.Л.	КФ-1	аспирант	361274
*				

Запись: 10 Всего: 11

Рис.14. Таблица *Преподаватели* открыта в режиме Таблицы для ввода записей

6. Подтвердите вставку записей. В результате таблица *Ведомость* будет иметь столько же записей, что и *Учетная карта*.

7. Перейдите в режим конструктора и проставьте в полях, где должны быть введены результаты зачетов (1\_р, 2\_р, 3\_р) Значение по умолчанию "з".

№ гр.	Куратор	Староста
А-1	ГОЛОВАНОВА Н.И.	АРСЕНТЬЕВ В.Р.
А-2	АРХИПОВА Е.Е.	БЕРЕЖНОЙ П.Р.
А-3		
Б-1	БЕДРОВ Р.Г.	АНТОНОВА Т.И.
Б-2	ЕРЕМИН Р.А.	ДРАГУНОВ А.А.
Б-3		
*		

Рис.15. Таблица *Группы* открыта в режиме Таблицы для ввода записей

Кафедра	Зав.каф.	№ ауд	Т/ф
КФ1	АНТОНОВ В.А.	А-121	435656
КФ2	ДАНИЛОВ А.И.	Б-403	433247
КФ3	ЗАЙЦЕВ А.С.	А-225	436428
*			0

Рис.16. Таблица *Кафедры* открыта в режиме Таблицы для ввода записей

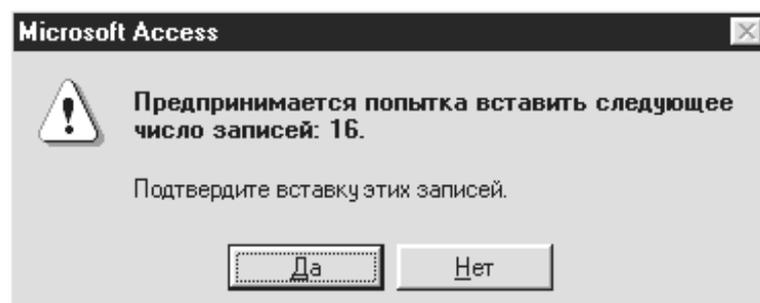


Рис. 17. Предупреждение Access о попытке вставить записи

8. Перейдите в режим таблицы и, сравнивая таблицу с рис. 13, исправьте те записи, где стоит "н".

9. Введите остальные оценки. Закройте таблицы *Учетная карта* и *Ведомость*.

10. Откройте таблицу *Преподаватели* в режиме Таблицы и введите в нее данные согласно рис. 14. Обратите внимание, как работает маска в поле

*Кафедра.* Воспользуйтесь списком подстановок для ввода должности преподавателя.

11. Откройте таблицу *Группы* в режиме Таблицы. Заполните ее согласно рис. 15. Можете воспользоваться буфером обмена для ввода фамилий кураторов и старост, копируя в него соответствующие фамилии из таблиц *Учетная карта, Преподаватели.*

12. Откройте таблицу *Кафедры* в режиме Таблицы. Заполните ее согласно рис. 16. Обратите внимание, как работает маска в поле *Кафедра.* Фамилии в поле *Зав.каф.* можно копировать из таблицы *Преподаватели.*

13. Закройте все таблицы. Обратите внимание, потребует ли программа от вас процедуры сохранения? Почему?

---

*Итак, на данном этапе создания базы данных вы обеспечили контроль и форматирование данных на этапе ввода, создали списки подстановок, научились работать сразу с несколькими таблицами, видоизменяя их расположение в окне. Кроме того, вы заполнили таблицы базы данных значениями, относящимися к студентам II и III курсов. Ввод значений для студентов I курса будем осуществлять позднее, используя более эффективные методы работы с программой Access.*

---

## **Организация связей между таблицами**

Когда между таблицами устанавливается связь, это означает, что величины из одной таблицы ставятся в соответствие величинам из другой таблицы.

Нами уже были рассмотрены определения таких важных понятий, как *первичный* и *внешний ключи*, вопрос выбора первичного ключа и рассмотрен основной принцип, позволяющий реализовать связь между реляционными таблицами: такая связь становится возможной благодаря наличию в этих таблицах полей с совпадающими значениями данных.

Существуют условия, которым должны соответствовать типы данных связываемых полей.

- Общие или связываемые поля должны быть одинакового типа.
- Если оба связываемых поля имеют числовой тип, они должны иметь и одинаковые значения свойства *Размер* поля.
- Поле с типом *Счетчик* можно связать с числовым полем, у которого свойство *Размер* поля имеет значение *Длинное целое*.

Поскольку при проектировании БД *Факультет* сразу предполагалось, что все таблицы этой БД будут связаны между собой, рассмотрим в качестве примера создание таких связей. Для этого необходимо открыть окно Схема данных и добавить в него те таблицы, которые будут связаны. Окно Схема данных – это удобное и наглядное графическое представление связей для открытой базы данных, где можно добавлять таблицы, определять тип связи и устанавливать параметры целостности данных между таблицами. Итак, следуя рекомендациям **ЗАДАНИЯ 7**, установите связи между таблицами БД *Факультет*.

### ЗАДАНИЕ 7. Установка связей между таблицами

1. Открыть базу данных и выполнить команду горизонтального меню Сервис>Схема данных или щелкнуть на кнопке  Схема данных панели инструментов. Откроется диалоговое окно Схема данных. Если БД уже открыта, то создавать или изменять связи между открытыми таблицами нельзя, поэтому перед началом работы их следует закрыть.

2. В том случае, если в окно Схема данных еще не добавлены никакие таблицы, сразу откроется окно Добавление таблицы. Если оно не появится, выполните команду горизонтального меню Связи>Добавить таблицу или щелкните на кнопке  Добавить таблицу на панели инструментов.

3. Дважды щелкните на имени таблицы (например, *Учетная карта*), которая должна быть включена в окно схемы данных.

4. Выполните шаг 3 требуемое число раз, чтобы включить в рабочую область окна Схемы данных все необходимые таблицы. Затем щелкните на кнопке Закрыть. Для нашего примера таких таблиц будет пять, и в итоге все они появятся в окне Схема данных в виде отдельных окон, где перечислены поля, содержащиеся в таблицах (рис. 18).

5. Создайте теперь связь между таблицами *Учетная карта* и *Ведомость*. Как отмечалось ранее, это связь типа «один к одному». Каждой записи первой таблицы соответствует одна (и не более) связанная запись второй таблицы (и, соответственно, наоборот).

В обеих таблицах роль первичного ключа выполняет поле *ФИО*. На этом поле таблицы *Учетная карта* нажмите левую клавишу мыши и, удерживая ее, перетащите на поле *ФИО* таблицы *Ведомость*. Затем отпустите левую кнопку мыши. (При таком порядке установления связи первую таблицу *Учетная карта* можно рассматривать как главную, а *Ведомость* – как подчиненную).

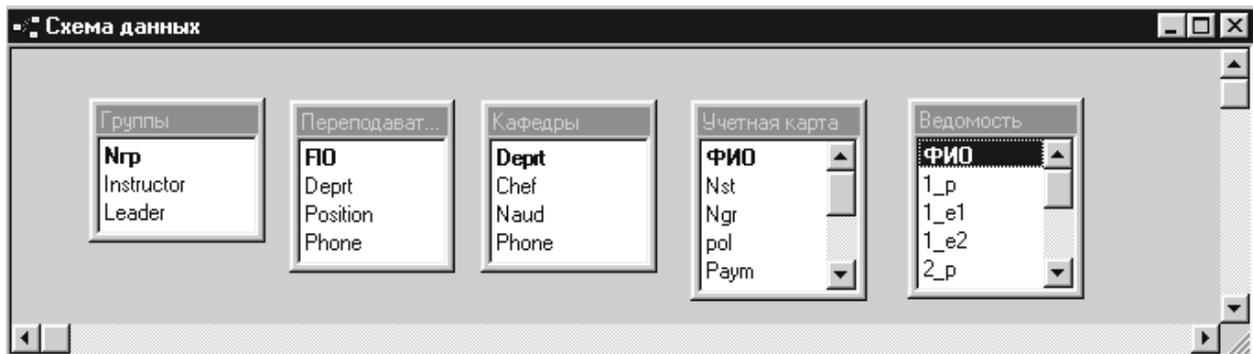


Рис.18. В окне Схема данных отображаются пять таблиц базы данных *Факультет*

6. После того, как поле перенесено, появится диалоговое окно Изменение связей (рис. 19).

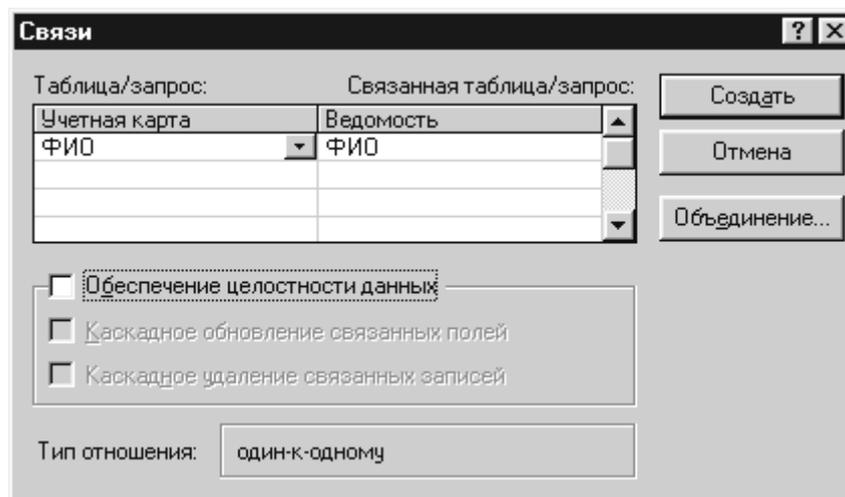


Рис. 19. Диалоговое окно Изменение связей позволяет определить параметры создаваемой связи

7. Щелкните в этом окне на кнопке Создать, в результате между двумя полями появится тонкая линия объединения (рис. 20).

8. Теперь создайте связь между таблицами *Учетная карта* и *Группы*. Здесь, как уже отмечалось, связь типа «один ко многим». Одна из таблиц – *Группы* – называется родительской, или главной (общее поле *Ngr* является в ней первичным ключом), а вторая – дочерней, или подчиненной (поле *Ngr* в ней – это внешний ключ). Теперь, как и в п.5, первичный ключ надо выделить и совместить со вторичным, установив тем самым связь между указанными таблицами.

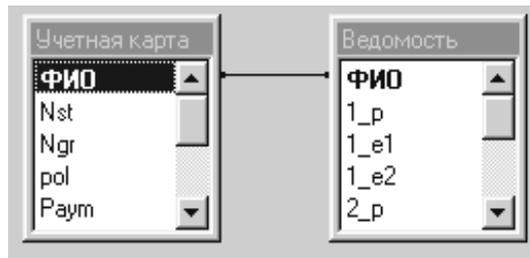


Рис. 20. Таблицы *Учетная карта* и *Ведомость* теперь связаны между собой

9. Теперь необходимо повторить шаги 5 – 7 для каждой пары таблиц, которые необходимо связать между собой. Сделайте это самостоятельно, получив в результате этих операций в окне *Схема данных* структуру связей, изображенную на рис. 21.

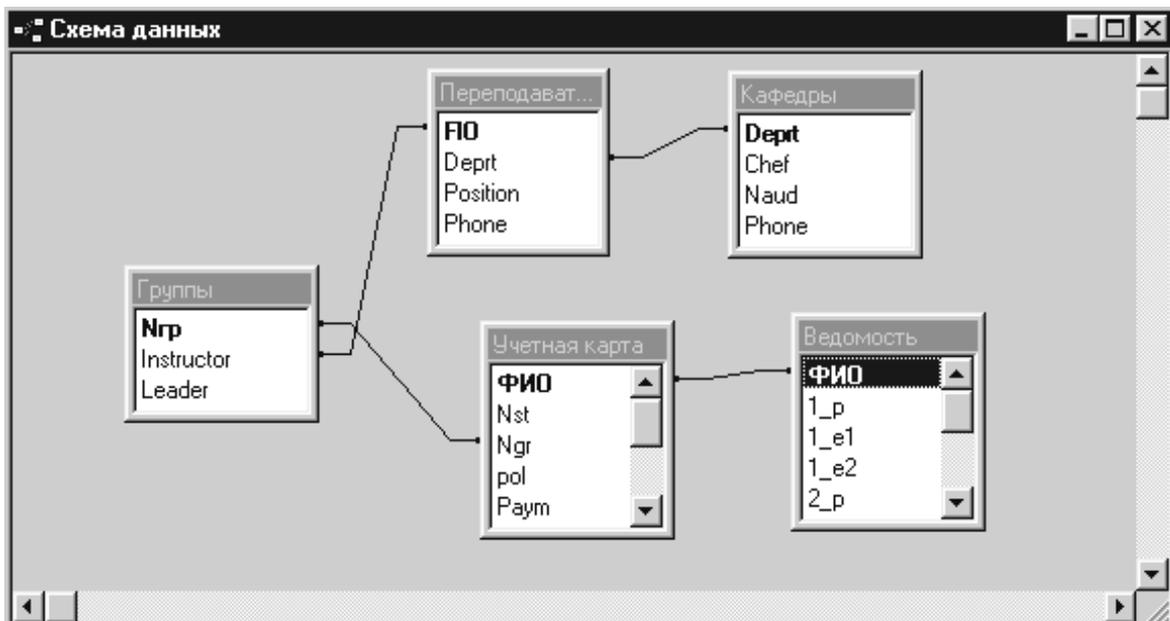


Рис. 21. В окне *Схема данных* отображены все связи между таблицами базы данных *Факультет*

10. Сохраните созданные связи, используя стандартные приемы Windows для сохранения. Но, даже, если Вы не сделаете этого, программа Access предложит вам сохранить созданные связи и при закрытии окна *Схема данных*.

## Обеспечение целостности данных в базе

При определении связи между полями двух таблиц в диалоговом окне Изменение связей имеется опция Обеспечение целостности данных (рис. 19). *Целостность данных* – это система правил, регулирующих взаимодействие между связанными таблицами и обеспечивающих корректность хранимых в таких таблицах данных. Целостность данных подразумевает, что данные, введенные в общее поле двух связанных таблиц, должны совпадать.

Если используются условия целостности данных, то Access будет поддерживать следующие правила.

- В поле внешнего ключа дочерней таблицы нельзя ввести значение, не содержащееся в поле первичного ключа родительской таблицы. Таким образом, если попытаться указать в поле *Ngr* номер группы, отсутствующий в списке *Группы* (поле *Ngr*), при сохранении записи программа Access выдаст сообщение об ошибке.

- Программа Access не позволяет удалить запись родительской таблицы, если в дочерней таблице имеются связанные с ней записи. Например, нельзя удалить из таблицы *Преподаватели* сведения о преподавателе, если он является куратором и о нем имеется запись в таблице *Группы*.

- Нельзя изменить значение первичного ключа в родительской таблице, если для данной записи существуют связанные записи в дочерней таблице. Например, если попытаться изменить название кафедры в поле *Deprt* таблицы *Кафедры*, то это сделать будет невозможно, т.к. в таблице *Преподаватели* существуют записи, в которых используется старое название (рис. 22).

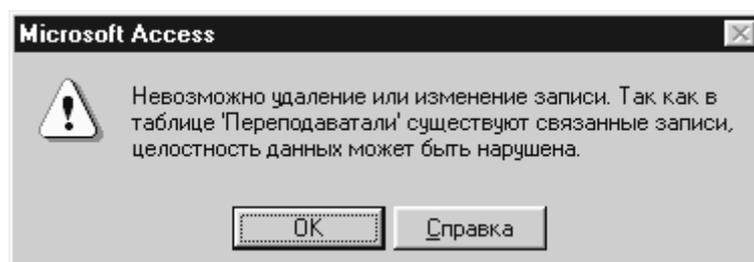


Рис. 22. Предупреждение Access о невозможности изменения данных, которое может нарушить их целостность

При выборе опции Обеспечения целостности в окне Изменение связей (рис. 19) становятся доступными еще две опции, которые позволяют активизировать автоматическое выполнение каскадного удаления и каскадного обновления данных.

- Если поставить флажок на опции Каскадное обновление связанных полей, то при любом изменении данных первичного ключа в родительской таблице (например, названия кафедры в поле *Deprt* таблицы *Кафедры*) автоматически будут обновляться соответствующие значения в поле связанной таблицы (поле *Deprt* таблицы *Преподаватели*). Благодаря внесению необходимых изменений в связанную таблицу целостность данных не будет нарушена. Если эта опция не включена, изменить значение ключевого поля первичной таблицы не удастся, как это уже было показано в предыдущем пункте.

- Когда установлен флажок в опции Каскадное удаление связанных полей, то при удалении записи в родительской таблице автоматически будут удаляться соответствующие записи в дочерней связанной таблице. Например, если такого рода связь установлена между таблицами *Учетная карта* и *Ведомость*, то при удалении записи из одной таблицы *Учетная карта* автоматически будет удалена такая же запись из таблицы *Ведомость*.

## **ЗАДАНИЕ 8. Редактирование и удаление связей между таблицами**

В предыдущем **ЗАДАНИИ 7** между всеми таблицами БД *Факультет* были установлены связи. Любая связь может быть отредактирована или удалена. Для это необходимо выполнить следующие действия:

1. Если таблицы, между которыми необходимо изменить связь, открыты, заройте их.

2. Открыть окно Схема данных.

3. В окне Схема данных подвести курсор к линии связи, которую надо откорректировать, и двойным щелчком открыть диалоговое окно Изменение связей (рис. 19).

4. Внесите необходимые изменения: установите целостность данных и каскадное обновление полей, проставив в соответствующих окнах флажки. После закрытия окна Изменение связей вы снова вернетесь в окно Схема данных.

5. Обратите внимание, как изменился вид линий связи: теперь по схеме данных можно определить тип связи (рис. 23).

6. После установления целостности связей всех таблиц откройте в режиме таблицы таблицу *Учетная карта* и *Ведомость*. Введите в первую таблицу в поле *ФИО* фамилии **Иванов И.И.** и **Петров П.П.**, другие поля заполнять необязательно (запись считается введенной, если курсор уже переведен в поле временной (новой) записи). Затем через буфер обмена перенесите эти фамилии в таблицу *Ведомость*.

7. Установите расположение окон Слева направо, используя пункт горизонтального меню Окно. Перейдите в таблицу *Учетная карта*. Удалите запись **Иванов И.И.** (используйте пункт Правка>Удалить запись гори-

зонтального меню). Внимательно прочтите предупреждение, которое выдает Access на ваши действия (рис. 24 а). Подтвердите удаление. Обратите внимание, что произошло с этой же записью в таблице *Ведомость*.

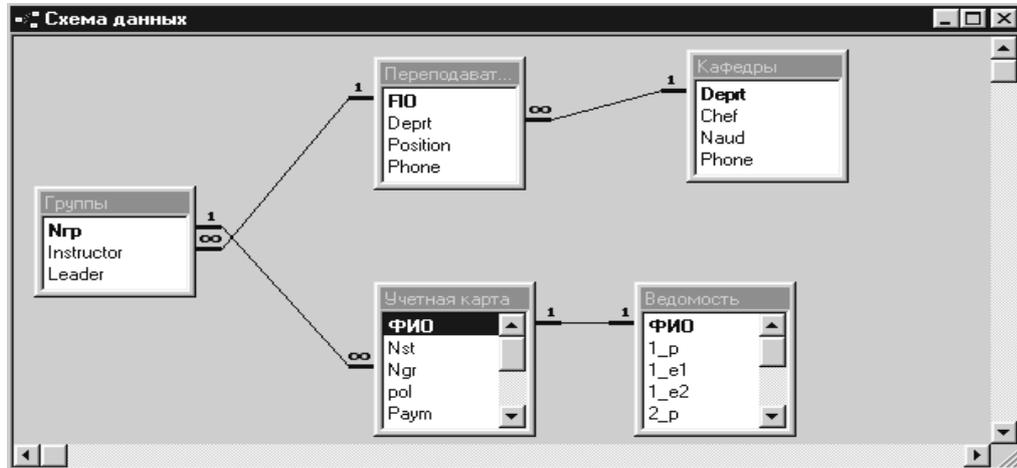


Рис. 23. Схема связей между таблицами, в которых обеспечивается целостность данных

8. Установите курсор в таблицу *Ведомость*. Удалите запись с фамилией **Петров П.П.** Обратите внимание на предупреждение, сделанное Access в этом случае (рис. 24 б). Самостоятельно удалите запись с фамилией **Петров П.П.** из таблицы *Учетная карта*. Закройте обе таблицы.

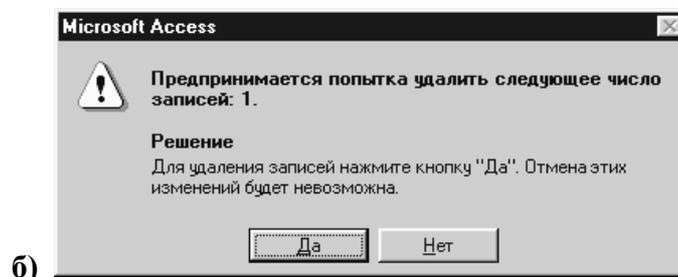
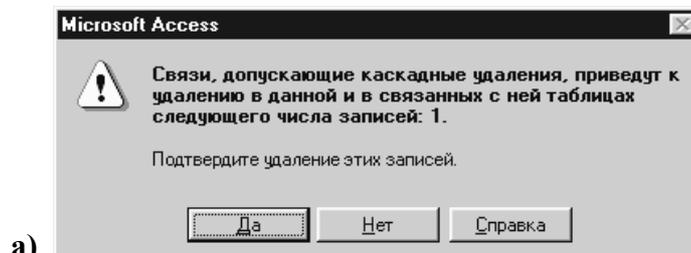


Рис. 24. Предупреждение Access при каскадном удалении записи  
а) в главной таблице. б) в подчиненной таблице

9. Откройте таблицы *Группы* и *Преподаватели*. и расположите их Слева направо.

10. Попробуйте удалить запись с фамилией **Голованова Н.И.** из таблицы *Преподаватели*. Access не может позволить вам этого сделать, иначе нарушается целостность данных (скопируйте запись с фамилией **Голованова Н.И.** в буфер обмена).

11. Удалите фамилию **Голованова Н.И.** из поля *Кураторы* таблицы *Группы*. А затем еще раз повторите шаг 10. На этот раз удаление записи должно пройти успешно.

12. Вставьте из буфера удаленную запись в таблицу *Преподаватели* и впишите эту же фамилию в поле *Кураторы* таблицы *Группы*. Закройте все таблицы.

Если понадобится удалить межтабличную связь, например, для того, чтобы изменить тип данных поля, участвующего в связи, необходимо выполнить следующие действия.

1. Щелкнуть на требуемой линии объединения в окне *Схема данных* (линия при этом станет более жирной) и нажать клавишу <Delete>. Есть и другой способ: щелкнуть правой кнопкой мыши на нужной линии связи и выбрать из появившегося контекстного меню команду *Удалить*.

2. На экране появится диалоговое окно *Access*, в котором необходимо будет подтвердить удаление связи.

---

*Итак, таблицы баз данных можно связывать между собой, используя общее поле. На данном этапе конструирования базы вы изучили, как создавать, редактировать и удалять связи между таблицами, при необходимости указывая опции обеспечения целостности данных, такие, как каскадное обновление связанных полей и каскадное удаление связанных записей.*

*Благодаря связям между таблицами можно создавать запросы, формы и отчеты, отображающие информацию из нескольких таблиц. Изучению этих возможностей Access и будут посвящены следующие задания.*

### **Контрольные вопросы**

1. Что такое база данных с точки зрения ее хранения на диске?
2. Как создать файл новой базы данных?
3. Можно ли изменить структуру таблиц после их сохранения? В чем заключаются эти изменения?
4. Что означает сохранение данных в режиме реального времени?

5. Что такое временная запись таблицы? Можно ли разместить новую запись между уже существующими записями или поместить ее в начале таблицы?
6. Какова реакция программы Access на ввод буквенных символов в поле числового типа?
7. Как осуществляется копирование записей через буфер обмена?
8. С какой целью делается форматирование данных в поле?
9. Какой специальный формат необходимо применить, чтобы все символы текстового поля изображались заглавными буквами?
10. Для каких типов полей разработаны встроенные форматы?
11. Для чего предусмотрено свойство Маска ввода? В чем принципиальное отличие опций форматирования и маски ввода?
12. С помощью какого свойства поля можно организовать проверку вводимых значений в числовое поле?
13. Из каких элементов складывается выражение при проверке условия на значение?
14. Для чего используется список подстановки?
15. Какие возможности предоставляют свойства поля Значение по умолчанию, Обязательное значение?
16. Когда Access потребует сохранения таблицы: при изменении свойств ее полей или при вводе данных? Что при этом сохраняется: значения в полях таблицы или ее структура?
17. Каким условиям должны соответствовать типы данных связываемых полей? Поля каких ключей используются в роли общего поля при реализации связи между реляционными таблицами?
18. Каковы условия поддержки целостности данных при связывании реляционных таблиц?
19. Как изменяются данные в ключевых полях, если при установлении целостности данных установлено Каскадное обновление связанных полей?

### **Оформление отчета**

Отчет о лабораторной работе должен содержать:

1. Описание в тетради основных этапов проделанной работы, их логическую последовательность.
2. Ответы на контрольные вопросы.

## Лабораторная работа № 2 «СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ЗАПРОСОВ»

### Цель работы<sup>1</sup>

Данная лабораторная работа предназначена для приобретения начальных навыков работы с запросами реляционной базы данных, а именно:

- а) использование простейших логических операндов,
- б) приобретение практики выборки заданной информации,
- в) изучение построения математического выражения в структуре запроса,
- г) изучение групповых операций в запросах.

### Назначение и типы запросов

Таблицы хранят данные. При работе с БД возникает необходимость работать сразу с несколькими таблицами, анализировать, выбирать и сортировать в них данные. Для этого в Access имеется гибкое и удобное средство – запросы. Для одной и той же таблицы можно создать множество разных запросов, каждый из которых сможет извлекать из таблицы лишь малую часть информации, но именно ту часть, которая необходима в данный момент. В результате работы запроса из общей исходной базы формируется *результатирующая таблица*, содержащая часть общей информации, соответствующую запросу.

Результат работы запроса – это группа записей, которые удовлетворяют заданному критерию запроса. Совокупности этих записей называются *динамическим набором записей* и отображаются в виде таблицы. Это временная таблица, которая не является объектом базы данных и хранится только в памяти компьютера. Если с момента последнего запуска запроса данные в исходной таблице были изменены, при выполнении запроса динамический набор данных будет включать уже обновленные данные. Таким образом, запрос – это временная таблица: данные в них не хранятся постоянно, а только временно вызываются из таблиц, по заранее заданному шаблону, в момент активизации запроса. Как результат, в базе данных постоянно хранятся только шаблоны вызова данных (временные таблицы удаляются после закрытия запроса), а сама информация не дублируется.

---

<sup>1</sup> Для выполнения Лабораторной работы №2 необходимо наличие в базе пяти таблиц, созданных в Лабораторной работе № 1.

Спектр возможностей, которыми обладает Access для обработки данных при помощи запросов, определяется разными типами запросов. Рассмотрим некоторые из них.

**Запрос на выборку** – тип запроса, принятый по умолчанию. Осуществляет выборку данных, соответствующих указанным условиям отбора, из одной или нескольких таблиц.

**Запрос с параметром** – это «интерактивный» тип запроса, при выполнении отображающий в собственном диалоговом окне приглашение ввести один или ряд параметров, например, условие отбора записей по определенному полю.

**Запрос на изменение** позволяют, например, удалить сразу несколько записей, удовлетворяющих определенному условию, создать новую таблицу по результатам запроса или скопировать данные из одной таблицы в другую.

**Перекрестный запрос** используется для расчетов и представления данных в структуре, облегчающий их анализ. Перекрестный запрос подсчитывает сумму, среднее число значений и другие статистические данные по двум наборам данных, один из которых определяет заголовки столбцов, а другой – заголовки строк.

При создании запросов используется специальный язык программирования SQL (Structured Query Language). Это формальный стандартизованный язык высокого уровня, содержащий средства непроцедурной обработки (не требующей программирования) спецификации запросов. Однако пользователи Access могут не изучать этот язык. Вместо него в Access есть простое средство – *бланк запроса по образцу*. С его помощью можно сформировать запрос простыми приемами, перетаскивая элементы запроса между окнами. Делать это удобнее и понятнее с помощью Конструктора запросов, т.е. вручную, хотя для опытных пользователей можно воспользоваться помощью Мастера запросов.

## Выражение и его компоненты

При рассмотрении ввода и редактирования данных в таблицы уже затрагивался вопрос использования выражений для проверки вводимых значений на уровне поля и таблицы. Внимание на самих выражениях не акцентировались, так как их использование было интуитивно понятным. Однако выражения – это мощное средство выполнения многих операций с данными в программе Access, особенно при построении запросов.

*Выражение* – это любая комбинация *операторов*, имен констант, констант в явном представлении, функций и имен полей, элементов управления или свойств, которые называются *операндами*. Результатом выражения является конкретное значение. С помощью выражений задаются значения многих свойств и аргументов макрокоманд, определяются условия отбора и вычисляемые поля в запросах или условия выполнения макрокоманд в макросах. Выражения используются также в конструкциях Visual Basic. Выражения можно вводить как вручную, так и с помощью Построителя выражений.

## Операторы

Операторы позволяют выполнить некоторые действия над одним или несколькими компонентами выражения. Программа Access поддерживает шесть типов операторов.

- Арифметические операторы. Выполняют привычные математические операции с числовыми значениями: сложения (+), вычитания (-), умножения (\*), и деления (/). Кроме того, к этой категории относят оператор целочисленного деления первого операнда на второй (\), оператор деления по модулю MOD (остаток целочисленного деления одного операнда на другой) и возведения в степень (^). В роли операндов могут быть как числа, так и значения полей или выражения.

Пример для оператора деления нацело:  $25,6 \setminus 3,8 = 6$  (числа сначала округляются до целых значений, а затем вычисляется частное двух чисел, которое также округляется до целых).

Пример арифметического выражения для формулы вычисляемого поля:  
**СреднийБалл: ([1\_e1]+[1\_e2])/2**

- Операторы сравнения. Служат для сравнения двух операндов и возвращают в зависимости от отношения между операндами, логические значения (**True** или **False**) или **Null** (если, хотя бы один из операндов имеет значение **Null**). К этому типу относятся следующие операторы: > (больше чем), < (меньше чем), >= (больше или равно), <= (меньше или равно), = (равно), <> (не равно).

- Логические (булевы) операторы. Так как в качестве операндов таких операторов могут быть значения, которые либо ложны, либо истинны, логические операторы возвращают логические значения (**True** или **False**) или **Null**. Как правило, используются для комбинирования результатов выполнения двух и более операций сравнения. Список логических операторов включает **And** (Логическое И), **Or** (Включающее ИЛИ), **Not** (Логическое НЕ), **Xor**

(Исключающее ИЛИ). Все логические операторы, за исключением **Not**, всегда работают с двумя операндами.

Например, **Not**”аспирант”. Будут отобраны все записи о преподавателях, имеющих в поле *Должность* любые значения, за исключением **аспирант**.

- Операторы конкатенации. Оператор **&** служит для объединения нескольких строк символов в одной строку. Обрабатывает все переменные как символьные строки.

- Операторы идентификации. Программа Access работает с двумя операторами идентификации: **!** (восклицательный знак) и **.** (точка). С помощью этих операторов можно обращаться к конкретным объектам, например, к полю таблицы. Символ «!» используется вместе с различными зарезервированными словами, например, **Forms**, указывая на то, что далее следует имя формы. Так, например, идентифицировать форму *Учетная карта*, используется выражение **Forms!**[Учетная карта], так как в базе данных могут быть другие объекты с таким именем, а именно: таблица *Учетная карта*. Синтаксис выражения следующий: **КлассОбъекта!ИмяОбъекта**.

Символ «.»(точка) отделяет имена объектов от их свойств или методов (синтаксис задан иначе: **КлассОбъекта!ИмяОбъекта.Свойство** или **КлассОбъекта!ИмяОбъекта.Метод**).

Например, **Forms!**[Ведомость]!**[1\_p].DefaultValue**.

- Прочие операторы. Это операторы языка SQL, такие как **Like**, **Is**, **In**, **Between...And**, с помощью которых можно упростить создание выражений. Возвращают значение **True** или **False**.

- ◆ **Is.** Используется в выражениях **Is Null** или **Is Not Null**. Определяет наличие или отсутствие значения **Null**, т.е. является ли объект пустым.

- ◆ **Like.** Проверяет, соответствует ли строковое значение заданному шаблону. Его ставят впереди заданного фрагмента, а до или после фрагмента, в этом случае, можно использовать 5 символов подстановки: **\***(любое число символов), **?**(любой одиночный символ), **#**(любая цифра), **[список]**(любой символ из списка) и **[!список]**(любой символ, не принадлежащий указанному списку). Пример использования этого оператора будет рассмотрен в **ЗАДАНИИ 11**.

- ◆ **In.** Проверяет, совпадает ли значение с одним из элементов, указанных в списке. Например, критерий отбора может иметь следующий вид: **In**(“A1”;”B1”). Согласно этому критерию будут отобраны записи, содержа-

щие в поле *№гp* значение **A-1** или **B-1** (тире в названии группы указывать не надо, т.к. этот символ заложен в маску ввода).

◆ **Between...And.** Определяет, принадлежит ли числовое значение заданному диапазону значений. Пример в **ЗАДАНИИ 11**.

## Литералы

Литералы – это используемые в Access значения в их явном представлении. Литералы бывают следующих типов.

- **Числовые.** Вводятся как ряд чисел, могут содержать знак разделителя (в десятичном числе) и знак «минус» (–) для отрицательных значений, символы «E» и «e», а также знак показателя степени (при экспоненциальной форме представления чисел). **Например:** 3,4567E-01, 12000,-25.

- **Текстовые (строковые).** Включают любые печатные символы (А–Я, А–Z, числа от 0 до 9, знаки пунктуации и специальные символы клавиатуры, а также непечатаемые символы, например, перевода каретки (задаются с помощью функции Chr()). Строковые литералы следует заключать в двойные кавычки (" "). **Например:** "Иванов".

- **Литералы даты и времени.** В программе Access знак номера (#) ставится до и после любой даты. Если при вводе в таблицу значение даты указывается в стандартной формате, распознаваемом Access, или определяется в качестве критерия отбора в бланке запроса, указывать эти знаки необязательно. **Например,** #01.03.00#, 15-январь-2004.

## Идентификаторы

Идентификаторы – это имена объектов (баз данных, таблиц, полей, запросов, форм и отчетов). Используемые в выражениях, идентификаторы возвращают определенные числовые или текстовые значения: т.е. представляют собой ссылку на текущее значение поля, элементы управления или свойства. Например, такой идентификатор, как **Forms![Ведомость]![1\_p].DefaultValue** определяет ссылку на значение свойства **Значение по умолчанию (DefaultValue)** элемента управления *1\_p* в форме *Ведомость*.

## Функции

Основное назначение функции – возвращать в выражение вместо имени функции вычисляемое значение. Для некоторых функций обязательным

является указание одного или нескольких аргументов, в роли которых могут выступать идентификаторы или значения, заключенные в скобки. В Access можно использовать как встроенные функции, так и определяемые пользователем. Ниже перечислены основные типы функций и некоторые примеры их использования.

- Функции даты и времени.

**Date()**. Отображает текущую дату в формате дд.мм.гг.

**DateAdd("d";15;[Абонемент]![Дата выдачи]).** Для БД *Библиотека* возвращает дату, на 15 дней отстоящую от даты, заданной значением поля *Дата выдачи* таблицы *Абонемент*.

**DateDiff ("d";[Дата возврата];[Дата выдачи]).** Возвращает значение, представляющее разницу числа дней между значениями полей *Дата возврата* и *Дата выдачи*.

**Year(#23.02.04#).** Возвращает число, представляющее год в указанной дате: 2004.

- Функции обработки текста.

**InStr("Андреева";"e").** Возвращает число, указывающее позицию первого вхождения одной строки "e" в другую строку "Андреева": 5.

**LCase([ФИО])** Возвращает строку, преобразованную к нижнему регистру.

**Left([ФИО],2).** Отображает два первых символа значения поля *ФИО*.

**Right([ФИО],5).** Отображает 5 последних символов значения поля *ФИО*.

- Функции преобразования типа данных.

**Val("1234.56").** Возвращает число, содержащееся в строке 1234.56.

**Str(123,45).** Возвращает строковое представление числа "123.45".

- Математические, тригонометрические, финансовые функции, функции ввода и управления и др. Название, назначение и формат этих функций можно посмотреть, если раскрыть окно **Построителя выражений**.

### **Использование Построителя выражений**

При создании запросов с вычисляемыми полями, а также в ситуациях, когда необходимо работать с длинными выражениями, включающими поля из нескольких таблиц, функции и другие компоненты, можно воспользоваться диалоговым окном **Построитель выражений** (рис. 25). Так, например, в бланке запроса **Построитель выражений** доступен для ячеек **Поле** и **Условие отбора**. Открыть его можно, щелкнув на команде **Построить** контекстного меню, вызываемого на нужной ячейке. Выражение, созданное в

окне Построителя выражений, будет размещено в той ячейке или поле, из которых Построитель был запущен.

Структурно окно построителя состоит из нескольких областей (Рис. 25).

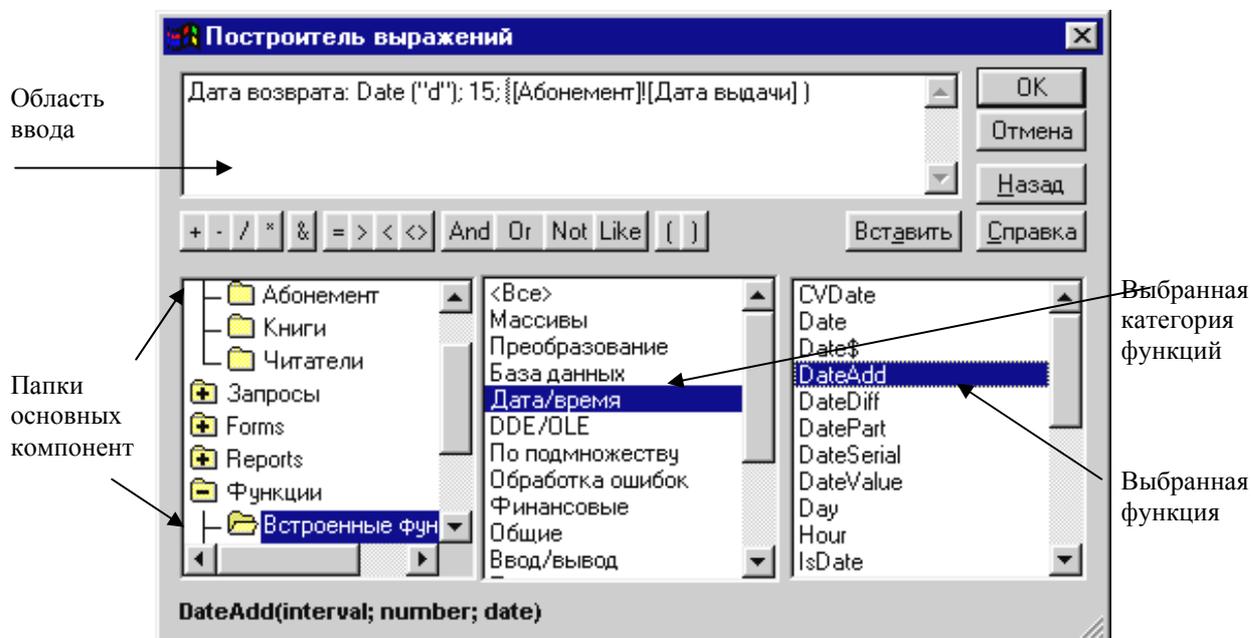


Рис. 25. Диалоговое окно построителя выражений

В верхней части диалогового окна находится поле выражения – область, в которой создается выражение. Здесь компоненты выражения можно ввести вручную, а можно – с помощью выбора из списков элементов и последующей вставки в выражение. Ниже расположен ряд кнопок, позволяющих включать в выражение математические и логические операторы, оператор конкатенации, а также скобки. После щелчка на любой из этих кнопок построитель вставит этот оператор в текущую позицию выражения.

Нижняя область окна разделена на три поля. В левом поле представлены папки с основными типами компонентов выражений, включая папки объектов текущей базы данных. Чтобы добавить в выражение поле из таблицы, сначала надо открыть папку таблицы и щелчком выбрать необходимый заголовок.

В средней секции отобразится список всех полей указанной таблицы. Дважды щелкнув на нужном поле, можно включить его в выражение. В результате в поле выражения появится имя поля, дополненное именем таблицы и восклицательным знаком.

В первом поле также содержатся списки встроенных функций, констант, операторов и общих выражений. Каждый из таких списков – это папка, включающая упорядоченные элементы. Щелкая мышью по элементам этих папок, в среднем столбце можно выбирать нужную категорию операторов или функций, в третьем столбце – сами операторы.

Познакомившись с основными инструментами, перейдем теперь к рассмотрению основных типов и способам построения запросов.

## Запросы на выборку

В запросах на выборку данные могут: отбираться по многим критериям; сортироваться; с ними могут производиться вычислительные операции.

Начнем создание запросов с простой задачи: дополнить таблицу *Ведомость* данными из таблицы *Учетная карта*, т.е. необходимо добавить номера студенческих билетов и номера групп.

### ЗАДАНИЕ 9. Построение запроса на выборку данных из нескольких таблиц

1. Создание запроса начинается с открытия вкладки Запросы диалогового окна База данных и щелчка на кнопке Создать. В открывшемся диалоговом окне Новый запрос задают ручной режим создания запроса выбором пункта Конструктор.

Создание запроса начинают с выбора тех таблиц базы, на которых будет основан запрос. В данном случае – это *Учетная карта* и *Ведомость*. Выбор таблиц выполняют в диалоговом окне Добавление таблиц. В нем отображаются все таблицы, имеющиеся в базе. Выбранные таблицы заносятся в верхнюю половину бланка запроса по образцу щелчком на кнопке Добавить (рис. 26). Наличие в диалоговом окне вкладки Таблицы и Запросы говорит о том, что запрос можно основывать не только на таблицах, но уже и на имеющихся запросах

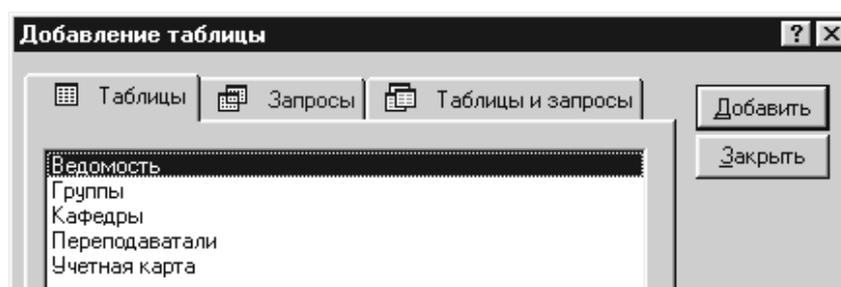


Рис. 26. Диалоговое окно Добавление таблицы.

2. После выбора таблиц, приступаем к работе с *Бланком запроса по образцу*. Он имеет две панели (рис. 27). На верхней панели расположены списки полей тех таблиц, на которых основывается запрос. Строки нижней панели определяют структуру запроса, т.е. структуру результирующей таблицы, в которой будут содержаться данные, полученные по результатам запроса.

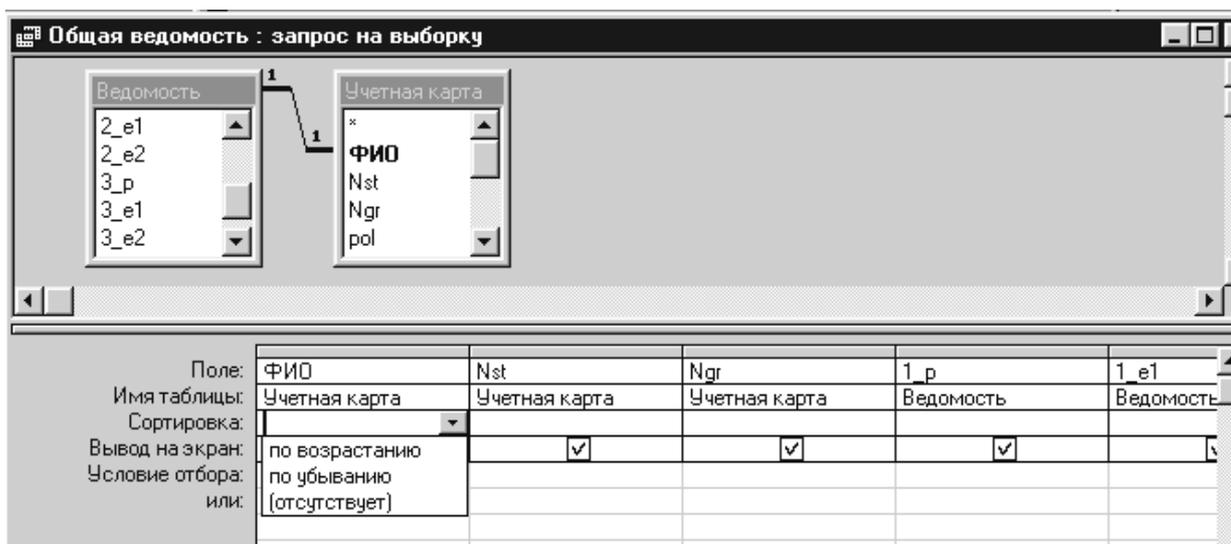


Рис. 27. Бланк запроса по образцу. В верхней части – выбранные таблицы. В нижней – выбранные (двойным щелчком или перетаскиванием) поля. В строке Сортировка для поля *ФИО* показан раскрывающийся список для назначения типа сортировки по данному полю

3. Строку Поле можно заполнить двумя способами: или перетаскиванием названий полей из таблиц в верхней части бланка, или двойным щелчком на этом поле. Каждому полю будущей результирующей таблицы соответствует один столбец бланка запроса по образцу. Добавьте поля *ФИО*, *Nst* и *Ngr* из таблицы *Учетная карта* перетаскиванием мышью, а поля с названиями зачетов и экзаменов из таблицы *Ведомость* – двойным щелчком. Строка Имя таблицы заполняется автоматически при установке поля.

4. Если щелкнуть на строке Сортировка, появится кнопка раскрывающегося списка, содержащего виды сортировки. В данном случае отсортируем результирующую таблицу по алфавиту фамилий.

5. Закройте запрос, при закрытии дайте ему имя *Общая ведомость*.

6. Запустите запрос на выполнение, используя команду горизонтального меню Запрос>Запуск, или щелкнув на значке Представление запроса  панели инструментов Конструктор запроса. Посмотрите, какие данные он выводит.

7. Закройте запрос.

8. Следуя инструкциям 1 – 8, объединив данные из трех таблиц *Группы*, *Преподаватели*, *Кафедры*, создать запрос *Кураторы*, который дает список кураторов групп с их местом работы: название и месторасположение кафедры. Отсортируйте список кураторов по алфавиту (рис. 28).

Куратор	№ гр.	Кафедра	№ ауд
АРХИПОВА Е.Е.	А-2	КФ-2	Б-403
БЕДРОВ Р.Г.	Б-1	КФ-1	А-121
ГОЛОВАНОВА Н.И.	А-1	КФ-1	А-121
ЕРЕМИН Р.А.	Б-2	КФ-2	Б-403

Рис. 28. Результаты выполнения запроса *Кураторы*, выполненного по трем таблицам *Группы*, *Преподаватели*, *Кафедры*. Записи отсортированы по алфавиту кураторов

### Построение запросов с условием отбора

Условиями отбора называют ограничения, которые накладываются на запрос или фильтр для отбора конкретных записей. Например, требуются сведения не обо всех студентах, а только о тех, кто обучается на бюджетной основе. Для этого следует указать условия отбора, которые позволят включать в набор записей только те, у которых в поле *Раут* (*Форма опл*) находится значение "Б" (рис. 29).

Поле:	ФИО	Раут
Имя таблицы:	Учетная карта	Учетная карта
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:	<input type="checkbox"/>	"Б"
или:		

Рис. 29. Для поля *Раут* указано Условие отбора для студентов, обучающихся на бюджетной основе. В результирующей таблице столбец *Раут* отражаться не будет, т.к. отсутствует галочка в строке Вывод на экран

Для того чтобы задать в бланке *Условие отбора* для конкретного поля, введите выражение в ячейку строки *Условие отбора* для этого поля. Причем поле, участвующее в отборе необязательно отражать в результирующей таблице. Для этого нужно убрать галочку в строке *Вывод на экран*.

Можно одновременно определить несколько условий отбора для одного поля или для разных полей. При вводе выражений в несколько ячеек *Условие отбора Access* объединяет их с помощью оператора **And** или **Or**. Если выражения находятся в разных ячейках одной и той же строки, используется оператор **And**. Если выражения находятся в разных строках бланка, применяется оператор **Or**.

Например, если требуется получить список студентов, не подлежащих призыву в армию, то в него войдут все девушки (поле *pol="ж"*), студенты имеющие льготы (поле *Lgoty= Да*), мужчины, дата рождения которых позднее, т.е. меньше 1 января 1985 года (на 2003 год). Тогда бланк запроса будет иметь вид, аналогичный показанному на рис. 30.

Поле:	ФИО	pol	Dtr	Lgoty
Имя таблицы:	Учетная карта	Учетная карта	Учетная карта	Учетная карта
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		"ж"	<#01.01.82#	Да
или:				

Рис. 30. Критерий запроса включает три условия для разных полей, объединенных оператором **OR**. Результирующие записи должны удовлетворять одному из трех условий

## ЗАДАНИЕ 10. Построение запросов с условием отбора

1. Сформировать запрос *Запрос1* на получение списка студентов-мужчин, обучающихся на коммерческой основе. В результирующей таблице

выведите только список фамилий. Рекомендация: условия для требуемых полей записывать в одной строке.

2. Сформировать запрос *Запрос2* на получение списка студентов-мужчин, дата рождения которых находится в интервале от 1.01.1983г. до 31.12.1983г. Рекомендация: условия для выбора можно задать с помощью операторов **>=#1.01.83# And <= #1.01.83#** или **Between #1.01.83# And #31.12.83#**

### Построение запросов с параметром

Если приходится часто выполнять один и тот же запрос на выборку, изменяя при этом условия отбора, то сэкономить время позволяет запрос с параметрами. Запрос с параметрами не требует каждый раз вносить изменения в бланк запроса; вместо этого выводится приглашение пользователю ввести условия отбора. Для каждого поля допускается определение набора параметров.

Для каждого поля, которое предполагается использовать как параметр, введите в ячейку строки *Условие отбора* текст приглашения, заключенный в квадратные скобки. Это приглашение будет выводиться при запуске запроса. Текст подсказки должен отличаться от имени поля, но может включать его.

Например, сделаем запрос на отсрочку от армии (рис. 30) более универсальным: возраст призыва будет меняться в зависимости от времени призыва. Для этого выполните следующее задание.

### ЗАДАНИЕ 11. Построение запросов с параметром

1. Используя буфер обмена, скопировать в базу данных запрос *Отсрочка от армии на 2003год*. Дать ему новое имя *Отсрочка от армии*.

2. Открыть новый запрос в режиме Конструктора. В ячейке *Условие ввода* ввести: **<[Ввести дату призыва дд.мм.гг.]** (рис. 31 а).

3. Для предварительного просмотра запроса с параметрами до его сохранения нажмите кнопку  на панели инструментов и введите значение параметра. Для возвращения в режим конструктора запроса нажмите кнопку  на панели инструментов. При выполнении запроса программа Access открывает диалоговое окно с введенной вами в квадратных скобках подсказкой (рис. 31 б). В это окно необходимо ввести нужную величину.

4. Активизируйте запрос несколько раз, задавая разные значения параметра. Посмотрите, как меняются результаты выполнения запроса.

5. Скопируйте *Запрос2* из **ЗАДАНИЯ 10**, дав ему имя *Запрос3*. Откройте его в режиме Конструктора.

6. Для того чтобы вывести приглашения "Введите начальную дату:" и "Введите конечную дату:" для определения диапазона отбираемых значений, введите *Between* [Введите начальную дату :] *And* [Введите конечную дату:] в ячейку строки *Условие отбора* в столбце поля *Dtr* (*Дата рожд*). Поработайте с этим запросом.

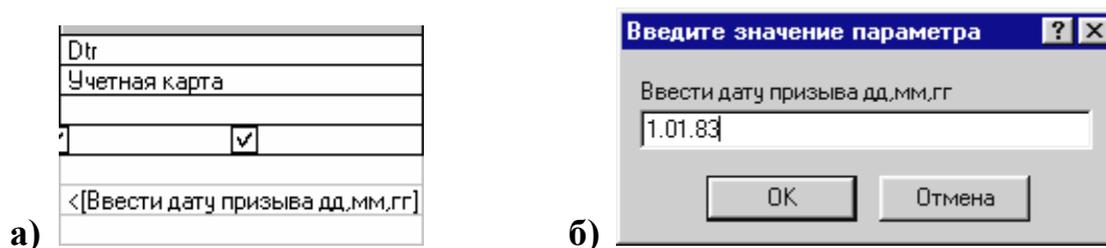


Рис. 31. а) Фрагмент бланка запроса, в котором параметром является условие отбора для поля *Dtr* (*Дата рожд*);  
б) Диалоговое окно для ввода параметра

Для того, чтобы находить в поле не конкретное значение, а только его фрагмент используют оператор *Like*. Например, если требуется отобрать студентов, фамилия которых начинаются на букву «А», в бланке запроса для поля *ФИО* можно записать условие *Like* "А\*".

7. Для таблицы *Ведомость* создать запрос, который выводил результаты экзаменов за I семестр для любой группы.

8. Для этого создайте новый запрос в режиме конструктора на основе запроса *Общая ведомость*. Добавьте поля *ФИО*, *Nst*, *1\_p*, *1\_e1*, *1\_e2*, *Ngr*. В последнем поле *Ngr* задайте условие отбора *Like* [Ввести номер группы]. Дайте ему название *Ведомость группы за I семестр*.

9. Проверьте, как работает данный запрос. Обратите внимание, что при вводе в поле параметра группы можно не соблюдать регистр букв и не надо набирать дефис. Например, чтобы получить ведомость группы А-2, надо ввести **а2**.

## Вычисления в запросах

Существует ряд вычислений, которые можно выполнить в запросе, например, найти сумму или среднее по значениям одного поля, перемножить значения двух полей или вычислить дату, отстоящую на три месяца от теку-

щей даты. Как результат, в таблице появляется новое *вычисляемое поле* – поле, определенное в запросе для вывода результата расчета выражения, а не для сохранения данных.

Значение вычисляемого поля пересчитывается при каждом изменении данных в выражении. Результаты вычислений, выводятся в поле, не запоминаются в базовой таблице. Вместо этого, вычисления снова производятся всякий раз, когда выполняется запрос, поэтому результаты всегда представляют текущее содержимое базы данных. Обновить вычисленные результаты вручную невозможно.

### Построение вычисляемых полей

Для расчетов с использованием формул, определяемых пользователем, требуется создать новое *вычисляемое поле* прямо в бланке запроса. Вычисляемое поле создается с помощью выражения, которое вводится в пустую ячейку поля в бланке запроса.

### ЗАДАНИЕ 12. Построение запросов с вычисляемыми полями

Построим простейший запрос, в котором выдается список группы и средний балл каждого студента (для II курса II семестра – это группы А-2 и Б-2).

1. Откройте окно запроса *Ведомость группы за I семестр* в режиме конструктора.

2. Уберите флажки в ячейке Вывод на экран для полей *Nst*, *Ngr*, *I\_e1*, *I\_e2*. В ячейку Поле свободного столбца поставьте курсор, вызовите контекстное меню и выберите команду Область ввода: в этом окне удобнее работать с длинными выражениями.

3. В Области ввода набрать выражение: **СреднийБалл: ([I\_e2]+[I\_e1])/2**. Закройте Область ввода. Поставить галочку в ячейке Вывод на экран вычисляемого поля: (рис. 32).

4. Запустите запрос (пункт 7 **ЗАДАНИЯ 9**), как работает данный запрос?

5. При проектировании таблиц мы оговаривали, что в таблице *Группы* количество студентов в группе и номер курса могут быть вычислены по уже имеющимся полям. Составьте самостоятельно запрос *Список групп*, в котором номер курса определяется в вычисляемом поле как разность между последней цифрой текущего года и последней цифрой номера группы. Например, для II полугодия 2004 года группа Б-1 будет отнесена к III курсу: (200)4-

(Б-)1=3. Причем, если берется I полугодие, то формула расчета должна включать добавление единицы. Этот факт реализуется *вводом параметра в выражение*:

**Курс: Val(Right\$(Date\$();1))-Val(Right\$([Ngr];1))+[Для I полугодия ввести 1, иначе 0]**

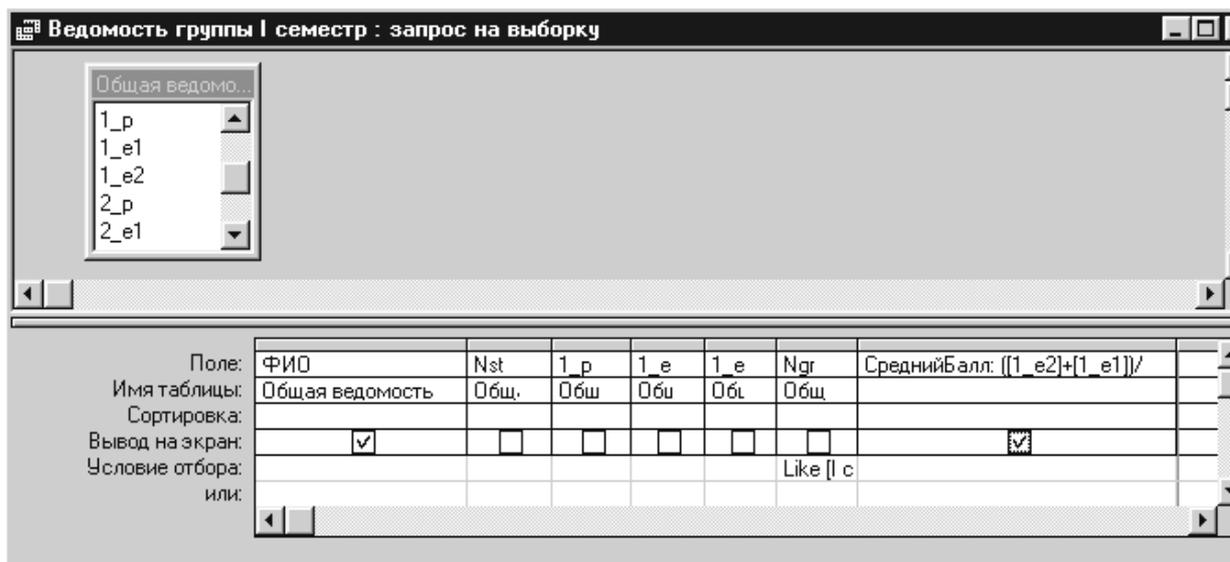


Рис. 32. Запрос с вычисляемым полем

6. Постройте указанный запрос на основе таблицы *Группы*. Вычисляемое поле поставьте после поля *Ngr*, используя команду горизонтального меню *Вставка>Столбец*. Проверьте, как работает запрос для разных значений параметра.

7. Используя выражение для вычисления номера курса пункта 5, создайте запрос *Список студентов по курсам* на основе таблицы *Учетная карта*, включив в него все поля, кроме *Phgr*. При этом модифицируйте выражение для вычисления курса так, чтобы полугодие вычислялось автоматически (для этого используйте функции *MID\$, DATE\$()*, чтобы определить текущий месяц года.)

### Групповые операции и вычисления. Итоговые запросы

При работе с базой данных не обойтись без запросов, позволяющих получить некоторую итоговую информацию, например, количество студентов в группе, количество студентов, имеющих задолженности за сессию, средний балл группы, сумму стипендии, выдаваемой за каждый месяц, процент преподавателей, имеющих ученую степень и т.д. Такие вычисления возможны благодаря поддерживаемым в программе Access средствам группировки за-

писей и встроенным статистическим функциям. Записи, результирующего набора организуются в группы по указанному полю (или полям), а затем к созданным группам могут быть применены различные итоговые вычисления.

Включить в запрос выполнение некоторых вычислений можно с помощью строки бланка запроса Групповая операция, которая изначально не видна в окне конструктора. Чтобы отобразить эту строку в режиме конструктора, следует выполнить команду Вид>Групповые операции.

Теперь в бланке запроса можно выбрать для любого поля нужный вариант вычислений. Групповые операции могут выполняться как для всех записей результирующего набора, так и для отдельных групп. Чтобы выполнить вычисления, для нужных полей в бланке запроса выбирается соответствующая опция из раскрывающегося списка строки Групповая операция.

Список опций поля Групповая операция включает 9 итоговых функций и три элемента: Группировка, Выражение, Условие.

- Группировка. Этот элемент указывает на поле, по которому результаты выполнения запроса будут организованы в группы для дальнейших итоговых вычислений.

- Sum. Суммируются все значения, содержащиеся в поле запроса.

- Avg. Вычисляется среднее арифметическое значение для всех чисел, содержащихся в выбранном поле.

- Min/Max. Отображается минимальное/максимальное из всех значений, содержащихся в поле запроса.

- Count. Вычисляется количество непустых значений в поле запроса.

- StDev. Вычисляется среднеквадратичное отклонение для значений в поле.

- Var. Вычисляется дисперсия распределения значений, содержащихся в указанном поле.

- First/Last. Отображается значение из первой/последней записи результирующего набора.

- Выражение. Этот элемент сообщает программе Access, что следует создать поле, значение которого будет вычисляться.

- Условие. Элемент, указывающий Access, что данное поле не участвует в группировке; условие отбора, вводимое в это поле, определяет какие записи будут участвовать в вычислениях.

### ЗАДАНИЕ 13. Построение итоговых запросов

В **ЗАДАНИИ 12** в пунктах 5 – 6 был создан запрос *Список групп*. Дополним этот запрос новыми вычисляемыми полями, в которых указано количество студентов в группе и средний балл по двум экзаменам.

1. Откройте запрос *Список групп* в режиме Конструктора.
2. Добавьте в верхнюю половину бланка запроса таблицу *Учетная карта*.
3. Поставьте курсор в поле *Instructor(Куратор)* и перетащите поле *Ngr* таблицы *Учетная карта*. Новое поле вставляется между имеющимися полями запроса.
4. Отобразите строку Групповая операция, выполнив команду горизонтального меню Вид>Групповые операции.
5. В ячейке нового поля строки Групповая операция щелкните на кнопке раскрывающегося списка справа от элемента Группировка и выберите опцию Count.
6. Сохраните запрос и посмотрите, как он работает. Обратите внимание, какое имя дает Access новому полю.
7. Перейдите в режим Конструктора и вызовите контекстное меню на вычисляемом поле. Перейдите к пункту Свойства и дайте новую Подпись **Количество**.
8. Теперь вычислите средний балл группы по каждому экзамену. Снова перейдите в режим Конструктора. Вставьте в верхнюю половину бланка запроса таблицу *Ведомость*. В свободные столбцы вставьте из новой таблицы поля с результатами экзаменов: *1\_e1, 1\_e2, 2\_e1, 2\_e2*.
9. Выберите среди групповых операций для этих полей функцию Avg. Посмотрите, как работает запрос.
10. Дайте новые подписи каждому полю (например, **СрБ\_1э1**) так, как это было сделано в пункте 7.
11. Измените формат вывода среднего балла так, чтобы высвечивался только один знак после десятичной запятой. Для этого воспользуйтесь пунктами **Формат (Фиксированный)** и **Число десятичных знаков (1)** из вкладки **Свойства поля**, вызываемой из контекстного меню.
12. Сохраните запрос и посмотрите результаты его работы. Они должны выглядеть так, как показано на рис. 33.
13. Самостоятельно постройте запрос по таблице *Преподаватели*, определяющий количество сотрудников по каждой должности. Используя ввод параметра, модифицируйте построенный запрос так, чтобы он определял количество сотрудников только одной категории.

Список групп : запрос на выборку									
	№ гр.	Курс	Кол-во	Куратор	Староста	Ср.Б_1э1	Ср.Б_1э2	Ср.Б_2э1	Ср.Б_2э2
▶	А-1	3	3	ГОЛОВАНОВА Н.И.	АРСЕНТЬЕВ В.Р.	4,3	3,3	4,0	4,0
	А-2	2	4	АРХИПОВА Е.Е.	БЕРЕЖНОЙ П.Р.	4,0	4,5	4,3	4,3
	Б-1	3	4	БЕДРОВ Р.Г.	АНТОНОВА Т.И.	3,8	3,8	3,5	4,0
	Б-2	2	5	ЕРЕМИН Р.А.	ДРАГУНОВ А.А.	4,2	4,0	4,0	4,0

Запись: 1 Всего: 4

Рис. 33. Результаты итогового запроса *Список групп*

### ЗАДАНИЕ 14. Построение итоговых запросов с использованием сложных условий отбора

Если возникает необходимость отфильтровать записи еще до выполнения вычислений, а условие отбора применяется не для поля группировки, в строке *Групповая* операция такого поля следует выбрать опцию *Условие*. Например, необходимо в каждой группе подсчитать количество студентов, сдавших экзамены за первый семестр без задолженностей и троек – количество «хорошистов». При этом вначале должны быть отобраны записи (студенты), удовлетворяющие указанному критерию, затем они должны быть сгруппированы по полю *№гр* и по каждой группе уже будет подсчитано количество попавших в нее студентов. Бланк такого запроса и результаты его работы представлены на рис. 34.

1. Создайте запрос, бланк которого показан на рис. 34. Обратите внимание, что программа Access автоматически снимает флажок опции *Вывод на экран* для полей, к которым применено условие отбора. Дайте запросу имя *Хорошисты групп*

2. Модифицируйте запрос *Хорошисты групп* так, чтобы он выдавал данные только для одной группы, т.е. введите параметр для названия группы.

3. Создайте запрос, определяющий в каждой группе количество студентов, имеющих задолженности за I семестр.

4. Создайте запрос, определяющий количество студентов, претендующих на повышенную стипендию (сдавших зачет и имеющих оценки «5» по двум экзаменам, например, по предметам 1\_э1, 2\_э2).

5. Создайте запрос, определяющий список студентов II курса (группы А-2 и Б-2), которые могут получать стипендию. Критерий для отбора таких студентов следующий: это студенты, которые обучаются на бюджетной основе и не имеют задолженностей, оценки по двум экзаменам II семестра (2\_э1 и 2\_э2) «4» и «5» ИЛИ имеют льготы (в поле *Lgoty* – да), которые дают им возможность получать стипендию, если среди оценок на экзаменах есть тройки. Дайте запросу имя *Список II курса на стипендию*.

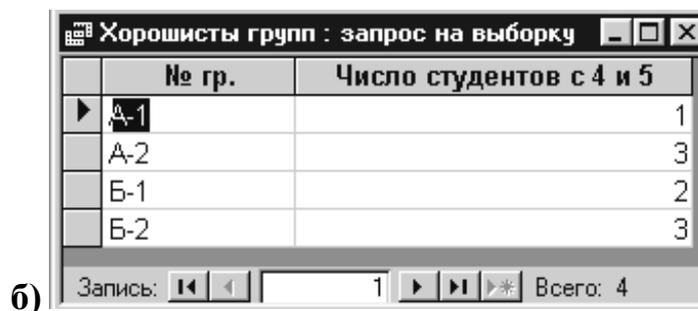
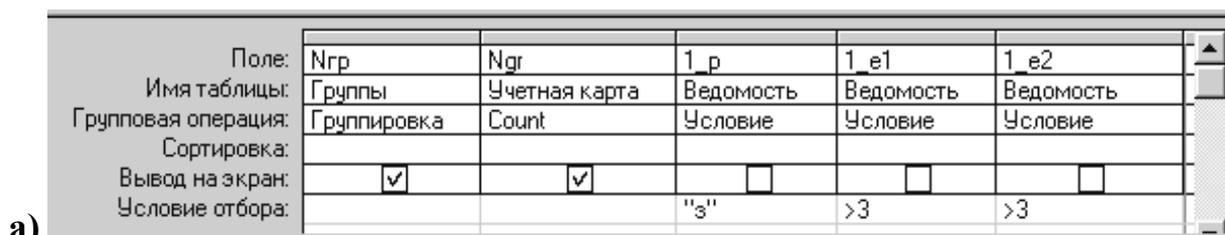


Рис. 34. а) Бланк запроса, в котором критерий отбора записей будет применен до выполнения вычислений. б) Результаты работы запроса: показано, сколько «хорошистов в каждой группе»

6. На основе запроса *Список на стипендию II курса* создайте запрос, определяющий количество студентов в каждой группе, получающих стипендию. Дайте полю с количеством студентов подпись *Количество*, а самому запросу – имя *Стипендия II курса*. Для реализации группировки необходимо в поле запроса добавить таблицу *Учетная карта*, которая даст возможность использовать имеющиеся между таблицами связи. (Поэкспериментируйте – посмотрите, как работает запрос, когда есть таблицы *Учетная карта*, и когда она отсутствует).

7. Усложним запрос, включив в него возможность подсчитывать общую сумму стипендии, выделяемой на каждую группу. Размер стипендии ввести как параметр. Для того, чтобы в выражении для стипендии записать формулу, необходимо ввести имя поля *Count\_Ngr*, в котором подсчитывается количество студентов в группе, получающих стипендию. Поскольку в бланке запроса это имя не видно (оно выдается только при просмотре результатов запроса, да и то в случае, если не дана другая подпись полю), то для записи выражения лучше воспользоваться Построителем выражений и взять требуемое поле из раскрывающегося списка полей запроса *Стипендия II курса* (рис. 35).

Как видно из этого рисунка, размер стипендии вводится параметром в само выражение, её значение будет запрашиваться при запуске запроса. Для того, чтобы запрос подсчитывал стипендию согласно построенному выражению, необходимо в строке *Групповая операция* поля *Стипендия* выбрать опцию *Выражение*.

8. Посмотрите, как работает созданный Вами запрос.

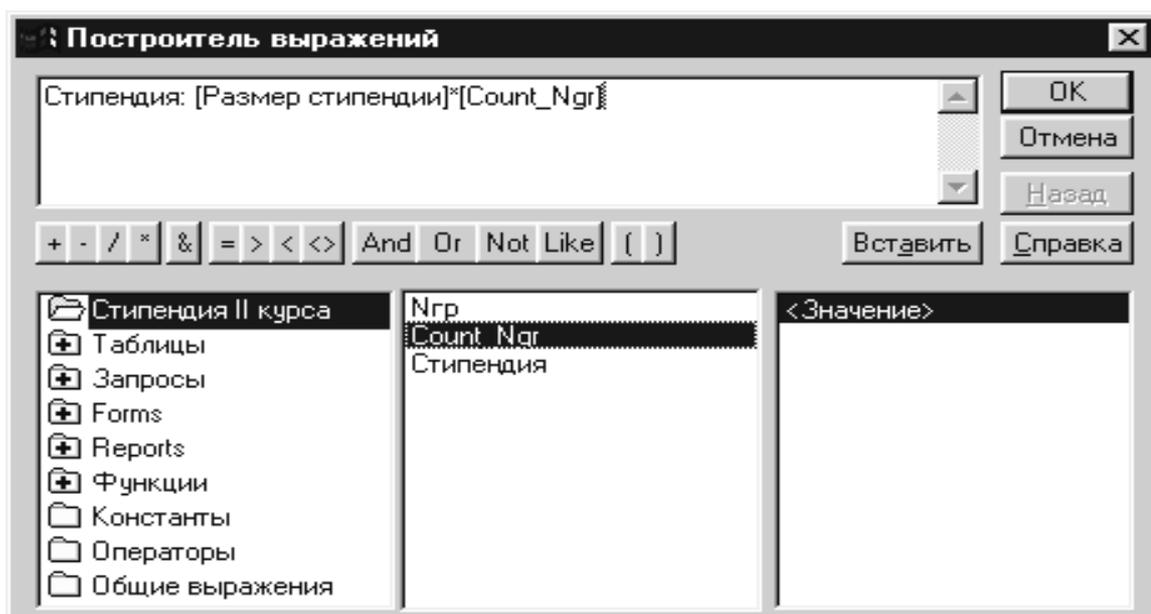


Рис. 35. Использование Построителя выражений для создания вычисляемого поля *Стипендия* на основе поля группировки *Count\_Ngr*. Обратите внимание, что запрос *Стипендия II курса*, открытый в базе в режиме конструктора, показан в первом столбце построителя отдельно от всех других запросов, что очень удобно при построении выражения.

### ЗАДАНИЕ 15. Построение сложных запросов с условиями отбора для вычисляемых полей

В запросе *Стипендия II курса* размер стипендии задавался одинаковым для всех категорий студентов. Во многих ВУЗах практикуется дифференцированная стипендия для разных категорий студентов. Допустим, что размер базовой стипендии равен N руб. Для отличников стипендия увеличивается вдвое, а для тех, кто получает стипендию, имея тройки, уменьшается вдвое. Составим запрос, в котором выдается список всех студентов II курса (группы А-2, Б-2) за II семестр (2\_з, 2\_э1, 2\_э2) с указанием начисленной стипендии. Создайте такой запрос по следующей схеме.

1. Создайте новый запрос в режиме Конструктора. Добавьте в бланк запроса таблицы *Учетная карта* и *Ведомость*.

2. Перенесите в нижнюю половину запроса поля *ФИО*, *Ngr*, *Форма опл*, *Льготы* из таблицы *Учетная карта* и поля *2\_з*, *2\_э1*, *2\_э2* из таблицы *Ведомость*.

3. В ячейке *Условие Отбора* поля *№гр* укажите выбор групп II курса любым способом, например, используя операторы **In ("A2";"B2")** или **"A2" OR "B2"**. (Можете предложить другой способ определения студентов одного курса?)

4. В новом столбце создайте вычисляемое поле логического типа *K*, определяющее что студент может получать базовую стипендию, а именно: он обучается на бюджетной основе, не имеет задолженностей за сессию и оценки за экзамены выше «3». Выражение, соответствующее данному критерию, можно записать так:

**К: [Paym]="б" And [2\_p]="з" And ([2\_e1]=4 Or [2\_e2]=4) And [2\_e1]>3 And [2\_e2]>3**

5. По аналогии в новом столбце запишите выражение для вычисляемого поля логического типа *L*, определяющее что студент может получать повышенную стипендию: он обучается на бюджетной основе, не имеет задолженностей за сессию и все оценки за экзамены «5».

6. В следующем столбце запишите выражение для вычисляемого поля логического типа *M*, определяющее что студент может получать половину базовой стипендии: он обучается на бюджетной основе, имеет льготы, сессию сдал без задолженностей и без двоек, но имеет тройки.

7. В следующем новом столбце создайте вычисляемое поле денежного типа *Стипендия*, указав в Свойствах поля формат поля *Денежный* с числом знаков 2. Выражение для этого можно записать, используя функцию управления *lif*. Это выражение будет достаточно громоздким, а именно:

**Стип: If([K]=-1; [Стипендия]; If([L]=-1; [Стипендия]\*2; If ([M]=-1; [Стипендия]/2; 0)))**

8. Посмотрите результаты работы запроса (рис. 36). Знак «-1» в логических полях *K, L, M* свидетельствует о том, к какой категории относится данный студент.

9. Сохраните запрос, присвоив ему имя *Дифференцированная стипендия II курса*.

10. Создайте с помощью Конструктора новую таблицу *Активисты факультета* (рис. 37). В этой таблице представлен список студентов, занимающихся общественной работой.

Категория этой работы оценивается как «1», если работа соответствует уровню факультета, и «0», если – уровню группы. Если категория общественной работы «1», то к стипендии добавляется *M* % от стипендии, которую получает данный студент.

11. Свяжите новую таблицу с таблицей *Учетная карта*, не устанавливая целостность данных.

Дифференцированная стипендия II курс : запрос на выборку											
ФИО	№ гр	Форма опл.	Льготы	2_з	2_з1	2_з2	К	L	M	Стип	
▶ БЕЗНОГОВ С.Ю.	Б-2	К	<input type="checkbox"/>	з	5	5	0	0	0	0,00р.	
БЕРЕЖНОЙ П.Р.	А-2	К	<input type="checkbox"/>	з	4	3	0	0	0	0,00р.	
ГАЛКИН И.Д.	А-2	Б	<input type="checkbox"/>	з	5	5	0	-1	0	800,00р.	
ДРАГУНОВ А.А.	Б-2	Б	<input type="checkbox"/>	з	4	4	-1	0	0	400,00р.	
ЕЛИСЕЕВА И.В.	Б-2	Б	<input checked="" type="checkbox"/>	з	3	5	0	0	-1	200,00р.	
ЛЕОНТЬЕВ М.В.	А-2	Б	<input type="checkbox"/>	н	4	4	0	0	0	0,00р.	
РОМАНОВА С.Л.	Б-2	Б	<input checked="" type="checkbox"/>	з	2	3	0	0	0	0,00р.	
СЕМЕНОВ А.Л.	А-2	Б	<input checked="" type="checkbox"/>	з	4	5	-1	0	0	400,00р.	
ЯКОВЛЕВ С.И.	Б-2	Б	<input type="checkbox"/>	з	3	3	0	0	0	0,00р.	
*			<input type="checkbox"/>								

Запись: 1 Всего: 9

Рис. 36. Результаты работы запроса на вычисление дифференцированной стипендии. Значение «-1» в логических полях *K, L, M* является показателем для начисления стипендии в зависимости от категории студента

Активисты факультета : таблица		
ФИО	Обществ. работа	Козф.
▶ АНДРЕЕВА О.С.	Культорг группы	0
АНТОНОВА Т.И.	староста Б-1	1
АРСЕНТЬЕВ В.Р.	староста А-1	1
БЕРЕЖНОЙ П.Р.	староста А-2	1
ВЕРХОВ А.В.	Профорг факультета	1
ДРАГУНОВ А.А.	староста Б-2	1
ЛЕОНТЬЕВ М.В.	Отвественный группы по общежити	0
РОМАНОВА С.Л.	Член сборной команды факультета	1
*		

Запись: 1 Всего: 8

Рис. 37. Таблица *Активисты факультета*

12. Модифицируйте запрос *Дифференцированная стипендия II курса* так, чтобы в нем вычислялась дополнительная стипендия, если категория общественной работы оценивается «1», и выдавалась суммарная стипендия (Рис. 38).

При введении вычисляемых полей не забудьте указать денежный формат полей.

ФИО	№ гр	Форм	Льготы	2_з	2_з1	2_з2	Кате	К	L	M	Стип	Надба	Сумма
ДРАГУНОВ А.А.	Б-2	Б	<input type="checkbox"/>	э	4	4	1	-1	0	0	400р.	40р.	440р.
РОМАНОВА С.Л.	Б-2	Б	<input type="checkbox"/>	э	2	3	1	0	0	0	0р.	0р.	0р.
ЛЕОНТЬЕВ М.В.	А-2	Б	<input type="checkbox"/>	н	4	4	0	0	0	0	0р.	0р.	0р.

Запись: 1 Всего: 3

Рис. 38. Результаты работы модифицированного запроса  
Дифференцированная стипендия II курса для случая 10 % надбавки  
к стипендии за общественную работу

Итак, нами был рассмотрен наиболее эффективный механизм получения актуальной информации из базы данных – запросы. В качестве источника данных для запросов могут служить таблицы и/или другие запросы.

Для создания сложных запросов можно использовать средства Конструктора запросов. Диалоговое окно Конструктора представляет пользователю бланк запроса, в котором можно выбирать поля из нескольких таблиц. При этом программа Access автоматически распознает установленные в базе данных межтабличные связи и отображает их в окне Конструктора в виде линии объединения.

Были также рассмотрены возможности использовать выражения при указании условий отбора записей, создании вычисляемых полей и т.д. При указании условий отбора записей можно вводить дополнительные критерии, комбинируя их с помощью логических операторов. Можно выполнять групповые вычисления как для всех записей результирующего набора, так и для отдельных групп.

Параметрический запрос позволяет варьировать условие отбора записей, не изменяя структуры запроса: при каждом запуске запроса пользователю предлагается ввести параметр (один или несколько) в специальном диалоговом окне.

### Контрольные вопросы

1. Для чего служат запросы на выборку?
2. Почему запросы не занимают в базе данных много места?
3. Перечислите все способы включения полей таблицы в бланк запроса.
4. Какие виды запросов были использованы при конструировании БД Факультет?

5. Как с помощью запроса провести сортировку записей по какому-либо полю?
6. Для чего в запрос вставляют ссылки на таблицы?
7. Как включают в запрос условия отбора?
8. Для чего служит оператор Like?
9. Какие преимущества дает запрос с параметром?
10. Что такое выражения и функции?
11. Как строится выражение в Access?
12. Какие действия допустимы с полями текстового типа? логического типа?
13. Как работает функция управления If?
14. Какие операции допустимы в вычисляемых полях? Сохраняются ли значения таких полей в файле базы данных?
15. Для чего используется Построитель выражений?
16. Для чего предназначена операция группировки? В чем суть итогового запроса?
17. Какие итоговые функции вы знаете?

### **Оформление отчета**

Отчет о лабораторной работе должен содержать:

1. Описание в тетради основных этапов проделанной работы, их логическую последовательность.
2. Ответы на контрольные вопросы.

## Лабораторная работа № 3 «СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ФОРМ И ОТЧЕТОВ»

### Цель работы<sup>2</sup>

Данная лабораторная работа предназначена для приобретения начальных навыков работы по созданию форм и отчетов в реляционной базе данных, а именно:

- а) построение автоформы и автоотчета,
- б) знакомство с режимом Мастеров формы и отчета,
- в) знакомство со структурой формы и отчета в режиме Конструктора,
- г) изучение приемов работы со связанными записями, объектами OLE,
- д) знакомство с приемами создания оболочек для баз данных.

### Назначение формы

Такие объекты базы данных, как формы, предназначены в первую очередь для работы одновременно только с одной записью. Формы избавляют от неудобств, которые неизбежно возникают при работе с таблицами и запросами, имеющими большое количество полей. В режиме формы можно всё внимание уделить одной записи, не пользуясь прокруткой для поиска нужного поля в длинной череде столбцов таблицы, а имея перед глазами всю информацию, например, по конкретному студенту или преподавателю. Экранные формы можно сделать точной копией бумажных бланков, с которых происходит ввод данных – анкет, заявлений, ведомостей и т.п.

Как и другие объекты Access, формы можно создавать вручную или автоматически, причем, несколькими способами. Если при изучении таблиц и запросов целесообразно было не пользоваться автоматическими средствами, то с формами дело обстоит несколько иначе. Они состоят из многочисленных элементов управления, и от того, насколько аккуратно эти элементы расположены на экране, зависит внешний вид формы. Автоматические средства позволяют создавать аккуратные формы и не задают пользователю лишних вопросов. Поэтому работу лучше начинать с них.

---

<sup>2</sup> Для выполнения Лабораторной работы № 3 необходимо наличие в базе пяти таблиц и запросов *Общая ведомость, Список студентов по группам, Дифференцированная стипендия II курса*, созданных в Лабораторных работах №1, 2

## Использование автоформы

Автоформы – самый простой вид автоматических форм. Автоформа помещает в новую форму все без исключения поля исходной таблицы или запроса. Для нашей базы данных *Факультет* автоформу можно использовать для ввода новых записей. Допустим, необходимо в базу данных ввести данные об абитуриентах, успешно сдавших вступительные экзамены и зачисленных на I курс. Номера студенческих билетов начинаются с цифры 3-01 и т.д. Создав автоформу, введите в таблицу несколько (пока не больше 1 – 2) новых записей с данными о студентах вашей группы, разделив их по вашему усмотрению на 2 группы. Для 2003/2004 учебного года это будут студенты групп А-3, Б-3.

### ЗАДАНИЕ 16. Добавление в базу данных записи, используя автоформу

1. Открыть вкладку *Формы* в окне базы данных и щелкнуть на кнопке *Создать* – откроется окно *Новая форма* (рис. 39), в котором доступно несколько вариантов форм.

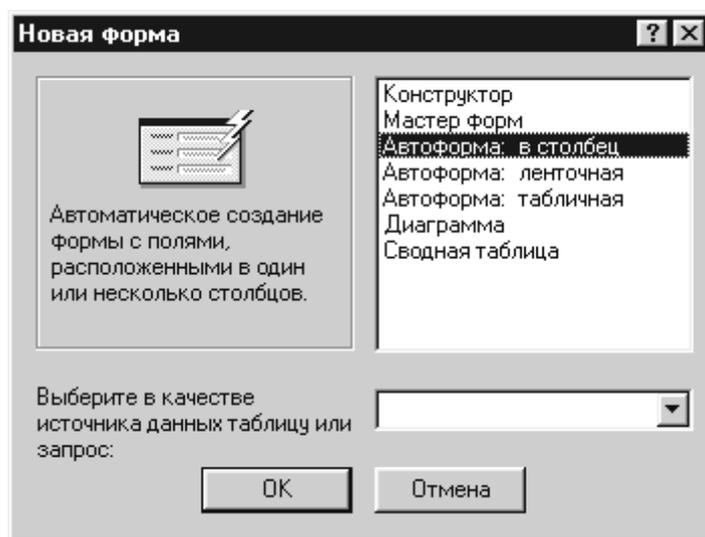


Рис. 39. Диалоговое окно *Новая форма* предлагает разные способы создания формы

2. Из списка, предложенного в окне *Новая Форма*, выберите подходящий тип автоформы (например, форма «в столбец») и щелкните на нем.

3. Из раскрывающегося списка, расположенного в нижней части окна, выберите имя таблицы или запроса, для которых вы хотите создать форму, и щелкните на нем, а затем – щелкните на кнопке ОК. В данном случае выберите таблицу *Учетная карта*.

4. На экране появится выбранный вами вид формы, и, если в таблице уже имеются записи, в форме появятся данные. Используя клавиши прокрутки <PageUp>, <PageDown>, можно просматривать каждую запись.

5. Пролистайте все записи до конца, пока в форме не появится пустая (новая) запись. Или нажмите кнопку  в строке Запись (внизу формы), чтобы сразу перейти к новой записи. Введите в нее данные об одном студенте.

6. Для перехода к следующей записи щелкните на кнопке  в строке Запись. Таким образом введите еще одну-две записи.

7. Сохраните форму с именем *Учетная карта*.

8. Перейдите в окно Таблицы, откройте таблицу *Учетная карта* и убедитесь, что в таблице появились новые записи.

9. Для того, чтобы ввести фамилии первокурсников в таблицу *Ведомость*, создайте запрос *Новые студенты*, в котором бы выводились фамилии этих студентов. Критерий отбора выберите самостоятельно: им может быть номер студенческого билета, начинающийся с цифры «3», или номер группы, заканчивающийся на «3». В случае первого выбора, можно порекомендовать поработать с функцией VAL(Nst), которая переведет символьное значение номера в числовое значение, тогда выбор можно будет реализовать сравнением с числом, например, для 2003/04 учебного года это 300.

10. Откройте запрос *Новые студенты* и таблицу *Ведомость* и выведите их на экран Слева направо. Затем, используя буфер обмена, введите фамилии новых студентов в таблицу *Ведомость*.

11. На основе таблицы *Учетная карта* создайте ленточную автоформу и оцените, какой вид формы удобнее для работы с данной таблицей.

## Создание форм с помощью Мастера

Формы и отчеты довольно сложные структуры баз данных, поэтому, обычно, их создают в режиме **Мастера** и дорабатывают в режиме **Конструктора**. В отличие от автоформы, Мастер создания форм готов учесть пожелания пользователя при выборе полей, макета и стиля отображения будущей формы. Мастер, кроме того, позволяет создавать формы, в которых пред-

ставлены поля из нескольких таблиц. В качестве примера, осуществим добавление записей в таблицы *Учетная карта* и *Ведомость*. Эта форма избавит от необходимости заполнять поля таблицы через буфер обмена.

### ЗАДАНИЕ 17. Добавление в базу данных записей, используя форму, созданную с помощью Мастера

1. В окне Формы окна базы данных щелкните на кнопке Создать, чтобы вызвать диалоговое окно Новая форма. В нем выберите Мастер форм.

2. Выберите из раскрывающегося списка Таблицы и запросы таблицу *Ведомость*. Список в области Доступные поля заполнится полями указанной таблицы (рис. 40).

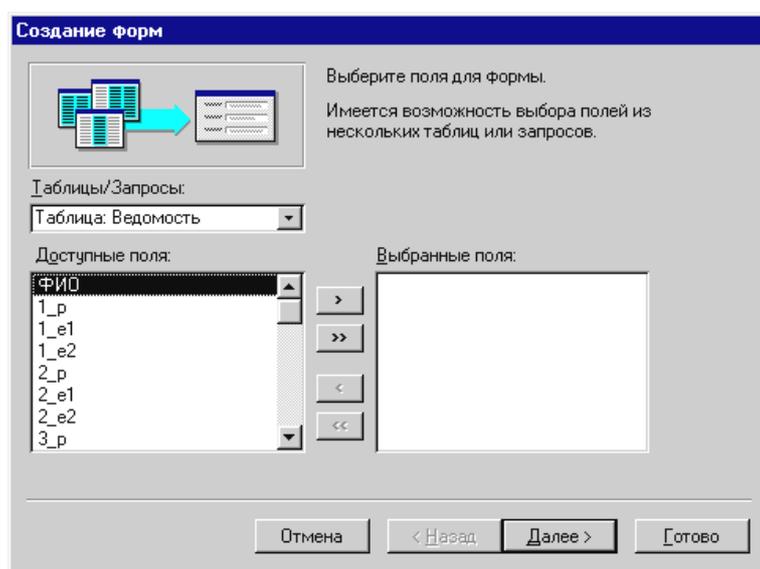


Рис. 40. В окне Мастера форм можно выбрать для будущей формы поля из нескольких таблиц

3. Используя кнопку со значком  или щелкнув дважды на имени поля *ФИО*, переместите его в список *Выбранные поля*.

4. Для включения полей из таблицы *Учетная карта* выберите ее имя из раскрывающегося списка. Все поля этой таблицы можно перенести за один раз, если нажать на кнопку со значком .

5. После того, как выбраны нужные поля, нажмите кнопку *Далее*. Новое окно Мастера предлагает выбрать внешний вид формы. Выберите тип *В один столбец*.

6. Следующее окно Мастера предлагает выбрать стиль для формы. Щелкнув по одному из стилей, можно предварительно просмотреть их, прежде чем выбрать наиболее подходящий. Нажав на кнопку Далее, перейдите к заключительному этапу создания формы – откройте форму для предварительного просмотра, дайте форме название *Учетная карта1* и щелкните на кнопку Готово. Мастер, создавая формы, автоматически присваивает им имена такие же, как и у таблиц. Если понадобится **изменить название**, можно воспользоваться командами меню Правка > Переименовать.

7. Новая форма похожа на ту, что была создана в **ЗАДАНИИ 16**, но у нее есть дополнительное поле *ФИО* из таблицы *Ведомость* (рис. 41). Теперь вводите данные.

8. Перейдите к новой записи (кнопка ) и введите значение в поле *ФИО* таблицы *Ведомость*. Обратите внимание, как только вы нажмете клавишу Enter, в поле *ФИО* таблицы *Учетная карта* появилась та же запись. Таким образом, вводя данные в ключевое поле одной таблицы, используя построенную форму, вы одновременно вводите эти же данные в другую таблицу, связанную с ней по этому полю.

Рис. 41. В форме *Учетная карта1* два поля *ФИО* из разных таблиц

9. Дополните список I курса еще 4 – 5 студентами вашей группы. Перейдите к просмотру таблиц и убедитесь, что в них появились новые записи.

## Конструктор форм

Если внешний вид формы далек от совершенства, заголовки полей оказались усеченными и еще многое в форме требует доработки, внести изменения в макет формы можно в режиме конструктора форм. С формой можно

работать в 3-х режимах. Переключение режимов можно осуществить через меню Вид, или через панель инструментов (рис. 42).

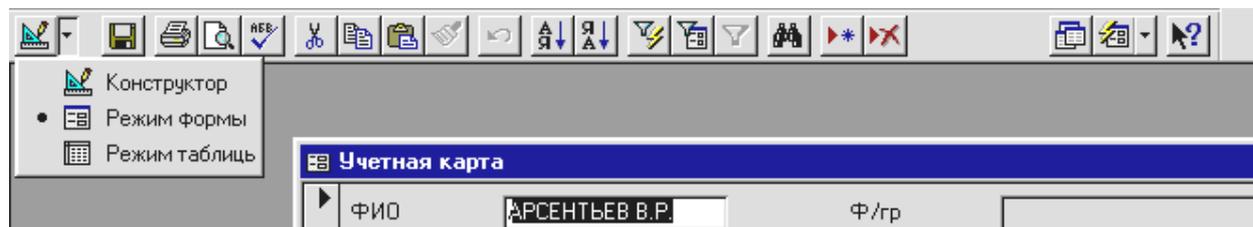


Рис. 42. Варианты представления формы

В режиме Конструктора возможно создание и изменение любых элементов формы. Для проверки "поведения" элемента необходимо переключиться в режим Формы. Режим Таблицы используется для табличного отображения источника записей формы (Элементы формы, такие как: кнопки, заголовок и примечание формы, рамки объектов, в этом режиме на экран не выводятся).

### Структура формы

Структуру формы составляют ее *разделы*, а разделы содержат *элементы управления*. Самый простой способ познакомиться с разделами формы состоит в том, чтобы посмотреть устройство уже готовой формы, например, *Учетная карта*, в режиме Конструктора (рис. 43). Рядом с формой в этом режиме открывается *панель элементов*, содержащая заготовки и инструменты для создания элементов управления формы.

В структуре формы видны три раздела: *раздел заголовка формы*, *область данных* и *раздел примечания* формы. В нашем случае заполнена только область данных.

Все, что содержится в области данных, является элементами управления. В форме *Учетная карта* присутствуют элементы управления только двух типов: *связанное поле* (то, что в него вводится, поступает и в одноименное поле таблицы, на базе которой создана форма) и *присоединенная надпись* (называется так, поскольку перемещается вместе со своим элементом управления). В данном случае содержание присоединенной надписи совпадает с названием связанного поля, но это можно изменить.

Фоновый рисунок, лежащий под элементами показывает размер рабочего поля. Размеры разделов и размеры рабочего поля можно менять методом перетаскивания.

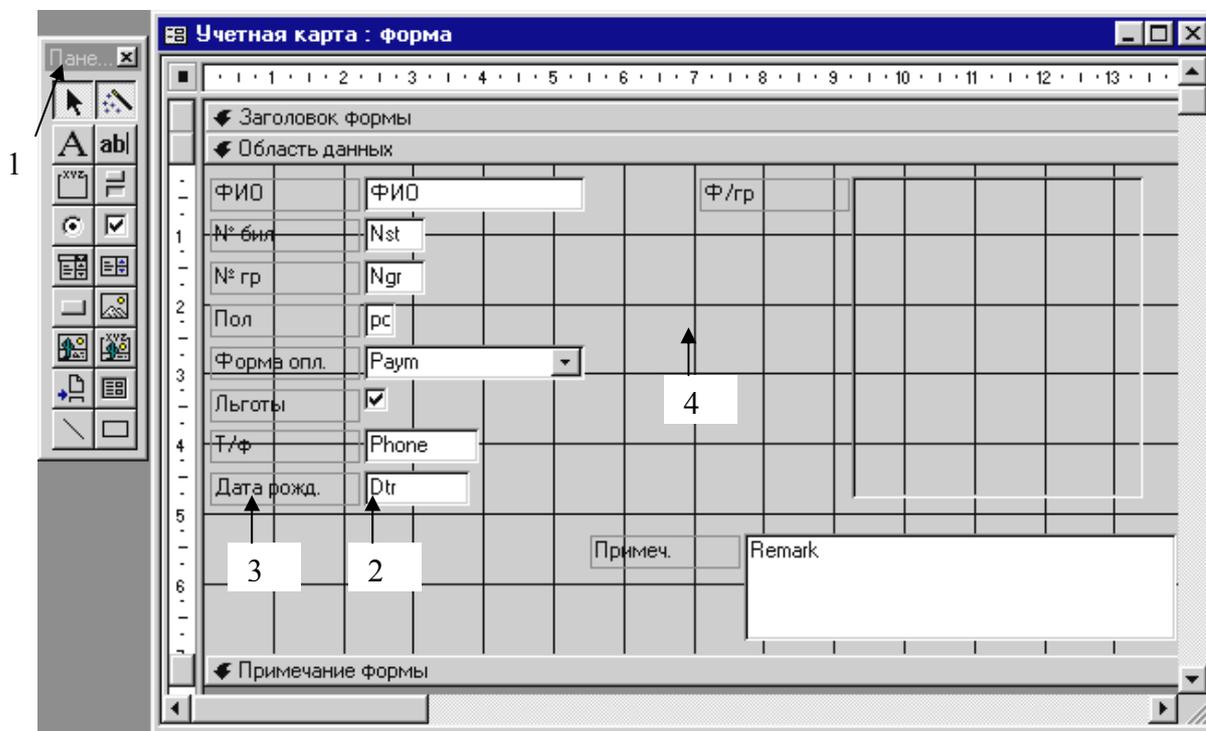


Рис. 43. Структура формы в режиме Конструктора. 1 – панель элементов, 2 – связанное поле, 3 – присоединенная надпись, 4 – фоновый рисунок

### Создание и редактирование связанных записей

Редактирование форм состоит в создании новых или изменении имеющихся элементов управления, а также в изменении их взаимного расположения. Усовершенствуем созданную Мастером форму *Учетная карта1*: дадим ей название, изменим границы связанных полей и названия присоединенных надписей.

#### **ЗАДАНИЕ 18. Усовершенствование вида формы *Учетная карта1***

1. Создание надписи к форме. Откройте форму в режиме Конструктора. Перетащите вниз разделительную границу между заголовком и областью данных. Щелкнув на значке **A** (Надпись) панели элементов, а затем на форме, получите текстовую рамку, в которую введите текст **Учетная карта студента**.

2. Для форматирования элемента выделите объект, щелкнув по нему мышью. При выделении элемента управления вокруг него появляется рамка с восемью маркерами (по углам и по центрам сторон рамки). Рамку можно сжимать или растягивать методом протягивания границ. При наведении на маркер указатель мыши меняет форму, принимая изображение открытой ладони. В этот момент рамку можно перемещать.

Когда объект выделен, измените параметры шрифта, размера, выравнивание текста Надписи, доступные через панель инструментов «Форматирование».

3. Редактирование связанных полей отличается от редактирования свободного поля Надписи. Присоединенная надпись перемещается вместе со связанным полем и образует с ним единое целое. Чтобы оторвать поле от присоединенной надписи, используйте левый верхний маркер рамки. При наведении на него указатель мыши принимает форму указательного пальца. В этот момент связанное поле можно оторвать от надписи и перемещать отдельно. Оторвите оба поля *ФИО* от их надписей и выровняйте их друг под другом.

4. Уменьшите размеры рамки поля для фотографии. Надпись **Фото** уберите, выделив ее и нажав клавишу DEL.

5. Уменьшите ширину поля *Форма opl*, размер Мемо-поля *Remrk*. Надпись к этому полю сделайте без сокращения **Примечания**.

6. Перейдите в режим Формы и посмотрите, как выглядит видоизмененная форма. Закройте форму, сохранив изменения.

7. Подготовьте ленточную форму для ввода результатов экзамена 1\_э1 для любой группы на основе запроса *Общая ведомость*. В режиме Конструктора придайте форме вид, показанный на рис. 44.

ФИО	№ бил	оценка
БЕЗНОГОВ С.Ю.	2-06	5
ДРАГУНОВ А.А.	2-03	4
ЕЛИСЕЕВА И.В.	2-04	5
РОМАНОВА С.Л.	2-07	3
ЯКОВЛЕВ С.И.	2-05	4

Рис. 4. Ленточная форма, построенная на основе запроса *Общая ведомость*. Обратите внимание, если изменять имя формы, в заголовке (строка заголовка) остается имя базового запроса

## Свойства полей. Внедрение и просмотр полей типа OLE

Для большинства объектов Windows есть возможность задания неких параметров (свойств). Самый простой способ для их изменения – это вызвать диалоговое окно щелчком правой клавиши мыши на заданном объекте – контекстное меню. Собственно, именно так задавались свойства полей таблицы, запроса в предыдущих заданиях. Работая с формой, можно также определять характеристики ее полей, выполнять активные действия при возникновении разных событий, например, проверять введенное значение, и т.д.

Как известно, поле объекта OLE служит для размещения внешнего объекта, соответствующего принятой в Windows концепции связывания и внедрения объектов. Объектом, как правило, является иллюстрация, для нашей базы данных – фотография студента.

Существует два типа полей для размещения объектов OLE. *Свободная рамка* объекта и *Присоединенная рамка* объекта. В первом случае рамка не связана ни с каким полем таблиц базы данных. Объект, находящийся в ней, выполняет роль иллюстрации и служит для оформления формы. С присоединенной рамкой связано одно из полей таблицы. В ней отображается содержимое этого поля. Это содержимое может меняться при переходе от одной записи к другой.

### ЗАДАНИЕ 19. Вставка фотографий студентов в таблицу *Учетная карта1*

Для выполнения этого задания создайте на диске в папке, где хранится БД *Факультет*, любой рисунок с помощью приложения Paint или преобразуйте файл, содержащий отсканированную фотографию, в файл с расширением .bmp

1. Откройте в режиме Конструктора форму *Учетная карта1*. Подведите курсор к связанному полю *Ф/гр* и, вызвав контекстное меню, выберите пункт *Свойства*.

2. На закладке *Макет* установите свойство *Установка размеров* – *По размеру рамки*.

3. Сохраните изменения и перейдите в режим *Формы*.

4. Вызовите контекстное меню на рамке объекта в форме и вберите команду *Добавить объект*. Далее выбрать команду *Создать из файла* и через кнопку *Обзор* выбрать папку, в которой находится рисунок с фотографией. После указания конкретного файла, нажать *ОК* – в поле появится созданный вами (или отсканированный) рисунок.

5. Перейдите к следующей записи и повторите действия пункта 5. с командой **Создать новый**, выбрав любой рисунок из MS Clip Gallery .

### Создание поля со списком

При конструировании таблиц мы уже задавали поле со списком для фиксированного значения списка. Формы позволяют осуществлять выбор из списка, который является столбцом какой либо таблицы. Используем это свойство для усовершенствования ввода данных в таблицу *Группы*. Ранее при заполнении этой таблицы данными фамилии старост и кураторов заносились через буфер обмена или вручную, что, естественно, не только неудобно, но и может привести к ошибкам в данных. Рассмотрим другой, более эффективный, способ заполнения таблиц.

### ЗАДАНИЕ 20. Создание поля со списком в форме *Группы*

1. Создайте автоформу в столбец для таблицы *Группы*. Откройте форму в режиме Конструктора.
2. Выделите поле *Куратор* и удалите его, используя клавишу <Delete>.
3. Из панели элементов щелчком мыши выберете значок  (Поле со списком) .
4. Подведите курсор мыши к предполагаемому месту вставки нового поля (учтите, что слева от него автоматически вставится заголовок). Нажмите левую клавишу мыши, и удерживая ее, растяните объект как рисунок (после того, как вы отпустите ее, у вас появится первое окно диалога).
5. Ознакомьтесь с ним и установите флажок напротив первой опции *Таблица или запрос...*. Нажав клавишу *Далее*, перейдите к следующему диалоговому окну. В нем выберите таблицу *Преподаватели*, откуда будут выбираться фамилии кураторов.
6. В следующем диалоговом окне выберите поле *ФИО*, благодаря которому можно будет осуществлять выборку.
7. Со следующим диалоговым окном просто ознакомьтесь.
8. В следующем выберите опцию *Сохранить* в поле и, используя кнопку вызова списка, выберите из него поле *Instructor*.
9. В последнем окне задайте подпись созданного поля **Куратор**.
10. Закройте форму с сохранением. Откройте ее и произведите ввод новой записи для групп А-3 и Б-3, выбрав кураторов из списка преподавателей.
11. Создайте аналогичное поле со списком для поля *Староста*, взяв для него два поля *ФИО* и *Ngr* из таблицы *Учетная карта*.

## Создание отчетов базы данных

Отчет в Access создается почти так же, как форма. Разница между этими двумя объектами базы данных состоит не в технологии их построения, а в их целевом назначении. Если с формой работают на экране, то отчет предназначен для печати и представляет ценность как материальная копия документа, полученного из базы данных.

Способы формирования отчета аналогичны способам построения форм: они могут быть созданы с помощью

- автоотчета,
- Мастера,
- конструктора,
- запросов.

### Использование функций автоотчета

*Автоотчет* – самый простой инструментарий, с помощью которого можно начать знакомство с отчетами. Все, что он позволяет сделать – это указать таблицу или запрос, которые должны стать источником данных для создаваемого отчета, а также выбрать макет будущего автоотчета («в столбец» или «ленточный»). Возможность выбора полей не предусмотрена: в автоотчет помещаются все поля исходной таблицы или запроса.

### ЗАДАНИЕ 21. Построение автоотчета для таблицы *Учетная карта*

1. В окне базы данных перейти на вкладку Отчеты. Нажать кнопку Создать на панели инструментов. В появившемся диалоговом окне Новый отчет выбрать категорию Автоотчет в столбец, а из раскрывающегося списка в нижней части окна таблицу *Учетная карта*. После завершения выбора нажать клавишу ОК.

2. На экране отобразится новый отчет в выбранном формате в режиме «Предварительный просмотр» (рис. 45).

3. Если Вас устраивает внешний вид нового объекта, выполните команду Файл>Сохранить.

Если внешний вид отчета не удовлетворяет пользователя, отчет можно доработать в режиме Конструктора, например, так же, как это было сделано с формой в **ЗАДАНИИ 18**.

<i>Учетная карта</i>	
<i>ФИО</i>	АНДРЕЕВА О.С.
<i>№ бил</i>	1-01

Рис. 45. Фрагмент отчета *Учетная карта* в режиме предварительного просмотра

### Использование Мастера отчетов

Мастер создания отчетов – это более гибкий инструмент создания внешнего вида и стиля оформления отчета, позволяющий выбирать поля из разных источников. Мастер допускает группировку записей по некоторым логическим категориям. Он позволяет быстро создавать структуру отчета, которую потом можно доработать в режиме конструктора.

В качестве примера, создадим отчет ведомости групп на экзамен 1\_э1 на основе запроса *Общая ведомость*.

### ЗАДАНИЕ 22. Создание отчета *Экзаменационная ведомость групп*

1. В окне базы данных создайте новый отчет с помощью Мастера.
2. Выбор полей. В окне *Новый отчет* из раскрывающегося списка *Таблицы и запросы* выберите запрос *Общая ведомость*. Заполните область *Доступные поля* полями *ФИО*, *Nst*, *Ngr*, *1\_э1*, используя кнопку со значком  или дважды щелкнув на имени поля.
3. Если потребуется включить поля из других таблиц, это можно будет сделать, выбрав из раскрывающегося списка другой источник данных, повторив все действия пункта 2.

4. Вид представления данных. В следующем окне, которое активизируется только в том случае, если включены поля из нескольких таблиц, для данного запроса Мастер позволяет сгруппировать информацию отчета на основе данных полей из любого источника, указанного в списке. В качестве подсказки справа отображается образец будущей страницы согласно схеме организации. Нажмите клавише Далее .

5. Группировка записей. В данном случае предложенная Мастером организация группировки по полю *Ngr* соответствует задаче построения ведомости группы (рис. 46 а).

Чтобы изменить предложенный Мастером вариант группировки, можно выбрать другие поля для группировки, используя уже известные приемы выбора полей. В данном случае этот этап можно пропустить.

6. Сортировка данных и вычисления. Следующее окно Мастера (Рис. 46 б) позволяет выбрать порядок сортировки для записей отчета, которые будут отображаться непосредственно в группах. Мастер позволяет выбрать до четырех полей сортировки. Чтобы задать поле, по которому будет производиться сортировка, выберите имя нужного поля – в данном случае *ФИО* – из раскрывающегося списка под номером 1. Справа от этого списка расположена кнопка, на которой указан порядок сортировки. Кнопка действует как переключатель: чтобы изменить принятый по умолчанию порядок сортировки по возрастанию на обратный, надо щелкнуть на данной кнопке.

Щелчок на кнопке *Итого*, расположенной под полями сортировки, открывает диалоговое окно, в котором можно установить флажки для применения к числовым полям некоторых статистических функций. В данном случае желательно вывести средний бал (*Avg*) для поля *I\_эI*.

7. Макет отчета. Новое окно предлагает выбрать макет отчета с помощью группы переключателей. Слева расположена область предварительного просмотра, в которой можно ознакомиться с видом макета для разных переключателей. Кроме того, в текущем окне Мастера можно выбрать ориентацию нового отчета (книжная или альбомная). Нажав кнопку *Далее*, перейдите в следующее окно.

8. Стилевое оформление отчета. Список всех стилей представлен справа, выбрав любой из них, можно предварительно оценить стиль будущего отчета.

9. На заключительном этапе Мастер предлагает указать заголовок для отчета. Введите **Экзаменационная ведомость**. Нажав на кнопке *Готово*, можно открыть отчет в режиме предварительного просмотра (рис. 47).

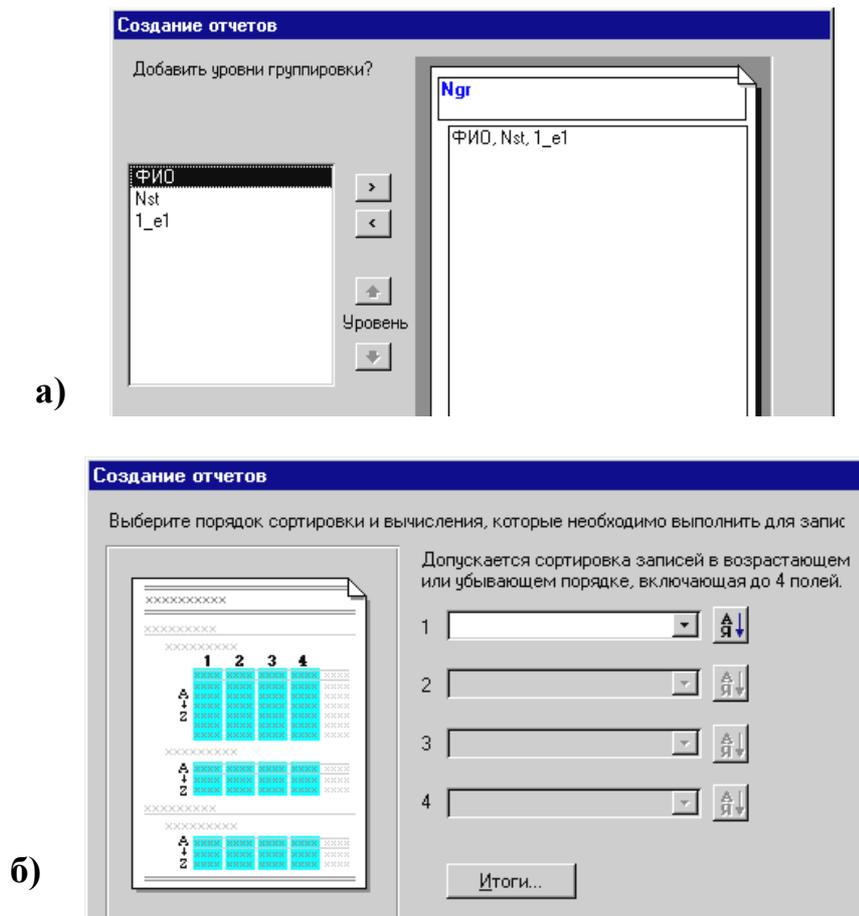


Рис. 46. а) Если отчет строится на основе нескольких таблиц, данные можно группировать; б) Выбор порядка сортировки для записей отчета внутри организованных групп

<i>Экзаменационная ведомость</i>			
<i>Ngr</i>	<i>A-1</i>		
	<u><i>ФИО</i></u>	<u><i>№ бил</i></u>	<u><i>1_эл</i></u>
	АНДРЕЕВА О.С.	1-01	4
	АРСЕНТЬЕВ В.Р.	1-02	4
	БЕЛОВ С.П.	1-03	5
<i>Итого для 'Ngr' = A 1 (3 записей)</i>			
<i>Avg</i>			3333333333333
<i>Ngr</i>	<i>A-2</i>		
	<u><i>ФИО</i></u>	<u><i>№ бил</i></u>	<u><i>1_эл</i></u>
	БЕРЕЖНОЙ П.Р.	2-01	5
	САВИЦА	2-02	4

Рис. 47. Фрагмент отчета *Экзаменационная ведомость* в режиме предварительного просмотра

## Настройка и форматирование отчета

Отчет, созданный с помощью автоотчета или Мастера, как правило, нуждается в доработке. Этот процесс во много аналогичен работе с формой в режиме Конструктора.

Построенный вами отчет желательно дополнить более информационным заголовком, который включал бы время сдачи экзамена, фамилию экзаменатора – все то, что обычно указывается на экзаменационных ведомостях. Латинские подписи необходимо заменить русскими названиями. Необходимо задать точное количество знаков после точки для среднего балла.

### ЗАДАНИЕ 23. Создание отчета Бланк экзаменационной ведомости

Прежде чем, приступать к доработке созданного отчета, модифицируйте запрос *Общая ведомость* так, чтобы он выдавал данные только для одной группы, т.е. введите параметр в строку *Условие* по номеру группы.

1. Создайте с помощью Мастера новый отчет *Бланк экзаменационная ведомость*. Включите в него поля *Depart(Кафедра)* из таблицы *Кафедры*, *ФИО* из таблицы *Преподаватели*. Сгруппируйте записи по полю *Ngr*. Введите расчет среднего балла. При выборе макета остановитесь на пункте по левому краю 1.

2. Чтобы усовершенствовать отчет, нужно открыть его в режиме Конструктора. Общая структура отчета похожа на структуру формы, показанную на рис. 43.

3. *Заголовок отчета* содержит информацию общего характера (как правило, совпадает с названием отчета), которая печатается один раз в начале отчета.

4. Увеличьте область заголовка, оттянув его границу на 3 см вниз. Измените заголовок на **Экзаменационная ведомость**.

5. Используя буфер обмена, перенесите из Заголовка группы связанное поле *Ngr* с его подписью, поставив его в нужное место в области Заголовка отчета. Измените подпись на **группы**.

6. Перенесите название предмета в Заголовок отчета, дав ему подпись **по предмету**. Вставьте надпись **за первый семестр**.

7. Вставьте поле, в котором будет вычисляться текущий учебный год:

**=Str(Year(Date()-1) & "/" & Str(Year(Date())))**

Формула учитывает тот факт, что обычно экзамены сдаются в январе и июне, поэтом текущая дата, задаваемая функцией *Date()*, относится ко второй

половине учебного года. Если ситуация не соответствует описанной, то подумайте, как видоизменить формулу.

Перенесите связанное поле *Dept* в Заголовок отчета, изменив его подпись на **Ведущая кафедра**. Аналогично перенесите поле *FIO*, изменив подпись к нему на **Экзаменатор**. Бланк конструируемого отчета представлен на рис. 48.

8. В Области данных разнесите на 3 см поля *Nst* и *l\_e1*. В области Заголовка группы поместите над полем *l\_e1* надпись **Оценка**.

9. В Области данных вставьте поле между полями номера билета и оценкой. Удалите присоединенную к нему подпись. Вызвав контекстное меню на данном поле, задайте ему Краткий формат даты. Введите в поле параметр **[Ввести дату]**. В области Заголовка группы поместите над новым полем надпись **Дата экзамена**.

Заголовок отчета													
<b>Экзаменационная ведомость</b>													
Группы		Ngr											
По предмету		l_e1		за I семестр		=Str(Year(Date())-1)							
Ведущая кафедра		Dept		Экзаменатор		FIO							
Верхний колонтитул													
Заголовок группы 'Ngr'													
FIO		№ бил		Дата экзамена		Оценка							
Область данных													
FIO		Nst		[Ввести дату]		l_e1							
Примечание группы 'Ngr'													
="Итого для " & "Группы" = " & " & [Ngr] & " (" & Count(*) & " " & If(Count(*)=1,"запись","записей") & ")"													
Средний балл						=Avg([l_e1])							

Рис. 48. Структура отчета *Бланк экзаменационной ведомости* в режиме Конструктора

10. Перейдите в область Примечания. Замените подписи **Ngr** на **Группы**, **Avg** на **Средний балл**. В вычисляемом поле для среднего балла через контекстное меню укажите Фиксированный формат с двумя знаками после точки.

11. Сохраните отчет и просмотрите, как он выглядит. Если необходимо, вернитесь в режим Конструктора. Измените формат надписей (шрифты, размеры), выровняйте надписи.

12. Для того, чтобы вручную не набирать названия ведущих кафедр и экзаменаторов, создайте самостоятельно для этих полей поля со списком, основываясь на рекомендациях **ЗАДАНИЯ 20**.

13. Чтобы построенный отчет можно в будущем использовать для построения экзаменационных бланков по любому предмету, его можно модифицировать, включив при построении Мастером все предметы из запроса *Общая ведомость*. Затем в режиме Конструктора удалить лишние предметы и создать бланк в соответствии с рекомендациями **ЗАДАНИЯ 23**. Чтобы поменять вид предмета, в режиме Конструктора через свойства поля надо выбрать новый экзамен, например, *I\_e2*, изменив для него соответствующие подписи в области *Заголовок* отчета. Сделайте такой отчет самостоятельно.

## **Завершение построения базы данных. Создание командных кнопок и кнопочных форм**

### **Элемент управления Кнопка**

▪ При работе с тем или иным объектом базы данных, часто необходимо обращаться к элементам горизонтального меню или панелей инструментов. Далек не каждый пользователь базы может грамотно воспользоваться этими элементами при работе с конкретным объектом. Поэтому в Access предусмотрен механизм запуска некоторого процесса через элемент управления Кнопка.

Кнопки чаще всего размещают в разделах заголовка или примечания формы. В качестве примера, создадим несколько элементов управления Кнопка для формы *Учетная карта*, которые позволяли производить поиск нужных записей в таблице, вызывать некоторые запросы, печатать текущую форму и выходить в окно базы данных, закрывая текущую форму.

### **ЗАДАНИЕ 24. Создание элементов управления Кнопка в форме Учетная карта**

1. Откройте форму *Учетная карта* в режиме Конструктора. Методом протягивания увеличьте область *Примечания* формы на 1,4 – 2 см. (Если

этот раздел не отображается, выполните команду Вид>Заголовок>Примечание формы)

2. Убедитесь, что кнопка Мастера  на панели элементов активизирована. Щелкните на инструменте Кнопка , а затем щелкните в том месте Примечания формы, где будет размещаться новый элемент Кнопка. Запустится процесс создания кнопки под руководством Мастера.

3. Создание кнопки поиска записей. В первом окне Мастера в области Категории щелкните на требуемой категории действия, в данном случае – *Переходы по записям*, а затем выберите в списке Действия *Поиск записи*. Нажмите кнопку Далее.

4. В следующем окне Мастера необходимо указать, что будет отображаться на кнопке: текст или рисунок. Выберите переключатель Рисунок и выберите стандартный рисунок Бинобль2, хотя, если установить флажок опции Показать все рисунки, можно будет выбрать любой другой рисунок. Нажав кнопку Далее, перейдите в следующее окно Мастера

5. В заключительном окне Мастера укажите имя новой кнопки, например, **Поиск записи**, и щелкните на кнопке Готово.

6. Аналогичным образом создайте Кнопку, реализующую вызов запроса *Список студентов по группам*. Пусть в кнопке отображается текст. Создайте еще две кнопки – **Печать формы** и **Закрытие формы**.

7. Выровняйте все значки кнопок *По верхнему краю*, предварительно выделив их, щелкните на левой линейке Формы в месте, близком к расположению значков и воспользуйтесь командами меню Файл>Выровнять.

8. Перейдите в режим Формы. Теперь она выглядит так, как показано на рис. 49.

Рис. 49. Форма *Учетная карта*, дополненная кнопочными элементами

9. Работа с кнопкой поиска записи. Команда поиска записи реализует простое средство поиска. Вызов этой команды активизирует диалоговое окно, которое позволит по фрагменту одного или нескольких полей найти нужную запись. Поработайте с кнопкой **Поиск записи**. Найдите, например, данные студента, фамилия которого начинается на **Бере**. Если Вы не помните фамилию полностью, воспользуйтесь, опциями диалогового окна или масками (\*, #, ?), действие которых было описано в разделе «Операторы».

Поработайте с другими кнопками, оцените их достоинства и недостатки. Обратите внимание на всплывающие подсказки на кнопках, отмеченных рисунками. Попробуйте создать другие кнопки, предлагаемые Мастером, или вызывающие любой созданный вами запрос.

### Создание кнопочных форм

После создания реальной базы данных, грамотно пользоваться ею может только создатель. Поэтому для ее эксплуатации рядовым пользователем необходимо создать оболочку управления с понятным интерфейсом (интерфейс – средства взаимодействия). Для этой цели подходят, так называемые, *кнопочные формы*. Идея в том, чтобы установить связь между определенным событием и щелчком мыши по графическому изображению кнопки. Однако, не все события можно вызвать с использованием кнопочных форм реализованных в СУБД ACCESS. Поэтому, для обхода этого препятствия, можно воспользоваться созданием макросов. Макрос является программой, содержащей макрокоманды т.е. вызов определенных действий с заданными параметрами. Запуск макроса можно выполнить из кнопочной формы.

### ЗАДАНИЕ 25. Создание кнопочной формы **Факультет**

1. В меню Сервис из списка Надстройки выберите команду Диспетчер кнопочных форм. Подтвердите ее создание (*на экране появилось диалоговое окно, в котором высветилось название кнопочной формы, данное ей по умолчанию*). Для изменения данной кнопочной формы щелкните по кнопке Изменить.

2. В следующем диалоговом окне, в верхнем поле, замените название кнопочной формы на название базы данных (*у вас есть кнопочная форма, но нет кнопок*). Щелкните по кнопке Создать.

3. В появившемся окне (изменение элемента кнопочной формы), выберите, используя кнопки выпадающего списка:

- заданную команду – Открытие формы в режиме редактирования;

- форму *Учетная карта*, которую следует открывать;
- и задайте (в верхнем поле) имя создаваемой кнопки (**Учетная карта**).

4. Следуя рекомендациям пункта 3, включите в кнопочную форму вызов отчета *Бланк экзаменационной ведомости*

5. Щелкните еще раз по кнопке *Создать*. Убедитесь, что в поле команда нет команды открытия запроса. Нажмите клавишу отмены. Закройте окна кнопочной формы.

6. Перейдите на вкладку *Формы*. Откройте кнопочную форму. Убедитесь, что созданные вами кнопки работают. Закройте кнопочную форму.

7. Создание макроса открывающего запрос. Перейдите на вкладку *Макросы*. Выполните команду *Создать В* появившемся диалоговом окне, в поле *Макрокоманда*, используя кнопку вызова списка, выберите команду *Открыть запрос*.

8. В поле *Аргументы макрокоманды* выберите имя открываемого запроса (пусть это будет запрос *Дифференцированная стипендия II курса*, его режим – таблица, и режим данных – чтение.

9. Закройте макрос, сохранив его под именем *Диф. стипендия II курса*.

10. Создание кнопочной команды, запускающей макрос. Выполните команду *Диспетчер кнопочных форм* из списка *Надстройки меню Сервис*. В появившемся диалоговом окне, нажмите кнопку *Изменить*. В следующем окне выполните команду *Создать*. Выполните настройки аналогично пункту 3.

11. Изменение параметров запуска. Теперь наша задача сделать недоступной, при открытии файла, окно базы данных, обеспечив при этом автоматический запуск кнопочной формы. Для этого необходимо проделать следующие действия:

- Выполнить команду *Параметры запуска* из меню *Сервис*
- В появившемся диалоговом окне, убрать флажок *Окно базы данных*.
- А, в окне *Форма*, выбрать *Кнопочная форма*.

Если потребуется все же открыть окно базы данных, достаточно воспользоваться кнопкой *База данных*  на панели инструментов или клавишей **F11**. Все внесенные изменения вступают в силу при следующем открытии базы данных.

12. Завершите конструирование *Кнопочной формы* внесением в нее кнопок **Выход из формы** и **Выход из Access**. Для этого откройте форму в режиме *Конструктора* и воспользуйтесь приемами **ЗАДАНИЯ 24**. Можно украсить *Кнопочную форму* любым рисунком (Рис. 50), используя приемы внедрения объектов типа OLE, описанные в **ЗАДАНИИ 19**.

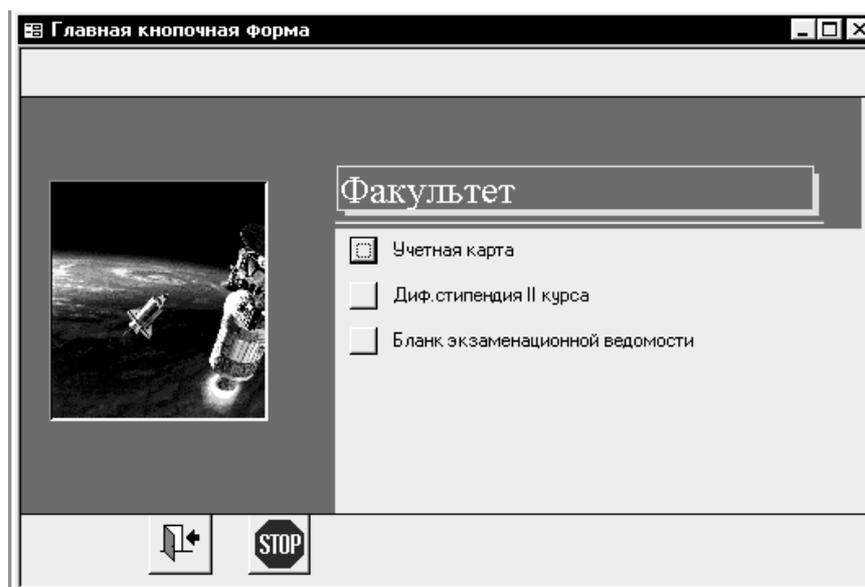


Рис. 50. Кнопочная форма для работы с базой данных *Факультет*

*В заключение краткого курса обучения работе в СУБД ACCESS были рассмотрены вопросы создания и настройки таких объектов базы данных, как Форма и Отчет. Форма – это наиболее гибкий способ просмотра и ввода данных в таблицы, Отчет – способ удобного представления данных для вывода их на печать. Существует несколько способов создания форм и отчетов. Независимо от способа создания, эти объекты могут быть доработаны инструментальными средствами Конструкторов форм и отчетов. Конструкторы позволяют не только удобно размещать в объекте специальные элементы управления, но и усовершенствовать его внешний вид, используя богатые возможности форматирования, доступные Конструктору.*

*Построением Кнопочной формы можно считать завершенным процесс создания базы данных, «в нулевом» приближении. Еще очень многие аспекты работы с базами данных остались не раскрытыми. Например, из всех типов запросов мы ограничились изучением только запросов на выборку, практически не затрагивали вопросы программирования, конвертирования баз данных из других программных продуктов и т.д. Все это – материал не одного пособия. Но, можно надеяться, что изучив основы работы в Access, вы получите творческий потенциал для применения своих знаний в будущей профессиональной деятельности, оценив по достоинству богатые возможности этого приложения Microsoft Office.*

## Контрольные вопросы

1. Для чего предназначены формы? Какие существуют методы для их создания?
2. Перечислите элементы управления и разделы форм.
3. Какие приемы редактирования можно применять к присоединенным элементам управления?
4. Как вызвать окно свойств полей? Какие параметры поля можно определять в этом окне?
5. Как называются элементы управления формы, позволяющие выбирать данные из заранее заготовленного списка или из какого-либо поля таблицы?
6. Можно ли использовать формы для вывода данных?
7. Какие функции выполняет элемент Кнопка?
8. Для чего предназначены отчеты? Какие способы построения отчетов вы знаете?
9. Для чего создается *Главная кнопочная форма*?
10. Для чего в кнопочной форме *Факультет* создавался макрос?

## Оформление отчета

Отчет о лабораторной работе должен содержать:

1. Описание в тетради основных этапов проделанной работы, их логическую последовательность.
2. Ответы на контрольные вопросы.

## СПИСОК ПРАКТИЧЕСКИХ ЗАДАНИЙ

ЗАДАНИЕ 1. Создание таблиц в режиме Конструктора .....	15
ЗАДАНИЕ 2. Ввод и редактирование данных в полях таблицы .....	20
ЗАДАНИЕ 3. Задание масок и форматов ввода для таблиц БД <i>Факультет</i> ..	26
ЗАДАНИЕ 4. Запись условий на значение в полях таблиц БД <i>Факультет</i> ..	28
ЗАДАНИЕ 5. Создание простого списка подстановок для таблицы <i>Преподаватели</i> .....	30
ЗАДАНИЕ 6. Ввод данных в таблицы .....	32
ЗАДАНИЕ 7. Установка связей между таблицами .....	36
ЗАДАНИЕ 8. Редактирование и удаление связей между таблицами .....	40
ЗАДАНИЕ 9. Построение запроса на выборку данных из нескольких таблиц .....	51
ЗАДАНИЕ 10. Построение запросов с условием отбора .....	54
ЗАДАНИЕ 11. Построение запросов с параметром .....	55
ЗАДАНИЕ 12. Построение запросов с вычисляемыми полями .....	57
ЗАДАНИЕ 13. Построение итоговых запросов .....	60
ЗАДАНИЕ 14. Построение итоговых запросов с использованием сложных условий отбора .....	61
ЗАДАНИЕ 15. Построение сложных запросов с условиями отбора для вычисляемых полей .....	63
ЗАДАНИЕ 16. Добавление в базу данных записи, используя автоформу .....	69
ЗАДАНИЕ 17. Добавление в базу данных записей, используя форму, созданную с помощью Мастера .....	71
ЗАДАНИЕ 18. Усовершенствование вида формы <i>Учетная карта1</i> .....	74
ЗАДАНИЕ 19. Вставка фотографий студентов в таблицу <i>Учетная карта1</i> ..	76
ЗАДАНИЕ 20. Создание поля со списком в форме <i>Группы</i> .....	77
ЗАДАНИЕ 21. Построение автоотчета для таблицы <i>Учетная карта</i> .....	78
ЗАДАНИЕ 22. Создание отчета <i>Экзаменационная Ведомость Групп</i> .....	79
ЗАДАНИЕ 23. Создание отчета <i>Бланк экзаменационной ведомости</i> .....	82
ЗАДАНИЕ 24. Создание элементов управления кнопка в форме <i>Учетная карта</i> .....	84
ЗАДАНИЕ 25. Создание кнопочной формы <i>Факультет</i> .....	86

## ОТВЕТЫ К ЗАДАНИЯМ

### ЗАДАНИЕ 1

Пункт 9. Список кураторов

Поле:	Instructor	Ngr	Dept	Naud
Имя таблицы:	Группы	Группы	Переподаватели	Кафедры
Сортировка:	по возрастанию	по возрастанию		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				
или:				

### ЗАДАНИЕ 10. Построение запросов с условием отбора

Пункт 1.

Запрос *Список студентов – мужчин, обучающихся на коммерческой основе.*

Поле:	ФИО	rol	Раум
Имя таблицы:	Учетная карта	Учетная карта	Учетная карта
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		"М"	"К"
или:			

### ЗАДАНИЕ 12. Построение запросов с вычисляемыми полями

Запрос *Список групп*

Поле:	Ngr	Kurs: Val(Right\$(	Ngr	Instructor	Leader
Имя таблицы:	Группы	Учетная карт	Группы	Группы	
Групповая операция:	Группировка	Группировка	Count	Группировка	Группировка
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>				
Условие отбора:					
или:					

Запрос *Список студентов по курсам*

**Kurs: Val(Right\$(Date\$;1))-Val(Right\$([Ngr];1))+  
 Pif(Val(Mid\$(Date\$;4;2))<=6;0;1)**

Поле:	ФИО	Nst	Ngr	Paym	Lgoty	Phon	Dt	Rema	Kurs: Val(Right\$(Da
Имя таблицы:	Учетная	Учет	Учет	Учет	Учет	Учет	Уч	Учет	
Сортировка:									
Вывод на экран:	<input checked="" type="checkbox"/>								
Условие отбора:									[Ввести номер курса]
или:									

### **ЗАДАНИЕ 13.** Построение итоговых запросов

Пункт 13. Запрос *Количество сотрудников по должностям*

<b>Кол-во сотрудников по должностям : запрос на выборку</b>		
Поле:	ФИО	Position
Имя таблицы:	Переподаватели	Переподаватели
Групповая операция:	Count	Группировка
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		Like [вести должности]
или:		

### **ЗАДАНИЕ 14.**

Пункт 5. Запрос *Список студентов II курса на стипендию*

<b>Список студентов II курса на стипендию : запрос на выборку</b>							
Поле:	ФИО	Ngr	Paym	Lgoty	2_p	2_e1	2_e2
Имя таблицы:	Учетная кат	Учетная к	Учетная	Учет	Ведомос	Ведомос	Ведомость
Сортировка:							
Вывод на экран:	<input checked="" type="checkbox"/>						
Условие отбора:		"a2" Or "62"	"Б"		"3"	>3	>3
или:		"a2" Or "62"	"Б"		"3"	>3	>3
		"a2" Or "62"	"Б"	Да	"3"	>2	>2

Пункт 6. Запрос *Стипендия II курса*

Поле:	Ngr	Ngr	Стипенди	Раум	2_р	2_е1	2_е2	Lgoty
Имя таблицы:	Группы	Список стд		Список студ	Список	Список	Список	Список студ
Групповая операция:	Группиров	Count	Выражен	Условие	Услови	Услови	Услови	Условие
Сортировка:								
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Условие отбора:				"б"	"з"	>3	>3	
или:				"б"	"з"	>2	>2	Да

**Стипендия: [Размер стипендии]\*[Count\_Ngr]**

**ЗАДАНИЕ 15.**

Модифицированный запрос *Дифференцированная стипендия II курса*

Поле:	ФИО	Ngr		Lgc	2	2	2	Категор	K	L	M:	Стип: III[	Ds: II	Сумм
Имя таблицы:	Учетн	Учетная ка		Учк	В	Ве	Ве	Активи						
Сортировка:														
Вывод на экран:	<input checked="" type="checkbox"/>													
Условие отбора:		In ("А2";"Б2")												
или:														

**K: [Раум]="б" And [2\_р]="з" And ([2\_е1]=4 Or [2\_е2]=4) And [2\_е1]>3 And [2\_е2]>3** – критерий для получения базовой стипендии.

**L:** [Paym]="б" And [2\_p]="з" And [2\_e1]=5 And [2\_e2]=5 – критерий для получения повышенной стипендии.

**M:** [Paym]="б" And [2\_p]="з" And [Lgoty]=Истина And (([2\_e1]=3 Or [2\_e2]=3) And [2\_e2]>2 And [2\_e1]>2 – критерий получения стипендии по льготам с тройками.

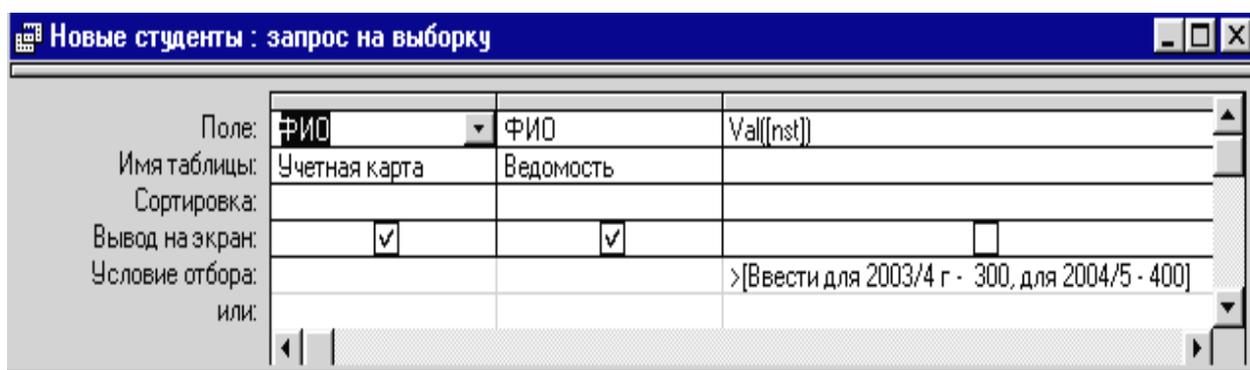
**Стип:** Иф([K]=-1;[Стипендия];Иф([L]=-1;[Стипендия]\*2;Иф([M]=-1; [Стипендия]/2; 0))) – вычисление величины стипендии.

**Ds:** Иф([категория]=1;[Процент]\*[Стип]/100;0) – надбавка за общественную работу.

**Сумма:** [Стип]+[Ds] – значение стипендии с надбавками.

## ЗАДАНИЕ 16.

Пункт 9. Запрос *Новые студенты*



## Рекомендуемая литература

1. Петрова Н.К. Основы работы в СУБД MICROSOFT ACCESS: Учеб. пособие. Казань: Казан. гос. энерг. ун-т, 2004.
2. Тимошок Т.В. Microsoft Access 2002. Самоучитель: М.: Издательский дом «Вильямс», 2003.
3. Дубнов П.Ю. Access 2002. Самоучитель. М.: ДМК Пресс, 2004.
4. Симонович С.В., Евсеев Г.А., Алексеев А.Г. Специальная информатика: Учеб.пособие. М.: АСТ-ПРЕСС: Инфорком-Пресс, 2000.
5. Евдокимов А.В. Лабораторный практикум по ACCESS'97: <http://cchheessss.boom.ru/index.html>, 2002.
6. Информатика: Базовый курс / Симонович и др. СПб.: Питер, 2002.
7. Леонтьев В.П. Новейшая энциклопедия компьютера 2003. М.: ОЛМА-ПРЕСС Образование, 2003.

**СОДЕРЖАНИЕ**

<b>ПРЕДИСЛОВИЕ .....</b>	<b>3</b>
<b>ОБЩИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ .....</b>	<b>4</b>
<b>Основные элементы интерфейса Access. Объекты Access .....</b>	<b>4</b>
<b>Основные свойства и параметры таблиц базы данных .....</b>	<b>6</b>
<b>Типы данных и их представление .....</b>	<b>7</b>
<b>Общее проектирование и цель создания БД.....</b>	<b>9</b>
<b>Определение связей между таблицами .....</b>	<b>13</b>
Контрольные вопросы.....	14
<b>ЛАБОРАТОРНАЯ РАБОТА № 1 «СОЗДАНИЕ НОВОЙ БАЗЫ ДАННЫХ» .....</b>	<b>15</b>
Создание пустой базы.....	15
Изменение типа полей и реорганизация таблиц.....	17
Ввод и редактирование данных в таблице.....	19
<b>Форматирование данных и контроль ввода информации .....</b>	<b>22</b>
Форматирование .....	22
Использование масок ввода и проверка значений .....	25
Проверка вводимых значений .....	27
Создание простого списка подстановки.....	30
Другие опции области Свойства поля .....	31
<b>Ввод данных в таблицы .....</b>	<b>32</b>
<b>Организация связей между таблицами.....</b>	<b>35</b>
<b>Обеспечение целостности данных в базе.....</b>	<b>39</b>
Контрольные вопросы.....	42
<b>ЛАБОРАТОРНАЯ РАБОТА № 2 «СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ЗАПРОСОВ».....</b>	<b>44</b>
<b>Назначение и типы запросов .....</b>	<b>44</b>
<b>Выражение и его компоненты .....</b>	<b>45</b>
Операторы.....	46

Литералы.....	48
Идентификаторы.....	48
Функции.....	48
Использование Построителя выражений.....	49
<b>Запросы на выборку.....</b>	<b>51</b>
<b>Построение запросов с условием отбора.....</b>	<b>53</b>
<b>Построение запросов с параметром.....</b>	<b>55</b>
<b>Вычисления в запросах.....</b>	<b>56</b>
Построение вычисляемых полей.....	57
Групповые операции и вычисления. Итоговые запросы.....	58
Контрольные вопросы.....	66
<b>ЛАБОРАТОРНАЯ РАБОТА № 3 «СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ФОРМ И ОТЧЕТОВ».....</b>	<b>68</b>
<b>Назначение формы.....</b>	<b>68</b>
<b>Использование автоформы.....</b>	<b>69</b>
<b>Создание форм с помощью Мастера.....</b>	<b>70</b>
<b>Конструктор форм.....</b>	<b>72</b>
Структура формы.....	73
Создание и редактирование связанных записей.....	74
Свойства полей. Внедрение и просмотр полей типа OLE.....	76
Создание поля со списком.....	77
<b>Создание отчетов базы данных.....</b>	<b>78</b>
Использование функций автоотчета.....	78
Использование Мастера отчетов.....	79
Настройка и форматирование отчета.....	82
<b>Завершение построения базы данных. Создание командных кнопок и кнопочных форм.....</b>	<b>84</b>
Элемент управления Кнопка.....	84
Создание кнопочных форм.....	86
Контрольные вопросы.....	89
<b>ОТВЕТЫ К ЗАДАНИЯМ.....</b>	<b>91</b>
<b>РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА.....</b>	<b>94</b>

Наталья Константиновна Петрова

## ОСНОВЫ РАБОТЫ В СУБД Microsoft ACCESS

Методическое пособие к лабораторным работам, практическим занятиям,  
расчетному заданию и самостоятельной работе студентов

(Кафедра информатики и информационных управляющих систем КГЭУ)

Редактор издательского отдела *О.В.Ханжина*

---

Изд. лиц. № 03480 от 8.12.00

Темплан издания КГЭУ 2004 г.

Подписано к печати

Формат 60 x 84/16

Гарнитура "Times"

Вид печати РОМ

Бумага "Business"

Физ. печ. л. 7

Усл. печ. л. 6,5

Уч-изд. л. 7,4

Тираж 200

Заказ

---

Издательский отдел КГЭУ  
420066, Казань, Красносельская, 51

---

Типография КГЭУ  
420066, Казань, Красносельская, 51