

Федеральное агентство по образованию  
ГОУ ВПО «Уральский государственный технический университет – УПИ»



**В.Ю. Кара-Ушанов**

# **КОМПЬЮТЕРНОЕ ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ В СРЕДЕ СУБД ACCESS**

В двух частях

## **Часть 1. Структура данных и запросы-выборки**

Учебное электронное текстовое издание  
Подготовлено кафедрой «Вычислительная техника»  
Научный редактор: проф., канд. физ.-мат. наук В.И. Рогович

Методические указания к лабораторному практикуму по дисциплинам «Базы данных», «Управление данными» для студентов специальностей 230101 – Вычислительные машины, комплексы, системы и сети, 230201 – Информационные системы и технологии.

Предназначены для студентов, изучающих в дисциплинах компьютерного цикла основы проектирования баз данных. На примерах компьютерных упражнений осваиваются методологические принципы проектирования баз данных и основы технологии работы в среде популярного Windows-приложения СУБД Access. Даются содержательные определения основных понятий, обозначаются проблемы проектирования баз данных и методы их решения. Рассматриваются приемы работы в среде Access по созданию таблиц и проектированию простейших запросов-выборок. Предлагаются контрольные задания для самостоятельных занятий.

© ГОУ ВПО УГТУ–УПИ, 2006

Екатеринбург  
2006

## КОНЦЕПЦИЯ БАЗ ДАННЫХ В ACCESS

### Модель данных Access

Access представляет собой систему управления базой данных (СУБД) офисного типа (т. н. xBase СУБД) и может представлять интерес для начинающего пользователя благодаря сравнительной простоте теоретических концепций и практической работы по проектированию баз данных. Кроме того, Access является разработкой фирмы Microsoft и представляет собой развивающуюся систему. Все это обуславливает широкое распространение этого популярного Windows-приложения.

Начинающий пользователь должен знать, что работа с Access выполняется на этапе реализации (рабочего проектирования) системы базы данных. В таком контексте Access представляет собой инструментальную среду, в которой реализуются результаты предшествующих этапов проектирования. Иными словами, не с Access начинается разработка базы данных, и начинающий пользователь должен знать, в какой момент потребуются соответствующие знания и навыки.

Для начинающего пользователя особый интерес может представлять инвариантная особенность Access, которая объединяет все версии этого приложения от самых первых до современных. Прежде всего, это *модель данных* [1–5] и *технология работы* в среде пакета [6–10].

Основным интегрированным объектом Access является база данных.

*База данных* (БД) – это организованная на машинном носителе совокупность взаимосвязанных данных, которая представляет сведения об объектах определенной предметной области (ПО), их свойствах и связях между ними.

Используемым типом структуры данных является *реляционная модель данных* (РМД) [1–5]. Термин «реляционная» происходит из теории множеств – теоретической основы баз данных, где обозначает специфический тип структуры данных, называемый *отношением* (отношение, по-английски *relation*, определяет название модели). Как тип структуры данных отношение допускает наглядное представление данных в виде двумерной таблицы.

Итак, основным элементом структуры БД в Access является *таблица*, с помощью которой моделируются объекты и связи между ними. Следовательно, БД в Access представляет собой совокупность взаимосвязанных двумерных таблиц.

Структура реляционной таблицы задается её заголовком и определяется составом и последовательностью полей-атрибутов, моделирующих свойства объектов и связей.

Содержание таблицы заключено в её строках. Каждая строка содержит данные об одном экземпляре объекта или связи и называется *записью*.

Для однозначного определения каждой записи (экземпляра объекта) таблица должна иметь уникальный ключ (*первичный ключ*). Ключ может состоять из одного или нескольких полей.

Таблицы в БД взаимосвязаны. Связь каждой пары таблиц, в которой одна является «отцом», а другая – «сыном», обеспечивается ключом связи (*внешним ключом*). Внешний ключ – это первичный ключ таблицы-отца, мигрировавший в таблицу-сына. Связи между таблицами могут быть двух типов: «один к одному» (1:1) или «один ко многим» (1:N).

Структура реляционной БД задается *схемой данных*. Схема данных включает в себя определение структуры каждой таблицы и определение связей между таблицами. Схема данных в Access создается специально для использования связей между таблицами при обработке данных из разных таблиц, а также для обеспечения целостности БД.

## СОЗДАНИЕ БАЗЫ ДАННЫХ

### *Основы технологии создания базы данных*

Работа с MS Access может начинаться по разным сценариям в зависимости от того, создана база данных или ее предстоит создать [6–9].

Access предоставляет два способа создания новой базы данных: «с нуля» или под управлением Мастера таблиц.

В нотации Access база данных представляет собой составной объект, которому соответствует один единственный файл с расширением **.mdb**.

При создании базы данных возможны разные варианты диалога, отличающиеся к тому же в разных версиях Access элементами интерфейса.

В версии Access 2000, например, создание БД возможно по следующему алгоритму:

1. После «запуска» Access в диалоговом окне выбрать «Новая база данных» и нажать **ОК**.

2. В диалоговом окне «Файл новой базы данных» в соответствующих полях выбрать *имя папки* (каталога), ввести *имя файла* БД и нажать экранную кнопку **Создать**.

Если вы уже находитесь в среде Access, тогда возможен такой сценарий:

1. В главном меню выполнить команду **Файл|Создать**. На экране появится диалоговое окно «Создание».

2. На закладке «Общие» выбрать иконку «База данных» и нажать **ОК**. На экране появится диалоговое окно «Файл новой базы данных».

3. В диалоговом окне «Файл новой базы данных» в соответствующих полях выбрать *имя папки* (каталога), ввести *имя файла* БД и нажать экранную кнопку **Создать**.

В обоих случаях на экране появится окно базы данных, подобное приведенному на рис. 1. В заголовке окна указывается имя БД. Закладки окна БД дают возможность приступить к проектированию объектов БД.

**Контрольное задание 1.** В своей рабочей папке создайте файл базы данных с именем «УЧЕБА».

Файл БД может содержать спецификации до 7 типов объектов-компонентов.

1. **Таблицы**, которые содержат хранимые данные.
2. **Запросы**, предназначенные для выборки или изменения данных одной или нескольких связанных таблиц.
3. **Формы** – экранные формы, предназначенные для ввода и просмотра взаимосвязанных данных в удобном для пользователя БД виде.
4. **Отчеты** – шаблоны печатных документов для вывода данных в удобном для пользователя БД виде.
5. **Страницы** – средства доступа к данным через Internet.
6. **Макросы** – средства автоматизации работы с БД.
7. **Модули** – компоненты прикладной программы автоматизации работы с БД, составленные на языке Visual Basic.

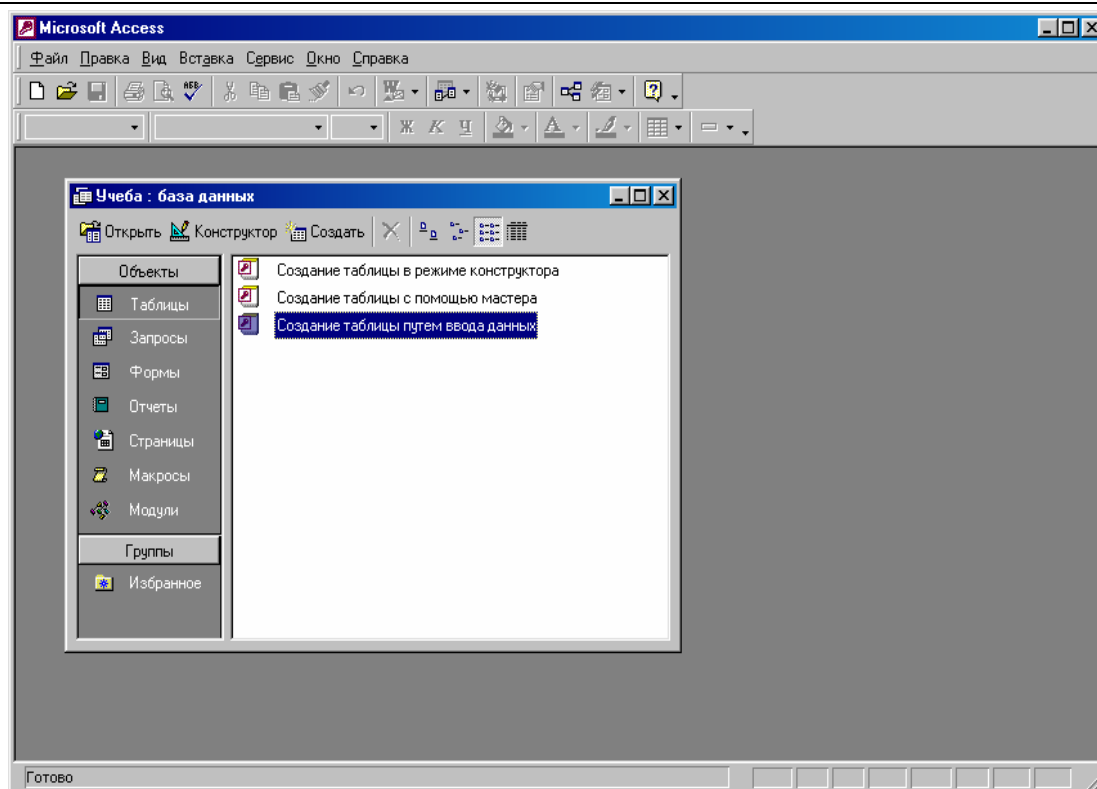


Рис. 1. Окно базы данных Access

### Объекты базы данных

Таблицы и запросы представляют собой сами данные в виде реальных таблиц, т.е. хранимых данных, или в виде виртуальных таблиц, которые представляются пользователю БД как реально существующие.

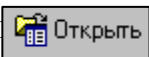
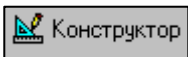
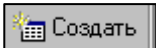
Формы и отчеты представляют собой средства разработки интерфейса конечного пользователя БД.

Страницы – средства доступа к данным через Internet.

Макросы и модули – это средства автоматизации работы с БД в эргономической среде пользователя.

Работа с БД может быть прервана в любой момент путем закрытия окна БД или файла БД (по команде **Файл|Закрыть**) и продолжена путем открытия файла БД (по команде **Файл|Открыть**).

Продолжение работы с БД всегда начинается с окна БД (рис. 1). Для работы с тем или иным объектом БД нужно выбрать соответствующую закладку и нажать экранную кнопку:

- для  **Открыть** – в результате проектирования объекта;
-  **Конструктор** – для изменения структуры объекта;
-  **Создать** – для создания объекта «с нуля».

## ПРОЕКТИРОВАНИЕ ТАБЛИЦ

### Разработка структуры таблицы

#### *Основы технологии формирования структуры таблицы*

Таблицы БД могут быть созданы одним из следующих способов [6–9]:


- 1) в среде Access в режиме конструирования «с нуля»;
- 2) как результат запроса того или иного типа к уже созданным таблицам;
- 3) в результате импорта таблиц в форматах других сред.

Рассмотрим более подробно основной первый способ. Второй способ рассмотрим в контексте проектирования запросов. Третий способ просто примем к сведению.

Создание таблицы «с нуля» выполняется в два этапа. Сначала разрабатывается её структура. Затем выполняется наполнение таблицы содержательной информацией.

При нажатии кнопки открывается окно «Новая таблица» с предложением выбрать способ создания новой таблицы. Наиболее актуальными являются:

▪ **Мастер таблицы** – для формирования структуры таблицы из полей заданных тематических таблиц;

▪ **Конструктор** – для  создания таблицы «с нуля» (основной способ).

При работе под управлением **Мастера** в диалоговом окне «Разработка таблицы» (рис. 2) предлагаются типовые примеры тематических таблиц в списке «Образцы таблиц» для переноса полей из списка «Образцы полей» для формирования структуры новой таблицы в списке «Поля новой таблицы». Для переноса полей в структуру новой таблицы достаточно выделить соответствующую таблицу и её поля и нажать экранные кнопки:

- > – для прямого переноса одного поля;
- >> – для прямого переноса всех выделенных полей;
- < – для обратного переноса одного поля;
- << – для обратного переноса всех выделенных полей.

При работе под управлением **Конструктора** формирование структуры новой таблицы выполняется в диалоговом окне (рис. 3). Для каждого поля в соответствующем столбце предлагается указать:

- 1) **имя поля** (набирается вручную в столбце «Поле», можно по-русски);

- 2) **тип данных** (выбирается из списка);
- 3) **описание-комментарий** (не обязательно);
- 4) **свойства** (в зависимости от типа данных).

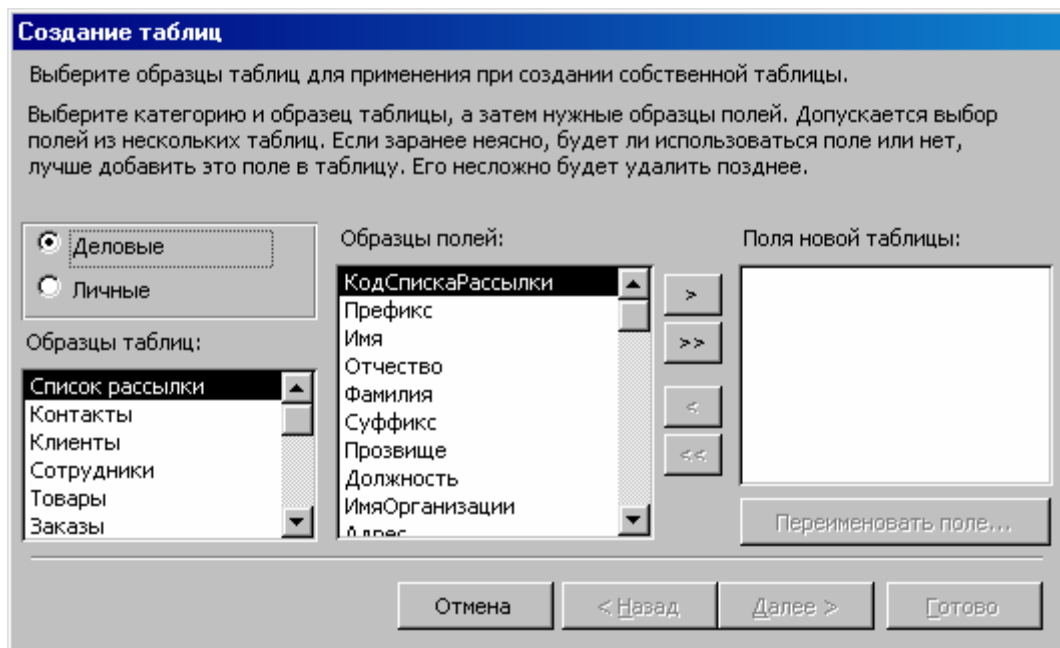


Рис. 2. Окно формирования структуры таблицы под управлением Мастера

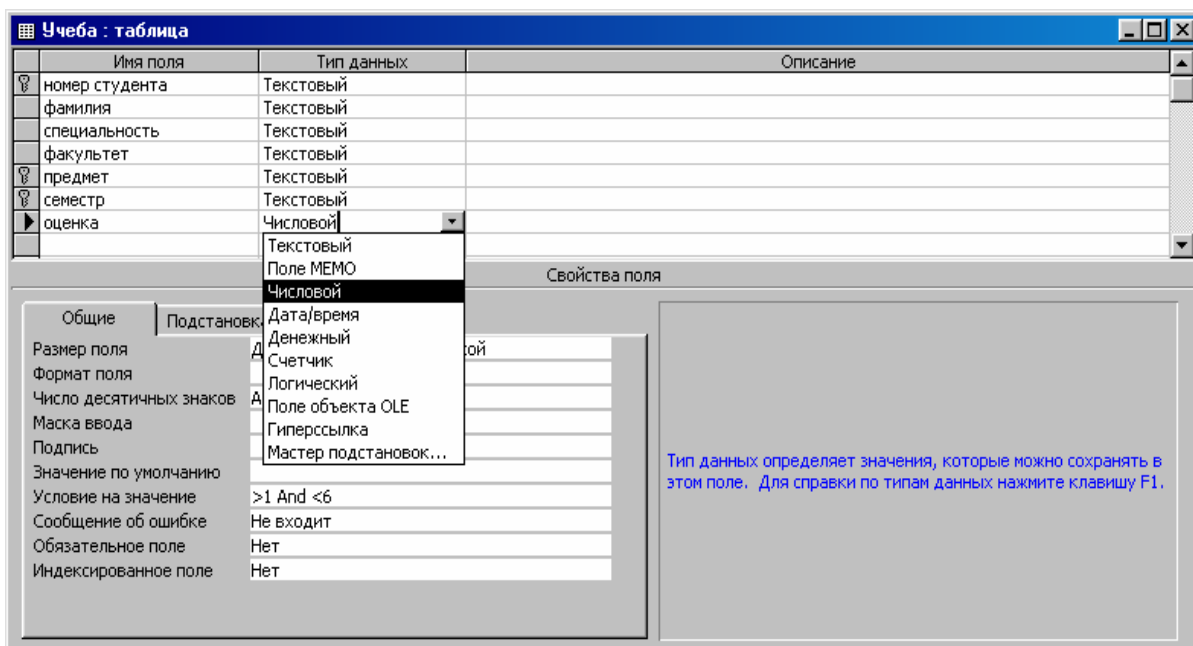


Рис. 3. Окно формирования структуры таблицы

Многообразие типов данных, predetermined in Access (табл.1), определяет разнообразие информации, которую можно хранить в БД.

Таблица 1

## Типы данных

Типы данных	Диапазон значений	Размер в байтах
Текстовый	0 ÷ 255	1
Числовой (байт)	0 ÷ 255	1
Числовой (целое)	-32768 ÷ 32767	2
Числовой (длинное целое)	-2147483468 ÷ 2147483467	4
Числовой (с плав. точкой)	-3.402823 10 <sup>38</sup> ÷ 3.402823 10 <sup>38</sup>	4
Числовой (двой- ной точности)	-1.797693134862232 10 <sup>308</sup> ÷ 1.797693134862232 10 <sup>308</sup>	8
Логический	Да / Нет	
Дата/Время	100 ÷ 9999 гг.	8
Денежный	15.4	8
Счетчик		4
Поле MEMO	64000	
OLE-объект		10 <sup>9</sup>
Гиперссылка	64000	

Большинство типов данных в комментариях не нуждаются. Поясним некоторые из них. Тип «Счетчик» может быть присвоен не более чем одному полю. Данные этого типа являются уникальными номерами записей и не могут быть изменены или удалены. Тип «Денежный» используется для описания данных, над которыми выполняются финансовые операции с высокой точностью вычисления. Точность представления данных этого типа составляет до 15 знаков в целой части и 4 – в дробной. Тип «поле MEMO» используется для полей, содержащих пространную текстовую информацию. Тип «поле объекта OLE» присваивается полям-объектам других приложений Windows (рисунок, фотография, звуковая запись, анимация и т. д.). Тип «Гиперссылка» используется для полей, содержащих адрес удаленного источника данных.

В нижней части окна на закладке «Общие» могут быть заданы свойства полей, которые зависят от типа данных:

- 1) *размер поля* (для числовых полей возможна детализация);
- 2) *формат поля* (внешняя форма представления данных) при помощи специальных символов форматирования;
- 3) *маска ввода* в виде символов, управляющих вводом данных;
- 4) *подпись* – псевдоним поля;



- 5) значение по умолчанию;
- 6) условие на значение в виде ограничений на допустимые значения поля;
- 7) сообщение об ошибке, сопровождающее нарушение условия на значение;
- 8) обязательное поле – поле, которое не может принимать не определенное значение (т.н. Null-значение);
- 9) индексированное поле – поле, по которому предполагается упорядочение строк таблицы.

Несколько слов об управлении вводом и отображением данных таблиц. В Access существует два свойства полей, использование которых приводит к похожим результатам: свойство *формат поля* и свойство *маска ввода*.

Свойство *формат поля* используется для отображения данных и влияет только на отображение значения, но никак не влияет на значение, хранимое в таблице. Формат отображения применяется только после сохранения введенных данных. Если вводом данных необходимо управлять, в дополнение к формату отображения данных или вместо него используется маска ввода. Если требуется, чтобы данные отображались так, как они были введены, свойство *формат поля* можно не устанавливать.

Маска ввода обеспечивает соответствие данных определенному формату, а также заданному типу значений, вводимых в каждую позицию. Маска ввода является удобным средством контроля вводимых значений.

Если для поля определены как формат отображения, так и маска ввода, то при добавлении и редактировании данных используется маска ввода, а формат поля определяет отображение данных после сохранения записи. При этом необходимо следить за тем, чтобы маска и формат не противоречили друг другу.

Управление вводом осуществляется при помощи символов маскирования, которые выполняют специфические функции (табл. 2).

### ***Теоретические предпосылки создания таблиц базы данных***

Реализации проекта базы данных предшествует большая аналитическая работа, которая выполняется на этапах инфологического и даталогического проектирования. На основании анализа семантики данных предметной области разрабатывается ее информационная модель, которая затем отображается в компьютерную среду с учетом структурных и процедурных возможностей СУБД реляционного типа. Разработка структуры БД, в том числе разработка структуры каждой таблицы, установление зависимостей между атрибутами, назначение ключе-

вых атрибутов, установление связей между таблицами, является результатом продолжительной аналитической работы [1]. Не нужно забывать, что Access – это только инструментальная среда для реализации информационной модели ПО и что обращаться к ней следует лишь тогда, когда проработаны особенности структуры данных ПО.

В качестве демонстрационного примера рассмотрим упрощенный вариант ПО, называемой «УЧЕБА». Допустим, что для спецификации этой, в общем-то, многогранной ПО в упрощенном варианте достаточно следующих атрибутов-свойств:

Таблица 2

**Управляющие символы маски ввода**

Символ	Описание
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 необязателен; пустые символы преобразуются в пробелы, допускаются символы плюс и минус)
L	Буква (от A до Z или от A до Я, ввод обязателен)
?	Буква (от A до Z или от A до Я, ввод необязателен)
A	Буква или цифра (ввод обязателен)
a	Буква или цифра (ввод необязателен)
&	Любой символ или пробел (ввод обязателен)
C	Любой символ или пробел (ввод необязателен)
.,:;- /	Десятичный разделитель и разделители тысяч, значений дат и времени. (Отображаемый символ зависит от настроек языка и стандартов на панели управления Windows)
<	Указывает перевод всех следующих символов на нижний регистр
>	Указывает перевод всех следующих символов на верхний регистр
!	Указывает заполнение маски ввода справа налево, а не слева направо. Заполнение маски символами всегда происходит слева направо. Восклицательный знак в маске ввода можно помещать в любую позицию
\	Указывает ввод любого следующего символа в качестве текстовой константы. Используется для отображения всех перечисленных в данной таблице символов как текстовых констант (например, \A выводится как символ «А»)
Пароль	Значение <i>Пароль</i> , заданное для свойства <i>Маска ввода (InputMask)</i> , создает поле для ввода пароля. Любой символ, введенный в поле, сохраняется как символ, но отображается как звездочка (*)

<i>номер студента</i>	- учётный номер студента, который однозначно идентифицирует его персону;
<i>фамилия</i>	- фамилия и инициалы студента;
<i>специальность</i>	- специальность, по которой обучается студент;
<i>факультет</i>	- факультет, на котором учится студент;
<i>предмет</i>	- учебная дисциплина, по которой студент в семестре сдает экзамен и получает оценку;
<i>семестр</i>	- учебный семестр, по итогам которого студент сдает экзамен и получает оценку;
<i>оценка</i>	- оценка за экзамен.

В реляционном представлении такой ПО можно поставить в соответствие «универсальное» отношение со схемой:

**Учёба (номер студента, фамилия, специальность, факультет, предмет, семестр, оценка)**

здесь подчеркнуты ключевые атрибуты.

Допустим также, что по условиям функционирования ПО УЧЕБА справедливы следующие семантические условия и соответствующие им функциональные (однозначные) зависимости (ФЗ) между атрибутами [6]:

1. Учетный номер студента, а не его фамилия, является уникальной характеристикой студента (т.е. допускаются однофамильцы):

*номер студента* → *фамилия*

2. Каждый студент учится только по одной специальности:

*номер студента* → *специальность*

3. Каждый студент учится только на одном факультете:

*номер студента* → *факультет*

4. По каждой специальности готовят специалистов только на одном факультете:

*специальность* → *факультет*

5. Каждый студент (номер студента) по учебному предмету в семестр может получить лишь одну итоговую оценку:

(*номер студента, предмет, семестр*) → *оценка*

Схема ФЗ ПО УЧЕБА приведена на рис. 4.

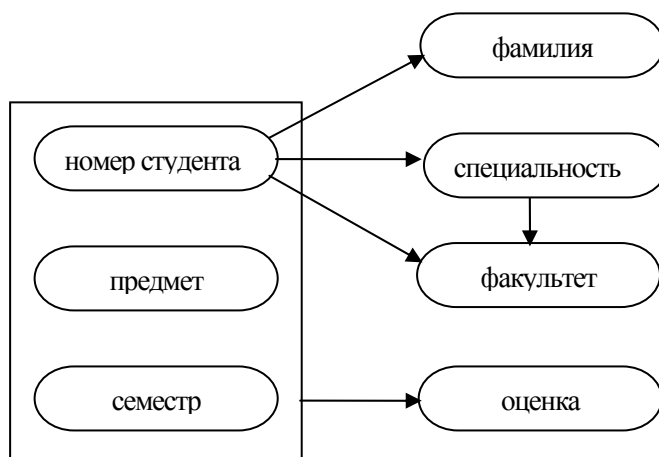

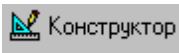



Рис. 4. Схема функциональных зависимостей ПО УЧЕБА

### *Основы технологии работы с таблицами*

Работа с таблицей в Access может выполняться в двух режимах:



1. Режим **Конструктор** используется при создании новой таблицы или при изменении структуры уже созданной таблицы. Начало работы в окне БД – по экранным кнопкам  или  (рис. 1) на закладке **Таблицы**.

2. **Режим таблицы** используется для добавления, удаления, изменения и просмотра данных в таблице. Начало работы в окне БД – по экранной кнопке  (рис. 1) на закладке **Таблицы**.

Переход из одного режима в другой может выполняться:

- 1) по командам Вид|Конструктор и Вид|Режим таблицы главного меню;
- 2) нажатием экранных кнопок на панели инструментов конструктора таб-

лиц:

-  – режим конструктора;
-  – режим таблицы.

**Контрольное задание 2.** После анализа ПО можно приступить к реализации проекта, например, в среде Access:

1. В БД УЧЕБА создайте в режиме **Конструктора** таблицу с именем «Учёба». При разработке её структуры воспользуйтесь спецификацией полей-атрибутов (табл. 3).

2. По завершении спецификации структуры новой таблицы выполните команду **Файл|Сохранить как...** и присвойте таблице имя «Учеба».

Таблица 3

**Спецификация полей-атрибутов отношения «Учёба»**

Наименование	Тип данных	Размер <sup>1</sup>	Маска ввода	Ограничения
номер студента	Текстовый	6	000000	Ключевой
фамилия	Текстовый	15	>L<????????? \ >L.L.;0	
специальность	Текстовый	6	000000	
факультет	Текстовый	2	<CC	Ключевой
предмет	Текстовый	20	<????????????????????	
семестр	Текстовый	2	99;0	
оценка	Числовой	целое		>1 And <6 <sup>2</sup>

3. Задайте в окне разработки структуры таблицы ключевые поля по следующему алгоритму:

1) удерживая клавишу **Ctrl** в нажатом состоянии, отметьте мышью в колонке слева от названия полей все ключевые поля (они должны выделиться цветом);

2) выполните команду **Правка|Ключевое поле** главного меню (в колонке слева от имени поля должно появиться изображение ключа).

По окончании формирования спецификация структуры таблицы «Учёба» должна выглядеть так, как показано на рис. 3.

**НАПОЛНЕНИЕ И РЕДАКТИРОВАНИЕ ТАБЛИЦЫ**

Пока таблица не заполнена данными, можно безбоязненно редактировать её структуру в режиме **Конструктора**:

1. Добавить новое поле:

1) выделить мышью в колонке слева от имени поля строку таблицы структуры;

2) выполнить команду **Вставка|Строки**.

2. Удалить поле:

1) выделить мышью в колонке слева от имени поля строку таблицы структуры;

2) выполнить команду **Правка|Удалить строки**.

3. Переименовать поле.

<sup>1</sup> Размер поля задается в списке свойств поля в нижней части диалогового окна.

<sup>2</sup> Свойство «условие на значение».

4. Переопределить тип поля.
5. Определить ключ (как обсуждалось выше).

Если таблица уже заполнена данными, редактирование её структуры может повлечь за собой потерю данных. Например, если при изменении типа данных конкретное значение невозможно преобразовать к новому типу данных, то оно удаляется и заменяется во всех содержащих это значение полях на пустое значение. Например, если поле преобразуется из типа «Текстовый» в тип «Числовой», то будет удалено любое значение, содержащее одну или несколько букв. При этом выводится предупреждение о числе удаляемых значений, и пользователь имеет возможность отменить преобразование.

Наиболее часто встречающиеся преобразования типов данных можно представить следующими четырьмя категориями: преобразование в текст других типов данных; преобразование текстовых данных в числовые, денежные, логические форматы или в формат дата/время; преобразование денежного типа данных в числовой, текстового в поле МЕМО и наоборот; а также изменение настройки свойства **Размер поля** для числовых полей.

**Контрольное задание 3.** Наполните таблицу «Учёба» содержательной информацией (табл. 4):

Таблица 4

**Информация для ввода в таблицу «Учёба»**

номер студента	фамилия	специальность	факультет	предмет	семестр	оценка
016510	Иванов И.И.	с2	ф2	экономика	9	3
021008	Иванов И.И.	с1	ф1	физика	2	4
029812	Петров П.П.	с1	ф1	кибернетика	4	4
031512	Николаева Н.Н.	с1	ф2	физика	1	5
028101	Андреев А.А.	с3	ф2	программирование	5	3
031512	Николаева Н.Н.	с2	ф2	математика	1	5
031512	Николаева Н.Н.	с2	ф2	математика	2	5
021008	Иванов И.И.	с1	ф1	математика	2	4

1) перейдите в **Режим таблицы**. Обратите внимание на контекстное изменение главного меню. В нем появились пункты «Формат» (для изменения размеров полей таблицы) и «Записи» (для работы с записями таблицы);

2) заполните шаблон таблицы «Учёба» данными (табл. 4). Обратите внимание, что расположение записей в таблице при вводе определяется не хроно-

логией ввода, а условиями упорядочения «по ключу» (в порядке возрастания ключа);

3) по завершении ввода сохраните результаты по команде **Файл|Сохранить** главного меню.

В **Режиме таблицы** можно выполнить простейшие интерактивные (без запоминания условий) запросы-выборки:

1. Поиск записи по заданному значению поля (выполняется по команде **Правка|Найти**).

2. Сортировка записей по значению поля (выполняется по команде **Записи|Сортировка**).

3. Выборка по условию (выполняется по команде **Записи|Фильтр**).

Проделайте самостоятельно по вашему усмотрению несколько упражнений на эту тему.

## ФОРМИРОВАНИЕ ЗАПРОСОВ

### Спецификация запросов

База данных представляет собой информационную модель ПО и предназначена для удовлетворения информационных запросов пользователей, работающих в этой ПО. После того как БД будет наполнена содержательной информацией, к ней можно обращаться с запросами. *Запрос* представляет собой спецификацию условий манипулирования данными, в результате которого создается то или иное представление о хранимой в БД информации. БД является объектом коллективного пользования, и каждый запрос выражает индивидуальное, локальное представление пользователей о структуре информации моделируемой ПО.

Запросы к БД можно разделить на две категории.

1. *Запросы-выборки* осуществляют только извлечение информации из одной или нескольких связанных таблиц и представление её в виде виртуальной (не существующей на самом деле) таблицы. Структура такого рода таблицы определяется локальным представлением о ПО конечного пользователя БД в контексте запроса.

2. *Корректирующие запросы* или запросы на добавление, удаление и изменение данных в БД. Это весьма значительные по своим последствиям запросы, так как их реализация приводит к изменению хранимой в БД информации. Такого рода запросы соответствуют функциям администрирования БД.

БД одной и той же ПО может быть представлена в виде одной таблицы («универсального отношения») или в виде совокупности связанных таблиц [1–5]. Наш учебный пример слишком прост, но и на нем, как увидим в последствии, это положение можно продемонстрировать. Построение БД в виде одной таблицы или в виде совокупности таблиц зависит от соотношения частот обращения к БД с корректирующими запросами или с запросами-выборками. Для первых предпочтительнее совокупность связанных таблиц. В теории БД есть на этот счет специальная методология, называемая нормализацией БД<sup>3</sup>. Для запросов-выборок предпочтительнее использование одной таблицы.

---

<sup>3</sup> Обсуждение этой проблемы выходит за рамки данного методического пособия.



Для спецификации запросов необходимы соответствующие языковые средства – *языки запросов*. Языки запросов – это языки высокого уровня, не процедурные (как Паскаль, Си, Бейсик и др.), а декларативные. Это значит, что в запросе достаточно указать, что нужно сделать, а реализация запроса скрыта от пользователя.

В Access (как и во многих других СУБД) используются языки запросов двух типов:

1. **QBE** (Query By Example) – запрос по образцу. Запрос формируется в режиме диалога путем заполнения таблицы специального вида. QBE-технология дает пользователю визуальное представление достаточно сложного запроса.

2. **SQL** (Structured Queries Language) – язык структурированных запросов. Запрос формируется в терминах команд этого языка (т. е. нужно знать язык SQL). В настоящее время SQL-технология запросов получила широкое распространение в системах БД на основе распределенных систем, в компьютерных сетях.


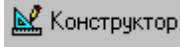
## ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ЗАПРОСА

### *Основы технологии проектирования запросов*

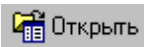
Основной технологией формирования запросов в Access является QBE-технология с возможностью отображения запроса в стиле SQL.

Работа с запросами в Access может выполняться в трех режимах:

1. Режим **Конструктора** (основной) используется при создании нового запроса или при изменении структуры уже созданного запроса в стиле QBE.

Начало работы в окне БД – по экранным кнопкам  Создать или  Конструктор (рис. 1) на закладке **Запросы**.


2. **Режим SQL** используется для просмотра уже созданного QBE-запроса, а также для создания нового запроса или при изменении структуры уже созданного в стиле QBE-запроса.

3. **Режим таблицы** используется для просмотра результатов запроса. Начало работы в окне БД – по экранной кнопке  Открыть (рис. 1) на закладке **Запросы**.

Переход из одного режима в другой может выполняться:

1) по командам Вид|Конструктор, Вид|Режим SQL и Вид|Режим таблицы главного меню;

2) нажатием экранных кнопок панели инструментов конструктора запросов:

 – режим конструктора;

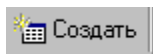
 – режим таблицы;

 – режим SQL.

### **Основной алгоритм построения запросов**

Формирование запроса выполняется примерно по следующему алгоритму:

1. В окне БД выбрать закладку **Запрос** и нажать экранную кнопку



2. В окне «Новый запрос» выбрать вариант **Конструктор**.

3. В окне «Добавление таблицы» выбрать по закладке источник данных к запросу: **Таблицы|Запросы|Таблицы и Запросы**.

4. Выбрать в списке конкретную таблицу или запрос и нажать экранную кнопку **Добавить** и в конце **Заккрыть**. Появится основное окно конструктора запросов в стиле QBE (рис. 5).

5. В окне конструктора оформить условия спецификации конкретного запроса (как это делается, рассмотрим позднее на конкретных примерах типов запросов). Обратите внимание, что в контекстном меню проявился новый пункт меню «Запрос».

6. Посмотреть результат запроса по команде **Запрос|Запуск**. На экране появится окно просмотра данных в **Режиме таблицы**.

7. Вернуться в режим **Конструктора** и принять решение, удовлетворяют ли вас результаты запроса или нет. При отрицательном исходе можно продолжить формирование запроса.

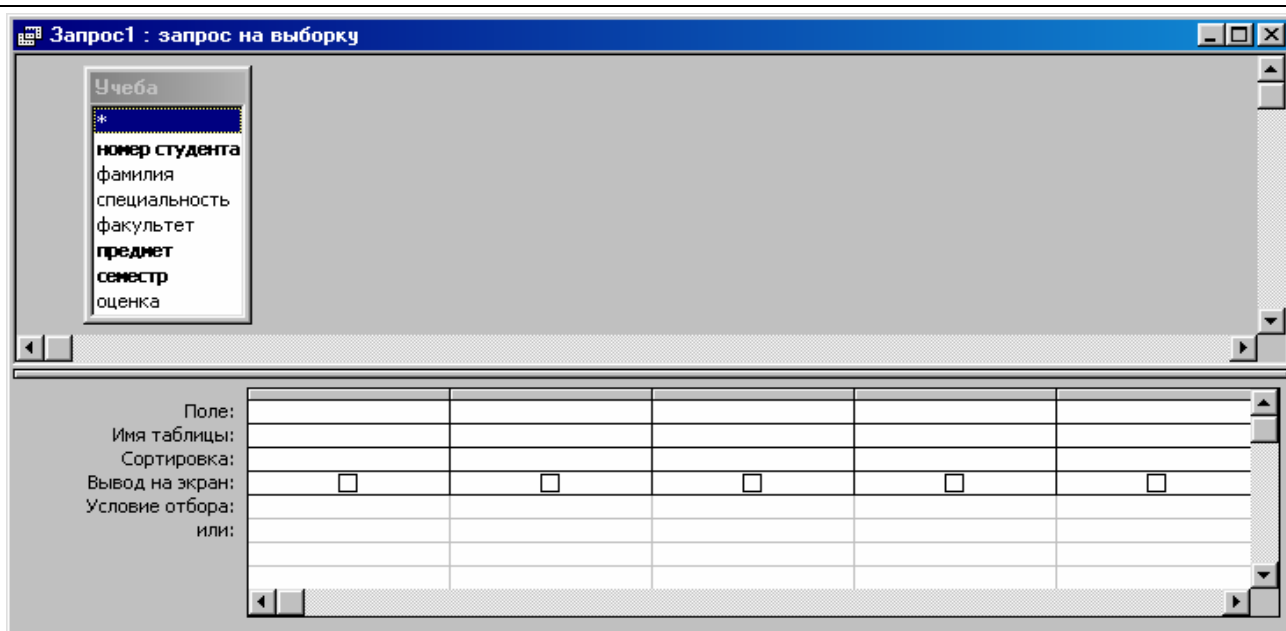


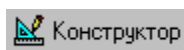
Рис. 5. Окно конструктора QBE-запросов

8. Если результаты запроса вас удовлетворяют, запрос можно сохранить по команде **Файл|Заккрыть** или **Файл|Сохранить как**, указав в диалоговом окне имя запроса, под которым он будет представлен в списке на закладке **Запросы** текущей БД.

9. Для воспроизведения результата запроса необходимо выбрать запрос по имени в списке в окне БД на закладке **Запросы** и нажать экранную кнопку



10. Для изменения структуры запроса необходимо выбрать запрос по имени в списке в окне БД на закладке **Запросы** и нажать экранную кнопку



## ЗАПРОСЫ ВЫБОРКИ

Технологию проектирования конкретных типов запросов рассмотрим на примере запросов к БД УЧЕБА, представленной одной таблицей «Учёба». Речь идет о детализации действий в п. 5 рассмотренного выше алгоритма. При сохранении запросов рекомендуется присваивать им содержательные имена, которые указывали бы на смысл, на семантику запроса.

### *Вывод всех полей*

**Контрольное задание 4.** Сформируйте запрос-выборку всех полей таблицы «Учёба» БД УЧЕБА:

- 1) перенесите мышью в строку «Поле» таблицы запроса имена *всех* полей из списка в верхней части таблицы<sup>4</sup>;
- 2) включите щелчком мыши по квадратикам в строке «Вывод на экран» режим просмотра соответствующих полей – спецификация запроса примет вид таблицы, приведенной на рис. 6;
- 3) завершите результаты формирования запроса так, как это обсуждалось в алгоритме формирования запроса на стр. 18. Результаты запроса должны быть представлены, как на рис. 7;
- 4) перейдите в SQL-режим по команде Вид|Режим SQL, чтобы получить представление о структуре SQL-запроса;
- 5) сохраните запрос под именем «Все поля».

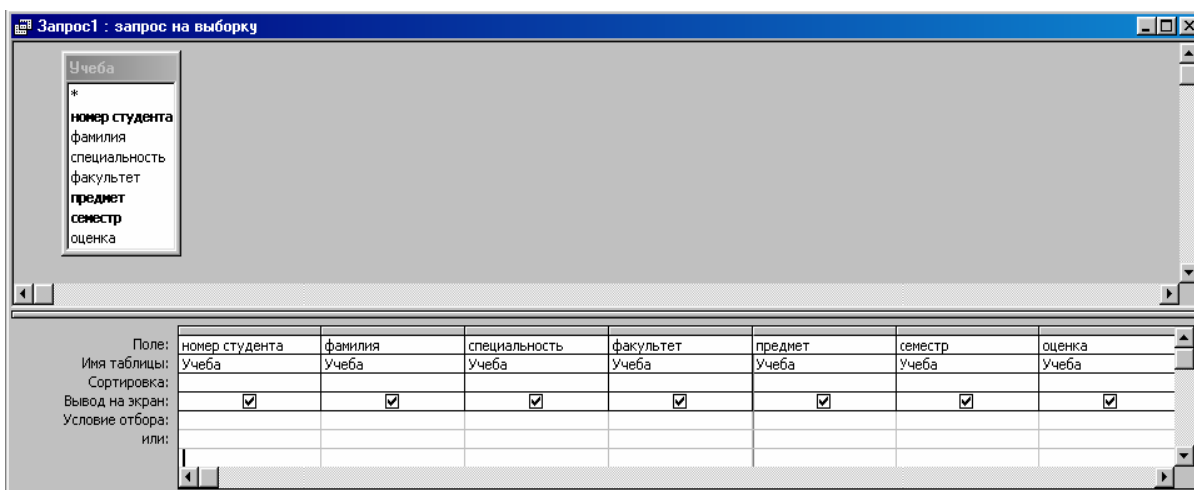


Рис. 6. Структура QBE - запроса «Все поля»

### ***Выборка полей***

В структуре запроса могут быть представлены не все поля, а только некоторые (вертикальная выборка). В результате в итоговой таблице могут присутствовать дубли строк. Это происходит по понятной причине: строки-дубли образуются потому, что поля, по которым они различались в исходной таблице, остались за пределами выборки.

**Контрольное задание 5.** Сформируйте запрос-выборку некоторых полей таблицы «Учёба» БД УЧЕБА.

<sup>4</sup> Того же результата можно добиться щелчком мыши в строке «Поле» и выбором имени поля из раскрывающегося списка.

номер студента	фамилия	специальность	факультет	предмет	семестр	оценка
016510	Иванов И.И.	с2	ф2	экономика	9	3
021008	Иванов И.И.	с1	ф1	математика	2	4
021008	Иванов И.И.	с1	ф1	физика	2	4
028101	Андреев А.А.	с3	ф2	программирование	5	3
029812	Петров П.П.	с1	ф1	кибернетика	4	4
031512	Николаева Н.Н.	с2	ф2	математика	1	5
031512	Николаева Н.Н.	с2	ф2	математика	2	5
031512	Николаева Н.Н.	с2	ф2	физика	1_	5
*						

Рис. 7. Результаты запроса «Все поля»

- 1) сформируйте запрос, в котором присутствовали бы следующие поля: «номер студента», «фамилия», «специальность», «факультет» (технология построения запроса такая же, как в примере «Все поля», отличается только количеством полей);
- 2) завершите результаты формирования запроса так, как это обсуждалось в алгоритме формирования запроса на стр. 18. Обратите внимание на присутствие в итоговой таблице дублирующих записей;
- 3) назовите запрос «Дубли».

### ***Выборка полей без дублей записей (проекция)***

Дубли записей в таблице являются её недостатком потому, что не несут никакой полезной информации. Избавиться от дублей можно, если настроить свойство запроса, которое запрещает совпадение значений данных в выбранных полях. Итоговая таблица по такому запросу представляет собой результат реляционной операции *проекции* исходной таблицы на выбранные поля-атрибуты.

**Контрольное задание 6.** Создайте запрос на выполнение проекции исходной таблицы:

- 1) сформируйте запрос, в котором присутствовали бы следующие поля: «номер студента», «фамилия», «специальность», «факультет» (технология построения запроса такая же, как в предыдущем примере);
- 2) в режиме **Конструктора** выполните команду **Вид|Свойства**;

3) в диалоговом окне «Свойства запроса» переключите мышью свойство «Уникальные значения» в состояние «Да» (рис. 8);

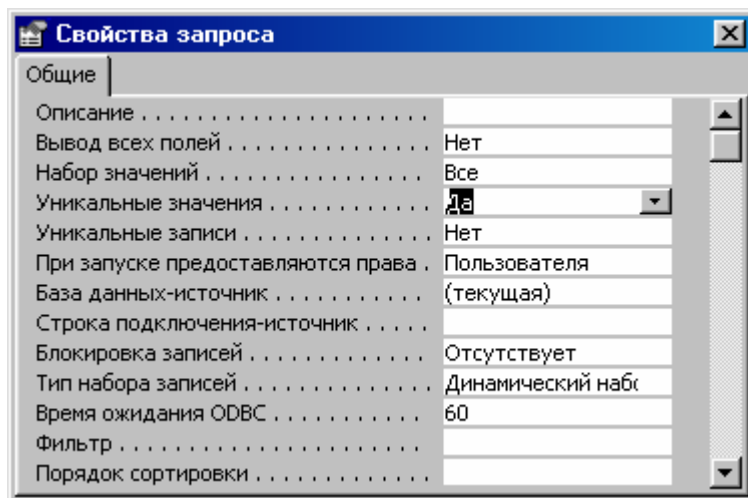


Рис. 8. Спецификация свойств запроса «без дублей записей»

4) завершите результаты формирования запроса так, как это обсуждалось в алгоритме формирования запроса на стр. 18;

5) назовите запрос «Проекция».

### *Запросы на основе операции селекции*

Очень распространенным типом запросов являются запросы выборки записей-строк по условию (горизонтальная выборка). Итоговая таблица в таком случае представляет собой результат реляционной операции *селекция*.

Условия отбора записываются в строке «Условие отбора» структуры запроса и могут состоять из литералов, констант и идентификаторов, соединенных операторами. Условия отбора можно записывать «по памяти» или при помощи специального инструмента, так называемого **Построителя выражений**. Доступ к этому средству осуществляется в контексте построения условия при нажатии правой клавиши мыши и выборе пункта **Построить** контекстного меню (рис. 9).

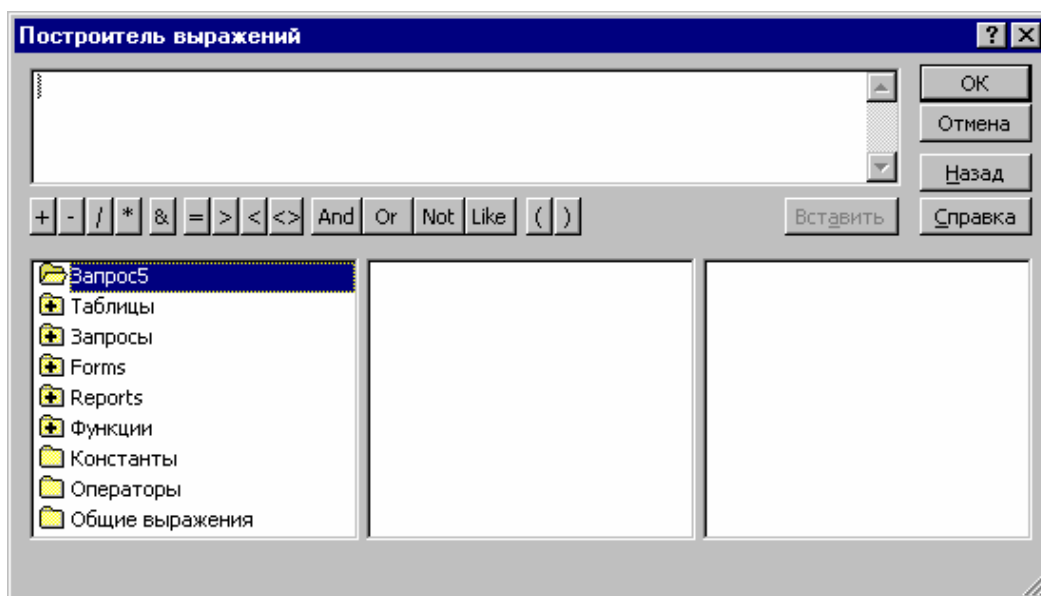


Рис. 9. Окно построителя выражений условий

### Выборка по значению поля

**Контрольное задание 7.** Для выбора всех данных о студентах, обучающихся на факультете «ф2»:

- 1) выберите все поля в структуру запроса, как в примере «Все поля»;
- 2) в поле «факультет» в строке «Условие отбора» наберите вручную значение «ф2» (рис. 10);
- 3) завершите результаты формирования запроса так, как это обсуждалось в алгоритме формирования запроса на стр. 18;
- 4) назовите запрос «Селекция».

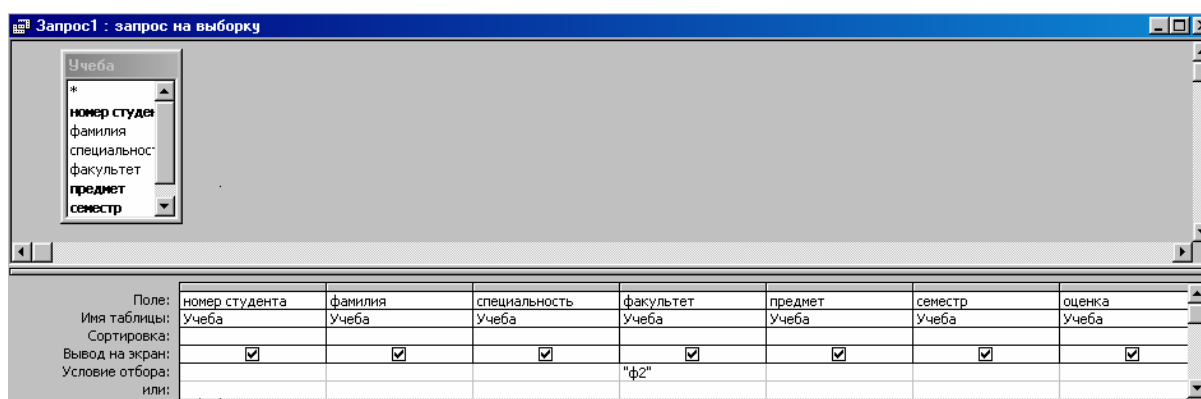


Рис.

10. Структура запроса-выборки по значению поля

### Параметрический запрос

Значение поля в условии запроса можно вводить не только непосредственно в шаблон структуры запроса, как в предыдущем примере, но и динамически в диалоге в процессе выполнения запроса. Для этой цели используется *параметрический запрос*.

**Контрольное задание 8.** Сформируйте параметрический запрос всех данных о результатах сдачи экзаменов по предмету, заданному его названием:

- 1) выберите все поля в структуру запроса, как в примере «Все поля»;
- 2) в поле «предмет» в строке «Условие отбора» наберите в *квадратных скобках* текст подсказки к вводу конкретного значения (рис. 11);

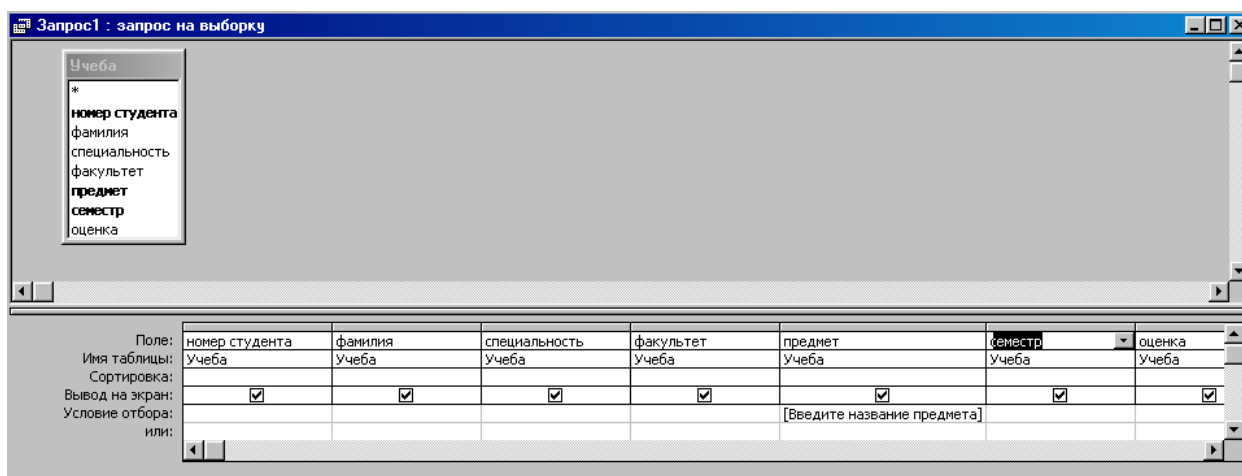


Рис. 11. Структура параметрического запроса

- 3) завершите результаты формирования запроса так, как это обсуждалось в алгоритме формирования запроса на стр. 18;

- 4) в поле диалогового окна, которое появится при выполнении запроса, введите конкретное значение, например «математика» (рис. 12);

- 5) сохраните запрос под именем «Параметрический».

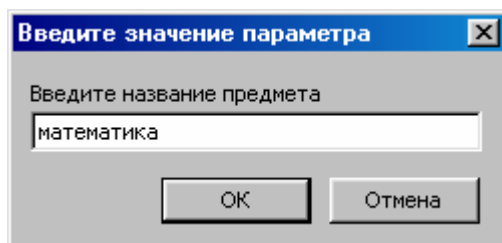


Рис. 12. Диалоговое окно параметрического запроса

Рассмотренный прием можно использовать в спецификации условий отбора с более сложной логикой.



### **Выборка по условию с операторами сравнения**

В спецификации условий выборки можно использовать операторы *сравнения* (=, <>, >, <, >=, <=). Для этого в шаблоне структуры запроса в строке «Условие запроса» в соответствующем поле указывается оператор сравнения.

**Контрольное задание 9.** Сформируйте запрос на выборку всех данных о студентах-младшекурсниках (1–3 курсы) может быть формализован условием «<7» по полю «семестр»:

- 1) выберите все поля в структуру запроса, как в примере «Все поля»;
- 2) в поле «семестр» в строке «Условие отбора» наберите «<7»;
- 3) завершите результаты формирования запроса так, как это обсуждалось в алгоритме формирования запроса на стр. 18;
- 4) назовите запрос «Меньше».

### **Выборка по условию с логическими операторами**

В спецификации условий выборки можно использовать *логические операторы* (Not, And, Or и т. д.). Речь идет о реализации сложных условий запроса. Для спецификации условия с **And** операнды логического выражения должны располагаться в строке «Условие запроса» на одном уровне в шаблоне структуры запроса. Для спецификации условия с **Or** операнды логического выражения должны располагаться в строке «Условие запроса» и последующих строках «или» на разных уровнях в шаблоне структуры запроса.

**Контрольное задание 10.** Выполните пример на тему And. Сформируйте запрос на выборку всех данных о результатах сдачи экзаменов студентами факультета «ф2» на оценки 4 и 5 (т.е. более 3).

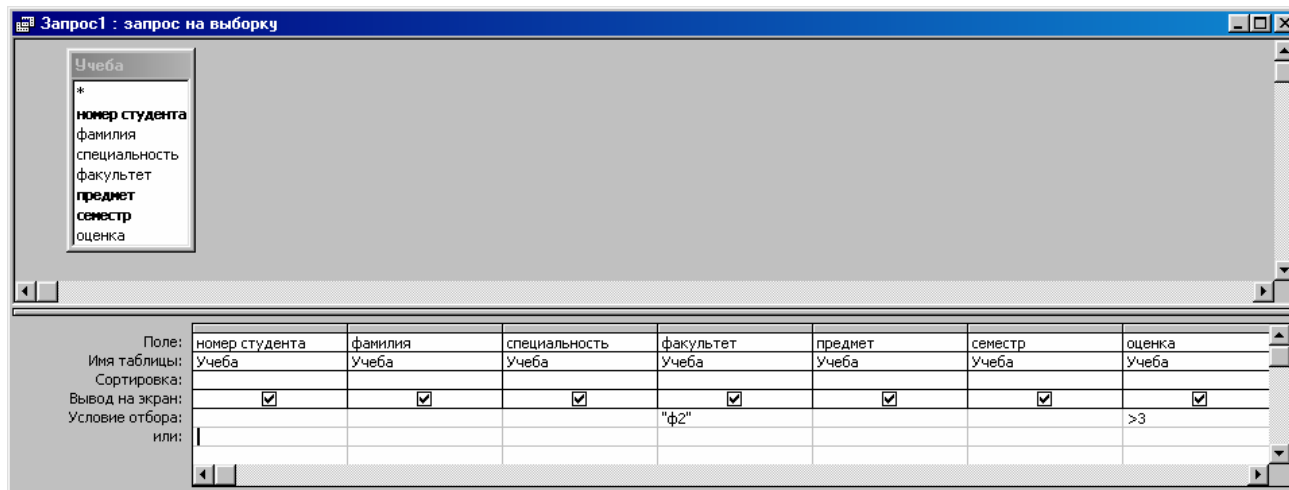
- 1) перенесите все поля в структуру запроса, как в примере «Все поля»;
- 2) в поле «факультет» в строке «Условие отбора» наберите «ф2», а в поле «оценка» **в той же строке** наберите «>3» (рис. 13);
- 3) завершите результаты формирования запроса так, как это обсуждалось в алгоритме формирования запроса на стр. 18;
- 4) назовите запрос «And».

**Контрольное задание 11.** Выполните пример на тему Or. Сформируйте запрос на выборку всех данных о результатах сдачи экзаменов студентами, которые или учатся на факультете «ф2», или получили оценки 4 и 5:

- 1) перенесите все поля в структуру запроса, как в примере «Все поля»;

2) в поле «факультет» в строке «Условие отбора» наберите «**ф2**», а в поле «оценка» в строке *или* наберите «>**3**»;

3) завершите результаты формирования запроса так, как это обсуждалось в алгоритме формирования запроса на стр. 18;



4) назовите запрос «Ог».

Рис. 13. Структура запроса с And

### ***Выборка по условию с оператором Between...And***

Оператор **Between** определяет принадлежность значения поля таблицы указанному диапазону значений и имеет следующий синтаксис применения: **Between «значение1» And «значение2»**, здесь «значение1» и «значение2» – выражения, задающие границы диапазона. Порядок следования операндов «значение1» и «значение2» значения не имеет.

Если значение поля попадает в диапазон, задаваемый выражениями «значение1» и «значение2» (включительно), оператор **Between...And** возвращает значение «истина»; в противном случае возвращается значение «ложь». Допускается проверка с помощью логического оператора **Not** обратного условия (т.е. того, что выражение не принадлежит диапазону значение1 – значение2).

**Контрольное задание 12.** Сформируйте запрос на выборку всех данных о результатах сдачи экзаменов студентами, которые сдали экзамены на 4 или 5 (**Between 4 And 5**):

- 1) перенесите все поля в структуру запроса, как в примере «Все поля»;
- 2) в поле «оценка» в строке «Условие отбора» наберите **Between 4 And 5** (рис. 14);

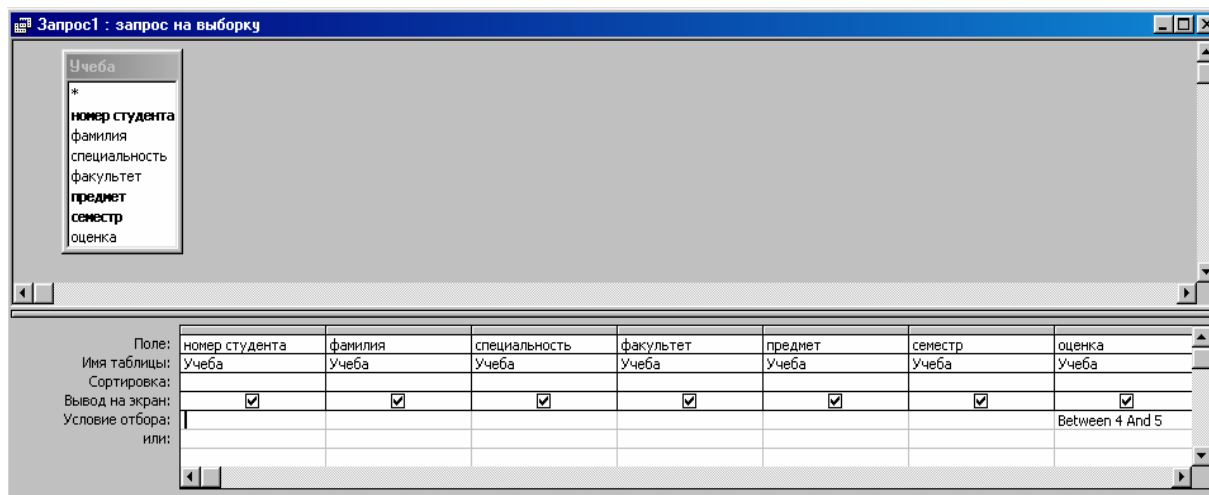


Рис. 14. Структура запроса с Between

3) завершите результаты формирования запроса так, как это обсуждалось в алгоритме формирования запроса на стр. 18;

4) назовите запрос «Between».

Границы диапазона можно задать параметрически по образцу **Between [от] And [до]**.

### ***Выборка по условию с оператором Like***

Оператор **Like** имеет синтаксис **Like «шаблон»** и используется для поиска в поле текстового типа значений, совпадающих с указанным образцом. Допускается как точное указание аргумента «шаблон» (например, Like «Иванов И.И.»), так и использование поддерживаемых операционной системой символов шаблона, определяющих диапазон значений (например, Like «И\*»).

При подстановке в выражение оператор Like позволяет сравнивать значение поля со строковым значением. Например, при вводе в строку «Условие отбора» таблицы запроса выражения Like «И\*» в запросе будут выведены все поля, содержимое которых начинается с буквы «И».

**Контрольное задание 13.** Сформируйте запрос на выборку всех данных о результатах сдачи экзаменов студентами, чьи фамилии заканчиваются на «ов» (Петров, Иванов и т.д.):

1) перенесите все поля в структуру запроса, как в примере «Все поля»;

2) в поле «фамилия» в строке «Условие отбора» наберите **Like «\*ов \*»** (после «ов» пробел);

3) завершите результаты формирования запроса так, как это обсуждалось в алгоритме формирования запроса на стр. 18 ;

4) назовите запрос «Like».

Запрос с Like, как любой запрос, может быть задан параметрически. В запросе с параметром допускается вывод для пользователя приглашения ввести образец поиска.

**Контрольное задание 14.** Сформируйте параметрический запрос на выборку всех данных о студентах, чьи фамилии начинаются с первых букв (одной, двух, сколько нужно по контексту поиска), задаваемых в диалоге:

1) перенесите все поля в структуру запроса, как в примере «Все поля»;  
2) в поле «фамилия» в строке «Условие отбора» наберите **Like [Введите первые буквы фамилии:]&«\*»** (рис. 15);

3) завершите результаты формирования запроса так, как это обсуждалось в алгоритме формирования запроса на стр. 18. При выполнении запроса будет открыто окно диалога с надписью над полем ввода «Введите первые буквы фамилии». Если пользователь введет в поле буквы «Ив», то в запросе будет выполняться поиск по шаблону «Ив\*» (т.е. будут найдены все фамилии, начинающиеся с букв «Ив»);

4) назовите запрос «Like параметрический».

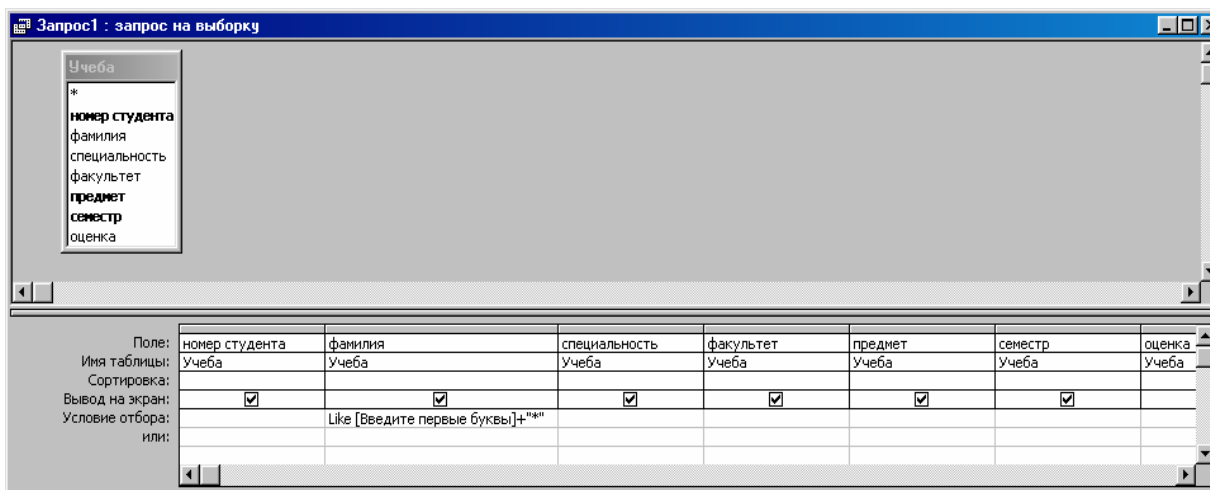


Рис. 15. Структура запроса с Like

### **Выборка по условию с оператором In**

Оператор **In** имеет синтаксис **In(«значение1»;«значение2»; ...)** и проверяет значение поля таблицы на совпадение с любым из элементов указанного списка значений. Если содержимое поля совпадает со значением одного из элементов списка, оператор **In** возвращает значение «истина»; в противном случае возвращается «ложь». Допускается проверка с помощью логического оператора

Not обратного условия (т.е. того, что выражение не принадлежит списку значений).

**Контрольное задание 15.** Сформируйте запрос на выборку всех данных о результатах сдачи экзаменов по базовым дисциплинам (физика, математика и др.):

- 1) перенесите все поля в структуру запроса, как в примере «Все поля»;
- 2) в поле «специальность» в строке «Условие отбора» наберите **In («физика»;«математика»)** (рис. 16);
- 3) завершите результаты формирования запроса так, как это обсуждалось в алгоритме формирования запроса на стр. 18;
- 4) сохраните запрос под именем «In».

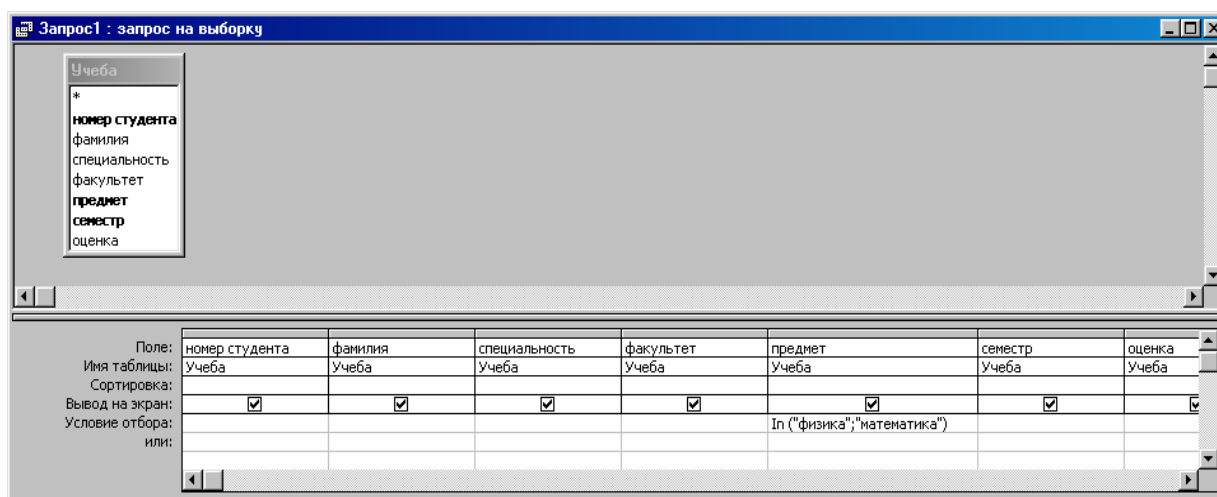


Рис. 16. Структура запроса с In

### **Выборка с вычисляемыми полями**

В структуре запроса наряду с полями реальных таблиц могут быть представлены данные, которые отсутствуют в реальных таблицах, но могут быть выведены из хранимых данных при помощи арифметических, логических и других операторов, а также встроенных функций. Типичными примерами такого рода данных являются вычисляемые (виртуальные) и агрегированные данные (количество, сумма, среднее, минимальное, максимальное значения и др.).

**Контрольное задание 16.** Сформируйте запрос-выборку с вычисляемыми данными из БД УЧЕБА, представленной одной таблицей «Учёба». Допустим, что по условиям функционирования нашей модельной ПО «Учёба» две первые цифры номера студента указывают на год его зачисления в вуз. Допустим также, что датой зачисления студента является 1 сентября года приема. До-

бавим в структуру запроса на выборку всех данных после поля «номер студента» вычисляемое поле «дата зачисления», построенное по следующей формуле:

**DateValue(«20»+Mid([Учеба]![номер студента];1;2)+«01.09»)**

Здесь функция **DateValue(«строка символов»)** преобразует строковое представление даты в данное типа дата.

Функция **Mid(«строка символов»; «начало»; «длина»)** вырезает из «строки символов», начиная с позиции «начало» (целое число), подстроку символов длиной «длина» (целое число) символов.

Оператор + выполняет сцепление (конкатенацию) строк символов.

Построение формулы для вычисления виртуального поля удобно производить под управлением **Построителя выражений** так, как это упоминалось на стр. 22.

Сформируйте запрос по следующему алгоритму:

- 1) перенесите все поля в структуру запроса, как в примере «Все поля»;
- 2) для добавления нового поля выделите поле, *слева* от которого будет выполнена вставка. В нашем случае это поле «фамилия». (Для выбора столбца следует поместить курсор в любую его ячейку или нажать кнопку в области маркировки полей в верхней части таблицы запроса);
- 3) выполните команду **Вставка|Столбцы**. (Для удаления поля следует выделить соответствующий столбец и выполнить команду **Правка|Удалить**);
- 4) в строке «Поле» наберите формулу для вычисляемого поля. Формулу можно вводить вручную или конструировать при помощи **Построителя выражений** (рис. 17) по принципу «снаружи внутрь»;
- 5) в списке «Функции» («Встроенные функции») в категории «Дата/время» найдите функцию **DateValue («stringexpr»)**;
- 6) аргумент функции «stringexpr» замените на сцепление трех строк символов: «20»+ Mid («stringexpr»; «start»; «length»)+ «01.09»;
- 7) функцию Mid найдете в списке «Функции» («Встроенные функции») в категории «Текстовые»: **Mid («stringexpr»; «start»; «length»)**;
- 8) аргумент «start» замените на «1» (с первой позиции), аргумент «length»– на «2» (подстрока длиной 2 символа);
- 9) аргумент «stringexpr» найдите в списке «Таблицы» («Учеба», поле «номер студента») **[Учеба]![номер студента]**;

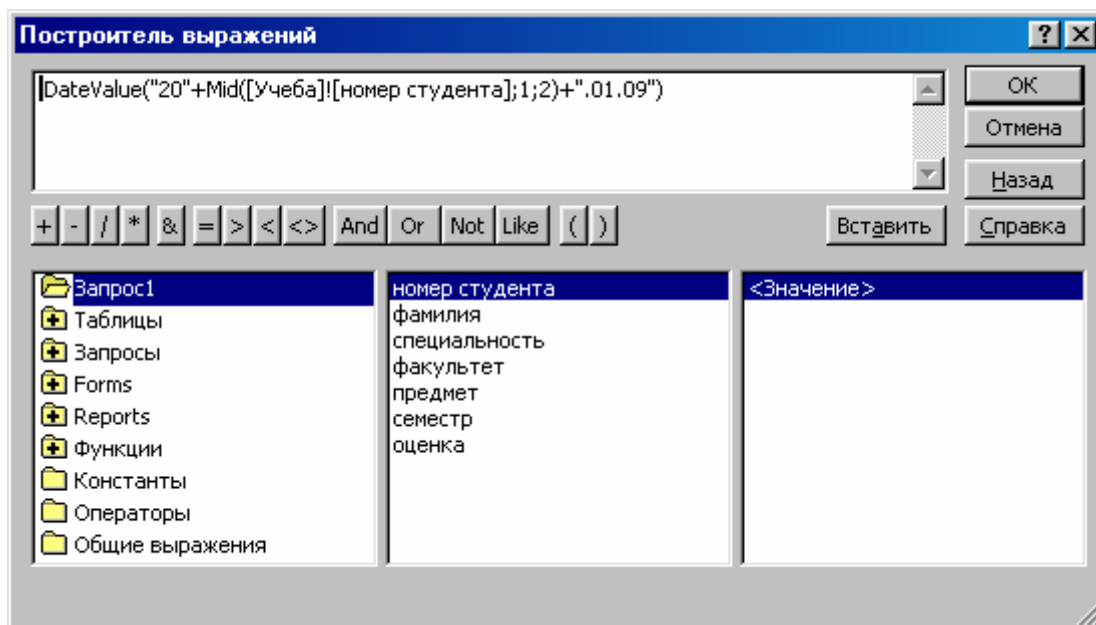


Рис. 17. Построение выражения для вычисляемого поля

10) в таблице результата запроса новое поле будет иметь имя «Выражение1». Чтобы присвоить ему содержательное имя, установите курсор в ячейку «Поле» и нажмите правую клавишу мыши;

11) в контекстном меню выполните команду **Свойства** и введите в диалоговом окне «Свойства поля» в строку «Подпись» имя поля «дата зачисления»;

12) завершите результаты формирования запроса так, как это обсуждалось в алгоритме формирования запроса на стр. 18;

13) результат запроса будет иметь вид, как на рис. 18;

14) сохраните запрос под именем «Вычисляемое поле».

номер студента	дата зачисления	фамилия	специальность	факультет	предмет	семестр	оценка
016510	09.01.01	Иванов И.И.	с2	ф2	экономика	9	3
021008	09.01.02	Иванов И.И.	с1	ф1	математика	2	4
021008	09.01.02	Иванов И.И.	с1	ф1	физика	2	4
028101	09.01.02	Андреев А.А.	с3	ф2	программирование	5	3
029812	09.01.02	Петров П.П.	с1	ф1	кибернетика	4	4
031512	09.01.03	Николаева Н.Н.	с2	ф2	математика	1	5
031512	09.01.03	Николаева Н.Н.	с2	ф2	математика	2	5
031512	09.01.03	Николаева Н.Н.	с2	ф2	физика	1	5
*							

Рис. 18. Результат запроса с вычисляемым полем

### Сортировка записей

Изначально записи-строки таблицы БД оказываются отсортированными по возрастанию ключа. Если по смыслу запроса требуется рассортировать записи по другому критерию, то формируют запрос на *сортировку*. Сортировка может выполняться по возрастанию или по убыванию значений поля. Сортировка может выполняться по нескольким полям. В этом случае порядок следования полей в структуре запроса определяет порядок сортировки (вложенная сортировка). Сначала выполняется сортировка по первому полю. В пределах внешней сортировки выполняется сортировка по второму полю и т. д.

**Контрольное задание 17.** Постройте запрос на сортировку результатов сдачи экзаменов в таблице «Учеба» по факультетам (по полю «факультет» по возрастанию), а в пределах факультета по специальностям (по полю «специальность» по убыванию):

- 1) перенесите все поля в структуру запроса, как в предыдущем примере;
- 2) в структуре запроса поменяйте местами поля «факультет» (первое поле) и «специальность» (второе поле);
- 3) в поле «факультет» в строке «Сортировка» мышью включите список способов сортировки и выберите «По возрастанию» (по алфавиту);
- 4) в поле «специальность» в строке «Сортировка» мышью включите список способов сортировки и выберите «По убыванию»;
- 5) завершите результаты формирования запроса так, как это обсуждалось в алгоритме формирования запроса на стр. 18;
- 6) сохраните запрос под именем «Сортировка».

### ***Выборка с группированием данных и вычислением функций агрегации***

Для просмотра интегральных характеристик данных по подмножествам записей удобно выполнять группирование данных. Для вычисления итоговых по группе характеристик (количество, сумма, среднее, минимальное, максимальное значения и др.) применяются *групповые функции*. Группировка данных может выполняться по нескольким полям, образуя группы и подгруппы. Так же как и в случае сортировки, порядок следования полей определяет порядок формирования групп и подгрупп. Особенность запроса на группировку данных состоит в том, что в структуре запроса можно использовать только те поля, по которым выполняется группировка, а также вычисляемые поля с функциями агрегации.



**Контрольное задание 18.** Допустим, что для каждого студента из БД УЧЕБА, представленной одной таблицей «Учёба», требуется подсчитать количество сданных экзаменов и средний балл по результатам сдачи всех экзаменов. Основанием для группирования является поле «номер студента» и «фамилия» (если мы хотим увидеть это поле в структуре запроса). Группировка по полю «фамилия» будет фиктивной, поскольку с каждым номером студента связана только одна фамилия. Вычисление групповых характеристик (количество и среднее значение) будет выполняться по полю «оценка».

Сформируйте запрос на выборку с группированием данных и вычислением групповых характеристик:

1) перенесите в структуру запроса поля «номер студента», «фамилия», «оценка», «оценка»;

2) выполните команду **Вид|Групповые операции** – в структуре запроса появится строка «Групповая операция». Группирование данных будет выполняться по полю «номер студента»; поле «фамилия» не создаст вложенной группы, так как уникальной характеристикой студента является его учетный номер. Поэтому значением групповой операции для этих полей будет значение по умолчанию «Группировка»;

3) в строке «Групповая операция» для первого поля «оценка» щелкните мышью и выберите из списка функцию подсчета количества Count, для второго поля – функцию вычисления среднего значения Avg (рис. 19);

4) в таблице результата запроса полям, по которым вычисляются групповые характеристики, присвойте содержательные имена;

5) установите курсор в ячейку «Поле» и нажмите правую клавишу мыши;

6) в контекстном меню выполните команду **Свойства** и введите в диалоговом окне «Свойства поля» в строку «Подпись» имена вычисляемых полей «количество экзаменов» и «средний балл». Результаты запроса будут иметь вид, как на рис. 20;

7) завершите результаты формирования запроса так, как это обсуждалось в алгоритме формирования запроса на стр. 18;

8) сохраните запрос под именем «Группировка».

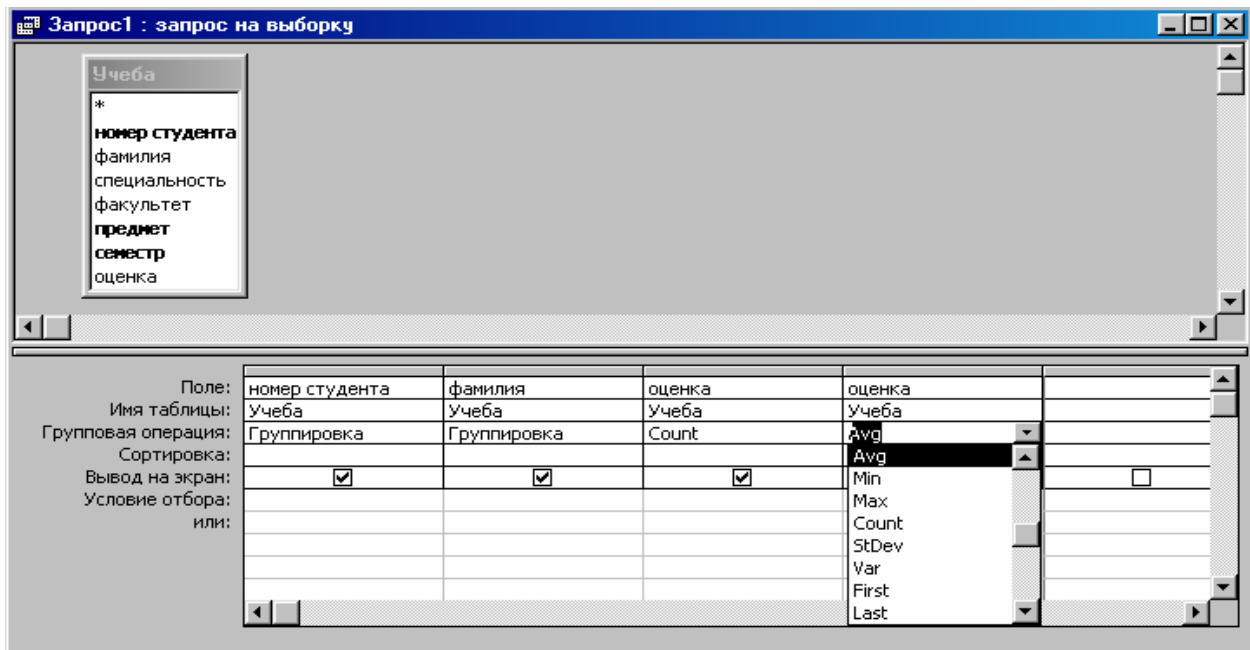


Рис. 19. Структура запроса на группирование данных

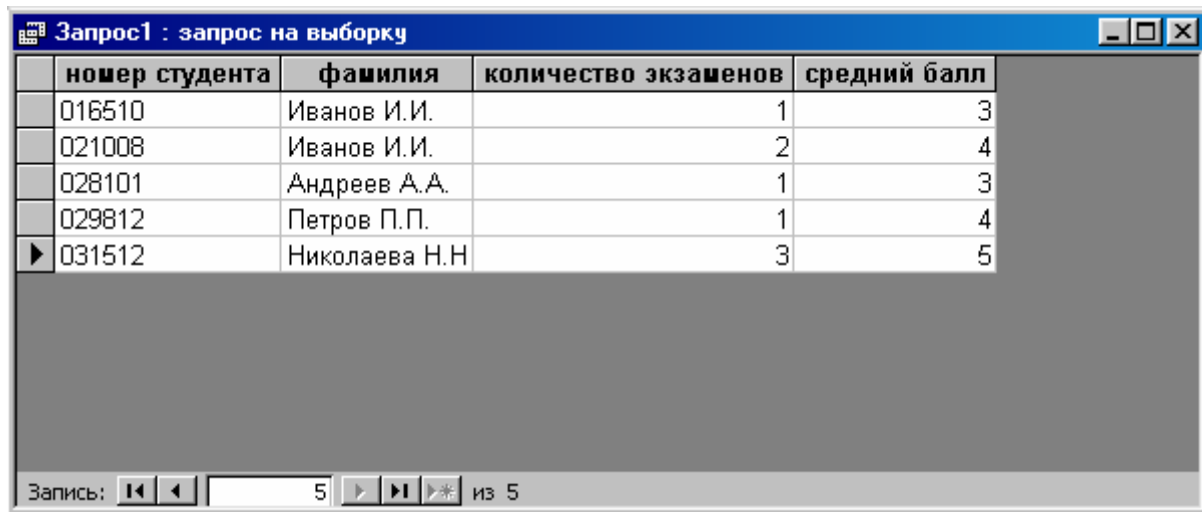


Рис. 20. Результаты запроса на группирование данных

**БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. Хомоненко А.Д. Базы данных : учебник для вузов / А.Д. Хомоненко, В.М. Цыганков, М.Г. Мальцев; под ред. проф. А.Д. Хомоненко. – Изд. 2-е, доп. и перераб. – СПб. : КОРОНА принт, 2002. 672 с.
2. Дейт К. Дж. Введение в системы баз данных : [пер. с англ.] / К. Дж. Дейт. – 6-е изд. – Киев; М.; СПб. : ИД«Вильямс», 1999. 848 с.
3. Коннолли Т. Базы данных: проектирование, реализация и сопровождение. Теория и практика : [пер. с англ.] / Т. Коннолли, К. Бегг, А. Страчан. – 2-е изд. – М. : ИД «Вильямс», 2000. 1120 с.
4. Ульман Дж. Основы систем баз данных / Дж. Ульман. – М. : Финансы и статистика, 1983.
5. Джексон Г. Проектирование реляционных баз данных для использования с микро-ЭВМ / Г. Джексон. – М. : Мир, 1991.
6. Бекаревич Ю.Б. MS Access 2000 за 30 занятий : практ. пособие / Ю.Б. Бекаревич, Н.В. Пушкина. – СПб. : ВHV, 2000. 492 с.
7. Вейскас Дж. Эффективная работа с Microsoft Access 2000 / Дж. Вейскас. – СПб. : Питер, 2000. 1040 с.
8. Иванов В.А. [Электронный ресурс].  
Режим доступа: <http://va-ivanov.narod.ru/access/index.html> .
9. Библиотека разработчика. Иллюстрированный самоучитель по Access: [Электронный ресурс].  
Режим доступа: [http://www.libray.narod.ru/Office/Access\\_2002/index.html](http://www.libray.narod.ru/Office/Access_2002/index.html) .
10. Компьютерное проектирование баз данных в среде СУБД Access. В 2 ч. Ч. 2. Нормализация структуры данных и корректирующие запросы: методические указания к лабораторному практикуму по дисциплинам «Базы данных», «Управление данными» / сост. В.Ю. Кара-Ушанов. – Екатеринбург : ГОУ ВПО УГТУ-УПИ, 2006. 46 с.

## ПРИЛОЖЕНИЕ

### Задания для самостоятельной работы

В предлагаемых ниже вариантах задач выполнить следующие задания:

- .1. Создать базу данных на основе одной таблицы.
- .2. Заполнить таблицу предлагаемыми данными.
- .3. Продумать и выполнить следующие типы запросов-выборок:
  - **проекция** на подмножество однородных атрибутов;
  - **параметрический запрос** на примере **селекции** по условию на основе операций сравнения (равно, не равно, больше, меньше, больше или равно, меньше или равно);
    - запрос-выборка, селекция по сложному условию на основе оператора конъюнкции **And**;
    - запрос-выборка, селекция по сложному условию на основе оператора дизъюнкции **Or**;
    - запрос-выборка, селекция по условию на основе оператора **Between...And**;
    - запрос-выборка, селекция по условию на основе оператора **Like**;
    - запрос-выборка, селекция по условию на основе оператора **In**;
    - запрос-выборка, содержащий **вычисляемое поле**;
    - запрос-выборка с **сортировкой** по двум-трем полям;
    - запрос-выборка с **группировкой** данных и вычислением статистических функций **Count, Sum, Avg, Min, Max** и др.

### Примеры задач

Самостоятельные упражнения выполнить на основе одного из нижеследующих примеров. При создании таблиц описание свойств полей задать самостоятельно, ориентируясь на фактические данные, приведенные в табл. П.1–П.5. Подчеркнуты ключевые поля.

Таблица П.1

**База данных Поставки<sup>5</sup>**  
**(номер поставщика, фамилия поставщика, город, код детали, название детали, количество за месяц)**

номер поставщика	фамилия поставщика	город	код детали	название детали	количество
п1	Смит	Лондон	д1	гайка	300
п1	Смит	Лондон	д2	болт	200
п1	Смит	Лондон	д3	винт	400
п1	Смит	Лондон	д4	винт	200
п1	Смит	Лондон	д5	кулачок	100
п1	Смит	Лондон	д6	шайба	100
п2	Джонс	Париж	д1	гайка	300
п2	Джонс	Париж	д2	болт	400
п3	Блейк	Париж	д2	болт	200
п4	Кларк	Лондон	д2	болт	200
п4	Кларк	Лондон	д4	винт	300
п4	Кларк	Лондон	д5	кулачок	400

<sup>5</sup> Пример взят из книги [2].

Таблица П.2

## База данных Больница

(номер пациента, фамилия пациента, адрес, номер хирурга, фамилия хирурга, дата операции, тип операции)

номер пациента	фамилия пациента	адрес	номер хирурга	фамилия хирурга	дата операции	тип операции
п1	Степанов	Екатеринбург	х2	Блинов	10.03.03	1
п1	Степанов	Екатеринбург	х3	Федоров	05.11.03	2
п3	Борисов	В. Пышма	х3	Федоров	10.11.03	2
п2	Иванов	В. Пышма	х4	Васильев	01.02.03	3
п4	Томин	Екатеринбург	х4	Васильев	10.02.03	4
п3	Борисов	В. Пышма	х5	Зонов	05.01.04	2
п5	Шмелев	Березовский	х3	Федоров	30.11.03	2

Таблица П.3

## База данных Библиотека

(номер абонента, фамилия абонента, номер книги, автор, название, год издания, дата выдачи, дата возврата)

номер абонента	фамилия абонента	номер книги	автор	название	год издания	дата выдачи	дата возврата
а1	Смирнов	к1	Шукшин	н1	1990	10.03.03	31.03.03
а1	Смирнов	к2	Булгаков	н2	1988	05.11.03	05.12.03
а3	Бобров	к7	Булгаков	н2	1988	10.11.05	30.11.05
а2	Иванов	к5	Чехов	н3	1980	01.02.03	20.02.03
а4	Тимофев	к4	Бунин	н5	1995	10.02.03	01.03.03
а3	Бобров	к6	Голстой	н7	1980	05.01.04	31.01.04
а5	Белов	к8	Бунин	н5	1995	30.11.03	30.12.03

Таблица П.4

**База данных Аренда (номер клиента, фамилия клиента, номер объекта, адрес объекта, плата, фамилия собственника, дата начала, дата конца)**

номер клиента	фамилия клиента	номер объекта	адрес объекта	плата	фамилия собственника	дата начала	дата конца
к1	Сергеев	е1	Екатеринбург	5000р	Богов	10.03.03	31.12.05
к1	Сергеев	е2	Екатеринбург	5000р	Фролов	05.11.03	05.11.04
к3	Григорьев	в7	Первоуральск	4000р	Фролов	10.11.03	31.12.05
к2	Иванов	в5	Первоуральск	3500р	Волков	01.02.03	01.05.03
к4	Костин	е4	Екатеринбург	4000р	Волков	10.02.03	10.03.03
к3	Григорьев	в6	Первоуральск	3000р	Зайцев	05.01.04	31.12.04
к5	Беляев	р8	Ревда	4000р	Фролов	30.11.03	30.05.04

Таблица П.5

**База данных Кадры (номер сотрудника, фамилия сотрудника, дата рождения, место работы,  
должность, прием, увольнение, учебное заведение, образование)**

номер со- трудника	фамилия со- трудника	дата рожде- ния	место ра- боты	должность	прием	увольнение	учебное за- ведение	образование
c1	Степанов	10.10.55	м1	д3	01.06.80	31.12.84	Вуз1	высшее
c1	Степанов	10.10.55	м1	д4	01.01.85	31.05.95	Вуз1	высшее
c3	Ковалев	15.05.66	м1	д2	01.07.86	31.08.92	Вуз2	высшее
c2	Петров	20.08.82	м1	д1	01.09.99		Школа1	среднее
c4	Шилов	25.06.70	м2	д2	01.07.90		Техникум1	сред.техн.
c3	Ковалев	15.05.66	м2	д3	01.09.92		Вуз2	высшее
c5	Кротов	18.04.80	м2	д2	01.10.99		Техникум2	сред.техн.
c1	Степанов	10.10.55	м1	д5	01.06.95		Вуз1	высшее



## ОГЛАВЛЕНИЕ

КОНЦЕПЦИЯ БАЗ ДАННЫХ В ACCESS .....	2
Модель данных Access .....	2
Создание базы данных.....	3
ПРОЕКТИРОВАНИЕ ТАБЛИЦ .....	6
Разработка структуры таблицы .....	6
Наполнение и редактирование таблицы.....	13
ФОРМИРОВАНИЕ ЗАПРОСОВ .....	16
Спецификация запросов .....	16
Проектирование структуры запроса .....	17
Запросы выборки.....	19
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	35
ПРИЛОЖЕНИЕ.....	36
Задания для самостоятельной работы.....	36
Примеры задач.....	36

**Учебное электронное текстовое издание**

**Кара-Ушанов Владимир Юрьевич**

**КОМПЬЮТЕРНОЕ ПРОЕКТИРОВАНИЕ  
БАЗ ДАННЫХ В СРЕДЕ СУБД ACCESS**

**Часть 1. Структура данных и запросы выборки**

**Редактор** *И.В. Коршунова*  
**Компьютерная верстка** *А.А. Гребенщикова*

**Рекомендовано РИС ГОУ ВПО УГТУ-УПИ  
Разрешен к публикации 06.09.06.**

**Электронный формат – PDF**

**Формат 60×90 1/8**

**Издательство ГОУ ВПО УГТУ-УПИ  
620002, Екатеринбург, ул. Мира, 19  
e-mail: sh@uchdep.ustu.ru**

**Информационный портал  
ГОУ ВПО УГТУ-УПИ  
<http://www.ustu.ru>**