

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Ордена Трудового Красного Знамени федеральное государственное бюджетное
образовательное учреждение высшего образования
Московский технический университет связи и информатики

Сосновилов Г.К., Воробейчиков Л.А.

**СРЕДСТВА РАЗРАБОТКИ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ
В СУБД ACCESS 2010
(Электронное пособие и лабораторный практикум)**

Учебное пособие

Москва 2017

УДК 004.625.4

Сосновиков Г.К., Воробейчиков Л.А., Средства разработки реляционных баз данных в СУБД ACCESS 2010. Электронное пособие и лабораторный практикум, 2017г., ЭБ МТУСИ, №27399.

В учебном пособии рассматриваются основные возможности и средства СУБД MS Access 2010 по организации баз данных. Пособие не предназначено для изучения теоретических основ организации реляционных баз данных, и основные сведения о них лишь в сжатом виде изложены в первом разделе пособия. Пособие предназначено для студентов, выполняющих цикл лабораторных работ с использованием лабораторного практикума.

Для студентов МТУСИ, обучающихся по направлениям 09.03.01, 09.03.02, 01.03.04.

Рецензенты: канд. техн. наук, доцент

1. ОБЩИЕ СВЕДЕНИЯ О РЕЛЯЦИОННЫХ БАЗАХ ДАННЫХ

1.1. Основные понятия и определения

База данных (БД) - это поименованная совокупность данных, относящихся к определенной предметной области.

Система управления базами данных (СУБД) - это комплекс программных и языковых средств, необходимых для создания, обработки баз данных и поддержания их в актуальном состоянии.

Почти все современные СУБД основаны на *реляционной модели* данных. Все данные в реляционной БД представлены в виде прямоугольных *таблиц*, каждая из которых содержит информацию о некотором *классе объектов*. Каждая строка таблицы содержит информацию только об одном объекте и называется *записью*. Столбец таблицы содержит однотипную для всех записей информацию и называется *полем*. Между отдельными таблицами устанавливаются *связи*, позволяющие работать с ними как с единым целым.

Для успешного функционирования базы данных важна правильная организация данных в ней. При определении структуры данных в базе важны следующие основные понятия, относящиеся к концептуальной (инфологической) модели БД:

Класс объектов или сущность - совокупность объектов, обладающих одинаковым набором свойств. Например, в базе данных о ВУЗе классами объектов являются студенты, преподаватели, дисциплины. Для каждой сущности в БД создается своя таблица, а для каждого экземпляра сущности – отдельная запись в таблице.

Свойство или атрибут - определенная часть информации о некотором объекте. Хранится в виде столбца (поля) таблицы. Например, *фамилия, номер зачетной книжки, год рождения* - это атрибуты сущности *Студент*.

Связь - способ, которым связана информация о разных классах объектов.

1.2. Таблицы и связи между ними. Первичные и внешние ключи

Основным структурным компонентом базы данных является таблица. При определении состава таблиц следует руководствоваться правилом: **в каждой таблице должны храниться данные только об одном классе объектов**. Например, в одной и той же таблице нельзя хранить анкетные данные студентов и наименования изучаемых дисциплин, т.к. это свойства

разных классов объектов. Если в базе данных должна содержаться информация о разных классах объектов, то она должна быть распределена по отдельным таблицам.

В примерах настоящего учебного пособия будет использоваться база данных **Успеваемость студентов** (рис. 2.5), содержащая информацию о студентах различных групп и факультетов, а также об их успеваемости по различным дисциплинам. При этом можно выделить следующие классы объектов и установить соответствующий им состав таблиц:

таблица **Факультеты**, содержащая информацию о факультетах, на которых обучаются студенты: код и наименование факультета, фамилию и инициалы декана;

таблица **Группы**, содержащая информацию об учебных группах, в которых обучаются студенты: код и номер группы, код факультета, год создания группы;

таблица **Студенты**, содержащая информацию об анкетных данных студентов и об их принадлежности к той или иной учебной группе: код студента и номер зачетной книжки, фамилию, инициалы и пол студента, набранные баллы ЕГЭ, код группы, дату рождения и город проживания, размер стипендии;

таблица **Дисциплины**, содержащая информацию об учебных дисциплинах, по которым фиксируется успеваемость студентов: код и название дисциплины, количество отводимых часов;

таблица **Оценки**, содержащая информацию о полученных оценках на экзаменах отдельными студентами по различным дисциплинам: код студента, код дисциплины, оценка данного студента по данной дисциплине.

Для ввода, редактирования и обработки данных, содержащихся в этих таблицах, между ними должны быть установлены связи. Для связывания таблиц в каждой из них должно быть выделено поле, однозначно идентифицирующее и уникальное для каждой записи (**ключевое поле** или **первичный ключ**). Первичный ключ может быть **простым** (из одного поля) и **составным** (из нескольких полей). Если среди полей таблицы нет естественных кандидатов на эту роль, или составной ключ содержит слишком много полей, то ключевое поле создается искусственным образом и представляет собой **автоинкрементное** поле, уникальные значения которого автоматически поддерживаются СУБД.

Связи между любыми двумя таблицами относятся к одному из трех типов: один-к-одному (1 : 1), один-ко-многим (1 : М) и многие-ко-многим (М : М).

При связи типа **“один-к-одному”** каждой записи в одной таблице соответствует не более одной записи в другой таблице. Этот вид связи встречается довольно редко, в основном в тех случаях, когда часть информации об объекте либо редко используется, либо является конфиденциальной (такая информация хранится в отдельной таблице, которая защищена от несанкционированного доступа). Связь **“один-к-одному”** устанавливается по одинаковому первичному ключу обеих таблиц.

При связи типа **“один-ко-многим”** каждой записи в одной таблице соответствует нуль, одна или несколько записей в связанной таблице. Это наиболее распространенный тип связи. Одна из связываемых таблиц представляет сторону "один", другая – сторону "многие". Для установления этого типа связи в таблице со стороны "многие" должно содержаться поле того же типа и длины, что и первичный ключ таблицы на стороне "один": так называемый **внешний ключ**. Каждая запись таблицы должна содержать в этом поле значение соответствующего первичного ключа таблицы на стороне "один".

Так, например, таблица **Факультеты** на стороне "один" будет иметь автоинкрементный первичный ключ с именем **Код**, а таблица **Группы** на стороне "многие" – внешний ключ с именем **КодФакультета**. Связь между таблицами будет установлена по этим полям.

При связи типа **“многие-ко-многим”** множеству записей в одной таблице соответствует множество записей в связанной таблице. Большинство современных СУБД непосредственно не поддерживают такой тип связи. Для его реализации такая связь разбивается на две связи типа **“один-ко-многим”**. Соответственно, для хранения информации потребуется уже три таблицы: две на стороне "один" и одна на стороне "много". Связь между этими тремя таблицами также осуществляется по ключевым полям.

Так, например, таблица **Оценки** реализует связь **“многие-ко-многим”** между сущностями **Студенты** и **Дисциплины** путем замены ее на две связи **“один-ко-многим”**: **Студенты – Оценки** и **Дисциплины – Оценки** (рис. 2.5).

1.3. Краткие сведения о СУБД MS Access

Первая версия MS Access была создана в 1993 г. фирмой Microsoft. MS Access - это функционально полная реляционная СУБД, работающая в среде Windows. MS Access позволяет создавать сложные базы данных, определять структуру таблиц и связи между ними.

В MS Access база данных включает в себя все объекты, связанные с хранимыми данными (таблицы, формы, отчеты, запросы). Все объекты MS Access хранятся в одном файле с расширением **.accdb**. В таблицах хранятся данные, которые можно просматривать и редактировать. Используя формы, можно выводить данные на экран в удобном виде, просматривать и редактировать их. Запросы позволяют быстро выбирать необходимую информацию из таблиц, а также изменять содержимое таблиц. С помощью отчетов можно создавать различные виды документов для вывода на печать.

В настоящем учебном пособии используется версия MS Access 2010, входящая в состав пакета Microsoft Office 2010.

2. СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ТАБЛИЦ

2.1. Начало работы с MS Access. Создание новой таблицы

После запуска Access нужно в окне *Microsoft Access* из списка доступных шаблонов выбрать шаблон **Новая база данных**, задать имя для файла БД и нажать кнопку **Создать** (рис. 2.1). После этого на экране появляется **основное окно базы данных** (рис. 2.2), из которого можно получить доступ ко всем ее объектам: таблицам, запросам, формам и т.д. В верхней части окна находится **главное меню** с вкладками **Файл, Главная, Создание** и пр.

Под главным меню располагается так называемая **лента** с кнопками и другими объектами управления работой СУБД, объединенными в группы. Вид ленты и состав объектов управления на ней зависит от выбранного пункта главного меню (выбранной вкладки) и текущего состояния процесса работы с СУБД.

Выбор того или иного действия (команды) с обрабатываемой базой данных в дальнейшем будет обозначаться последовательностью наименований вкладок и управляющих объектов ленты (групп и кнопок), разделенных наклонной чертой. Например, **Создание/Таблица** будет обозначать выбор вкладки **Создание** с последующим нажатием кнопки **Таблица**.

Для создания новой таблицы нужно выполнить команды **Создание/Таблица** или **Создание/Конструктор таблиц**. В первом случае свойства таблицы и ее полей задаются объектами ленты, при выборе второго способа Access выводит **окно Конструктора таблицы**, в котором задаются имена, типы и свойства полей для создаваемой таблицы.

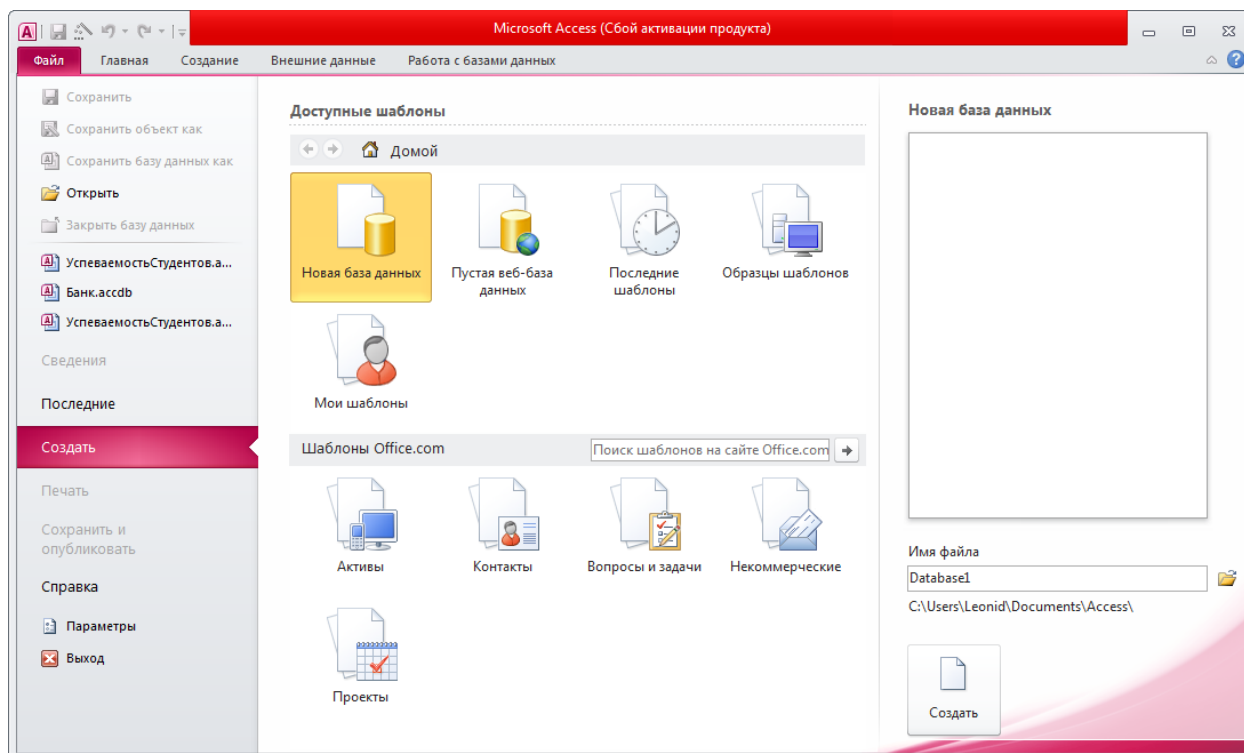


Рис. 2.1. Диалог загрузки Access

Имена полей рекомендуется выбирать, исходя из смыслового соответствия хранящимся в них атрибутам таблицы. В качестве имен могут быть использованы слова и группы слов любого языка, в том числе – русского.

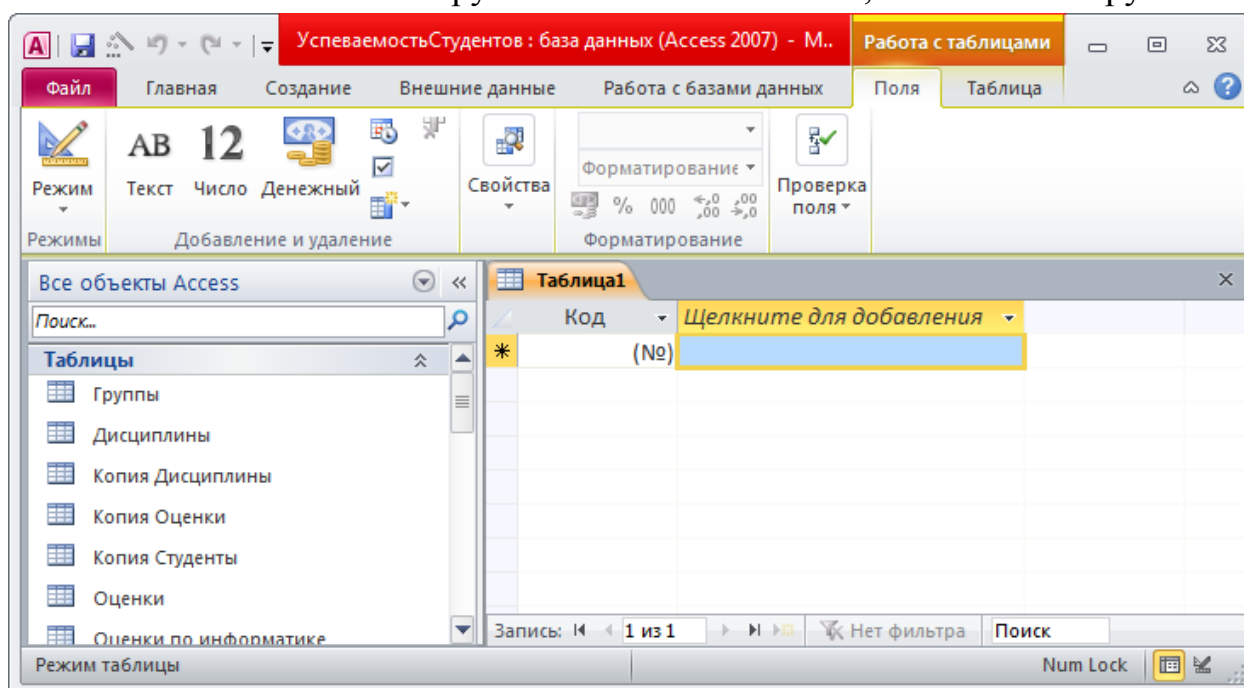


Рис. 2.2. Основное окно Access

Тип данных поля должен соответствовать хранящемуся в нем атрибуту и может быть выбран из следующего списка основных типов:

- **Текстовый.** Алфавитно-цифровые данные (до 255 символов).
- **Поле Метод.** Алфавитно-цифровые данные - предложения, абзацы, тексты (до 64 000 символов).
- **Числовой.** Различные числовые данные (имеет несколько форматов).
- **Дата/время.** Дата и (или) время в одном из предлагаемых Access форматов.
- **Денежный.** Денежные суммы, хранящиеся с 8 знаками в десятичной части. В целой части каждые три разряда разделяются запятой.
- **Счетчик.** Уникальное длинное целое, создаваемое Access для каждой новой записи в возрастающей последовательности. Редактировать значения такого поля нельзя.
- **Логический.** Логические данные, имеющие значения *Истина* или *Ложь*.
- **Поле объекта OLE.** Картинки, диаграммы и другие объекты OLE из приложений Windows.
- **Гиперссылка.** В полях этого типа хранятся гиперссылки, которые представляют собой путь к файлу на жестком диске либо адрес в Интернете.

Кроме вышеперечисленных типов данных в списке есть элемент **Мастер подстановок**, который позволяет представить значения полей в виде простого или комбинированного списка. Дополнительные свойства такого поля представлены на вкладке **Подстановка** окна конструктора таблиц. Использование подстановок подробно рассматривается в разделе 2.5.

Каждое поле обладает индивидуальными свойствами, по которым можно установить, как должны сохраняться, отображаться и обрабатываться данные. Набор свойств поля зависит от выбранного для него типа данных. Для определения свойств поля используется бланк **Свойства поля** в нижней части окна конструктора таблиц.

Свойства полей устанавливаются с помощью вкладок **Общие** и **Подстановка** в нижней половине окна конструктора таблиц. Рассмотрим эти свойства.

Размер поля. Для текстового поля это максимальное число символов (до 255). По умолчанию устанавливается размер 255 символов. При выборе размера надо учитывать, что в поле размером 20 символов нельзя будет ввести текст, длиной 30 символов. С другой стороны, установка слишком большой длины может привести к бесполезному увеличению размера файла базы данных, если значения, хранимые в поле, будут существенно меньше его установленной длины.

Для числовых полей размер выбирается из списка возможных числовых типов данных. Числовой тип данных, в свою очередь, выбирается исходя из формы хранимых чисел (целые или с плавающей точкой) и диапазона возможных значений.

Формат устанавливает вид данных на экране, например, для типа Дата/время можно выбрать представление даты вида 30 июня 2007 г. Или 30.06.2007.

Число десятичных знаков определяет количество цифр после десятичной точки для дробных чисел. Это значение влияет только на вид представления числовых величин, а не на способ их хранения в памяти.

Маска ввода позволяет задать шаблон для ввода, который дает некоторую гарантию правильности ввода данных. Она применяется для полей типа Дата/время и текстовых. Маска ввода выводит на экран символы-местозаменители (дефис, скобки и т.п.) и показывает, сколько символов нужно ввести. Например, для даты маска ввода может выглядеть так: __.__.____. Данная маска кодируется набором символов 00.00.0000. Заполнитель 0 означает, что разрешается вводить только цифры, причём их ввод является обязательным. При выборе свойства Маска ввода справа появляется кнопка, по которой запускается мастер создания маски.

Значение свойства Подпись используется в качестве заголовка столбца в режиме таблицы.

Значение по умолчанию обеспечивает при добавлении новой записи автоматический ввод значения, заданного этим свойством.

Свойство Условие на значение позволяет задать диапазон допустимых значений поля, а свойство Сообщение об ошибке – текст сообщения, которое будет появляться при вводе значений поля вне заданного диапазона.

Свойство Обязательное поле имеет два значения: Да и Нет. Если установить значение Да, Access потребует обязательного ввода какого-либо значения в данное поле.

Свойство Пустые строки определяет, допускается ли ввод в данное поле пустых строк.

Свойство Индексированное поле определяет, будет или нет проводиться индексация по данному полю. Индексация состоит в создании списка номеров записей, упорядоченных в соответствии со значениями поля. Наличие индекса ускоряет операции поиска и сортировки, но требует дополнительного места на диске.

Для значения Да свойства Индексированное поле возможны две установки. Установка Совпадения не допускаются означает, что данный

индекс является уникальным, то есть повторение значений данного поля в различных записях невозможно. Установка *Допускаются совпадения* означает, что данный индекс не является уникальным, и значения данного поля в различных записях могут повторяться. Так, например, в таблице **Группы** индекс по полю *НомерГруппы* должен быть уникальным, не допускающим совпадения номеров групп в различных записях, а индекс по полю *КодФакультета* – неуникальным, так как различные группы могут относиться к одним и тем же факультетам.

Просмотреть и отредактировать созданные индексы возможно в режиме конструктора таблиц при нажатии кнопки *Индексы*.

Для сохранения структуры таблицы нужно ввести команду *Файл/Сохранить* и в окне *Сохранение* ввести имя таблицы. Другим способом сохранения таблицы является щелчок правой кнопкой мыши на заголовке таблицы и выбор команды *Сохранить* из контекстного меню.

2.2. Определение ключевых полей

Любая таблица базы данных в обязательном порядке должна иметь первичный ключ, однозначно идентифицирующий каждую запись. В таблице не может быть нескольких записей с одним и тем же значением ключа. В многотабличной базе данных ключевые поля используются также для установления связей между таблицами.

Первичным ключом таблицы может быть любое поле, однозначно идентифицирующее ее записи. Так, например, в таблице **Студенты** таким полем мог бы быть номер зачетной книжки, а в таблице **Группы** – номер группы. Однако в этом случае всегда возникает вопрос: может ли при определенных условиях значение ключа быть изменено? Представим, например, что руководство ВУЗа решило изменить систему нумерации групп. В таком случае в базе данных придется выполнить эту перенумерацию не только в таблице **Группы**, но и во всех связанных с ней таблицах, например, в таблице **Студенты**.

Поэтому более предпочтительным является использование в качестве первичного ключа специально создаваемого автоинкрементного поля типа “счетчик”, значения которого автоматически формируются при вводе новых записей и не могут быть изменены пользователем. Такие ключи обладают рядом важных преимуществ:

- снимается проблема обновления внешних ключей в связанных таблицах, так как первичный ключ типа “счетчик” не может быть изменен;
- разработчик базы данных получает полную свободу в изменении содержимого и свойств неключевых полей таблицы, так как такие изменения уже не влияют на связанные таблицы;
- поле типа “счетчик” имеет относительно небольшую длину (8 байтов), что позволяет экономно использовать память в связанных таблицах;
- упрощаются и унифицируются связи между таблицами.

Для определения поля как ключевого нужно, находясь в конструкторе таблиц, установить курсор в строке с именем этого поля и нажать на кнопку **Ключевое поле**. После этого в строке рядом с именем выбранного поля появится изображение ключа. Для отмены выбора ключевого поля достаточно еще раз нажать кнопку **Ключевое поле**, находясь в строке с ключевым полем. Для переопределения ключевого поля надо выполнить это действие, находясь в строке с любым другим полем таблицы.

В таблицах, реализующих связь “многие-ко-многим”, первичный ключ часто бывает составным, состоящим из двух или более внешних ключей. Так, в таблице **Оценки** первичный ключ состоит из двух внешних ключей **КодСтудента+КодДисциплины**. Такой ключ позволяет избежать дублирования оценок студентов по одной и той же дисциплине.

Чтобы определить составной первичный ключ таблицы, надо выделить ключевые поля в столбце ключей с помощью мыши и клавиши **Shift**, а затем нажать кнопку **Ключевое поле**. При этом изображения ключа появляются во всех соответствующих строках.

Если к моменту сохранения создаваемой таблицы ключевое поле не определено, то Access предлагает и в случае подтверждения автоматически создает первичный ключ, вставив перед первым указанным в структуре полем поле **Код** и задав для него тип “счетчик”.

Типы данных полей таблицы и их свойства выбираются, исходя из природы хранимых в них данных. Так, если над содержимым поля могут производиться вычисления, его тип целесообразно установить числовым. Полям, в которых хранятся денежные суммы, лучше задавать денежный тип. Поля логической природы с двумя возможными значениями “Да” и “Нет” естественно делать логического типа.

На рис. 2.3 приведено окно с таблицей **Студенты** в режиме конструктора. Типы полей в таблице выбраны в соответствии с приведенными рекомендациями.

2.3. Определение индексов

Как уже отмечалось в разделе 2.1, создание индексов по отдельным полям таблицы позволяет ускорить выполнение операций поиска и сортировки по этим полям, а также выполнение запросов, в которых участвуют индексируемые поля. Кроме того, создание уникальных индексов позволяет избежать повторения значений в этих полях. Следует иметь в виду, что по ключевому полю (полям) таблицы автоматически создается уникальный индекс с именем *PrimaryKey*.

Студенты	
Имя поля	Тип данных
Код	Счетчик
НомерЗачетки	Текстовый
ФИО	Текстовый
Пол	Текстовый
БаллыЕГЭ	Числовой
КодГруппы	Числовой
ДатаРождения	Дата/время
Город	Текстовый
Стипендия	Денежный

Рис. 2.3 Типы полей таблицы “Студенты”

Индексы занимают дополнительное место на диске. Кроме того, на их модификацию при обновлении данных в таблице тратится дополнительное время. Потому индексы следует создавать только по тем полям, по которым будет производиться поиск и сортировка данных, а также по полям, участвующим в запросах к базе данных.

Уникальные индексы следует создавать по тем полям или их комбинациям (составные индексы), которые не должны повторяться при вводе данных в таблицу, если такая уникальность не обеспечивается первичным

ключом таблицы. Например, таблица **Студенты** в базе данных **Успеваемость студентов** имеет ключевое поле **Код** типа “счетчик”, а поле **НомерЗачетки** является не ключевым. Очевидно, что по номеру зачетки может производиться поиск и сортировка данных, и этот атрибут может использоваться в различных запросах к БД. Поэтому по полю **НомерЗачетки** следует создать индекс, причем он должен быть уникальным во избежание появления в таблице **Студенты** нескольких записей о студентах с одним и тем же номером зачетки.

Индекс по некоторому полю может быть создан путем соответствующей установки свойства **Индексированное поле** в конструкторе таблиц (см. раздел 2.1). Кроме того, индексы могут быть созданы, отредактированы и просмотрены в окне **Индексы** (рис. 2.4), появляющемся при нажатии кнопки **Индексы** в конструкторе таблиц. Использование окна индексов является единственной возможностью для создания составных индексов по нескольким полям.



Рис. 2.4. Окно индексов таблицы “Студенты”

2.4. Установка связей между таблицами

После определения структуры таблиц БД и определения ключевых полей необходимо установить связи между таблицами. Для определения связей нужно перейти в окно базы данных и выполнить команду **Работа с базами данных/Схема данных** или, находясь в конструкторе таблиц, нажать кнопку

Схема данных. После этого Access открывает окно **Схема данных**. По щелчку правой кнопкой мыши появляется контекстное меню, в котором надо выбрать команду **Добавить таблицу**, в появившемся окне **Добавление таблицы** выделить имена таблиц, добавляемых в схему данных, нажать на кнопку **Добавить**. После этого окно **Добавление таблицы** можно закрыть.

В окне **Схема данных** появятся изображения всех указанных таблиц со списками их полей. Нужно выделить ключевое поле таблицы со стороны “один”, нажать кнопку мыши, перетащить ее курсор на соответствующее поле в связываемой таблице, после чего кнопку мыши отпустить. В результате появится диалоговое окно **Изменение связей**.

В этом окне Access заполнил первую строку именами полей, по которым связывались таблицы. Чтобы в связанных таблицах не нарушалась целостность данных, нужно включить флажок **Обеспечение целостности данных**. После этого Access сделает невозможным:

- запись в таблицу со стороны “многие” такого значения внешнего ключа, которого нет в таблице со стороны “один”;
- удаление из таблицы со стороны “один” записи с таким ключом, который присутствует как внешний ключ в таблице со стороны “многие”.

Например, если таблицей со стороны “один” является таблица **Факультеты**, а таблицей со стороны “многие” – **Группы**, то после включения обеспечения целостности будет невозможно добавить в таблицу **Группы** запись о группе факультета, которого нет в таблице **Факультеты**. В то же время становится невозможным удаление записи о некотором факультете из таблицы **Факультеты**, если в таблице **Группы** присутствуют записи о группах этого факультета.

После включения обеспечения целостности данных в окне **Изменение связей** становятся доступными две дополнительные опции: **Каскадное обновление связанных полей** и **Каскадное удаление связанных полей**.

Если выбрать первую опцию, то при изменении какого-либо значения первичного ключа в таблице со стороны “один” Access автоматически обновит значения соответствующего внешнего ключа в соответствующих записях всех связанных таблиц. Как уже говорилось, если первичным ключом является поле типа “счетчик”, то изменение первичного ключа невозможно, поэтому выбор этой опции не имеет смысла.

Выбор второй опции при удалении одной из записей в таблице со стороны “один” приведет к удалению тех записей в таблицах со стороны “многие”, которые имеют такое же значение ключа. Например, если из таблицы

Факультеты удалить запись об одном из факультетов, то записи обо всех группах удаленного факультета в таблице **Группы** будут удалены автоматически. Поэтому к выбору данной опции следует относиться с осторожностью. При отказе от каскадного удаления при каждой попытке удаления записи таблицы со стороны "один" Access будет сообщать о наличии записей с соответствующим внешним ключом в связанных таблицах и предлагать пользователю удалить эти записи. Так, например, если таблица **Группы** содержит записи о группах некоторого факультета, то удаление записи об этом факультете из таблицы **Факультеты** станет возможным только после удаления всех таких записей из таблицы **Группы**.

На рис. 2.5 приведена схема данных БД **Успеваемость студентов**, описанной в разделе 1.2.

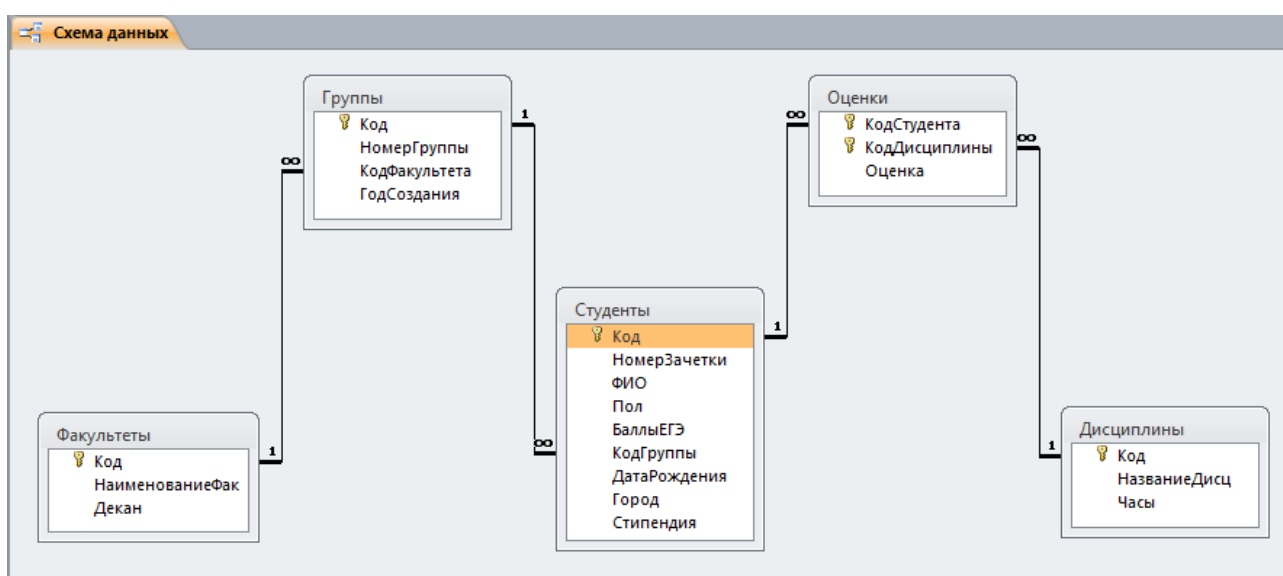


Рис. 2.5. Схема данных БД "Успеваемость студентов"

2.5. Использование подстановок при определении структуры таблиц и вводе данных в них

Мастер подстановок позволяет формировать для нужного поля список значений, который может содержать данные другой таблицы или запроса, либо состоять из фиксированного набора значений. В обоих случаях **Мастер подстановок** облегчает ввод данных и обеспечивает их достоверность.

Если первичными ключами являются поля типа "счетчик", то использование подстановок является единственным разумным способом организации ввода данных в связанные таблицы. В этом случае вообще

отпадает необходимость видеть первичные и внешние ключи таблиц и оперировать с ними.

Так, например, чтобы не вводить в поле **КодФакультета** таблицы **Группы** ни о чем не говорящие целые числа (значения первичных ключей из таблицы **Факультеты**), следует определить для этого поля подстановку наименования факультета с выбором его из комбинированного списка. Тогда при вводе данных в таблицу **Группы** пользователю будет достаточно выбрать из списка нужное название факультета, а его код будет подставлен в таблицу автоматически.

Для вызова **Мастера подстановок** нужно открыть соответствующую таблицу (в нашем примере - **Группы**) в режиме конструктора, выбрать поле подстановки (в нашем примере - **КодФакультета**), перейти в колонку **Тип данных** и из списка доступных типов полей выбрать элемент **Мастер подстановок**. Если между таблицами уже установлена связь, то Access предложит разорвать ее.

Первое диалоговое окно **Мастера подстановок** предлагает выбрать источник формирования списка: на основе данных таблицы/запроса или фиксированный набор значений. В данном случае нужно выбрать первый вариант.

В следующем окне из приведенного списка таблиц следует выбрать таблицу, являющуюся источником данных для списка. В нашем примере такой таблицей является таблица **Факультеты**.

В третьем окне **Мастера подстановок** из списка доступных полей нужно выбрать поля, значения которых будут фигурировать в списке подстановки. В нашем примере надо выбрать поля **Код** и **НаименованиеФак**.

В следующих окнах можно задать порядок сортировки элементов списка (например, по наименованию факультета), установить ширину столбцов и скрыть ключевой столбец, задать подпись для поля подстановки и включить проверку целостности данных.

После этого следует нажать кнопку **Готово** для завершения процесса создания комбинированного списка подстановки. При этом между таблицами автоматически создается соответствующая связь. На вкладке **Подстановка** в окне свойств поля подстановки можно увидеть результаты работы **Мастера**. Следует заметить, что создать комбинированный список в поле подстановки можно и без использования **Мастера**, путем непосредственного выбора и ввода свойств поля на вкладке **Подстановка**.

Другая возможность использования **Мастера подстановок** - это создание списков с фиксированными значениями. Этой возможностью следует

пользоваться в тех случаях, когда подставляемые значения берутся не из таблицы, а представляют собой конечный набор значений некоторого атрибута. Пусть, например, стипендия студента может принимать только 3 возможных значения: 0 – отсутствует, 1000 руб. – обычная и 1500 руб. – повышенная. Тогда вместо ввода этих значений в соответствующее поле можно создать в этом поле список подстановки с тремя фиксированными значениями.

2.6. Сортировка данных

При открытии таблицы Access выводит записи в последовательности, определенной значениями первичного ключа этой таблицы. Если ключ не определен, то записи выводятся в последовательности их ввода. Для изменения последовательности записей в таблице используется сортировка.

Для управления сортировкой необходимо открыть или перевести таблицу в **Режим таблицы**. В этом режиме на вкладке **Главная** доступны объекты управления раздела **Сортировка и фильтр**. Для сортировки таблицы по значениям только одного поля используются кнопки **По возрастанию** и **По убыванию**. Для отбора записей, удовлетворяющих тем или иным условиям на значения в выбранном поле, используется большая кнопка **Фильтр**. Для отмены или восстановления условий отбора используется малая кнопка **Фильтр**.

Сортировка и отбор записей таблицы по нескольким полям выполняется с помощью **расширенного фильтра**, для установки которого открывается окно при выборе одноименного пункта из списка **Дополнительно**.

Сортировать таблицу по нескольким полям имеет смысл только тогда, когда в поле, по которому сортируются записи в первую очередь, есть много повторяющихся значений. Сортировка по второму полю будет проводиться уже только среди этих записей.

Например, если отсортировать таблицу **Группы** по возрастанию в поле **КодФакультета** в первую очередь и по возрастанию в поле **НомерГруппы** - во вторую, то в пределах каждого факультета группы будут упорядочены по их номерам.

2.7. Поиск и замена данных

С увеличением количества записей поиск определенной информации усложняется. Access помогает значительно упростить и ускорить этот процесс.

Для проведения поиска в таблице нужно открыть эту таблицу, перейти на любую запись в поле, по которому предполагается вести поиск, выбрать вкладку **Главная** и нажать кнопку **Найти** в одноименной группе объектов управления.

В появившемся после этого окне **Поиск и замена** необходимо ввести в поле **Образец** искомое значение. В списке **Поиск в:** следует выбрать область поиска: **Текущее поле** или **Текущий документ**; в списке **Совпадение** – уровень совпадения искомого значения с образцом: **Поля целиком**, **С любой частью поля** или **С начала поля**; в списке **Просмотр** – направление просмотра при поиске: **Все**, **Вверх** или **Вниз**.

Расширить круг поиска можно, применив символы подстановки: * - любое количество символов; ? - только один символ. Например, по образцу ***ова** будут найдены все фамилии, заканчивающиеся на “ова”: Петрова, Иванова, Смирнова и т.д.

Для замены образца на требуемое значение необходимо в окне **Поиск и замена** выбрать вкладку **Замена** и в поле **Заменить на:** ввести заменяющее значение.

3. СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ФОРМ

3.1. Основные сведения о формах

Формы являются основным средством организации интерфейса пользователя в приложениях Access. Хорошо разработанные формы позволяют работать с приложением даже неподготовленному пользователю. При этом следует отметить, что в формах, а также запросах в режиме таблицы и отчетах в режиме представления, не должны фигурировать коды записей (поля типа “счетчик”), значения которых не подлежат редактированию и ни о чем не говорят пользователю.

Чаще всего формы создаются в следующих целях:

- ввод и редактирование данных – это наиболее распространенный способ использования форм. Формы обеспечивают вывод на экран данных в удобном для пользователя виде;
- управление ходом выполнения приложения;
- вывод сообщений – с помощью форм можно вывести на экран информацию, предупреждение или сообщение об ошибках;

- печать информации – несмотря на то, что для печати информации чаще всего используются отчеты, можно напечатать информацию, содержащуюся в форме.

3.2. Создание форм инструментом Форма

Работа с формами выполняется инструментами группы **Формы** вкладки **Создание** (рис. 3.1):

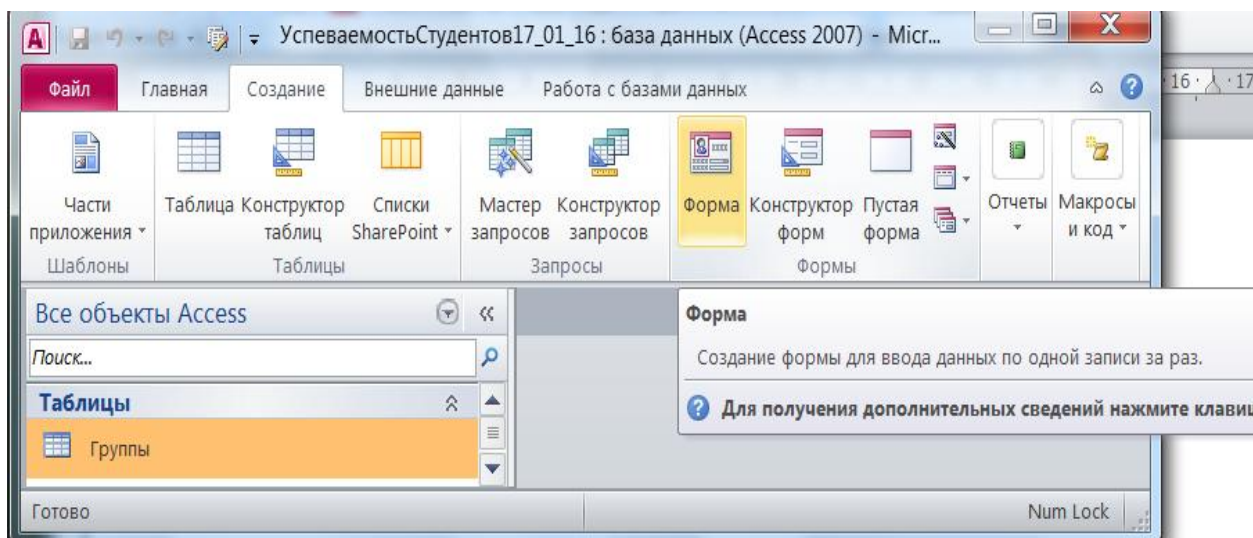


Рис. 3.1. Окно инструментов для создания формы

С помощью инструмента **Форма** создается форма для выбранного источника данных для отображения одной записи.

Например, выберем в списке таблиц таблицу **Группы** и нажмем кнопку **Форма**. Для выбранной таблицы автоматически создается форма (рис. 3.2). На форме отображаются и соответствующие записи студентов текущей группы из таблицы **Студенты** за счет связи между этими таблицами (см. раздел 2.4).

Как видно из рисунка, при создании формы с помощью инструмента **Форма** в нее включаются все поля основной и связанной таблицы, в том числе поля типа “счетчик”. То же самое происходит и при использовании некоторых других инструментов, которые могут быть выбраны из списка **Другие формы** на ленте вкладки **Создание/Формы**. Исключить лишние поля из формы можно в режиме **Конструктора** (см. раздел 3.4)

[illegible]

Рис. 3.2. Окно с формой *Группы*, созданной инструментом *Форма*

3.3. Использование Мастера для создания форм

Формы, созданные с помощью **Мастера**, более разнообразны по стилю оформления, могут содержать выбранные поля, в том числе и из нескольких связанных таблиц.

Для запуска **Мастера форм** нужно выполнить команду **Создание/Мастер форм** и далее последовательно выбирать соответствующие параметры создаваемой формы:

- на первом шаге необходимо выбрать таблицы и их поля для будущей формы;
- на втором шаге предлагается задать внешний вид формы;
- на третьем шаге нужно ввести имя создаваемой формы.

3.4. Создание и редактирование форм в режиме Конструктора

Изменять форму можно, используя режим **Макета** и режим **Конструктора**. Переключаться между режимами работы с формой можно командами группы **Режим** на вкладке **Главная**, командами контекстного меню или кнопками в правом нижнем углу формы.

В режиме **Макета** на форме отображается информация из таблицы или запроса, и при этом можно редактировать элементы управления.

Создавать и редактировать формы любой степени сложности позволяет только **Конструктор** форм. В этом режиме можно, в том числе, отредактировать формы, созданные **Мастером** или другими инструментами (например, инструментом **Форма**).

Любая форма в Access состоит из различных объектов, которые имеют характерные для них свойства. Для каждого объекта можно определить действия, выполняемые при наступлении определенных событий. Процесс создания формы в режиме **Конструктора** состоит в размещении объектов в областях формы и определении их свойств, связанных с ними событий и выполняемых действий.

Для создания формы в режиме **Конструктора** нужно выполнить команду **Создание/Конструктор форм**. При открытии окна **Конструктора** форма содержит только один раздел – область данных, и форма создается путем размещения элементов управления на пустом бланке конструктора.

Для внесения изменений в существующую форму нужно перейти в режим конструктора командой **Конструктор** контекстного меню или щелчком по соответствующей кнопке. Например, форма, изображенная на рис. 3.2, при переходе в режим **Конструктора** выглядит так, как показано на рис. 3.3. При этом форма может быть отредактирована с использованием инструментов, рассматриваемых ниже.

3.5. Разделы формы

Бланк конструктора формы состоит из трех разделов. В разделе **Заголовок формы** находится эмблема и название формы. Раздел **Примечание формы** на рис. 3.3 пуст. В этих разделах, помещаются элементы, которые не меняются при переходе от записи к записи. Данные разделы можно размещать на бланке или убирать с бланка конструктора командой контекстного меню **Заголовок/Примечание формы**.

Размеры разделов можно изменять, перетаскивая границу между разделам и на линейке, расположенной на бланке конструктора слева. Размеры самой формы также можно изменять, перетаскивая ее границы.

В разделе **Область данных** для каждого поля таблицы помещена надпись с названием поля таблицы и текстовое поле, в котором отображается содержимое поля соответствующей записи таблицы.

Новые элементы управления помещаются на форму с помощью инструментов, размещённых в группе **Элементы управления**.

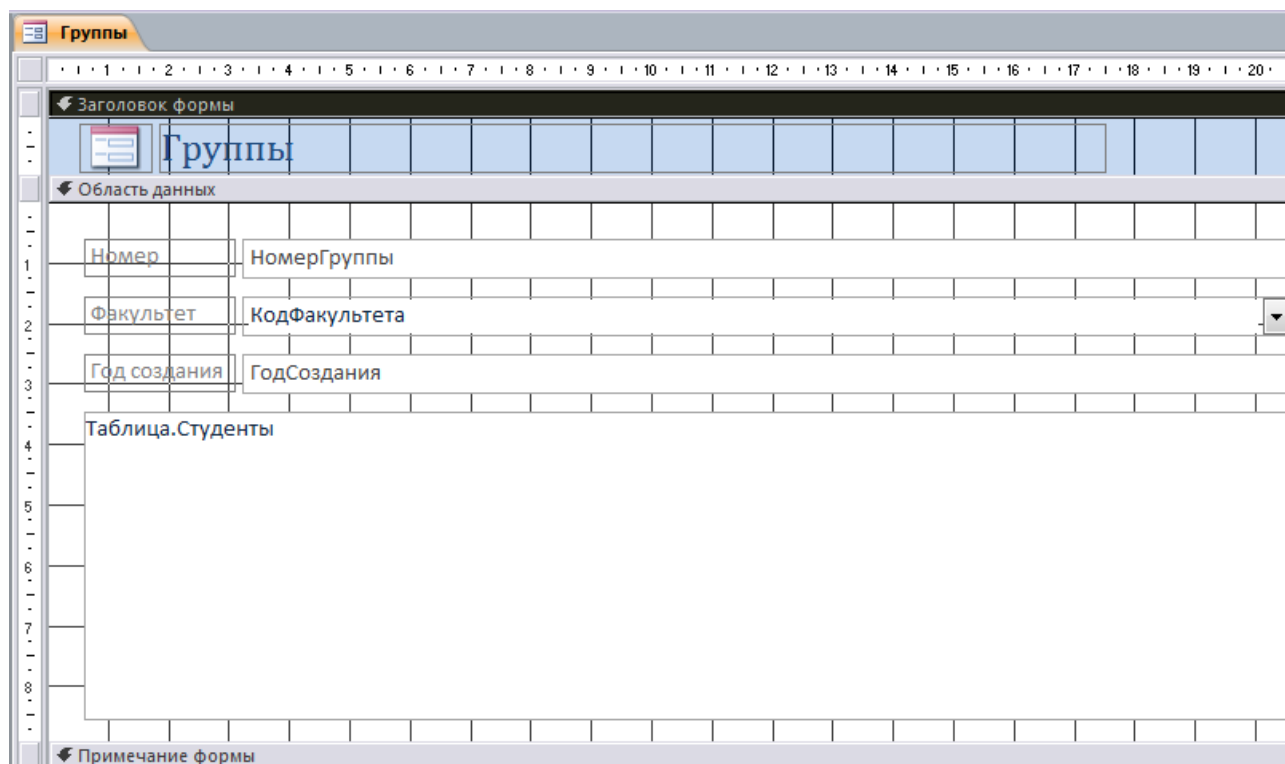


Рис. 3.3. Окно с отредактированной формой *Группы* в режиме *Конструктора*

3.6. Панель элементов и список полей

Для размещения на форме нового элемента управления (ЭУ) его нужно выбрать щелчком мыши на панели элементов, перейти на бланк конструктора, при этом курсор должен принять форму крестика, и нарисовать прямоугольник, удерживая левую кнопку мыши. В этом прямоугольнике будет размещен созданный ЭУ.

Ниже перечислены основные кнопки ЭУ на панели элементов и указано их назначение.

- **Выбор объектов.** Позволяет выделить определённый элемент управления или группу элементов управления, если держать нажатой клавишу *Shift*.
- **Текстовое поле** для ввода и редактирования текста.
- **Надпись.** Описательный текст.
- **Кнопка.** Используется для выполнения заданных действий при нажатии.
- **Вкладка.** Позволяет разделить форму на несколько вкладок.
- **Группа переключателей.** Содержит набор кнопок, флажков или выключателей.

- **Поле со списком.** В поле можно ввести новое значение или выбрать существующее из списка.
- **Линия.** Используется для графического разделения групп элементов на форме.
- **Выключатель.** Может быть в двух состояниях: *включено* и *выключено*.
- **Список.** Содержит значения, из которых можно сделать выбор.
- **Прямоугольник.** Используется для графического объединения групп элементов на форме.
- **Флажок.** Используется для отображения и управления логическими полями таблиц.
- **Переключатель.** Несколько переключателей обычно объединяются в группу и позволяют выбрать одно из нескольких взаимоисключающих значений.
- **Подчинённая форма/отчёт.** В форму добавляется информация из дополнительной таблицы.

Элементы управления на бланке конструктора выделяются щелчком мыши. Если выделен какой-либо элемент, становятся доступными соответствующие объекты закладки **Формат**, на которой расположены инструменты для выбора шрифта, способов выравнивания, цвета и т.д.

При размещении в форме полей из базовой таблицы удобно пользоваться **Списком полей**, окно с которым можно вывести нажатием кнопки **Добавить поля**. Чтобы поместить какое-либо поле в форму, достаточно просто перетащить его левой кнопкой мыши из **Списка полей** в нужный раздел формы.

На рис. 3.4 показано окно конструктора форм с панелью элементов и кнопкой **Добавить поля**.

3.7. Свойства объектов формы

Все объекты формы характеризуются своими свойствами, которые можно настроить в соответствии с определенными требованиями.

Для того чтобы получить доступ к свойствам объекта формы, следует сначала выделить этот объект в окне конструктора формы, а затем в контекстном меню выполнить команду **Свойства**, либо нажать кнопку **Страница свойств** на панели инструментов. В результате на экране появится окно со свойствами и событиями выбранного объекта.

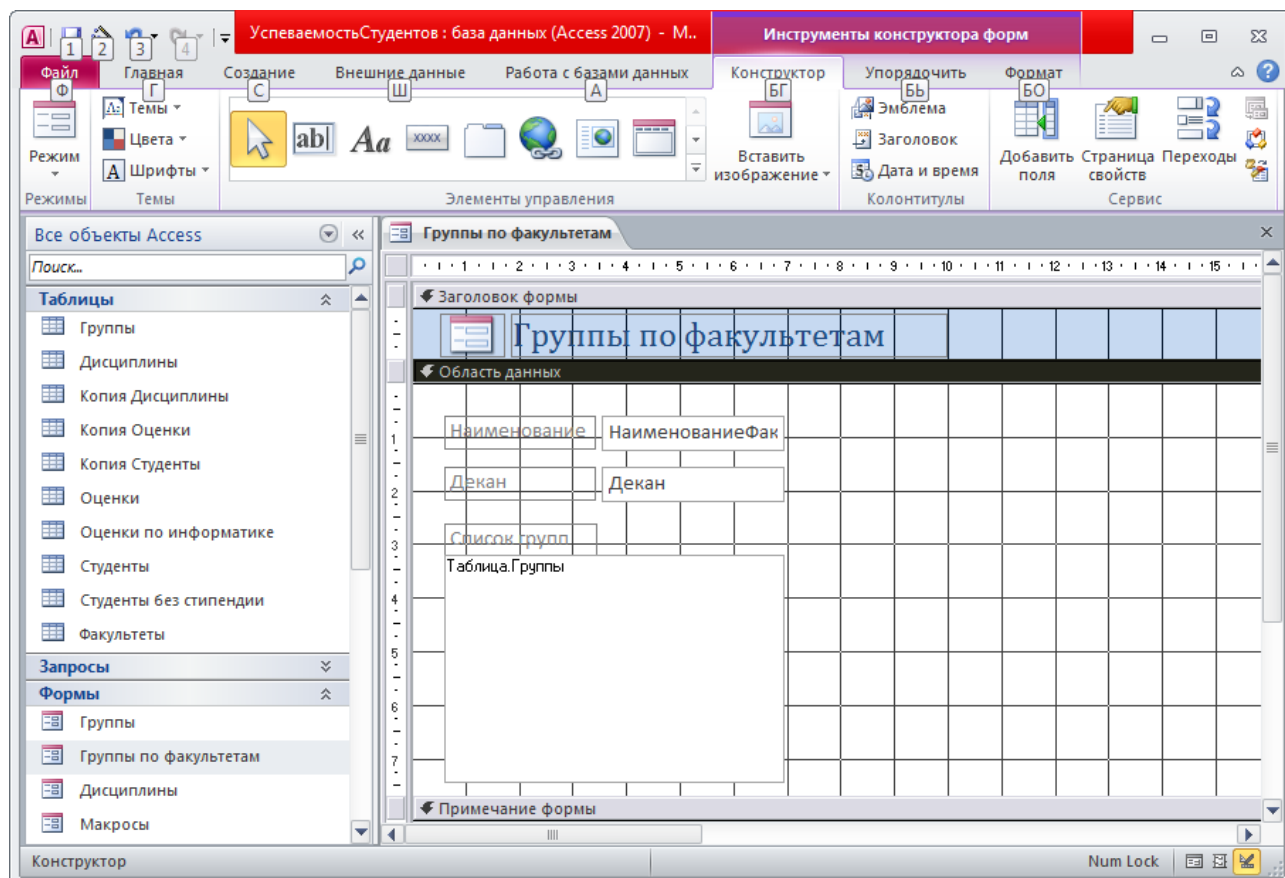


Рис. 3.4. Окно конструктора форм с панелью элементов

При размещении в форме нового объекта для него устанавливаются значения свойств, принятые по умолчанию. Некоторые свойства наследуются из базовой таблицы.

Каждый объект формы имеет свое имя, которое используется в дальнейшем для ссылок на данный объект. По умолчанию новым объектам присваивается имя, состоящее из названия элемента и номера объекта, который определяется порядком создания, например, **Поле1** или **Надпись3**. Не допускается существования в одной форме двух объектов с одинаковыми именами, однако в разных формах объекты могут иметь одинаковые имена.

Если объект формы создается путем переноса поля из списка полей, то в его свойство **Имя** копируется значение свойства **Имя поля**. Например, если из списка полей в форму перенесено поле **НаименованиеФак**, то этому объекту в форме будет дано имя **НаименованиеФак**.

Объекты формы, созданные путем переноса из списка полей, называются связанными. С их помощью можно просматривать или менять значение в связанном поле. Список полей в конструкторе выводится командой **Добавить поля**. Перед перетаскиванием поля из **Списка полей** предварительно можно

выбрать какой-либо элемент управления на панели элементов. Тогда в результате перетаскивания будет создан объект соответствующего типа, связанный с полем, например, **переключатель** или **список**. По умолчанию при перетаскивании поля на форму создаётся текстовое поле.

3.8. Создание управляющих кнопок

Кнопки используются в форме для выполнения определенного действия или ряда действий. Например, можно создать кнопку, которая будет открывать таблицу, запрос или другую форму. Можно создать набор кнопок для перемещения по записям таблицы, если не устраивают стандартные средства перемещения, предусмотренные в форме.

В Access предусмотрено более 30 стандартных кнопок, при создании которых достаточно воспользоваться **Мастером** по созданию кнопок, для использования которого должен быть включен режим использования мастеров на панели быстрого доступа. В этом случае при создании в форме объекта типа **Кнопка** открывается первое из окон **Мастера**, в которых выбираются категория кнопки, выполняемые ею действия, ее внешний вид и имя.

3.9. Управление объектами формы

В процессе создания формы все ее объекты можно перемещать, удалять, изменять их свойства. Для этого нужный объект, прежде всего, следует выделить. Выделенный объект имеет маркеры выделения в виде небольших квадратиков, расположенных по углам и серединам сторон.

Для выделения одного объекта достаточно просто щелкнуть по нему мышью. Выделив несколько объектов, можно управлять их общими свойствами.

Можно создать *группу* объектов, выделяя их щелчками мыши при нажатой клавише **Shift**. Тогда возможно перемещать одновременно все объекты группы. Ряд действий по выравниванию, рассмотренных ниже, применим только к группам объектов.

Для изменения размера объекта нужно переместить один из маркеров выделения до достижения нужного размера. Для перемещения выделенного объекта нужно установить указатель мыши на его границу так, чтобы указатель принял форму открытой ладони. После этого следует нажать кнопку мыши и переместить объект на новое место.



Рис. 3.5. Объекты поля таблицы в форме

Поле ввода и его подпись связаны друг с другом. Тем не менее, иногда нужно перемещать их порознь. Для этого используются метки перемещения – большие квадраты в левом верхнем углу каждого связанного объекта (рис. 3.5). Если подвести к ним указатель мыши, то он примет форму указательного пальца руки. В этом положении можно переместить каждый из связанных объектов отдельно.

Для удаления объекта формы нужно его выделить. После этого следует нажать клавишу **Delete**.

Команды управления размерами, взаимным расположением и выравниванием объектов формы находятся на вкладке **Упорядочить**.

Для изменения размеров или взаимного расположения объектов формы используется команда **Размер или Интервал**. Может быть выбран способ задания размера: **по размеру данных** или **по узлам сетки**. Если выделена группа элементов управления, то становятся доступными дополнительные команды установки размеров: **по самому высокому**, **по самому низкому**, **по самому широкому**, **по самому узкому**.

Взаимное расположение объектов, включенных в группу, определяется интервалами по вертикали и горизонтали. Для изменения интервалов нужно выполнить команду и группы **Интервал**.

Команда **Выровнять** позволяет задать способ выравнивания объектов формы. Для одиночного объекта доступна одна команда - **По сетке**, для группы объектов доступны дополнительные команды выравнивания: **Слева**, **Справа**, **Сверху**, **Снизу**.

3.10. Изменение порядка обхода объектов формы

Во время ввода данных в форме переход от одного объекта к другому при нажатии клавиши **ТАВ** осуществляется в соответствии с заданным в экранной форме порядком. Для определения этого порядка используется команда **Переходы**. При выполнении команды на экране откроется окно

диалога **Последовательность перехода**. В этом окне нужно расположить объекты в том порядке, в котором должен осуществляться их обход в форме. Для перемещения объекта на другое место в этом списке следует выделить его и, нажав кнопку мыши, переместить на новое место.

3.11. Разработка сложных форм

Создать форму, позволяющую просматривать и редактировать записи только одной таблицы, довольно просто. Но на практике требуются формы, позволяющие производить одновременный ввод или просмотр данных, хранимых в двух или более связанных таблицах. Например, при выводе информации о результатах экзаменов, было бы неплохо сразу размещать необходимую информацию о студентах. Эти данные реально хранятся в двух разных таблицах (**Студенты** и **Оценки**), связанных отношением “один-ко-многим”.

Access позволяет создавать многотабличные иерархические формы, в которых можно вводить, редактировать и просматривать все эти данные одновременно.

Иерархической называется форма, содержащая поля из главной и одной или нескольких подчиненных форм. Подчиненные формы используются для отображения данных из таблиц с отношением “один-ко-многим”. Наиболее простым способом создания иерархической формы является использование **Мастера форм**.

Как видно из рисунка 3.6, иерархическая форма состоит из двух частей. В верхней половине окна формы находятся поля таблицы **Студенты**, т.е. таблицы со стороны “один”. В нижней половине находится подчиненная форма, содержащая поля из таблицы **Оценки** (таблица со стороны “много”).

Для создания такой формы нужно:

- перейти на вкладку **Создание** в окне базы данных и нажать кнопку **Мастер форм**;
- в появившемся окне в качестве источника записей формы указать таблицу **Экзамены** (в иерархических формах базовой всегда является таблица со стороны “много”);
- в следующем окне **Мастера форм** определить главную и подчиненную таблицы. Здесь в качестве главной следует выбрать таблицу **Студенты**. В правой части данного окна диалога отображаются поля таблиц: в верхней части поля главной, а в нижней – поля подчиненной таблицы;

Списки экзаменов, сданных студентами

Номер зачетки

ФИО

Список сданных экзаменов

Дисциплина	Дата сдачи	Оценка
Информатика	12.01.2016	3
Вычислительная математика	14.01.2016	4
*		

Запись: 1 из 2

Нет фильтра

Поиск

Номер зачетки

ФИО

Список сданных экзаменов

[illegible]

Рис. 3.6. Пример иерархической формы

- выбрать опцию **Подчиненные формы**;
- далее следует выбрать внешний вид подчиненной формы: табличный или ленточный;
- на следующем шаге **Мастера форм** следует выбрать стиль, задать имена форм, а также указать режим, в котором будет открыта форма после завершения ее создания.

4. СОЗДАНИЕ И ВЫПОЛНЕНИЕ ЗАПРОСОВ-ВЫБОРОК

4.1. Понятие запроса–выборки

Запросы являются основным инструментом выборки и обработки данных в таблицах БД.

Различают средство формирования запросов по образцу (QBE – Query By Example) и структурированный язык запросов (SQL – Structured Query Language). С помощью инструкций языка SQL реализуется любой запрос в Access. В то же время Access позволяет создавать запросы, не прибегая к записи инструкций языка SQL. Простейшие запросы могут быть созданы с помощью **Мастера**; практически любой запрос можно создать в режиме **Конструктора**, путем заполнения бланка запроса в соответствующем окне.

При создании запроса по образцу Access сам автоматически создает эквивалентную инструкцию SQL, которую можно увидеть, переключившись в режим SQL. Конструктор позволяет создавать запросы простым и удобным способом, а просмотр этих запросов в режиме SQL позволяет понять и освоить синтаксис основных инструкций языка SQL, реализованного в Access.

Основным видом запроса является **запрос на выборку**. Результатом выполнения этого запроса является новая таблица, которая существует до закрытия запроса. Структура такой таблицы определяется выбранными из одной или нескольких взаимосвязанных таблиц полями. Записи формируются путем объединения записей таблиц, на которых построен запрос. Способ объединения записей таблиц указывается при определении их связи в схеме данных или при создании запроса. Условия отбора, сформулированные в запросе, позволяют фильтровать записи, составляющие результат объединения таблиц.

Запросы могут служить источниками записей для других запросов, форм, отчетов. С помощью запроса можно собрать полные сведения для формирования некоторого документа из нескольких таблиц, и далее использовать его для создания формы — электронного представления этого документа. Если форма или отчет создаются мастером на основе нескольких взаимосвязанных таблиц, то для них в качестве источника записей автоматически формируется запрос.

4.2. Создание запроса

Для создания запроса следует выполнить команду **Создание/Запросы/Мастер запросов** или **Создание/Запросы/Конструктор запросов**. При выборе **Мастера запросов** открывается окно **Новый запрос** (рис.4.1), где можно выбрать вид создаваемого запроса.

Простой запрос. Запускается мастер, помогающий создать запрос на выборку из одной или нескольких таблиц без возможности отбора записей по заданным условиям.

Перекрёстный запрос. Специальный запрос для анализа данных по двум или более признакам.

Повторяющиеся записи. В этом режиме запускается мастер, который просматривает выделенную таблицу или запрос и отбирает все записи, содержащие повторяющиеся значения в заданном поле. Его можно использовать для проверки возможности установления связей между таблицами.

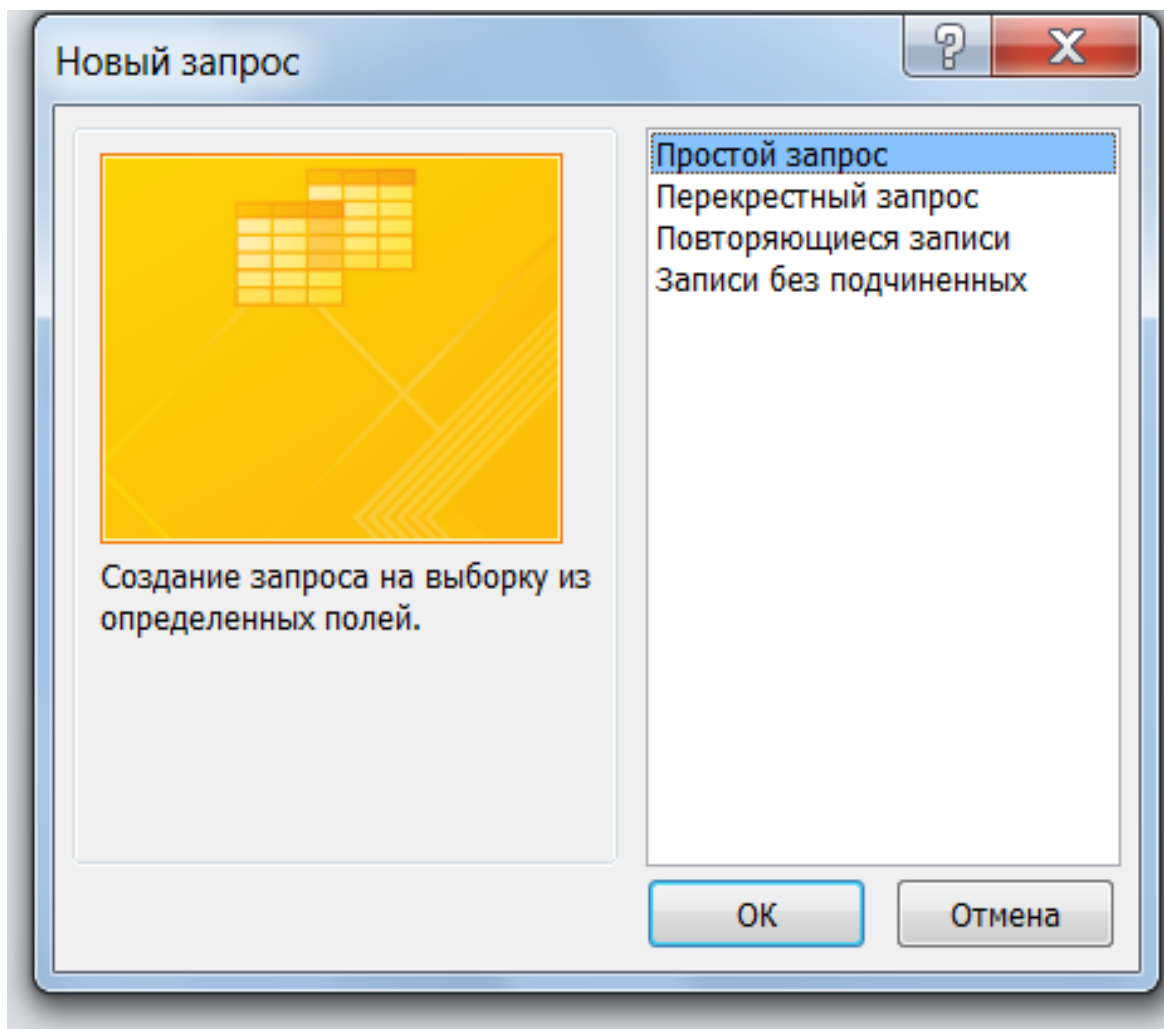


Рис. 4.1 . Диалоговое окно **Новый запрос**

Записи без подчиненных. Данный запрос используется для сравнения двух таблиц и поиска всех записей в первой таблице, для которых не имеется подчинённых записей во второй таблице.

При выборе простого запроса появится окно **Создание простых запросов** (рис. 4.2). В окне следует выбрать необходимую таблицу (таблицу-источник) из списка **Таблицы и запросы** и выбрать поля данных из списка **Доступные поля**. Если запрос формируется на основе нескольких таблиц, необходимо повторить действия для каждой таблицы-источника.

В том случае, если среди выбранных полей есть поля числового типа, в следующем окне можно выбрать подробный или итоговый отчет и щелкнуть на кнопке **Далее**. В появившемся окне необходимо задать имя запроса и выбрать один из вариантов дальнейшего действия: **Открыть запрос для просмотра данных** или **Изменить макет запроса**, и нажать кнопку **Готово**. Если выбрать

Открыть запрос для просмотра данных, на экране появится окно с результатами выполнения запроса (запрос в режиме таблицы).

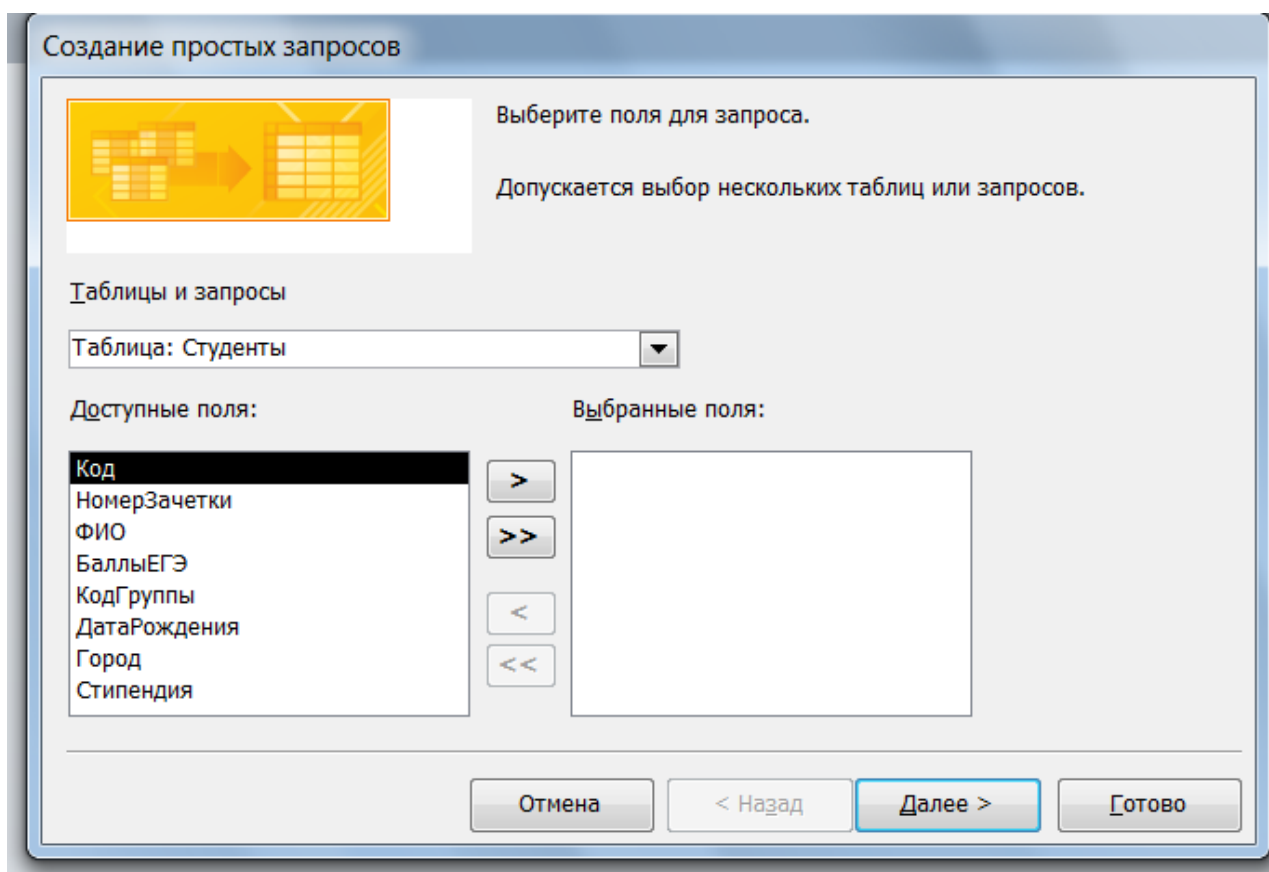


Рис. 4.2. Окно создания простых запросов

4.3. Создание нового запроса с помощью Конструктора запросов

При выборе **Конструктора запросов** можно увидеть окно конструктора и всплывающее диалоговое окно **Добавление таблицы**. Окно конструктора разделено на две части. В верхней части отображаются таблицы с их именами и списками полей.

Нижняя часть является бланком запроса, или, как его называют, QBE – областью (запрос по образцу). Здесь указываются параметры запроса и данные, которые нужно отобразить, а также определяется способ их отображения на экране.

Например, на рис. 4.3 представлено окно запроса с выбранной таблицей **Студенты**. Результатом запроса является вывод номера зачетки и ФИО тех студентов, балл ЕГЭ которых больше 100.

Для перемещения из верхней панели окна в нижнюю и обратно можно использовать клавишу F6.

4.4. Включение полей в запрос

Добавить нужные поля в бланк запроса можно путем перетаскивания их имен из списка, находящегося в верхней части окна конструктора в строку бланка **Поле**.

Еще один способ – двойной щелчок по имени поля. Например, на рис. 4.3 в бланк запроса включены поля **НомерЗачетки**, **ФИО** и **БаллыЕГЭ** из таблицы **Студенты**.

В строке **Вывод на экран** флажком помечаются те поля, которые должны быть выведены на экран.

Поля, включаемые в запрос, наследуют свойства, заданные для соответствующих полей таблицы. Можно изменить эти свойства в **Окне свойств**, нажав кнопку **Страница свойств** группы **Показать или скрыть**. Для изменения доступны следующие свойства: **Описание** (текст, содержащий описание объекта), **Формат поля** (представление данных на экране), **Число десятичных знаков** (для числовых данных), **Маска ввода**, **Подпись** (заголовок столбца).

Запрос1

Студенты

- *
 - Код
 - НомерЗачетки
 - ФИО
 - БаллыЕГЭ
 - КодГруппы
 - ДатаРождения

Поле:	НомерЗачетки	ФИО	БаллыЕГЭ
Имя таблицы:	Студенты	Студенты	Студенты
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			> 100
или:			

Рис 4.3 . Окно конструктора запроса. Пример ввода условия.

Удалить поле из бланка запроса можно клавишей **Delete** или командой **Удалить столбцы**. Чтобы удалить таблицу, следует маркировать ее в верхней части окна конструктора запроса, выполнив щелчок по имени, и нажать **Delete**.

4.5. Установка критериев отбора записей. Запрос с параметром

При создании запроса можно задать критерии отбора, вследствие чего по запросу будет осуществлен отбор только нужных записей.

Чтобы отобразить записи с конкретным значением поля, нужно ввести это значение в данное поле в строке бланка QBE **Условие отбора**.

Для создания запроса с несколькими критериями используют различные логические операции. Можно задать несколько условий отбора, соединенных логическим оператором ИЛИ (or) или И (and), для некоторого поля одним из двух способов (на примере оператора ИЛИ (or)):

1) Ввести все условия в одну ячейку строки **Условие отбора**, соединив их логическим оператором ИЛИ (or). В этом случае будут выбраны данные, удовлетворяющие хотя бы одному из условий, как, например, показано на рис. 4.4.

2) Ввести второе условие в отдельную строку. В этом случае, чтобы запись была выбрана, достаточно выполнения условий хотя бы в одной из строк. При такой записи условия также будут выбраны студенты с баллом ЕГЭ 120 или 123.

Логическая операция И (and) используется в том случае, когда для выбора записи должны быть выполнены оба условия. Например, по условию **>200 and <250** будут выбраны студенты с баллом ЕГЭ в соответствующем диапазоне.

Чтобы объединить операцией И несколько условий отбора, относящихся к разным полям, следует поместить их в одной строке. Например, на рис. 4.5. показано, как можно задать условие для выбора студентов, проживающих в городе **Москва** и с баллом ЕГЭ **120**.

Операторы И, ИЛИ применяются как отдельно, так и в комбинации. Следует помнить, что условия, связанные оператором И, выполняются раньше условий, объединенных оператором ИЛИ.

Оператор Between позволяет задать диапазон значений. Например:

Between 10 and 20

Оператор In позволяет задавать используемый для сравнения список значений. Например:

In ("первый", "второй", "третий")

Студенты

*

Код

НомерЗачетки

ФИО

БаллыЕГЭ

КодГруппы

ДатаРождения

Город

Стипендия

Поле:	НомерЗачетки	ФИО	БаллыЕГЭ
Имя таблицы:	Студенты	Студенты	Студенты
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			120 Or 123
или:			

Рис 4.4. Пример записи условия с использованием оператора *или (or)*

Оператор Like полезен для поиска образцов в текстовых полях, причем можно использовать следующие шаблоны:

- * — обозначает любое количество символов;
- ? — любой одиночный символ;
- # — указывает, что в данной позиции должна быть цифра.

Например, для выбора фамилий, начинающихся с буквы П и с окончанием “ов” можно записать:

Like П*ов

Можно включить в условие отбора дату и время, при этом их значения должны быть заключены между символами #. Например:

#10 мая 1998#
>#31.12.96#

В Access существует ряд функций, которые помогут задать условия отбора для даты и времени. Например, функция DateDiff используется для определения числа интервалов времени между двумя датами. DateDiff можно использовать для вычисления числа дней между двумя датами или числа недель между сегодняшним днем и концом года. Первый параметр функции -

строковое выражение, определяющее интервал времени в годах, в месяцах и т.д.

Студенты				
* Код НомерЗачетки ФИО БаллыЕГЭ КодГруппы ДатаРождения Город Стипендия				
Поле:	НомерЗачетки	ФИО	БаллыЕГЭ	Город
Имя таблицы:	Студенты	Студенты	Студенты	Студенты
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			120	"Москва"
или:				

Рис 4.5. Пример реализации логической операции И

Например, для вычисления числа полных лет (без учета дня рождения) можно использовать функцию DateDiff с параметрами:

DateDiff("yyyy";[Студенты]![ДатаРождения];Date())

В данном случае значение параметра "yyyy" определяет измерение интервала времени в годах. Соответствующий запрос приведен на рис. 4.6.

Функция Date\$() возвращает текущую дату в соответствии с системной датой и системным временем на компьютере пользователя. Перечень всех функций даты и времени можно найти в построителе выражений.

Можно создавать запросы с параметрами. При этом ввод условия отбора осуществляется в процессе выполнения запроса. Для этого в строку **Условие отбора** вводится не конкретный критерий выбора, а сообщение пользователю о необходимости ввода критерия. Сообщение должно быть заключено в квадратные скобки. Само сообщение в квадратных скобках может отсутствовать, квадратные скобки обязательны.

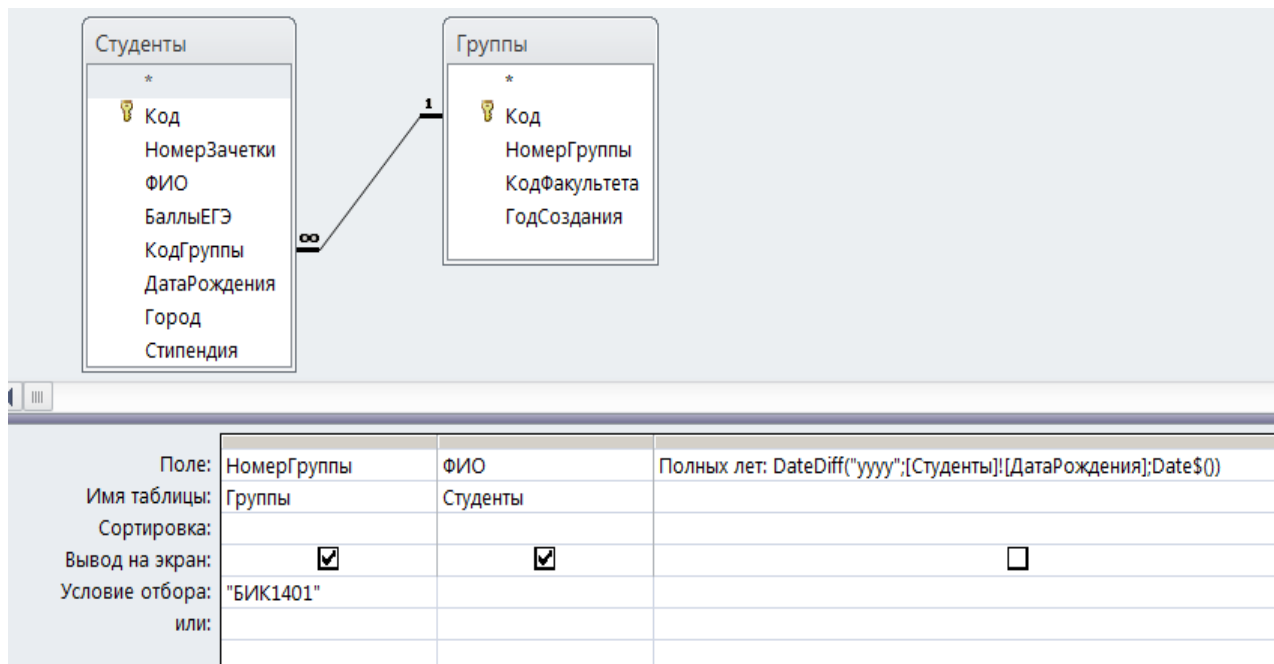


Рис.4.6.Запрос, вычисляющий число полных лет каждого студента группы.

При выполнении параметрического запроса в дополнительном окне с именем **Введите значение параметра** необходимо ввести конкретное значение параметра. Пример такого запроса приведен на рис. 4.7. При каждом выполнении такого запроса можно выводить список студентов конкретной группы.

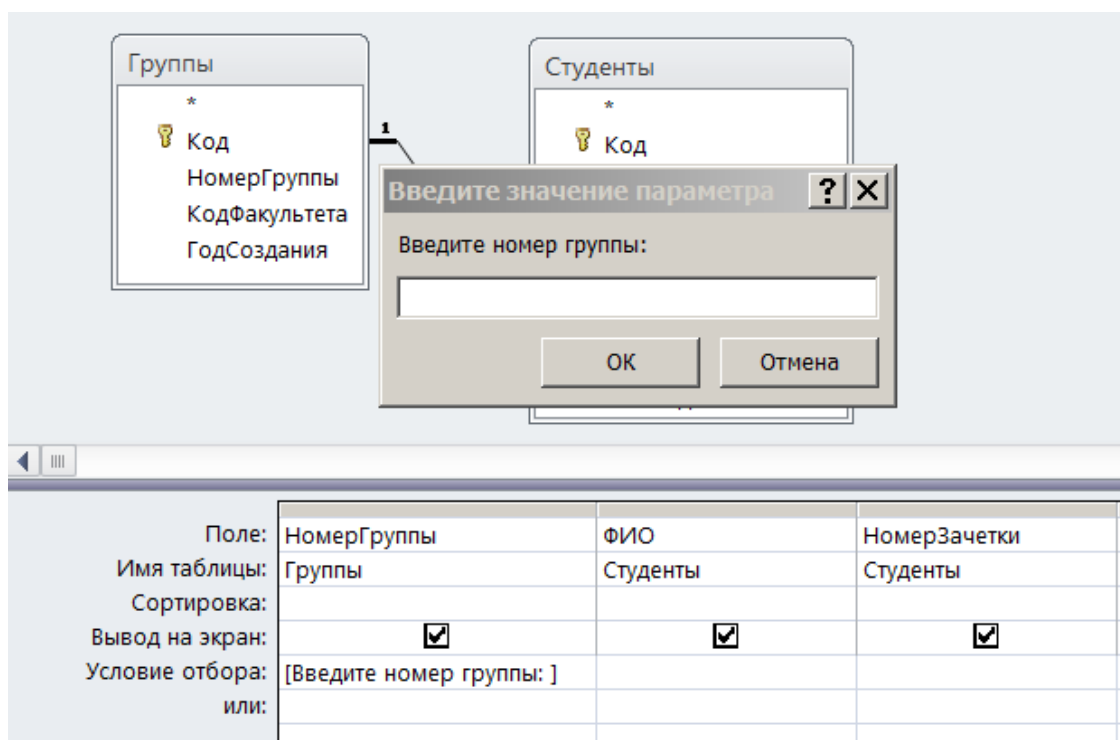


Рис.4.7.Пример параметрического запроса

4.6. Вычисляемые поля

При конструировании запроса можно задать вычисления над любыми полями таблицы и сделать вычисленное значение новым полем в запросе. Для этого в строке **Поле** бланка QBE вводится формула для вычисления, причем имена полей заключаются в квадратные скобки. Например:

$$= [\text{Оклад}] * 0.15$$

В выражениях можно использовать следующие операторы:

- арифметические: * умножение; + сложение; - вычитание; / деление; ^ возведение в степень;

- соединение частей текста при помощи знака &, например:

$$= [\text{Фамилия}] \& \text{""} \& [\text{Имя}]$$

В кавычки заключен пробел для того, чтобы запись не была слитной.

4.7. Использование построителя выражений

При создании выражений для вычисляемых полей можно использовать **Построитель выражений**. Для этого нужно щелкнуть по пустому полю в бланке QBE, а затем в группе **Настройка запроса** нажать кнопку **Построитель**, при этом откроется окно **Построитель выражений**.

В верхней части окна расположена пустая область ввода, в которой создается выражение. Можно самим ввести выражение, но проще использовать различные элементы выражений, выбирая их из списка, расположенного под областью ввода.

Все имена объектов, из которых строится выражение для вычисления, заключаются в квадратные скобки, причем перед именем поля вставляется восклицательный знак (!), разделяющий имя поля и имя таблицы.

4.8. Итоговые запросы (запросы с групповыми функциями)

Групповые операции в запросах Access позволяют выделить группы записей с одинаковыми значениями в указанных полях и вычислить итоговые данные для каждой из групп по другим полям, используя одну из групповых функций. Например, если необходимо получить по каждой группе средний балл ЕГЭ и количество студентов в группе, то достаточно проводить группировку по полю **НомерГруппы** и использовать соответствующие групповые функции по полям **БаллыЕГЭ** и **Код** таблицы **Студенты**.

Группировка необходима, если надо получить уникальные (не повторяющиеся) записи.

Итоговые запросы существенно отличаются от обычных запросов на выборку. В них поля делятся на 2 типа:

- поля, по которым осуществляется группировка данных;
- поля, для которых проводятся вычисления.

Кроме этих полей в запрос могут включаться поля, по которым задаются условия отбора.

Для создания итогового запроса после включения полей таблиц в запрос следует нажать кнопку **Итоги** на панели инструментов. В результате этого в бланке запроса появится строка **Групповая операция**.

Если для соответствующего поля из списка выбрать функцию **Группировка** (рис. 4.8), то при выполнении запроса записи группируются по значениям в этом поле, но итог не подводится. Если же для некоторого поля выбрать одну из групповых операций, то при выполнении запроса по этому полю будет подведен соответствующий итог.

Групповые функции применимы, прежде всего, к полям с типом данных **Числовой, Денежный, Дата/время**.

Access предоставляет 9 функций, обеспечивающих выполнение групповых операций. Можно задать нужную функцию, набрав на клавиатуре ее имя в строке **Групповая операция** или выбрав ее из раскрывающегося списка (рис. 4.8).

Поле:	Имя таблицы:	Групповая операция:	Сортировка:	Вывод на экран:	Условие отбора:	или:
Город	Студенты	Count		<input checked="" type="checkbox"/>		
Код	Студенты			<input checked="" type="checkbox"/>		

Рис 4.8. Выбор функции из списка в бланке QBE

Имеются следующие групповые функции:

Sum - вычисляет сумму всех значений заданного поля (для числовых или денежных полей), в записях, отобранных запросом;

Avg - вычисляет среднее значение заданного поля для отобранных запросом записей (для числовых или денежных полей);

Min - выбирает минимальное значение заданного поля в записях, отобранных запросом;

Max - выбирает максимальное значение заданного поля в записях, отобранных запросом;

Count - вычисляет количество записей, отобранных запросом, в которых значения данного поля отличны от Null (от пустого значения);

First - определяет значение указанного поля в первой записи;

Last - определяет значение указанного поля в последней записи;

StDev - вычисляет стандартное отклонение значений, которые содержатся в указанном поле;

Var - вычисляет дисперсию значений, которые содержатся в указанном поле.

Например, на рис. 4.9 приведен запрос с группировкой по полю **Номер зачетки** с использованием групповой операции **Avg**, в результате чего выводится средняя оценка по экзаменам для каждого студента. Результат запроса приведен на рис. 4.10. Обратите внимание на включения заголовка **Средняя оценка по экзаменам** в поле **Оценка**.

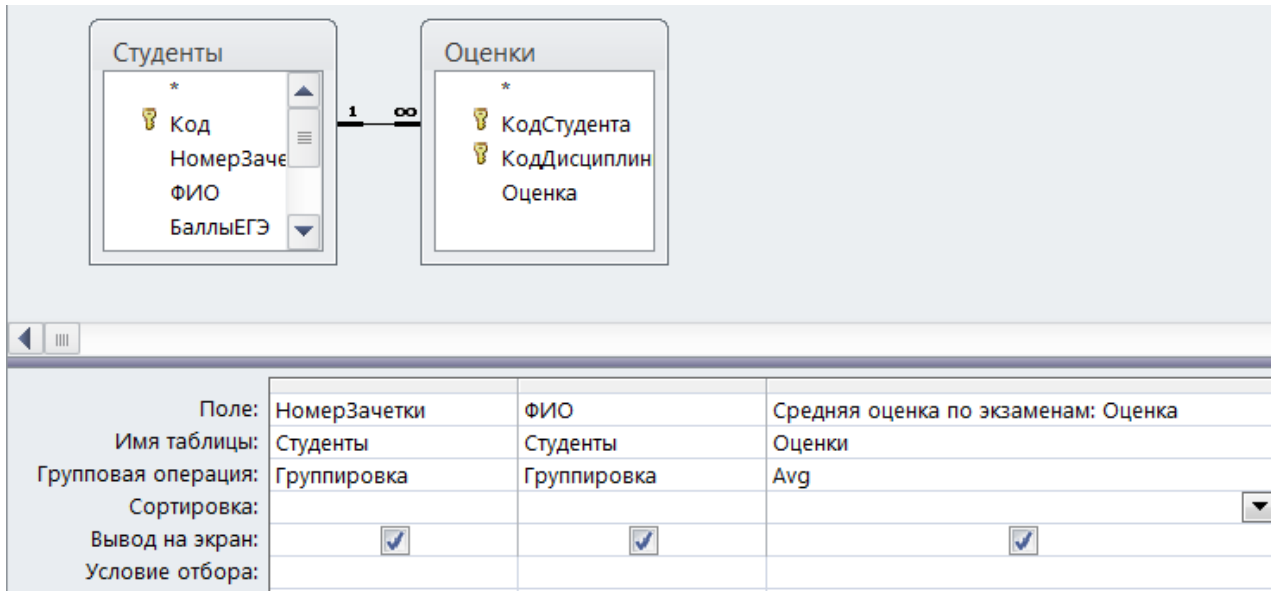


Рис. 4.9. Запрос **Средняя оценка по экзаменам**

Номер зачетки ▾	ФИО ▾	Средняя оценка по экзаменам ▾
1234567890	Петров П.П.	4
3456345345	Иванов И.И.	4,5
9876543210	Сидоров С.С.	4

Рис. 4.10. Результат выполнения запроса **Средняя оценка по экзаменам**

После группировки по какому-то признаку, и использования какой-либо групповой функции (например, среднего - **Avg**), можно выбрать максимальное значение результата (например, среднего) следующим образом. Сначала упорядочить записи по убыванию (возрастанию), установив соответствующую сортировку в строке **Сортировка**. Затем, путем выбора вкладки **Работа с запросами/Конструктор** в списке **Возврат**, указать количество выводимых (с начала) записей равное единице.

4.9. Перекрестные запросы

С помощью перекрестного запроса можно более наглядно представить данные итоговых запросов, предусматривающих группировку по нескольким признакам (по двум, в частности). В этом случае значения полей по первому признаку группировки могут стать заголовками строк, а по второму - заголовками столбцов.

Для создания перекрестного запроса удобно использовать **Мастер запросов** и выбрать **Перекрестный запрос**. Мастеру необходимо указать исходную таблицу, которая может быть таблицей из базы данных или являться результатом выборки данных из нескольких таблиц. На основе исходной таблицы или запроса **Мастер запросов** создает итоговую выборку, в которой группирует и сортирует данные по полям, используемым в качестве строки и столбца перекрестной таблицы, а также вычисляет итоговое значение по заданному полю данных.

В перекрестном запросе отображаются результаты статистических расчетов (такие как суммы, количество записей и средние значения), выполненных по данным из одного поля таблицы. Эти результаты группируются по двум наборам данных, один из которых расположен в первом столбце таблицы, а второй - в верхней строке.

После выбора **Мастера запросов**, на последующих шагах следует выбрать таблицу, поля в качестве строк и в качестве столбцов, вычисляемое поле, функцию для подсчета итогов.

Например, для создания перекрестного запроса с помощью **Мастера запросов** на базе таблицы **Студенты**, можно выбрать в качестве строк поле **Город**, в качестве столбцов поле **КодГруппы**, а вычисляемое поле – **НомерЗачетки**, и использовать для подсчета функцию **Число**. В результате получается перекрестный запрос, показывающий распределение студентов по городам.

Мастер запросов может выбрать поля только из одной таблицы. Если нужны поля из разных таблиц при использовании **Мастера запросов**, то необходимо создать отдельный запрос с разными полями, на основе которого и будет построена перекрестная таблица.

Перекрестный запрос можно построить и без использования **Мастера запросов** в **Конструкторе запросов**. В этом случае выбираются две таблицы, выполняется **Работа с запросами/Перекрестный**, далее в качестве заголовков строк и заголовков столбцов выбираются соответствующие поля. На рис.4.11.

приведен запрос в режиме конструктора, результатом которого является распределение количества студентов по городам и группам.

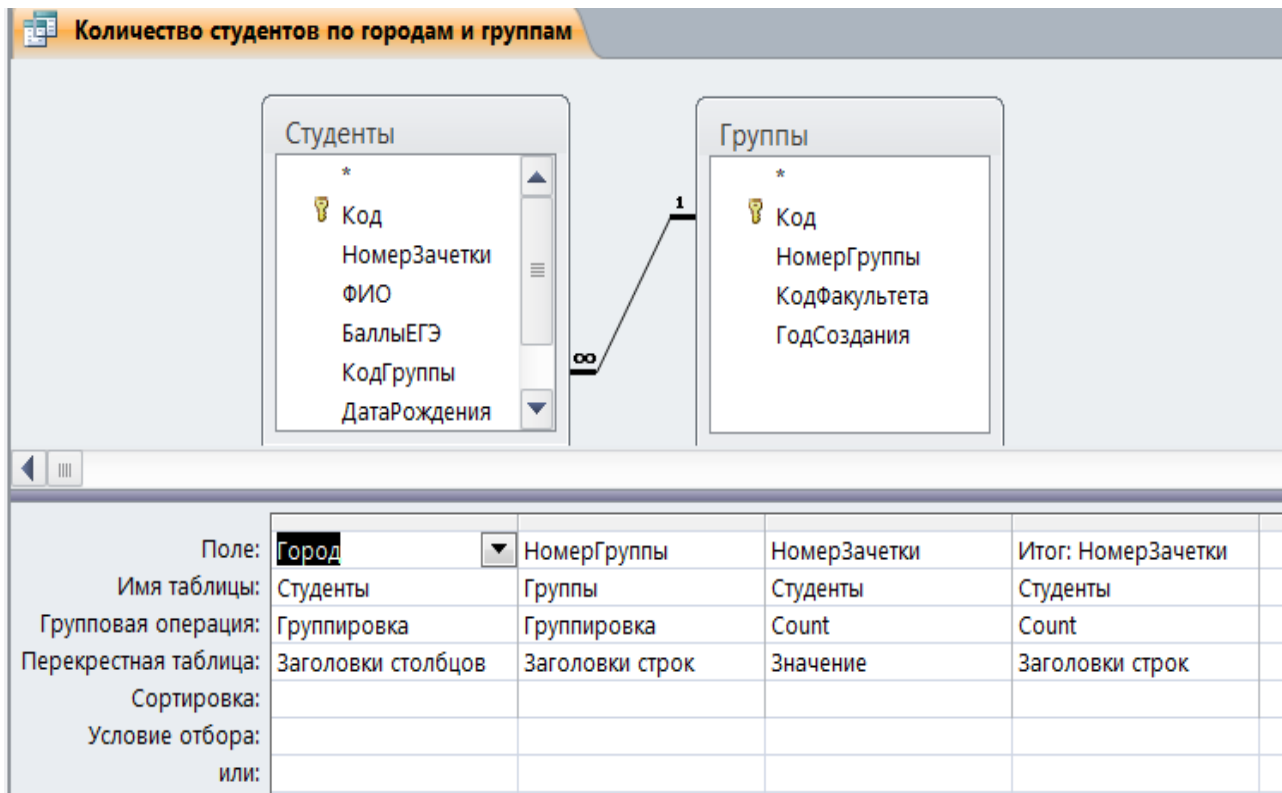



Рис. 4.11. Запрос «Количество студентов по городам и группам».

Если запрос находится в режиме Конструктора, то его можно выполнить по кнопке  панели инструментов. Результат выполнения запроса приведен на рис. 4.12.

Номер	Москва	Серпухов
БИК1502	2	
БИН1502		1
БИН1503	1	
БУТ1401	3	

Рис. 4.12. Результат выполнения запроса «Количество студентов по городам и группам».

5. СОЗДАНИЕ И ВЫПОЛНЕНИЕ ЗАПРОСОВ-ДЕЙСТВИЙ

5.1. Понятие запроса-действия

Для быстрого изменения, добавления, создания или удаления наборов данных из базы данных используются управляющие запросы или запросы-действия.

Создание запросов-действий аналогично созданию запросов-выборок. Отличие лишь в том, что в режиме конструктора запросов необходимо указать тип действий над выбранными записями. Существует всего четыре типа запросов-действий: создание таблицы, обновление, удаление и добавление записей. Access отмечает запросы-действия специальными значками.

Создание таблицы. Этот запрос позволяет сохранить набор данных в виде таблицы.

Обновление. Используется для того, чтобы обновить значения некоторых полей для всех записей.

Удаление. Используется для удаления записей.

Добавление. Используется для добавления данных в имеющуюся таблицу.

В процессе конструирования запроса, чтобы обезопасить себя от каких-либо случайностей, можно перед выполнением запроса-действия создать резервную копию используемой таблицы. Создание запросов на обновление, добавление и удаление может привести к нарушению целостности данных, поэтому упражняться лучше на созданной копии таблицы. Новый запрос-действие желательно создать вначале как запрос на выборку и, выполнив его, проверить правильность отбора записей.

5.2. Запросы на создание таблицы

Когда активно используется база данных, количество записей в ней растет, что ведет к замедлению работы с ней, так как тратится много времени на переписывание данных с диска в оперативную память и обратно. Целесообразно периодически удалять не актуальные данные из базы, но делать это надо осторожно, так как удаленные из таблиц записи восстановить невозможно. Выходом может служить перенос части записей в архивную таблицу, чтобы при необходимости старые записи можно было восстановить.

Пусть, например, сводную информацию по годам о студентах без стипендии необходимо хранить в течение некоторого времени. Для этого

создается простой запрос, в группе **Тип запроса** выбирается **Создание таблицы**. Access выведет на экран диалоговое окно «Создание таблицы», где нужно указать имя таблицы, в которой будут сохранены результаты выборки (рис. 5.1).

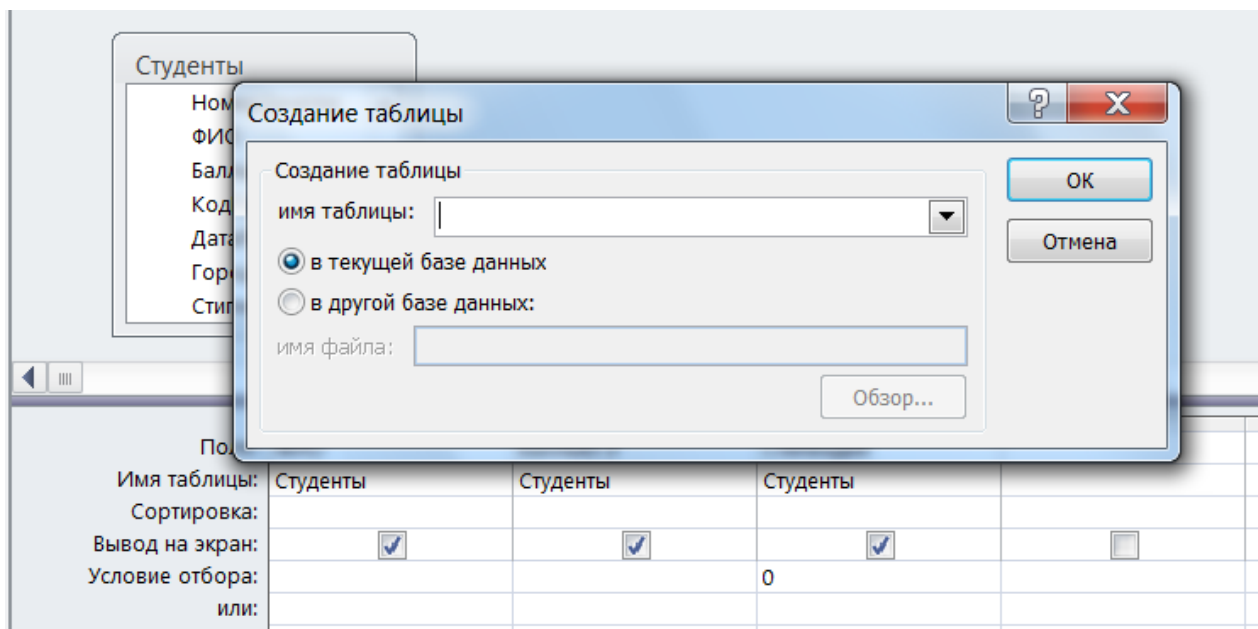


Рис. 5.1. Вид окна запроса на создание таблицы

После ввода имени таблицы и выполнения запроса Access выведет окно подтверждения операции (рис.5.2).

5.3. Запросы на обновление записей

Этот тип запросов-действий предназначен для изменения значений некоторых полей для выбранных записей. После входа в режим **Конструктора запроса** нужно выбрать команду **Обновление**. После этого в бланке запроса появится строка **Обновление**. Эта строка используется для того, чтобы задать новые значения для выбранных полей (рис.5.3.). После выполнения запроса Access выведет окно подтверждения операции (рис. 5.4).

5.4. Запросы на удаление записей

Этот вид запросов служит для удаления группы записей базы данных, удовлетворяющих определенным условиям. Обычный запрос можно

преобразовать в запрос на удаление с помощью команды **Удаление** в режиме **Работа с запросами**. При этом в бланке запроса появится строка **Удаление**, где можно ввести условие для удаления записей (рис. 5.5).

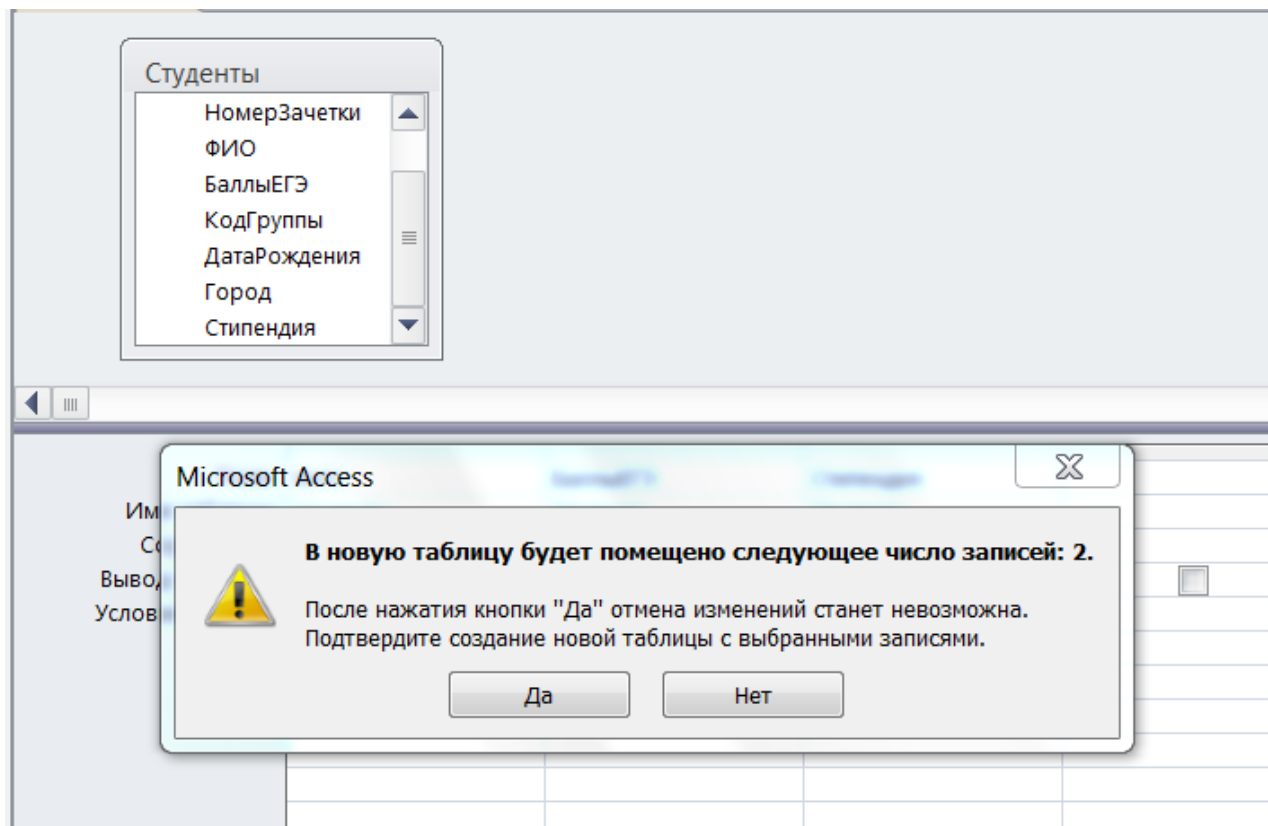


Рис. 5.2. Окно подтверждения операции создания таблицы

В режиме **Конструктора запроса** необходимо указать таблицу, в которой будут проводиться удаления. Затем можно определить критерии и просмотреть результаты выполнения запроса на выборку. Если в результате запроса присутствуют только подлежащие удалению записи, следует преобразовать его в запрос на удаление.

Удаление записей может привести к потерям данных, если, например, таблица является частью отношения “один-ко-многим” и в качестве условия обеспечения целостности данных использована опция **Каскадное удаление связанных полей**.

Например, надо удалить какой-то факультет (удалить соответствующую запись из таблицы **Факультеты**). С таблицей **Факультеты** отношением “один–ко-многим” связана таблица **Группы**. Если для обеспечения целостности данных выбрана опция **Каскадное удаление связанных полей**, в

результате удаления записи о факультете будут удалены записи обо всех группах этого факультета.

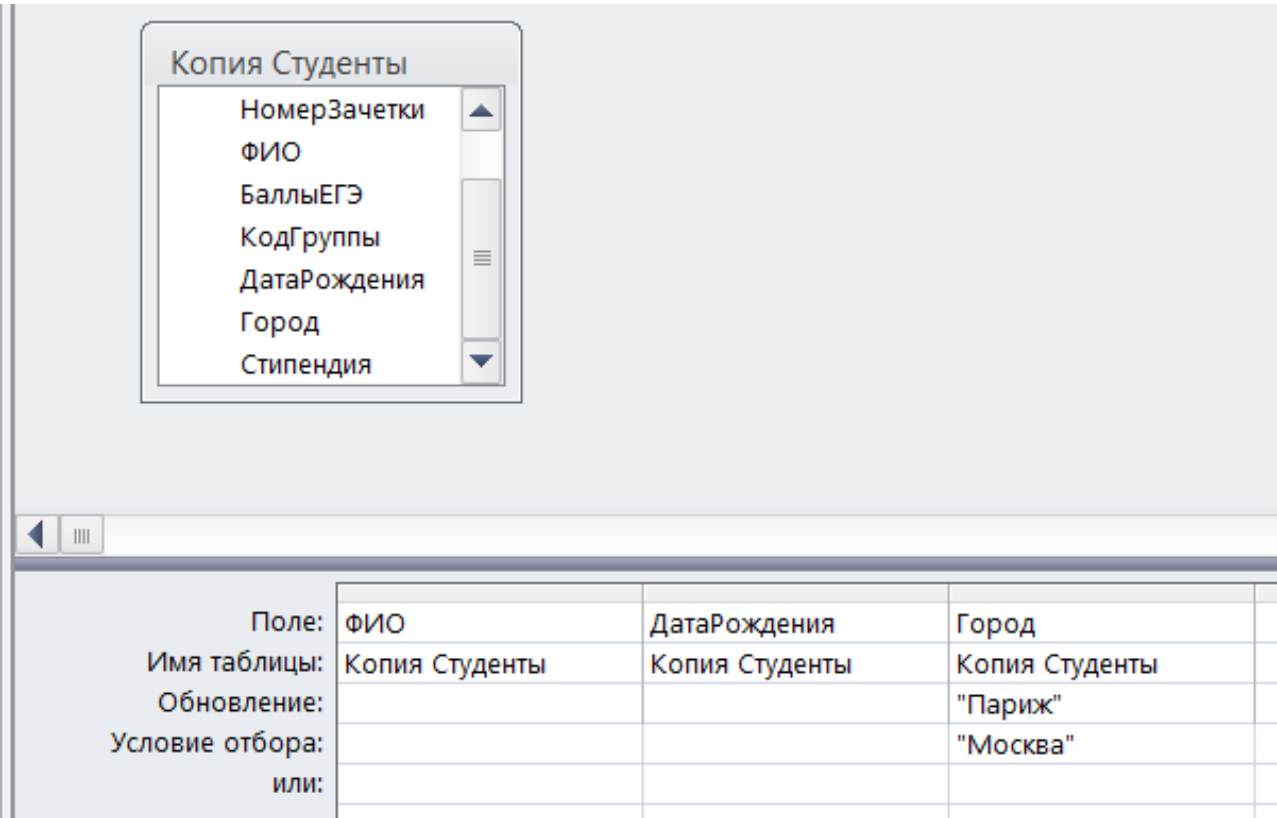


Рис. 5.3. Вид окна запроса на обновление записей

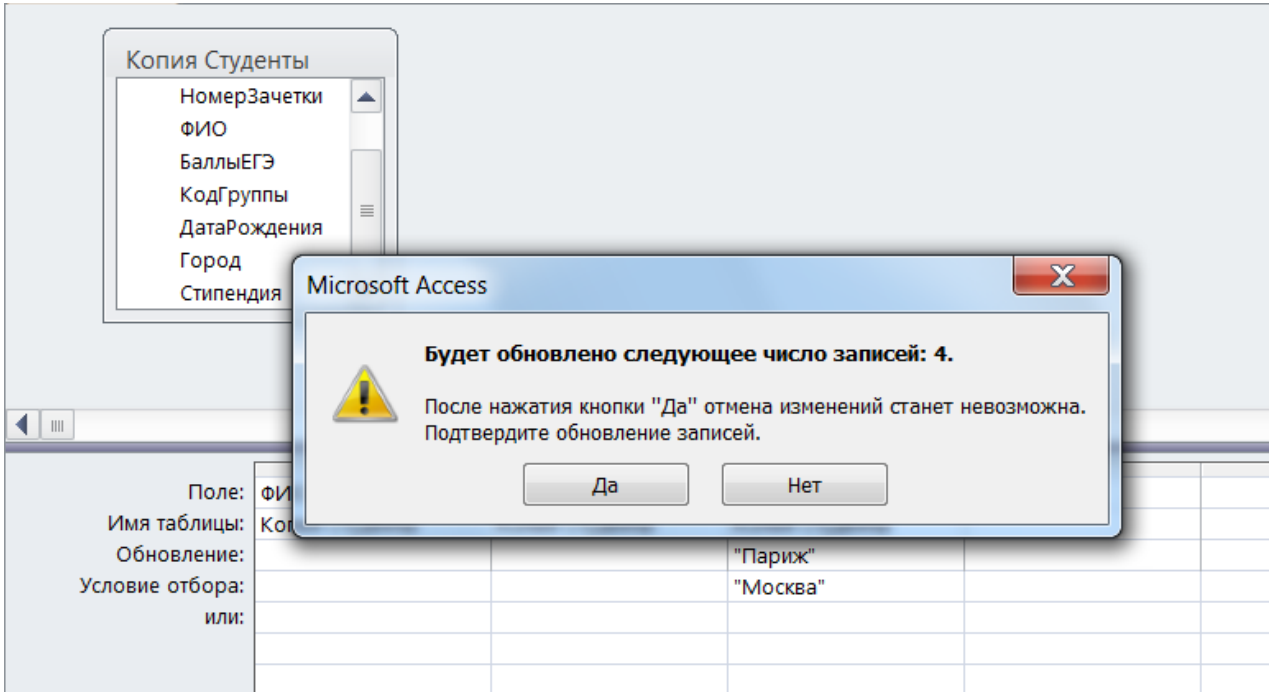


Рис. 5.4. Окно подтверждения операции обновления

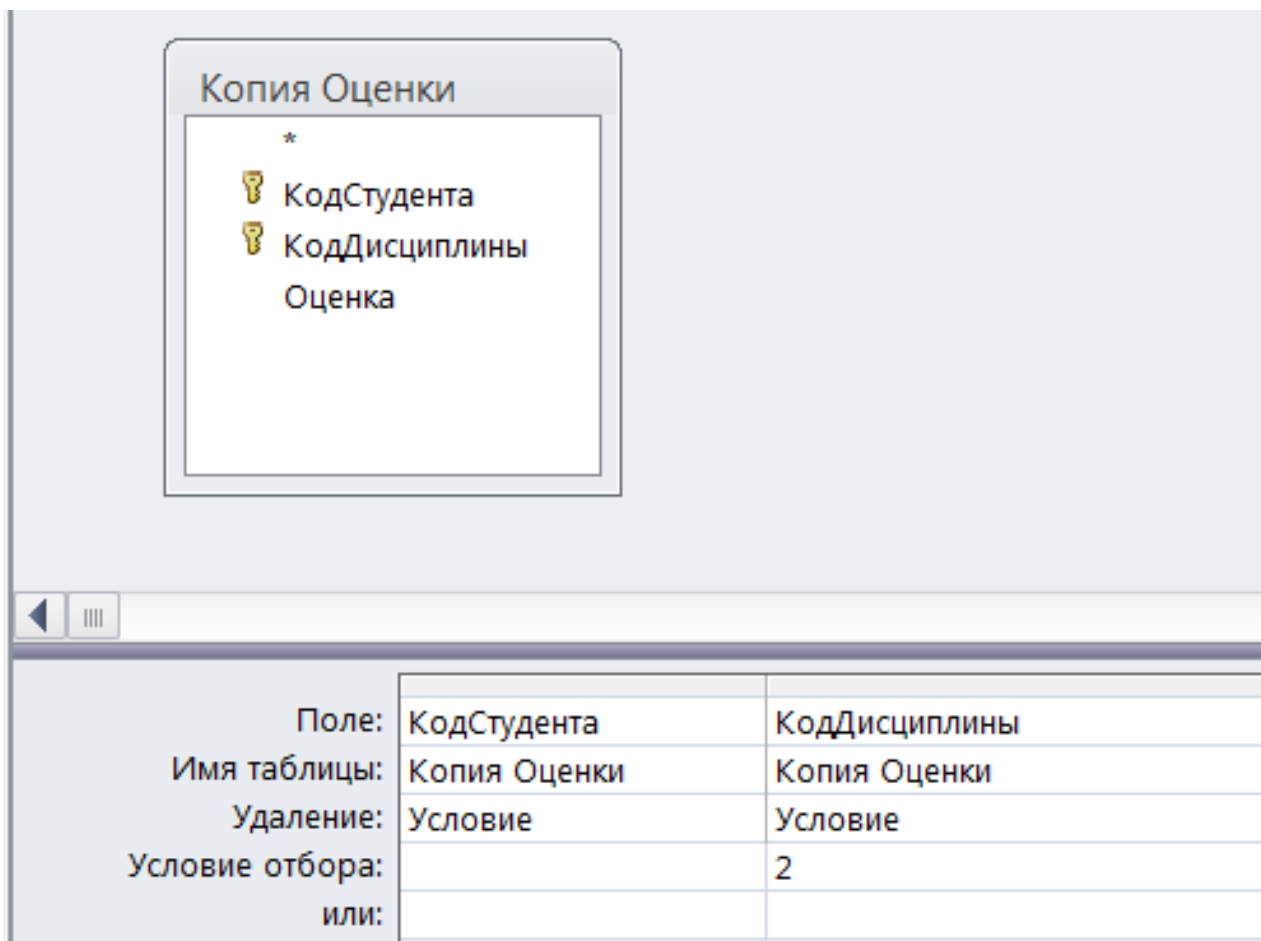


Рис. 5.5. Вид окна запроса на удаление записей

Если установлена опция **Каскадное удаление связанных полей**, и вы пытаетесь удалить записи в режиме просмотра таблицы, то получите предупреждение о возможном удалении данных в других таблицах. А при использовании запросов на удаление вы не увидите предупреждения. Access выдаст только информацию о количестве удаляемых записей, но ничего не сообщит про каскадное удаление.

5.5. Запросы на добавление записей

Запрос на добавление необходим, если нужно добавить новые записи в уже существующую таблицу, используя данные из других источников.

С помощью запроса на добавление записей можно, например, добавлять одновременно группу записей, можно скопировать выбранные записи и поместить их в другую таблицу. Можно также использовать запрос на добавление для переноса данных в используемую базу данных из другого источника (из какой-либо другой базы данных).

Запрос на добавление выполняется обычно в 2 этапа:

1) Выбор данных для добавления.

Для отбора добавляемых записей нужно создать запрос на выборку. При необходимости оценки результата можно выполнить созданный запрос и просмотреть результат, переключившись в **Режим таблицы**.

2) Преобразование запроса на выборку в запрос на добавление.

В режиме **Конструктора** нужно активизировать команду **Добавление** и в открывшемся окне диалога в поле **Имя таблицы** ввести имя таблицы, к которой требуется присоединить данные из результирующего набора записей запроса. Далее следует выполнить запрос. В специальном диалоговом окне Access укажет, сколько записей будет добавлено к целевой таблице, и потребует подтвердить выполнение этой операции.

Запрос на добавление нельзя отменить. Если сделана ошибка, придется либо восстанавливать базу данных с помощью резервной копии, либо исправлять ошибку вручную или с помощью запроса на удаление.

Пусть, например, в БД Успеваемость студентов (рис. 2.5) необходимо добавить в таблицу **Оценки** результаты экзамена по некоторой дисциплине для некоторой группы. Это может быть сделано путем последовательного добавления по одной записи в соответствующей форме. Однако более эффективный способ состоит в групповом добавлении сразу всех записей с условно нулевой оценкой с последующим редактированием оценок в форме.

На рис. 5.6. показан запрос на такое добавление после преобразования его из запроса-выборки. После ввода номера группы и названия дисциплины в таблицу **Оценки** будет добавлена группа записей по количеству студентов в заданной группе.

Запрос на добавление

Дисциплины
 *
 Код
 НазваниеДисц
 Часы

Группы
 *
 Код
 НомерГруппы
 КодФакультета
 ГодСоздания

Студенты
 *
 Код
 НомерЗачетки
 ФИО
 Пол
 БаллыЕГЭ
 КодГруппы
 ДатаРождения
 Город

1 — ∞

Поле:	Код	Код	Оценка: 0	НомерГруппы	НазваниеДисц
Имя таблицы:	Студенты	Дисциплины		Группы	Дисциплины
Сортировка:					
Добавление:	КодСтудента	КодДисциплины	Оценка		
Условие отбора:				[Введите № группы]	[Введите дисциплину]
или:					

Рис. 5.6. Запрос на добавление группы записей

6. СОЗДАНИЕ ОТЧЕТОВ НА ОСНОВЕ ТАБЛИЦ И ЗАПРОСОВ

6.1. Основные сведения об отчетах

Отчеты представляют собой наилучшее средство представления информации из базы данных в виде печатного документа. По сравнению с другими методами вывода данных на печать отчеты обладают двумя принципиальными преимуществами:

- отчеты предоставляют широкие возможности для группировки, промежуточных вычислений и получения общих итогов для больших наборов данных;
- отчеты могут быть использованы для получения красиво оформленных документов.

В основном отчеты применяются для просмотра «картины в целом», поэтому часто базируются на многотабличных запросах.

Для создания отчета в Access 2010 нужно выбрать один из инструментов группы **Отчеты** на вкладке **Создание** (рис. 6.1):

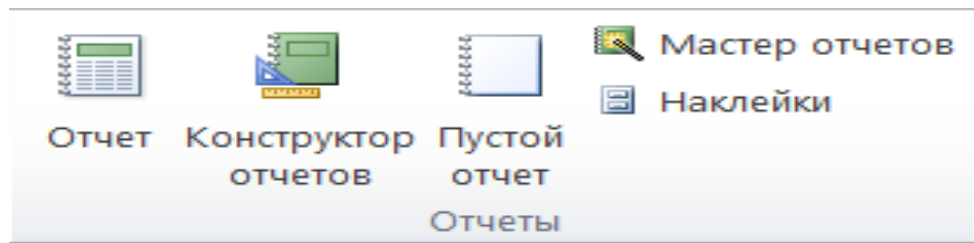


Рис 6.1 Инструменты группы **Отчеты**

Отчёт - создание простого отчета на основе данных из запроса или таблицы.

Конструктор - позволяет создавать отчет с нуля на бланке отчета.

Пустой отчет - создание пустого отчета, в который можно вставить новые поля.

Мастер отчетов - запуск мастера, помогающего настроить отчет при его создании;

Наклейки - предназначены для наклеивания на конверты при рассылке писем адресатам, информация о которых хранится в базе данных.

Отчет может находиться в четырех режимах, выбор которых производится командами из списка кнопки **Режим** (в левом верхнем углу) на вкладке **Главная** (рис.6.2), кнопками в правом нижнем углу окна отчета или командами контекстного меню.

Режим **Представление отчета** предназначен для просмотра отчета в среде Access. В этом режиме можно отфильтровать данные для отображения только заданных строк, найти нужные данные, скопировать текст отчета или его часть в буфер обмена.

В режиме **Предварительный просмотр** дается вид отчета, который получится после его вывода на принтер. В этом режиме можно увеличивать масштаб для просмотра деталей или уменьшать его для проверки размещения данных на странице, изменять параметры страницы.

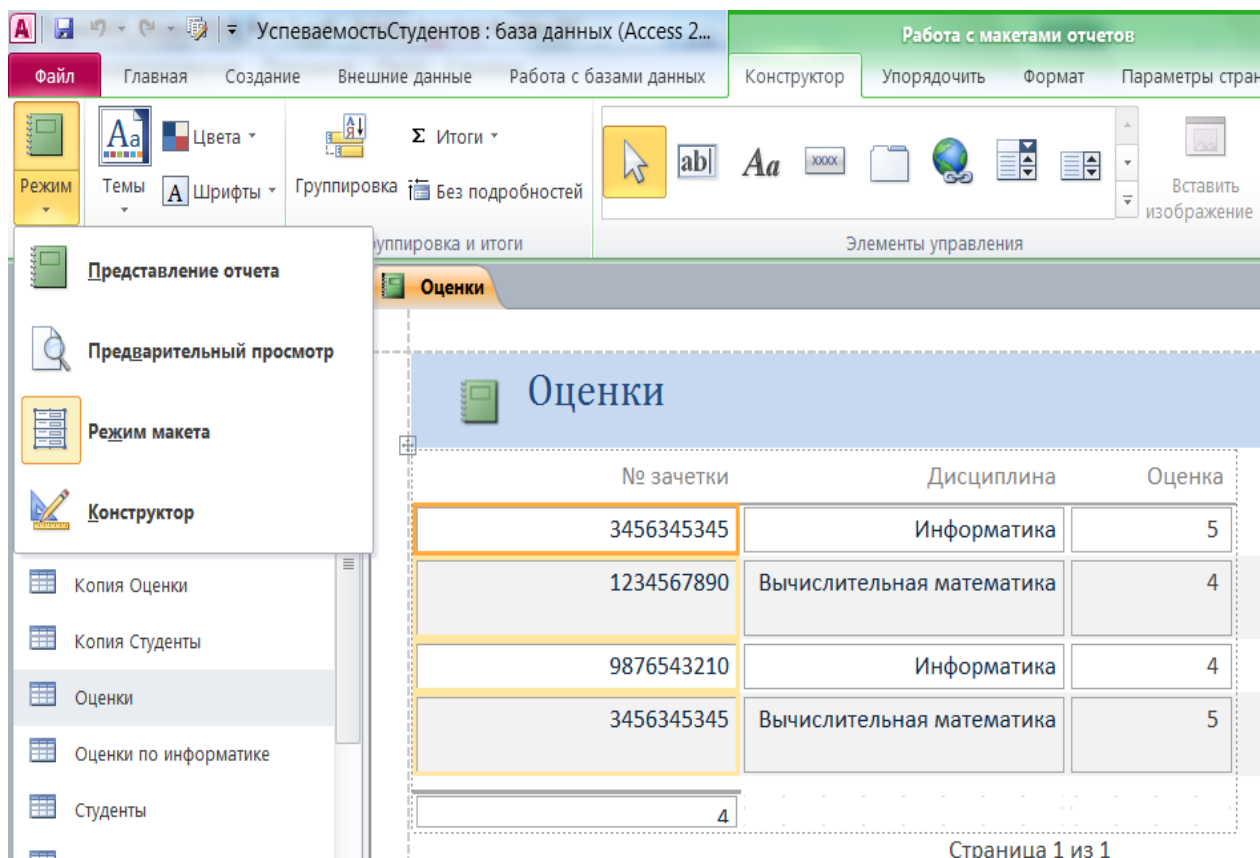


Рис. 6.2. Режимы работы с отчетом

В Access существуют два представления, в которых можно вносить изменения в отчет: режим **Макета** и режим **Конструктора**. Режим **Макета** является наиболее удобным для внесения изменений в отчет, поскольку пользователь сразу видит данные отчета. В этом режиме предусмотрено большинство инструментов, необходимых для его настройки. В нем можно изменить ширину столбцов, поменять их местами, добавить или изменить уровни группировки и итоги. Можно также разместить в макете отчета новые поля, а также задать свойства отчета и элементов управления.

В режиме **Конструктора** отображаются разделы отчета и предусмотрены дополнительные инструменты и возможности разработки. Если не удастся выполнить изменения в режиме **Макета**, то можно перейти в режим **Конструктора**. В определенных случаях в Access отображается сообщение о том, что для внесения изменений следует переключиться в режим **Конструктора**.

6.2. Разделы отчета

Ниже перечислены типы разделов и указано назначение каждого из них.

Заголовок отчета. В заголовок включается информация, обычно помещаемая на обложке, например, эмблема компании, название отчета или дата. Если в заголовке отчета помещен вычисляемый элемент управления, использующий статистическую функцию, то результат функции рассчитывается для всего отчета. Заголовок отчета печатается перед верхним колонтитулом.

Верхний колонтитул. Верхний колонтитул используется в тех случаях, когда нужно, чтобы название отчета повторялось на каждой странице.

Заголовок группы. Используется для печати названия группы. Например, если отчет сгруппирован по изделиям, в заголовках групп можно указать их названия. Если поместить в заголовок группы вычисляемый элемент управления, использующий статистическую функцию, то результат функции будет рассчитываться для текущей группы. Заголовков групп может быть несколько в зависимости от добавленных уровней группирования.

Область данных. В этом разделе размещаются элементы управления, составляющие основное содержание отчета.

Примечание группы. Примечание группы можно использовать для печати сводной информации по группе. Примечаний групп может быть несколько, в зависимости от добавленных уровней группирования.

Нижний колонтитул. Используется для нумерации страниц и для печати постраничной информации.

Примечание отчета. Примечание отчета можно использовать для печати итогов и другой сводной информации по всему отчету.

6.3. Однотабличный отчет

Наиболее простым способом создания отчета на основе таблицы или запроса является использование средства **Отчет**. С помощью этого средства отчет формируется без диалога с пользователем и выводит все поля выбранного источника. Лишние поля могут быть удалены в режиме **Конструктора**.

После выбора таблицы и выполнения запроса Access создаст отчет и отобразит его в режиме **Макета**. Например, на рис. 6.3 в отчете представлены все записи таблицы **Группы**. Размещение полей таблицы - источника записей отчета в разделах отчета - представлено в режиме **Конструктора** на рис. 6.4.

Группы			9 января 2017 г.
			13:56:26
Номер	Факультет	Год создания	
БИК1401	ОТФ-1	2014	
БИК1402	ОТФ-1	2014	
БИК1501	ОТФ-1	2015	
БИК1502	ОТФ-1	2015	
БИК1503	ОТФ-1	2015	
БИН1501	ОТФ-2	2014	
БИН1502	ОТФ-2	2014	
БИН1503	ОТФ-2	2015	
БИН1504	ОТФ-2	2015	
БУТ1401	ОТФ-1	2014	
10			

Страница 1 из 1

Рис. 6.3. Отчет в режиме **Макета**

Заголовок отчета											
	Группы									=Date()	
										=Time()	
Верхний колонтитул											
Номер		Факультет			Год создания						
Область данных											
НомерГруппы	КодФакультета				ГодСоздания						
Нижний колонтитул											
										= "Страница " & [Page] & " из " & [Pages]	
Примечание отчета											
=Count(*)											

Рис. 6.4. Отчет в режиме **Конструктора**

В области данных отчета размещены выбранные поля таблицы в табличном виде. Размещение заголовков столбцов в верхнем колонтитуле обеспечивает вывод их на каждой странице отчета. Для добавления номера

страницы в нижнем колонтитуле в вычисляемом элементе сформировано выражение ="Страница " & [Page] & " из " & [Pages].

В заголовке отчета размещены: эмблема, название отчета, текущая дата и время. Для добавления в отчет текущей даты использована встроенная функция Date() из категории **Дата/время**. В свойствах этого элемента на вкладке **Макет/Format** в строке **Формат поля** установлен формат отображения даты — **Длинный формат даты**. Для добавления в отчет текущего времени использована встроенная функция Time().

В режиме **Макета** легко привести созданный отчет в соответствие с заданными требованиями. Ориентируясь на фактические данные отчета, можно отрегулировать ширину столбцов, высоту строк, изменить их порядок, добавить уровни группировки и итоговые значения. Можно добавить в отчет новые поля, а также изменить свойства отчета и входящих в него элементов управления.

6.4. Использование Мастера отчетов. Многотабличный отчет.

Мастер отчетов позволяет в режиме диалога с пользователем создать многотабличный отчет путем выбора необходимых таблиц (запросов) и полей, определения полей группировки, итоговых значений для записей. Создание отчета мастером является простой процедурой, а полученный отчет без больших усилий может быть приведен к желаемому виду.

После запуска **Мастера отчета** на экране откроется окно диалога, в котором необходимо определить поля будущего отчета (рис. 6.5).

Необходимо нажать кнопку раскрытия списка **Таблицы и запросы** и из списка таблиц базы данных выбрать таблицу (таблицы), для которой создается отчет, или запрос, на основании которого строится отчет. При этом в списке **Доступные поля** появляется перечень всех полей выбранной таблицы или запроса. Необходимо из данного перечня перенести в список **Выбранные поля** те поля, которые надо поместить в создаваемый отчет. Завершив выбор полей, необходимо нажать кнопку **Далее** для перехода к следующему шагу. На втором шаге создания отчета с помощью **Мастера** необходимо определить, требуется ли сгруппировать данные по какому-то из полей (рис. 6.6).

Если поля не группировать, отчет произведет итоговые вычисления по всем полям с числовым типом данных для всей таблицы или запроса, на которых он основан. Можно для группировки выбрать одно поле. В этом случае отчет обеспечит для группы промежуточные вычисления, а для таблицы целиком — итоговую сумму. Можно применять до четырех группировок, вложенных одна в другую.

Поля, по которым будет осуществляться группировка, помещаются в верхней части правого списка в отдельной рамке и выделяются на экране синим цветом. Access предлагает свой вариант группировки данных. Можно согласиться с предложенным вариантом или задать свой вариант.

Установив группировку данных, можно изменить интервал группировки, для этого необходимо нажать кнопку **Группировка**. Появляется диалоговое окно **Интервалы группировки**. Данное окно позволяет проводить группировку по диапазону значений в записи, что предпочтительнее, чем группировка по отдельным записям. Если, например, используется поле с датой как основа для группировки, данные можно сгруппировать в отдельные группы для каждого года или провести группирование по месяцам этих данных в поле. Виды диапазонов, которые можно задавать, зависят от типа данных. Диалоговое окно **Интервалы группировки** включает в себя поля, на основе которых проводится группировка. Справа от каждого поля в окне имеется раскрывающийся список, который можно использовать для выбора соответствующего интервала.

Для перехода к следующему окну диалога надо нажать кнопку **Далее**. В этом окне диалога задается порядок сортировки записей внутри каждой группы (до четырех полей) и вычисления, выполняемые для записей, на задание которых можно перейти по кнопке **Итоги** (рис.6.7).

Для числовых полей можно вывести в отчет среднее, сумму, минимальное или максимальное значения. Для возврата в окно сортировки необходимо нажать кнопку ОК. На следующих двух шагах создания отчета с помощью мастера необходимо определить вид макета отчета и стиль оформления. На заключительном шаге создания отчета можно задать имя отчета и выбрать один из двух вариантов дальнейшей работы с отчетом: просмотр отчета или изменение структуры отчета (рис. 6.8).

Итоги

Какие итоговые значения необходимо вычислить?

Поле	Sum	Avg	Min	Max
ГодСоздания	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Показать

☒ данные и итоги

☐ только итоги

☐ Вычислить проценты

OK

Отмена

Рис.6.7. Окно **Итоги** - выбор итоговых значений

Факультеты

Наименование	Декан	НомерГруппы	Факультет	Год создания
ОТФ-1	Соколов С.С.	БУТ1401	ОТФ-1	2014
		БИК1402	ОТФ-1	2014
		БИК1401	ОТФ-1	2014
		БИК1503	ОТФ-1	2015
		БИК1502	ОТФ-1	2015
		БИК1501	ОТФ-1	2015
ОТФ-2	Шакин В.Н.	БИН1504	ОТФ-2	2015
		БИН1503	ОТФ-2	2015
		БИН1502	ОТФ-2	2014
		БИН1501	ОТФ-2	2014

9 января 2017 г.

Стр. 1 из 1

Рис.6.8 Многотабличный отчет в режиме макета

Многотабличные отчеты, так же как формы, могут состоять из главного отчета и включаемого в него подчиненного отчета. Для каждого из этих отчетов в качестве источника данных выбирается своя таблица или запрос, построенный на нескольких таблицах. Чтобы обеспечить соответствие записей, выводятся в подчиненном отчете, записям в главном отчете, устанавливается связь подчиненного отчета с главным.

6.5. Использование вычисляемых значений

Обширные возможности отчетов Access во многом определяются их способностью выполнять простые и сложные вычисления с данными базовых таблиц или запросов. Access предоставляет десятки встроенных функций, предназначенных для обработки данных или для добавления информации в отчет.

В качестве значения свойства любого элемента управления, позволяющего выводить данные, можно задать выражения любой сложности с использованием арифметических операторов. Кроме того, в вычислениях можно использовать любые встроенные функции. При желании можно воспользоваться помощью построителя выражений. Чтобы сообщить Access, что в свойстве **Данные** вы хотите использовать выражение, ввод надо начинать со знака равенства (=).

Например, чтобы разместить в области данных поле **Надбавка**, на панели элементов выбирается кнопка **Поле** и указывается ее размещение в области данных. Вызывается окно **Свойств** и в свойстве **Данные** вводится = [Стипендия]*0,5, т.е. надбавка составляет 50% для всех записей.

6.6. Вычисление промежуточных итогов для групп и общего итога

Еще одна задача, которую обычно приходится решать при составлении отчетов, это получение итоговых величин по группам данных. Для получения промежуточных итоговых значений для групп в эти разделы можно поместить дополнительные свободные элементы управления, в которых используются выражения, построенные с применением итоговых функций: Sum, Min, Max, Avg, Count, First, Last, StDev и Var.

Выражение в разделе группы, использующее итоговую функцию, не должно ссылаться на элемент управления, размещенный в области данных. Так, например, нельзя создать вычисляемое поле в области данных и сослаться на него в выражении, использующем итоговую функцию. Но можно повторить

вычисляемое выражение в итоговом поле. Например, если вычисляемое поле в области данных с именем **Цена** содержит выражение = [Цена за единицу]*[Количество], то в разделе примечаний группы необходимо использовать выражение =Sum([Цена за единицу]*[Количество]), а не =Sum([Цена]).

Используя раздел примечаний отчета, можно получить общие итоги по всем записям для любого поля или выражения. В примечании отчета, как и в примечаниях групп, допустимы любые итоговые функции.

6.7. Вычисление процентов

В любом отчете, группирующем и суммирующем данные, можно определить долю внешнего промежуточного или общего итога, приходящуюся на конкретную сумму. Это возможно в силу того, что Access дважды просматривает данные отчета. При первом проходе Access вычисляет выражения в детальных строках, итоговые значения для групп, общие итоги для всего отчета и определяет длину отчета, а при втором вычисляет все выражения, ссылающиеся на итоги, рассчитанные во время первого прохода. Следовательно, в области данных и разделах, содержащих итоги для групп, можно создать выражение для вычисления процентов от общих сумм или промежуточных сумм, рассчитанных для группировки более высокого уровня. Создав выражение для нахождения процентов от суммы, надо установить свойство **Формат поля** в значение **Процентный**. В режиме просмотра можно убедиться в том, что в итогах по группе отразится процент от общего итога.

6.8. Вычисление итогов с накоплением

Наряду с вычислением итогов в разделах групп можно создавать итоги с накоплением в области данных. Свойство **Сумма с накоплением** любого числового поля, размещенного в области данных, позволяет вычислять итоги, которые обнуляются в начале каждой группы или накапливаются на протяжении всего отчета. Разместив в области данных поле, надо установить для свойства **Данные** – значения поля, для которого вычисляется сумма, а для свойства **Сумма с накоплением** - значение **Для группы**.

7. ИСПОЛЬЗОВАНИЕ МАКРОСОВ В ACCESS

7.1. Понятие макроса и макрокоманды

База данных Access это не только набор таблиц, хранящих взаимосвязанные данные, но и набор средств для работы с данными, таких как запросы, формы, отчеты.

Для автоматизации выполнения задач, связи различных объектов, создания, редактирования и автоматизации более сложной логики приложений необходимо использовать средства программирования.

Объектно-ориентированный язык программирования VBA (Visual Basic for Applications), являясь общим средством программирования для всего семейства Microsoft Office, позволяет создать в среде Access целостные графические диалоговые приложения пользователя с большими возможностями по управлению и контролю за их выполнением, решить и автоматизировать выполнение самых сложных задач.

Простейшим языком программирования, позволяющим добавлять функциональные возможности и автоматизировать выполнение задач приложения, является язык **макросов**. Использование макросов для автоматизации управления реакцией приложения на действия пользователя в формах или отчетах позволяет создавать полноценные интерактивные приложения без написания кода на VBA.

Макросы обеспечивают выполнение части команд, доступных в VBA, и для большинства пользователей создание макроса оказывается проще, нежели написание кода VBA.

Макрос — программа, состоящая из последовательности макрокоманд (макрос от слова "макрокоманда"). Часто макросом называют набор из одной или более команд, выполняющих определенные, часто используемые операции, например, открытие форм или печать отчетов.

Макросы могут быть полезны для автоматизации часто выполняемых действий. Если записать эти действия в виде макроса, то они будут выполняться автоматически при запуске макроса. Например, при нажатии пользователем кнопки можно запустить макрос, который распечатает отчет или выведет на экран форму.

Макрокоманда — это инструкция, ориентированная на выполнение определенного действия над объектами Access и их элементами. Макрокоманда — основной компонент макроса, инструкция, которая самостоятельно или в комбинации с другими макрокомандами определяет выполняемые в макросе действия (иногда макрокоманды называют просто командами макроса). Серия

макрокоманд, из которых состоит макрос, выполняется каждый раз при его запуске.

Макрос может быть наряду с другими объектами представлен как отдельный объект (изолированный макрос), который отображается в области навигации в группе **Макросы** (Macros). В такой макрос, в свою очередь, можно вложить один или несколько вложенных макросов. Кроме того, макрос, связанный с любым событием в форме, отчете или элементе управления, может быть внедрен в форму или отчет (внедренный макрос). При этом он не отображается как объект в группе **Макросы**, а становится компонентом формы или отчета.

Изолированный макрос может выполняться в ответ на многочисленные виды событий, возникающих в формах, отчетах и их элементах управления. Внедренный макрос всегда связывается с событием и сохраняется в форме или отчете. События наступают, прежде всего, при выполнении определенных действий пользователя с объектами. Примерами событий являются: изменение данных в поле, открытие или закрытие формы или отчета, нажатие кнопки в форме и просто передача фокуса от одного поля к другому. Сохранение внедренных макросов вместе с формами и отчетами упрощает управление объектами приложения.

7.2. Создание макроса

Для создания макроса в окне базы данных нужно выполнить **Создание/Макрос**. Появится окно конструктора макросов. Макрокоманда в макрос выбирается из списка **Добавить новую макрокоманду**.

Добавить новую макрокоманду или условный оператор очень просто. Макрокоманду можно выбрать в раскрывающемся списке, из контекстного меню по щелчку правой кнопкой мыши либо в области **Каталог макрокоманд**, расположенной в правой части конструктора макросов.

Выполнение каждой макрокоманды зависит от ее аргументов (некоторые макрокоманды не имеют аргументов). Аргументы вводятся в специально отведенные для этого поля, расположенные в нижней части окна макроса. Аргументы могут вводиться с помощью клавиатуры, однако, по возможности, лучше выбирать их из списка, чтобы избежать некорректных значений.

На рис. 7.1. показан созданный макрос **Ввод записи в форму по факультетам**. В него включены две макрокоманды: **Открыть форму** и **На Запись**. При выполнении этого макроса происходит открытие формы, заданной параметром **Имя формы**, и переход к записи, указанной в параметре **Запись**.

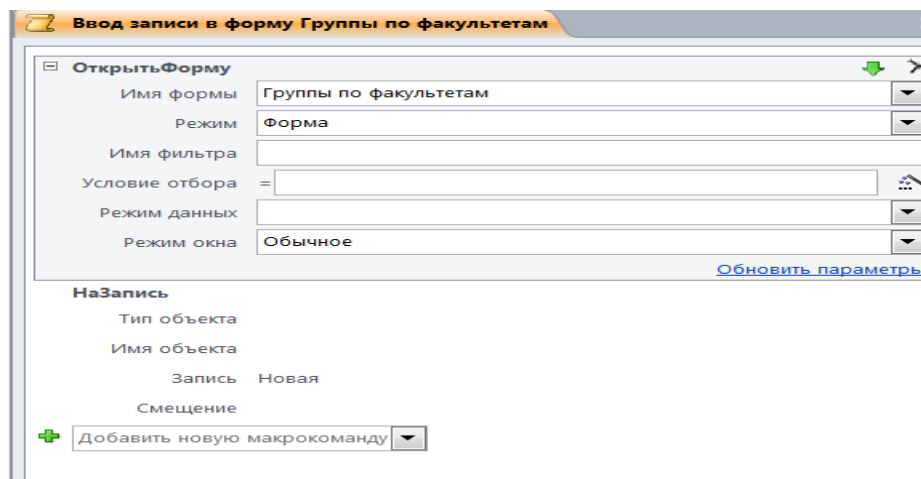


Рис.7.1. Пример макроса с двумя макрокомандами


Существует следующий прием быстрого создания макроса, выполняющего действия над конкретным объектом базы данных. Нужно выбрать объект в окне базы данных и переместить его с помощью мыши в ячейку макрокоманды в окне макроса.

Для этого следует выбрать в окне базы данных соответствующую вкладку, выбрать объект и переместить его значок с помощью мыши в ячейку макрокоманды. Например, для того чтобы создать макрос, открывающий форму, следует переместить с помощью мыши значок формы из окна базы данных в ячейку макрокоманды.

При переносе значка макроса в ячейку макрокоманды вводится макрокоманда, запускающая этот макрос, а при переносе значков других объектов (таблиц, запросов, форм, отчетов или модулей) в макрос добавляется макрокоманда, открывающая этот объект.

7.3. Запуск макроса



Выполнение макроса может начинаться по команде пользователя, при вызове из другого макроса или процедуры обработки события, а также в ответ на событие в форме, отчете или элементе управления. Например, можно назначить запуск макроса на кнопку в форме, в результате чего макрос будет выполняться при нажатии этой кнопки. Допускается также создание специальной команды меню или кнопки на панели инструментов, запускающей макрос; определение сочетания клавиш, нажатие которых запускает макрос, а также автоматический запуск макроса при открытии базы данных.

После создания макроса следует его сохранить. Запустить макрос на выполнение можно путем нажатия кнопки . Кроме того макрос можно запустить путем двойного клика на имени макроса или при выполнении контекстной команды **Выполнить**.

7.4. Выполнение макроса по шагам

В Access предусмотрена возможность выполнения макроса в пошаговом режиме. Для этого предназначена кнопка  **Конструктора макросов**.

В этом режиме перед выполнением каждой макрокоманды происходит останов, который сопровождается появлением диалогового окна **Пошаговое исполнение макроса**.

Для выполнения макроса по шагам следует нажать кнопку  **По Шагам** на панели инструментов и затем кнопку . Появится панель **Пошаговое исполнение макроса** (рис. 7.2.). Нажимая кнопку **Шаг** можно проследить выполнение последующей макрокоманды.

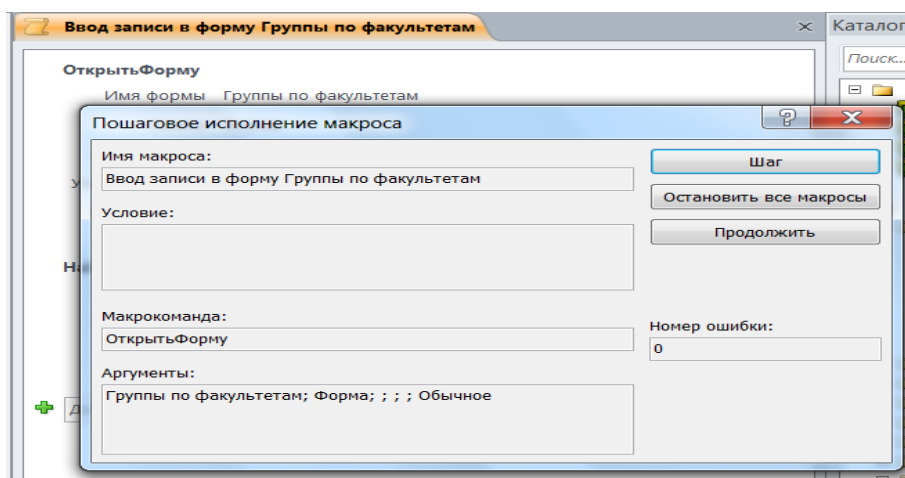


Рис.7.2. Пошаговое выполнение макроса

Кнопка **Шаг** позволяет выполнить следующую макрокоманду, а кнопка **Продолжить** – все последующие шаги без остановок, то есть выключить пошаговый режим. Режим пошагового выполнения позволяет найти логические ошибки, допущенные при записи макроса.

7.5. Вложенные и внедренные макросы

Access 2010 предоставляет возможность создавать в макросе вложенные макросы. Вложенный макрос имеет имя и может содержать любые макрокоманды.

В предыдущих версиях также предоставлялась возможность объединения макросов в группы. Работать с группой часто оказывается удобнее, чем с несколькими отдельными макросами, поскольку группа макросов в области навигации отображается как один объект. Целесообразно объединять несколько макросов в одном, если они связаны с решением одной задачи или используются при работе с одним объектом.

В окне **Каталог макрокоманд** имеется инструмент **Вложенный макрос**, с помощью которого можно создавать в составе макроса наборы макрокоманд, имеющие собственное имя. Приведем пример создания такого макроса. Создадим два набора макрокоманд – **Оценки студентов** и **Факультеты** (рис.7.3). Сам макрос назовем **Макрос2**.

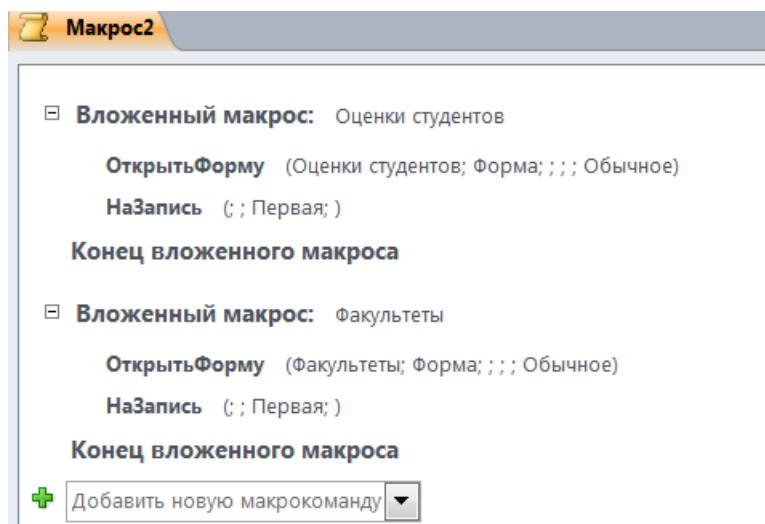



Рис. 7.3. Вложенный макрос

При запуске макроса командой **Выполнить** будет выполнен только первый набор макрокоманд, названный в рассматриваемом примере **Оценки студентов**.

Для запуска любого вложенного макроса из макроса **Макрос2** нужно выполнить **Работа с базами данных/Выполнить макрос**, затем из выпавшего списка выбрать нужный вложенный макрос. Имена вложенных макросов будут, соответственно, **Макрос2.Оценки студентов** и **Макрос2.Факультеты**.

Внедренный макрос всегда связывается с событием и сохраняется в форме или отчете. В лабораторной работе будет пример создания такого макроса.

7.6. Создание кнопки для запуска макроса в форме или отчете

Можно разместить кнопку на выбранной форме или отчете. Для этого следует открыть форму (или отчет) в режиме конструктора и воспользоваться инструментом  **Кнопка** на **Панели элементов**. По нажатии на кнопку следует указать место для кнопки в форме (или отчете). В **Окне свойств** для элемента управления **Кнопка** выбрать вкладку **События/Нажатие кнопки/Построитель** (нажав кнопку с троеточием). В открывшемся диалоговом окне **Построителя** можно выбрать элемент **Макросы**, выбрать внедренный макрос, который будет запускаться при открытии выбранной формы, или, например, отчета.

Кнопку, запускающую макрос, можно также создать, перетаскив с помощью мыши название макроса из окна базы данных в окно конструктора формы (или отчета) в область заголовка.

7.7. Условные макрокоманды

В макросе макрокоманды выполняются в порядке их расположения. Однако для реализации алгоритма в программах часто необходимо нарушить последовательность их выполнения в зависимости от некоторых условий. Условия позволяют определить порядок передачи управления между макрокомандами в макросе и обеспечивают выполнение определенных ветвей алгоритма. Например, в макросе проверяется значение поля в форме на соответствие заданным условиям, и для одних значений может потребоваться вывести сообщение, а для других значений произвести вывод отчета.

При работе с конструктором макросов можно инструментом **Если** вставить в состав макроса условную макрокоманду, которая будет выполняться при истинности заданного условия.

Для этого следует из каталога макрокоманд при создании макроса выбрать **Управление/Если** и ввести соответствующее условие в шаблон макрокоманды. При выполнении лабораторной работы это рассматривается более подробно.

7.8. Макрос AutoExec

Специальный макрос **AutoExec** автоматически выполняется при загрузке базы данных. Таким способом можно автоматизировать операции, которые должны всегда выполняться в момент загрузки базы данных. В процессе открытия базы данных Access проводит поиск макроса с этим именем и, если такой макрос существует, автоматически запускает его.

Для создания макроса **AutoExec** существует два способа:

- создать макрос, содержащий макрокоманды, которые требуется выполнить при открытии базы данных и сохранить его под именем **AutoExec**;
- в окне базы данных выделить ранее созданный макрос и переименовать макрос, присвоив ему новое имя - **AutoExec**.

При следующем открытии базы данных Access автоматически запустит этот макрос.

8. ГЛАВНАЯ КНОПОЧНАЯ ФОРМА

Главная кнопочная форма позволяет создавать удобный интерфейс для работы с базой данных, обеспечивая быстрый доступ к таблицам, формам, отчетам и другим объектам базы данных нажатием соответствующих кнопок на кнопочной форме.

Кнопочная форма есть не что иное, как меню для работы в базе данных. В меню может быть несколько вложенных подменю. Для включения диспетчера кнопочных форм в Access 2010 надо выполнить **Файл/Параметры/Панель быстрого доступа**. Затем на вкладке **Настройка панели быстрого доступа** из раскрывающегося списка слева выбрать строку **Вкладка «Работа с базами данных»** и в списке команд выделить **Диспетчер кнопочных форм**, а затем кликнуть по кнопке **Добавить**. Теперь надо в меню найти панель быстрого доступа (она сверху в левом углу), на которой и будет соответствующая пиктограмма диспетчера кнопочных форм.

В диалоге **Диспетчера кнопочных форм** перечисляются все страницы кнопочной формы. По умолчанию создается одна страница, имя которой **Главная кнопочная форма**. Кнопка **Создать** позволяет добавить на кнопочную форму еще одну страницу.

Для создания кнопок на существующей странице кнопочной формы нужно нажать кнопку **Изменить**. Появится диалог **Изменение страницы кнопочной формы**.

Для создания нового элемента кнопочной формы нужно нажать кнопку **Создать** в окне **Изменение страницы кнопочной формы**. Нажав кнопку **Изменить**, можно отредактировать существующий элемент кнопочной формы. Существующий элемент удаляется с помощью кнопки **Удалить**. Взаимное расположение элементов кнопочной формы изменяется с помощью кнопок **Вверх** и **Вниз**.

В поле **Текст** вводится надпись, поясняющая назначение создаваемой кнопки, в списке **Команда** выбирается команда, которая будет выполняться при нажатии кнопки. Если выбрана команда, требующая аргумента, появляется список для его выбора. Например, для команды **Открыть форму для изменения** список возможных аргументов будет называться **Форма**. В нем следует выбрать форму, которая будет открыта при нажатии создаваемой кнопки. Аналогично добавляются другие кнопки на кнопочную форму.

На рис. 8.1. приведен пример кнопочной формы.

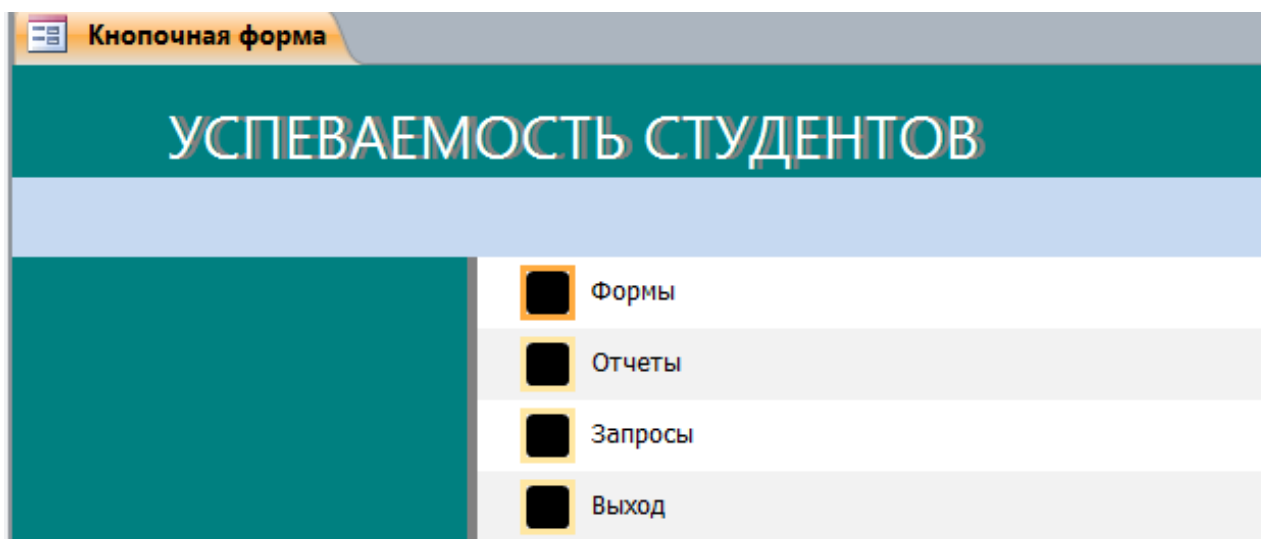


Рис. 8.1. Пример кнопочной формы

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Лабораторная работа 1

СОЗДАНИЕ И РЕДАКТИРОВАНИЕ ТАБЛИЦ. СОРТИРОВКА И ПОИСК ДАННЫХ.

Порядок проведения занятия

1. Запустить MS Access.
2. Создать новую базу данных в файле с именем **УспеваемостьСтудентов**.
3. Создать в режиме Конструктора следующую структуру таблицы **Факультеты**:

Имя поля	Тип поля	Свойства поля
КодФакультета	Счетчик	Ключевое поле
НаименованиеФак	Текстовый	Размер – 5; обязательное; индексированное (совпадения не допускаются)
Декан	Текстовый	Размер – 25; обязательное; неиндексированное

4. Перейти в режим таблицы и ввести в таблицу несколько записей.
5. Убедиться в невозможности изменения ключевого поля **Код**.
6. Отсортировать таблицу по возрастанию поля **НаименованиеФак**.
7. Создать в режиме Конструктора следующую структуру таблицы **Группы**:

Имя поля	Тип поля	Свойства поля
КодГруппы	Счетчик	Ключевое поле
НомерГруппы	Текстовый	Размер – 7; обязательное; индексированное (совпадения не допускаются); маска ввода - >LLL9999 (3 прописные буквы и 4 цифры)
КодФакультета	Числовой	Размер – длинное целое; обязательное; индексированное (допускаются совпадения); формат – фиксированный с 0 дес. знаков
ГодСоздания	Числовой	Размер – целое; обязательное; индексированное

		(допускаются совпадения); формат – фиксированный с 0 дес. знаков; маска ввода – 9999 (4 цифры); значение по умолчанию – 2014 ; условие на значение - Between 2010 And 2017 ; сообщение об ошибке – Недопустимый год создания группы
--	--	---

8. Добавить таблицы **Факультеты** и **Группы** в схему данных базы.

9. С помощью Мастера подстановок создать подстановку с комбинированным списком в поле **КодФакультета** из таблицы **Факультеты** (см. п. 2.5).

10. Открыть окно индексов (см. п. 2.3) и убедиться в наличии всех индексов, заданных при формировании структуры таблицы.

11. Открыть окно схемы данных и убедиться в появлении связи между таблицами.

12. Двойным щелчком по связи открыть окно **Изменение связей** и установить флажок **Обеспечение целостности данных**, если это не было сделано в п. 9.

13. Перевести таблицу **Группы** в режим таблицы и ввести несколько записей, выбирая из комбинированного списка различные факультеты.

14. Отсортировать таблицу по факультетам и номерам групп, используя для этого расширенный фильтр.

15. Создать в режиме Конструктора следующую структуру таблицы **Студенты**:

Имя поля	Тип поля	Свойства поля
КодСтудента	Счетчик	Ключевое поле
НомерЗачетки	Текстовый	Размер – 10; обязательное; индексированное (совпадения не допускаются)
ФИО	Текстовый	Размер – 25; обязательное; индексированное (допускаются совпадения)
БаллыЕГЭ	Числовой	Размер – целое; обязательное; индексированное (допускаются совпадения) ; формат – фиксированный с 0 дес. знаков; маска ввода – 999 (3 цифры);
КодГруппы	Числовой	Размер – длинное целое; обязательное; индексированное (допускаются совпадения) ;

		формат – фиксированный с 0 дес. знаков
ДатаРождения	Дата/время	Краткий формат даты; маска ввода формируется конструктором; обязательное; индексированное (допускаются совпадения)
Город	Текстовый	Размер – 20; обязательное; индексированное (допускаются совпадения) ; по умолчанию - Москва
Стипендия	Денежный	0 дес. знаков; маска ввода – 9 999р. ; обязательное; индексированное (допускаются совпадения) ; по умолчанию - 0

16. Добавить таблицу **Студенты** в схему данных базы.

17. С помощью Мастера подстановок создать подстановку с комбинированным списком в поле **КодГруппы** из таблицы **Группы** (см. п. 2.5).

18. Открыть окно индексов и убедиться в наличии всех индексов, заданных при формировании структуры таблицы.

19. Открыть окно схемы данных и убедиться в появлении связи между таблицами **Группы** и **Студенты**.

20. Двойным щелчком по связи открыть окно **Изменение связей** и установить флажок **Обеспечение целостности данных**, если это не было сделано в п. 17.

21. Перевести таблицу **Студенты** в режим таблицы и ввести 10-15 записей, выбирая из комбинированного списка различные группы.

22. Отсортировать таблицу по номерам зачетных книжек.

23. Найти в таблице **Студенты** все записи, удовлетворяющие следующим условиям:

- студенты, чьи фамилии начинаются с определенной буквы;
- студенты, обучающиеся в одной определенной группе.

24. Найти записи для студентов, проживающих в Москве, и очистить поле **Город** в этих записях с помощью операции **Поиск и замена** (см. п. 2.7).

25. Создать в режиме Конструктора следующую структуру таблицы **Дисциплины**:

Имя поля	Тип поля	Свойства поля
КодДисциплины	Счетчик	Ключевое поле
НазваниеДисц	Текстовый	Размер – 30; обязательное; индексированное (совпадения не допускаются)
Часы	Числовой	Размер – целое; необязательное; неиндексированное ; формат – фиксированный с 0 дес. знаков; маска ввода – 999 (3 цифры);

26. Перейти в режим таблицы и ввести в таблицу несколько записей.

27. Отсортировать таблицу по возрастанию поля **НазваниеДисц.**

28. Создать в режиме Конструктора следующую структуру таблицы

Оценки:

Имя поля	Тип поля	Свойства поля
КодСтудента	Числовой	Размер – длинное целое; обязательное; индексированное (допускаются совпадения) ; формат – фиксированный с 0 дес. знаков
КодДисциплины	Числовой	Размер – длинное целое; обязательное; индексированное (допускаются совпадения) ; формат – фиксированный с 0 дес. знаков
Оценка	Числовой	Размер – байт; обязательное; индексированное (допускаются совпадения) ; формат – фиксированный с 0 дес. знаков; маска ввода – 9 (1 цифра); условие на значение - Between 2 And 5 ; сообщение об ошибке – Недопустимая оценка

29. Выделить поля **КодСтудента** и **КодДисциплины** и создать первичный ключ по этим полям. Это позволит избежать повторного ввода результатов экзаменов для каждого сочетания **Студент-Дисциплина**.

30. Открыть окно индексов и убедиться в наличии всех индексов, заданных при формировании структуры таблицы.

31. Добавить таблицы **Дисциплины** и **Оценки** в схему данных базы.

32. С помощью Мастера подстановок создать подстановку с комбинированным списком в поле **КодСтудента** из таблицы **Студенты** (см. п. 2.5). Для удобства идентификации студентов при вводе данных использовать в подстановке поля **НомерЗачетки** и **ФИО** таблицы **Студенты**.

33. С помощью Мастера подстановок создать подстановку с комбинированным списком в поле **КодДисциплины** из таблицы **Дисциплины** (см. п. 2.5).

34. Открыть окно схемы данных и убедиться в появлении связей между таблицами **Оценки** и **Студенты**, **Оценки** и **Дисциплины**.

35. Двойным щелчком по каждой из этих связей дважды открыть окно **Изменение связей** и установить флажки **Обеспечение целостности данных**, если это не было сделано в п. 32-33. Для связи **Студенты-Оценки** установить флажок **Каскадное удаление связанных записей**, чтобы при удалении записи о студенте из таблицы **Студенты** автоматически удалялись оценки этого студента из таблицы **Оценки**.

36. Перевести таблицу **Оценки** в режим таблицы и ввести 10-15 записей, выбирая из комбинированных списков различных студентов и различные дисциплины.

37. Открыть окно схемы данных и убедиться, что в результате выполнения работы оно имеет вид, показанный на рис. 1.4.

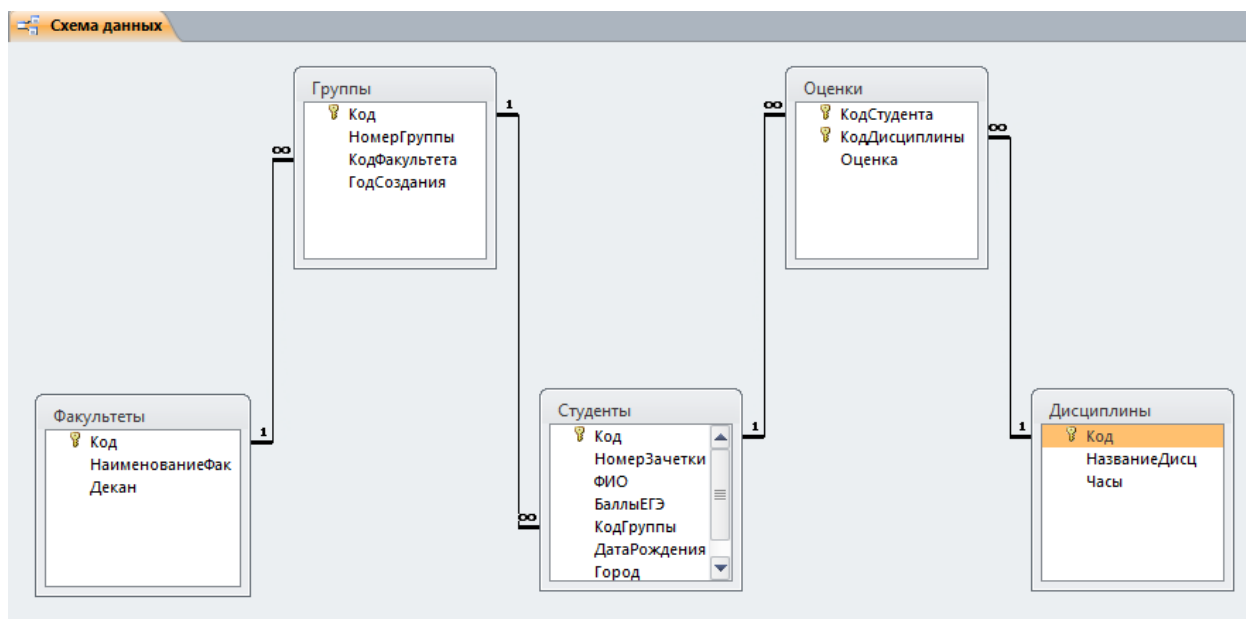


Рис. 1.4. Окно схемы данных

Содержание отчета

1. Тема и цель занятия.
2. Скриншоты всех таблиц в режиме конструктора и в режиме таблицы.
3. Скриншот схемы базы данных.

Контрольные вопросы

1. Что понимают под классом объектов, свойствами объектов?
2. Типы связей "один-к-одному", "один-ко-многим" и "многие-ко-многим": как они реализуются в реляционных базах данных? Как устанавливаются связи в СУБД Access?
3. Что такое ключевое поле? Какие требования предъявляются к нему? Как в СУБД Access определяется ключевое поле?
5. Для чего используются индексы таблиц? Как определяются индексы в Access?
6. Перечислите типы данных, поддерживаемые Access.
7. Для чего и как используется Мастер подстановок?
8. Как отсортировать или отобразить записи по одному полю? По нескольким полям?
9. Как произвести поиск и замену данных в Access?

Лабораторная работа 2

СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ФОРМ

Порядок проведения занятия

1. Создайте форму для ввода и редактирования записей в таблице

Факультеты:

- выберите в списке таблиц базы данных таблицу **Факультеты**;
- на вкладке меню **Создание/Другие формы** выберите режим **Таблица**;
- в появившемся окне с формой выберите функцию сохранения и сохраните форму, дав ей имя **Факультеты**; при этом в окне навигации появляется заголовок **Формы** и строка с именем созданной формы.

2. Используйте созданную форму для редактирования списка факультетов, обеспечив наличие не менее 2-3 записей в таблице.

3. Упорядочьте записи в списке по возрастанию наименования факультетов.

4. С помощью кнопок перемещения по записям перейдите на следующую запись, на последнюю запись, на предыдущую запись, на первую запись.

5. Создайте форму для ввода и редактирования записей в таблице

Дисциплины:

- выберите в списке таблиц базы данных таблицу **Дисциплины**;
- на вкладке меню **Создание/Другие формы** выберите режим **Разделенная форма**;
- в появившемся окне с формой выберите функцию сохранения и сохраните форму, дав ей имя **Дисциплины**; при этом в окне навигации появляется новая строка с именем созданной формы.

6. Используйте созданную форму для редактирования списка дисциплин, обеспечив наличие не менее 5 записей в таблице. Редактирование текущей записи следует производить в верхней части формы, а добавление и удаление записей – в списке записей в нижней части формы.

7. Упорядочьте записи в списке по возрастанию названия дисциплин.

8. Создайте форму для ввода и редактирования записей в таблице

Группы, сгруппированным по факультетам:

- выберите в списке таблиц базы данных таблицу **Факультеты**;
- на вкладке меню **Создание/Формы** выберите и нажмите кнопку **Форма**; при этом появляется окно с формой, в верхней части которой находятся поля с атрибутами текущего факультета, а в нижней – список групп текущего факультета;
- перейдите в режим **Конструктор** и измените заголовок формы на **Группы по факультетам**;
- сохраните форму под таким же именем;

- перейдите в **Режим формы** и отредактируйте списки групп по факультетам, обеспечив наличие не менее 5 групп на каждом факультете;
- с помощью кнопок перемещения выполните перемещения, как по факультетам, так и по группам выбранного факультета.

9. Создайте форму для ввода и редактирования записей в таблице

Студенты в общем списке:

- на вкладке меню **Создание** выберите и нажмите кнопку **Мастер Форм**;
- в появившемся окне выберите таблицу **Студенты**, из списка доступных полей выберите все поля кроме поля **Код** и нажмите кнопку **Далее**;
- в появившемся окне выберите внешний вид формы **Ленточный** и нажмите кнопку **Далее**, а в следующем окне введите имя формы **Студенты общий список** и нажмите кнопку **Готово**;
- на экране появляется окно с формой, созданной **Мастером форм**, а форма с именем **Студенты общий список** появляется в списке окна навигации;
- для подгонки ширины полей, их выравнивания, редактирования надписей и т.п. используйте **Режим макета** и **Конструктор**; обратите внимание на то, что в поле **КодГруппы** **Мастер форм** автоматически поместил комбинированный список для выбора номера группы.

10. Добавьте несколько записей в список студентов, выбирая номер группы из комбинированного списка.

11. Выполните перемещение по списку студентов, сортировку по различным полям формы, отбор записей о студентах некоторой группы.

12. Создайте форму для ввода и редактирования записей в таблице

Студенты, сгруппированным по группам:

- на вкладке меню **Создание** выберите и нажмите кнопку **Мастер Форм**;
- в появившемся окне выберите сначала таблицу **Студенты**, из списка доступных полей выберите все поля кроме поля **Код** и **КодГруппы**;
- затем выберите таблицу **Группы**, из списка доступных полей выберите поля **НомерГруппы** и **Код факультета** и нажмите кнопку **Далее**;
- в появившемся окне выберите вид представления **Группы** и **Подчиненные формы** и нажмите кнопку **Далее**;
- в следующем появившемся окне выберите внешний вид подчиненной формы **Ленточный** и нажмите кнопку **Далее**;
- в следующем окне задайте имя основной формы **Студенты по группам** и нажмите кнопку **Готово**;
- на экране появляется окно с двумя формами, созданными **Мастером форм**, а их имена (**Студенты по группам** и **Студенты подчиненная**

форма) появляются в списке окна навигации; в основной форме расположены поле с номером группы и комбинированный список с наименованием факультета, в подчиненной форме – список студентов текущей группы с полями для ввода и редактирования данных;

- для подгонки расположения форм в окне, ширины полей, их выравнивания, редактирования надписей и т.п. используйте **Режим макета** и **Конструктор**; обратите внимания на то, что хотя созданный набор форм предназначен для редактирования списка студентов и их атрибутов, вы можете при необходимости редактировать и атрибуты групп: номер и факультет; если вы хотите запретить такое редактирование в этой форме, следует в режиме **Конструктор** установить для этих объектов свойству **Блокировка** значение **Да**.

13. Используя созданную форму, добавьте и отредактируйте записи в таблице **Студенты**, обеспечив наличие не менее 5 студентов в каждой группе, а также следующие условия, используемые в дальнейших лабораторных работах для обеспечения достаточного количества записей, выдаваемых по запросам:

- баллы ЕГЭ у различных студентов должны принимать значения, более или менее равномерно заполняющие 3 диапазона: менее 200, от 200 до 250 и более 250;
- годы дат рождения различных студентов должны более или менее равномерно принимать значения трех последовательных лет (например, 1997, 1998, 1999);
- город местожительства различных студентов должен более или менее равномерно принимать 3 различных значения (значение по умолчанию **Москва** и два произвольно выбранных города;
- стипендии различных студентов должны более или менее равномерно принимать 3 различных значения (значение по умолчанию **0** и два других значения, соответствующие обычной и повышенной стипендии).

14. Создайте форму для ввода и редактирования записей в таблице **Оценки**, сгруппированном по студентам:

- на вкладке меню **Создание** выберите и нажмите кнопку **Мастер Форм**;
- в появившемся окне выберите сначала таблицу **Оценки**, из списка доступных полей выберите поля **КодДисциплины** и **Оценка**;
- затем выберите таблицу **Студенты**, из списка доступных полей выберите поля **НомерЗачетки** и **ФИО** и нажмите кнопку **Далее**;
- в появившемся окне выберите вид представления **Студенты** и **Подчиненные формы** и нажмите кнопку **Далее**;
- в следующем появившемся окне выберите внешний вид подчиненной формы **Табличный** и нажмите кнопку **Далее**;
- в следующем окне задайте имя основной формы **Оценки студентов** и нажмите кнопку **Готово**;

- на экране появляется окно с двумя формами, созданными **Мастером форм**, а их имена (**Оценки студентов** и **Оценки подчиненная форма**) появляются в списке окна навигации; в основной форме расположены поля с номером зачетки и ФИО студента, в подчиненной форме – список сданных экзаменов текущего студента с комбинированным списком для выбора дисциплины и полем для ввода оценки;
- для подгонки расположения форм в окне, ширины полей, их выравнивания, редактирования надписей и т.п. используйте **Режим макета** и **Конструктор**; обратите внимания на то, что хотя созданный набор форм предназначен для редактирования списка сданных экзаменов, вы можете при необходимости редактировать и атрибуты студентов: *номер зачетки* и *ФИО*; если вы хотите запретить такое редактирование в этой форме, следует в режиме **Конструктор** установить для этих объектов свойству **Блокировка** значение **Да**.

15. Используя созданную форму, добавьте и отредактируйте записи в таблице **Оценки**, обеспечив наличие не менее 3 сданных экзаменов каждым студентом по разным дисциплинам. Оценки должны более или менее равномерно принимать все допустимые значения.

17. Создать форму для ввода и редактирования записей в таблице **Оценки**, сгруппированную по дисциплинам. Порядок создания формы аналогичен п.14 с учетом изменившейся группировки записей.

Содержание отчета

1. Тема и цель занятия.
2. Скриншоты всех форм в режиме конструктора и в режиме формы.

Контрольные вопросы

1. С какой целью создаются формы? Назовите виды форм в Access.
2. Назовите инструменты создания форм.
3. Перечислите основные разделы форм и их назначение.
4. Для чего используется раздел **Данные** в форме?
5. Как разместить в форме объект?
6. Как используется при создании форм **Список полей**?
7. Как изменить порядок обхода объектов в форме?
8. Как построить форму по двум связанным таблицам?

Лабораторная работа 3

РАБОТА С ДАННЫМИ С ИСПОЛЬЗОВАНИЕМ ЗАПРОСОВ. ЗАПРОСЫ-ВЫБОРКИ

Порядок проведения занятия

Следует запустить Access и открыть базу данных **Успеваемость Студентов**, созданную в предыдущих работах. После создания соответствующих запросов их необходимо сохранять под именами, отражающими смысл запроса. Следует научиться создавать запросы как с помощью **Мастера запросов**, так и **Конструктора запросов**.

Создание однотабличных запросов

1. Определить ФИО студента с максимальным значением балла ЕГЭ.
2. Получить данные конкретного студента, ФИО которого вводится с клавиатуры.
3. Выбрать ФИО студентов, дата рождения которых до определенной вводимой даты.

Создание многотабличных запросов.

4. Выбрать студентов конкретной группы, у которых балл ЕГЭ меньше 200.
5. Выбрать список номеров групп одного конкретного факультета.
6. Выбрать ФИО студентов, проживающих в одном из городов и обучающихся на одном из факультетов.

Создание запросов с вычисляемыми полями.

7. Вычислить возраст каждого студента одной конкретной группы (использовать функцию DateDiff).
8. Подсчитать надбавку 15% от стипендии тем студентам, которые имеют только отличные оценки.

Создание запросов с групповыми функциями.

9. Вычислить средний балл ЕГЭ каждой группы.
10. Вычислить количество студентов каждой группы.
11. Определить номер группы, у которой средний балл ЕГЭ наименьший.

Создание перекрестных запросов.

12. Создайте перекрестный запрос с помощью **Мастера запросов** на базе таблицы **Студенты**, выбрав в качестве строк **Город**, в качестве столбцов - **КодГруппы**, а вычисляемое поле – **НомерЗачетки** и используя для подсчета функцию **Число**. Для этого, после выбора **Мастера запросов**, на последующих шагах следует выбрать таблицу, поля в качестве строк и в качестве столбцов, вычисляемое поле, функцию для подсчета итогов.

Сохраните запрос под именем **Распределение студентов по городам**; просмотрите запрос в режиме конструктора.

Недостатком этого запроса является использование в заголовках столбцов кодов групп, а не номеров групп. Причина в том, что **Мастер запросов** может выбрать поля только из одной таблицы. Если нужны поля из разных таблиц (а поле **Номер группы** находится в таблице **Группы**) при использовании **Мастера запросов** необходим дополнительный запрос с разными полями, на основе которого и будет построена перекрестная таблица.

13. Построим запрос с использованием номеров групп в **Конструкторе запросов**. Для этого:

- выполним **Создать/Конструктор запросов**;
- выберем две таблицы: **Студенты** и **Группы**;
- выполним **Работа с запросами/Перекрестный**; в качестве заголовков строк – **НомерГруппы**, столбцов – **Город**, вычисляемое поле с помощью функции **Count**, итог – функция **Count** по полю **НомерЗачетки** (рис. 3.1).
- сохраним запрос под именем **Количество студентов по городам и группам**.

14. Постройте перекрестный запрос на базе дополнительного запроса, для чего создайте простой запрос, включив в него поля **НаименованиеФак** из таблицы **Факультет**, **Город** и **НомерЗачетки** из таблицы **Студенты**. Имя запроса: **ФакультетЗапрос**.

Постройте перекрестный запрос с помощью **Мастера запросов** на базе запроса **ФакультетЗапрос**; поле **НаименованиеФак** выберите в качестве заголовка строк, поле **Город** - заголовка столбцов, для подсчета количества студентов используйте функцию **Count** по полю **НомерЗачетки**. Выполните запрос и проанализируйте результат.

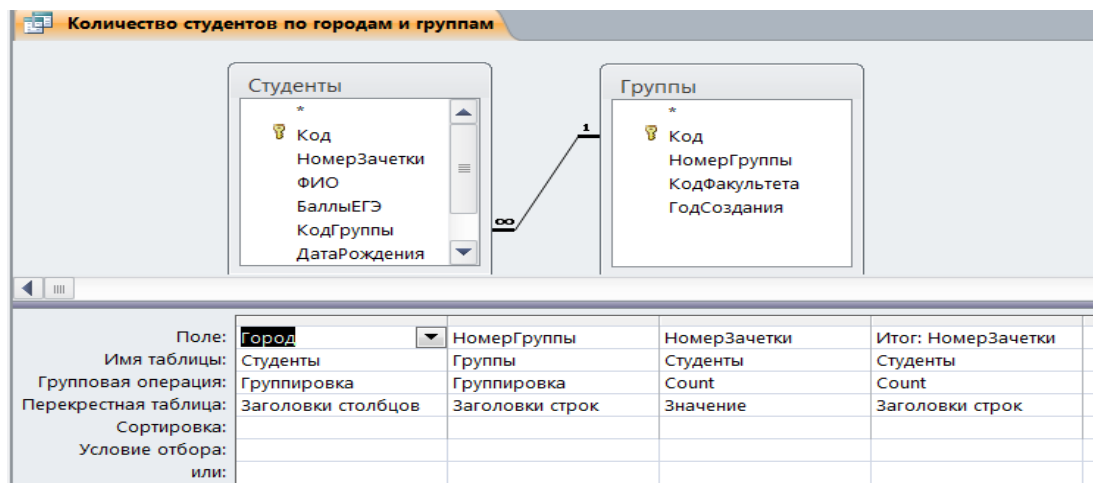


Рис. 3.1. Перекрестный запрос

Содержание отчета

1. Тема и цель занятия.
2. Скриншоты всех запросов в режиме конструктора и в режиме SQL и результатов их выполнения.

Контрольные вопросы

1. Что такое запрос?
2. Назовите элементы окна конструктора запросов.
3. Что такое бланк QBE?
4. Где записываются критерии условия выбора для запроса?
5. Назначение итоговых запросов.
6. Назначение построителя выражений.
7. Как вычислить сумму значений заданного поля?
8. Как осуществить сортировку записей в запросе?
9. Какие виды вычислений можно произвести в итоговых полях?

Лабораторная работа 4

РАБОТА С ДАННЫМИ С ИСПОЛЬЗОВАНИЕМ ЗАПРОСОВ. ЗАПРОСЫ-ДЕЙСТВИЯ.

Порядок проведения занятия

Запустите Access и откройте базу данных **Успеваемость Студентов**, созданную в предыдущих работах.

Разработка запросов на создание таблицы

1. Сконструируйте запрос-выборку к таблице **Студенты**, выбрав тех, кто не получает стипендии. Выполните запрос и просмотрите результат.

2. Перейдите в режим **Конструктора** и преобразуйте запрос в запрос для сохранения результатов в виде отдельной таблицы. Для этого нужно выполнить **Создание таблицы** и ввести наименование будущей таблицы **Студенты без стипендии**. Просмотрите полученную таблицу.

3. Создайте еще одну таблицу. Сначала сконструируйте простой запрос, включив в него все поля таблицы **Оценки**. Для этого после добавления в запрос таблицы **Оценки** надо выбрать в списке полей таблицы верхний символ * (рис. 4.1).

4. Дополним этот запрос для получения оценок только по одной дисциплине. В таблице **Оценки** есть код дисциплины, а не наименование, что неудобно для отбора оценок: знание кода дисциплины для пользователя необязательно. Для отбора оценок, например, по вычислительной математике, добавим в запрос таблицу **Дисциплины** и включим в запрос поле **НазваниеДисц**. В строке **Условие отбора** поставим в поле наименование требуемой дисциплины (например, вычислительной математики), а в строке **Вывод на экран** выключим флажок, чтобы наименование дисциплины не выводилось. Таким образом, запрос будет выводить только данные из таблицы **Оценки** по заданной дисциплине (рис. 4.1).

Выполним этот запрос и убедимся в правильности результатов.

5. В группе **Тип запроса** выберем **Создание таблицы** и зададим имя создаваемой таблицы, например, **Оценки По Вычмату**. Выполним этот запрос и посмотрим созданную таблицу. Мы получили список оценок только по одной дисциплине.

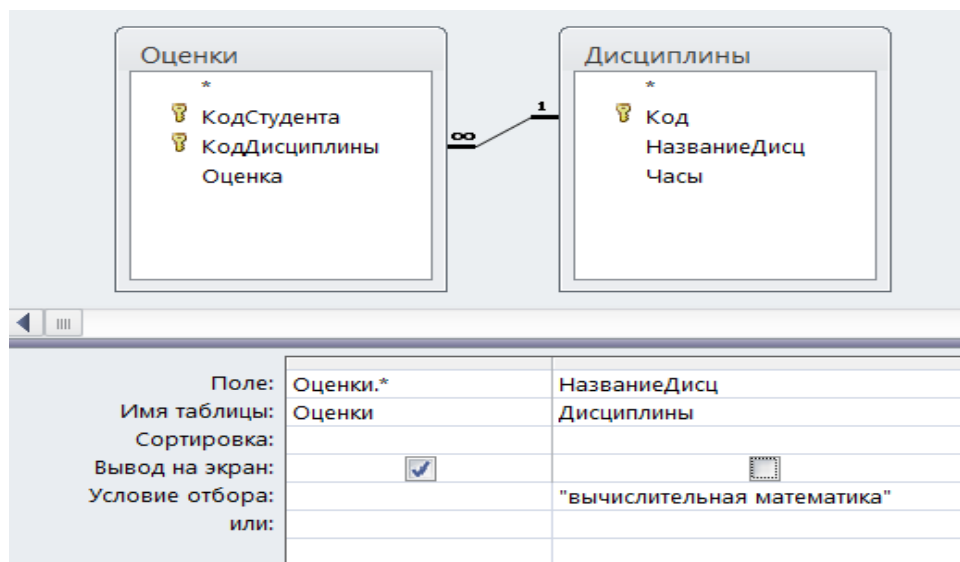


Рис. 4.1. Запрос для выборки оценок по дисциплине «вычислительная математика»

Разработка запроса на удаление записей

6. Разработаем запрос на удаление записей с оценками по вычислительной математике. Будем упражняться с копией таблицы **Оценки**, для чего ее надо создать. Надо выделить таблицу, выполнить копирование, затем вставку и указать имя таблицы (рис. 4.2)

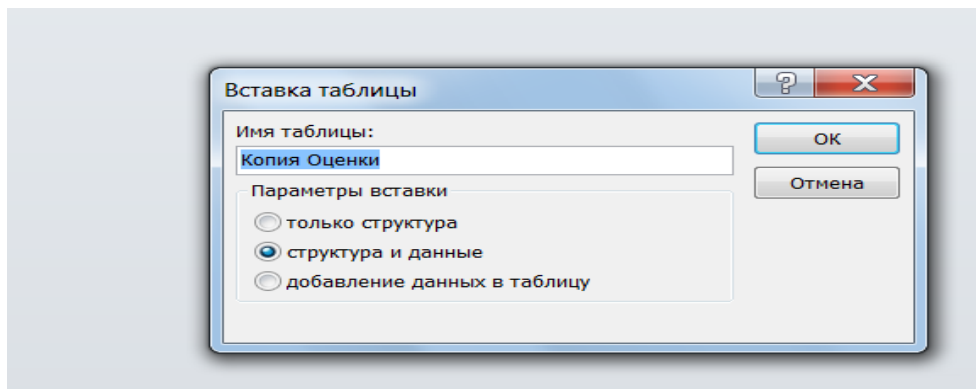


Рис.4.2. Вставка таблицы **Копия Оценки**

7. Удалим записи об оценках по вычислительной математике из копии таблицы, а затем попробуем их вернуть (добавить) из сохраненной таблицы **Оценки По Вычмату**. Для этого создадим запрос-выборку к таблице **Копия Оценки** для получения списка оценок по вычислительной математике. Подключим таблицу **Дисциплины** и установим связь между таблицами по коду дисциплины (рис. 4.3).

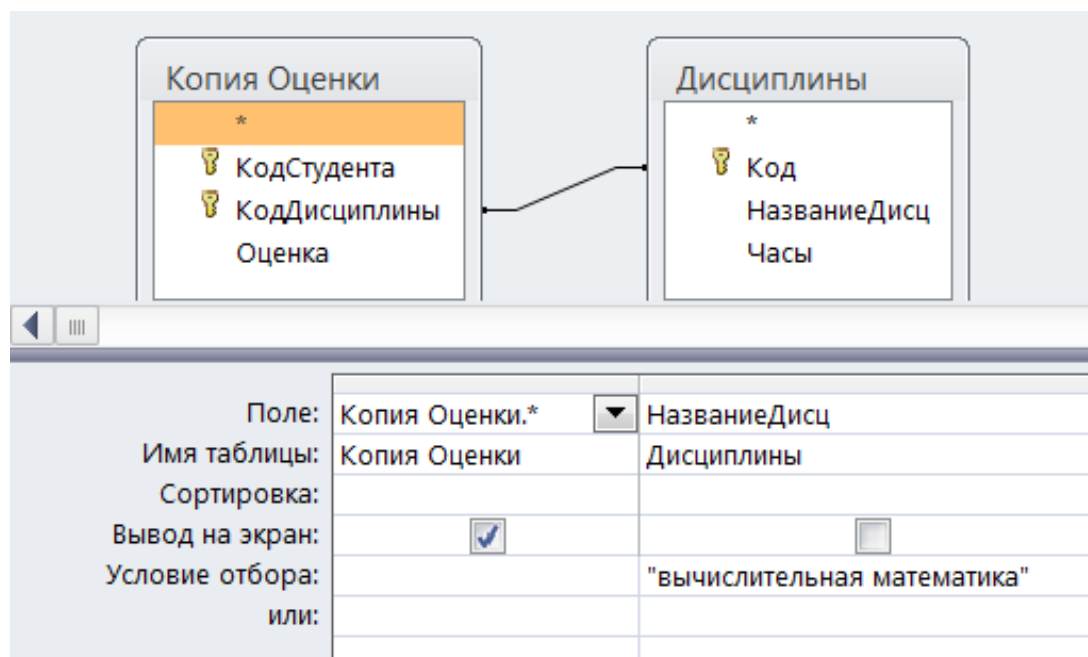


Рис. 4.3. Подключение таблицы **Дисциплины**

8. Выделим связь и установим в окне **Параметры объединения** объединение только тех записей, в которых поля обеих таблиц совпадают (рис. 4.4).

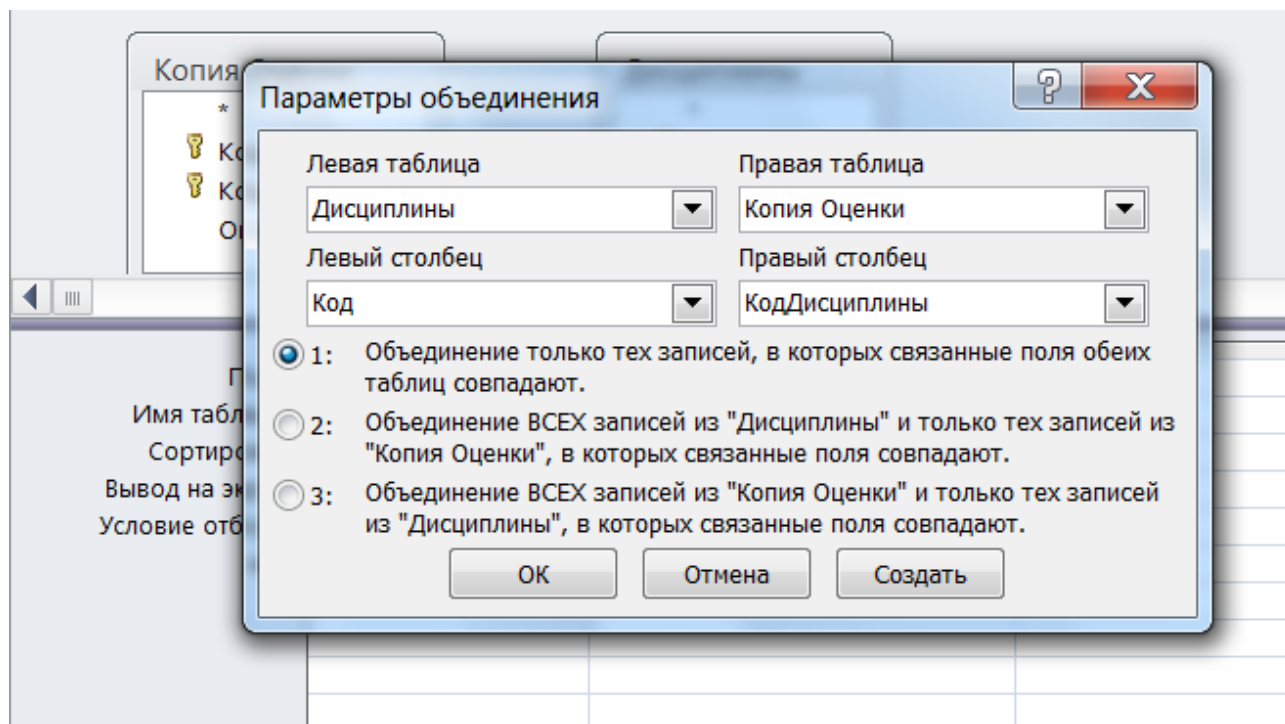


Рис. 4.4. Установка параметров объединения

9. Преобразуем запрос в запрос-удаление, нажав **Работа с запросами/Удаление**. В строке **Удаление** появляется запись **Из**, указывающая на имя таблицы, из которой производится удаление. Подтвердим удаление и выполним его (рис. 4.5). Убедимся, что из копии таблицы эти записи удалены. После этого таблицу следует закрыть.

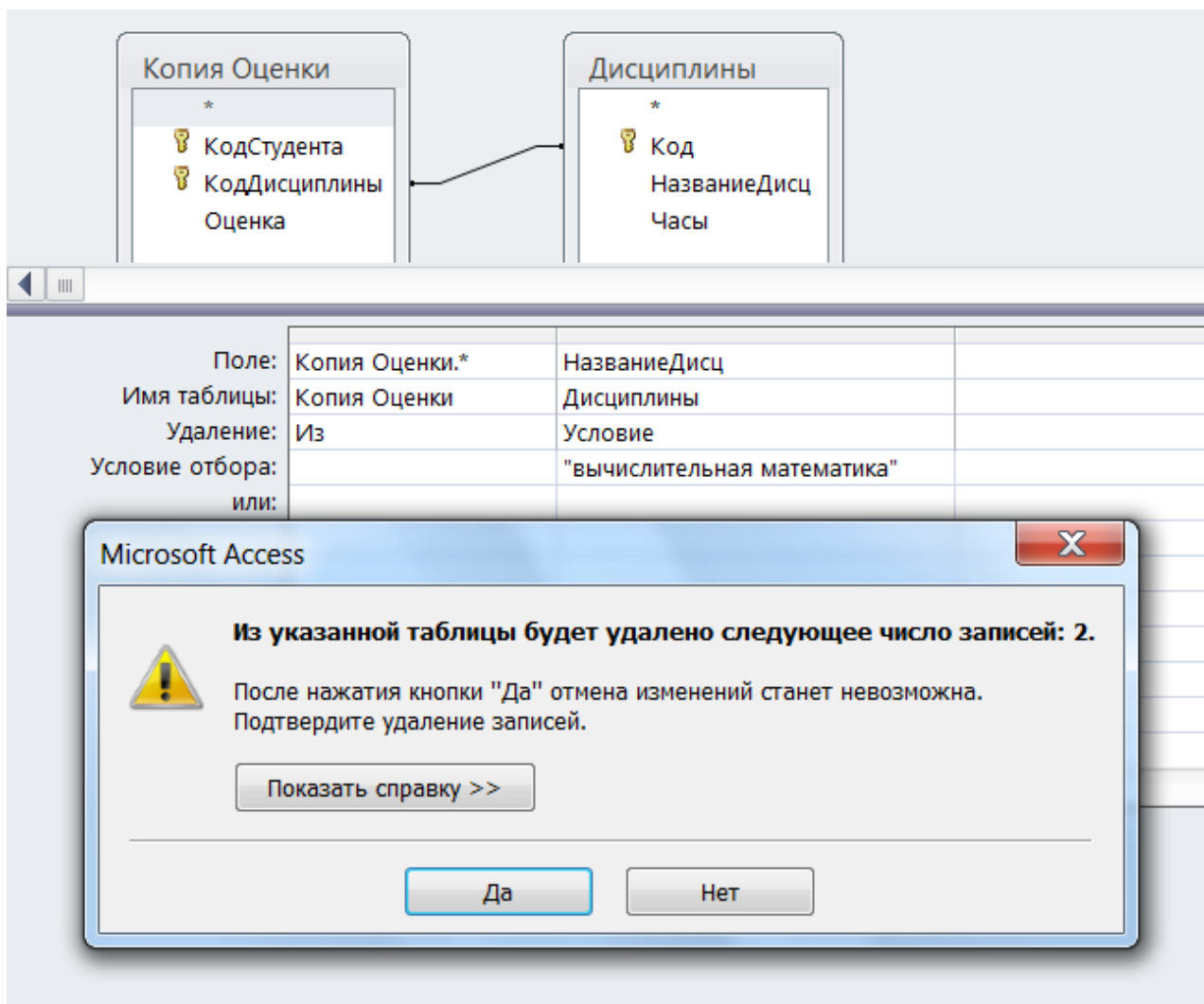


Рис.4.5. Дополнительное окно при удалении из таблицы **Копия Оценки**

Разработка запроса на добавление записей

С помощью запроса на добавление записи из одной таблицы можно поместить в другую таблицу. Добавим записи из таблицы **Оценки По Вычмату** обратно в таблицу **Копия Оценки**. Для этого:

10. Создадим запрос, в который включим все поля таблицы **Оценки По Вычмату**, и выполним его.

12. В окне **Добавление** выберем таблицу, в которую будут добавляться соответствующие записи – **Копия Оценки**. Выполним запрос, при этом выводится сообщение о количестве добавляемых записей и предлагается подтвердить запрос (рис. 4.6). Убедимся, что записи добавились в таблицу.

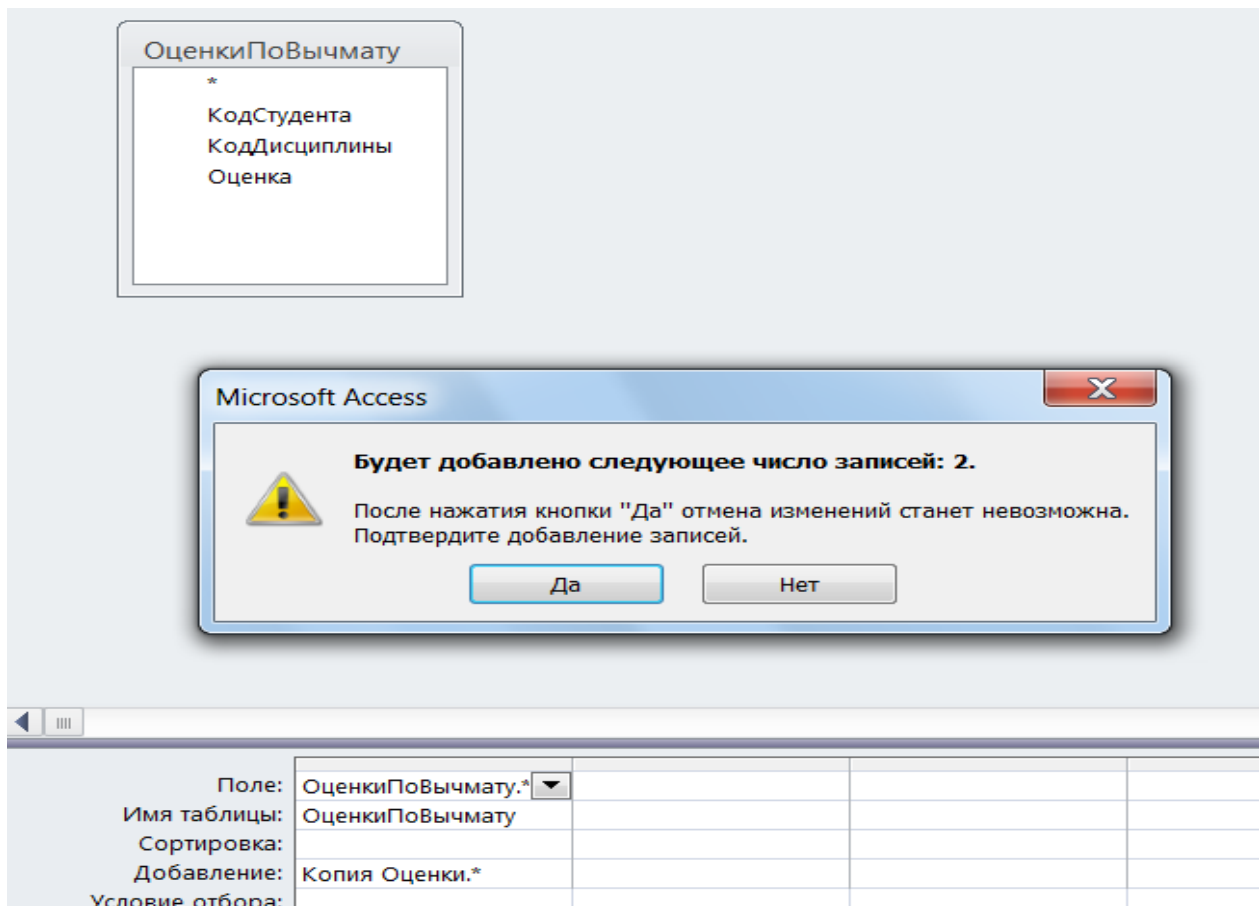


Рис.4.6. Сообщение о количестве добавляемых записей

13. Добавим в таблицу **Оценки** группу записей - результаты экзамена по одной дисциплине в одной группе. Сначала добавим группу записей с условно нулевой оценкой для всех студентов данной группы, а затем можно будет все нулевые оценки изменить на оценки, полученные данными студентами за экзамен.

Предварительно необходимо изменить условие на значение поля **Оценка**: вместо **Between 2 And 5** необходимо установить **Between 0 And 5**.

Составим параметрический запрос-выборку с параметрами **Номер группы** и **Дисциплина**. Затем преобразуем его в запрос-добавление и выполним. Просмотрим результат запроса. Количество добавленных записей должно соответствовать количеству студентов в данной группе.

Отредактируем оценки на произвольные оценки за экзамен. После выполнения данного пункта задания необходимо вернуть старое условие на значение поля **Оценка**.

Разработка запроса на обновление записей

Замените в таблице **Студенты** одно название города на другое. Для этого:

14. Выберите в таблице **Студенты** всех, кто проживает в определенном городе, например, Москве. Сделав запрос, проверьте правильность отбора записей.

15. Перейдите в режим **Конструктора** и преобразуйте запрос в запрос на обновление данных, выбрав **Обновление** в меню **Работа с запросами**. В строке **Обновление** задайте новое значение для поля **Город** (например, перепишите его с заглавной буквы или введите сокращенное название города и т.п.).

16. Выполните запрос и просмотрите результаты выполнения в режиме просмотра таблицы **Студенты**.

17. Выполните запрос-обновление таблицы **Студенты** для восстановления первоначальных данных.

Содержание отчета

1. Тема и цель занятия.
2. Скриншоты всех запросов в QBE и SQL.
3. Скриншоты обработанных таблиц до выполнения запросов и после.

Контрольные вопросы

1. Какие типы запросов имеются в Access?
2. Как создать запрос на обновление части данных в таблице?
3. Как удалить часть данных в таблице?
4. Какие проблемы могут возникнуть при использовании запросов на удаление?
5. Как создать таблицу для постоянного хранения результатов определенного запроса?
6. Как посмотреть результат запроса?

Лабораторная работа 5

СОЗДАНИЕ ОТЧЕТОВ НА ОСНОВЕ ТАБЛИЦ И ЗАПРОСОВ

Порядок проведения занятия

1. Откройте базу данных **Успеваемость Студентов**.
2. Создайте **отчет № 1** с результатами экзаменов по разным дисциплинам. Будущий отчет в режиме **Представления отчета** приведен на рис. 5.1.

РЕЗУЛЬТАТЫ ЭКЗАМЕНОВ			
Название	Номер Группы	ФИО	Оценка
<u>Информатика</u>			
	БИК1502	Иванов И.И.	3
Итого для БИК1502 (1 запись)			
Среднее по группе			3,00
	БИН1503	Фомин Т.А.	2
		Кузнецов Т.И.	5
		Сидоров С.С.	4
Итого для БИН1503 (3 записей)			
Среднее по группе			3,67
<u>Вычислительная математика</u>			
	БИН1502	Петров П.П.	4
Итого для БИН1502 (1 запись)			
Среднее по группе			4,00
16 января 2016 г.			Стр. 1 из 1

Рис. 5.1. Отчет №1 Результаты экзаменов в режиме Представление отчета

- Выполните **Создание/Мастер отчетов**. Последовательно выберите из таблицы **Дисциплины** поле **НазваниеДисц**, из таблицы **Группы** поле **НомерГруппы**, из таблицы **Студенты** поле **ФИО** и из таблицы **Оценки** поле **Оценка**;
- На втором диалоге, который появляется только в случае, если для отчета выбрано несколько таблиц, можно установить группировку. Для нашего случая выберем таблицу **Дисциплины**;
- На следующем шаге добавьте для группировки поле **НомерГруппы**
- Выберите порядок сортировки по полю **ФИО**, по кнопке **Итоги** - вычисление среднего (**Avg**) по полю **Оценки**;
- Выберите вид макета для отчета ступенчатый;
- Сохраните отчет под именем **Результаты экзаменов**.
- Просмотрите отчет на экране, анализируя его содержимое (группировка записей, вид итогов, элементы страницы отчета и т.д.). Очевидно, что полученный отчет необходимо доработать, чтобы он принял вид как на рис.5.1. На рис. 5.2 приведен возможный заключительный вид отчета в режиме конструктора.
- Перейдите в режим **Конструктора** данного отчета. Если какие-то элементы отчета не умещаются на экране, попробуйте в режиме предварительного просмотра поменять разметку страницы с книжной на альбомную.
- Произведите в отчете следующие изменения:

- ✓ Надписи **Верхнего колонтитула** и соответствующие им надписи и поля из **Заголовка группы 'Номер группы'** разместите компактно и левее;
- ✓ В **Примечании группы 'Номер группы'** заменить надпись **Avg** на **Среднее по группе**;
- ✓ В полях раздела **Примечание группы "Номер группы"** в свойствах выберите **Тип границы – Отсутствует**;
- ✓ Чтобы получить итоги только по группам, а не по дисциплинам, удалите в **Примечании группы "Код"** все поля и надписи, а высоту полосы сделайте равной 0;
- ✓ Отредактируйте соответствующие подписи. Добавьте подчеркивания и линии из инструмента конструктора отчетов. Необходимо подогнать поля отчета в соответствии с образцом. Просмотрите отчет и сохраните.

Результаты экзаменов									
Заголовок отчета									
РЕЗУЛЬТАТЫ ЭКЗАМЕНОВ									
Верхний колонтитул									
Название		Номер Группы			ФИО		Оценка		
Заголовок группы 'Код'									
НазваниеДисц									
Заголовок группы 'НомерГруппы'									
НомерГруппы									
Область данных									
					ФИО		Оценка		
Примечание группы 'НомерГруппы'									
="Итого для " & "" & [НомерГруппы] & " (" & Count(*) & "" & If(Count(*)=1									
Среднее по группе					=Avg([Оценк				
Примечание группы 'Код'									
Нижний колонтитул									
=Now()					="Стр. " & [Page] & " из " & [Pages]				
Примечание отчета									

Рис. 5.2. Отчет № 1 **Результаты экзаменов** в режиме **Конструктора**

Примечание: многие операции с отчетом можно выполнять в режиме макета.

3. Создайте **отчет № 2** на основе запроса.

- Создайте базовый запрос к таблицам **Студенты – Оценки - Дисциплины**, включив в него поля **НомерЗачетки, ФИО,**

НазваниеДисц, Оценка. Сохраните запрос под именем **Данные для отчета по успеваемости;**

- В качестве режима создания отчета выберите **Мастер отчетов;**
- На первом шаге **Мастера отчетов** выберите все поля исходного запроса для включения в отчет;
- На втором шаге выберите тип представления данных - по таблице **Студенты;**
- На третьем шаге не надо добавлять уровень группировки, нажмите **Далее;**
- На четвертом шаге выберите сортировку по полю **НазваниеДисц** и вид итогов - Среднее (**Avg**) по полю **Оценка;**
- На последующих шагах **Мастера** задайте стиль оформления отчета и его название **Результаты экзаменов студентов (на основе запроса).**
- Отредактируйте отчет, следует подогнать поля отчета левее.
- Добавьте в режиме конструктора новое вычисляемое поле **Надбавка.** Для этого:
 - ✓ В области верхнего колонтитула поместите надпись **Надбавка;**
 - ✓ В области **Примечание группы** 'Номер зачетки' поместите свободное поле и удалите присоединенную надпись для него;
 - ✓ Вызовите **Окно свойств** поля и с помощью **Построителя выражений** в свойстве **Данные** введите
=Иф(Avg([Оценка])=5;150;Иф(Avg([Оценка])>=4;100;0))
Таким образом, надбавка будет начисляться в зависимости от результатов сессии. В **Окне свойств** в свойстве **Макет** установите соответствующее число десятичных знаков для вывода;
- Отредактируйте отчет, замените **Avg** в **Примечании группы** 'Номер зачетки' на **Среднее студента.**

4. Создайте **отчет №3** по двум таблицам с включением в базовый отчет подчиненного отчета. В отчет предполагается выводить состав студентов по каждой группе. Вид будущего отчета представлен на рис. 5.3.


 Состав групп		11 марта 2016 г. 11:52:55									
Группа	БИК1401										
<table> <tr> <th>ФИО</th><th>Дата рождения</th><th>Стипендия</th></tr> <tr> <td>Иванов И.И.</td><td>15.04.1998</td><td>1 000р.</td></tr> <tr> <td>Петров П.П.</td><td>03.05.1998</td><td>1 200р.</td></tr> </table>			ФИО	Дата рождения	Стипендия	Иванов И.И.	15.04.1998	1 000р.	Петров П.П.	03.05.1998	1 200р.
ФИО	Дата рождения	Стипендия									
Иванов И.И.	15.04.1998	1 000р.									
Петров П.П.	03.05.1998	1 200р.									
Группа	БИК1402										
<table> <tr> <th>ФИО</th><th>Дата рождения</th><th>Стипендия</th></tr> <tr> <td>Сидоров С.С.</td><td>30.08.1999</td><td>1 000р.</td></tr> <tr> <td>Кузнецов А.И.</td><td>30.08.1997</td><td>0р.</td></tr> </table>			ФИО	Дата рождения	Стипендия	Сидоров С.С.	30.08.1999	1 000р.	Кузнецов А.И.	30.08.1997	0р.
ФИО	Дата рождения	Стипендия									
Сидоров С.С.	30.08.1999	1 000р.									
Кузнецов А.И.	30.08.1997	0р.									

Рис. 5.3 Отчет №3 **Состав групп** в режиме **Предварительного просмотра**

- В качестве основной таблицы отчета необходимо выбрать таблицу **Группы**, а таблицу **Студент** – в качестве источника данных для подчиненного отчета со списком студентов.
- Выберем таблицу **Группы** и выполним **Создание/Отчет**. Перейдем в режим конструктора для данного отчета.
- Очистите **Верхний колонтитул** от всех полей и надписей, а в **Области данных** оставьте только поле для **Номера группы** и добавьте для него надпись **Группа**. Сохраните отчет под именем **Состав групп**.
- Создайте подчиненный отчет в **Мастере отчетов**. В качестве источника данных выберите таблицу **Студенты**.
- В подчиненном отчете в области данных должны быть включены поля **ФИО**, **ДатаРождения** и **Стипендия**, а в заголовок из верхнего колонтитула перенесены соответствующие надписи. Выделите все подписи полей в верхнем колонтитуле и перетащите их в заголовок. При необходимости отрегулируйте высоту заголовка.
- Поскольку отчет будет использоваться только как подчиненный, можно удалить оба колонтитула. Для этого нажмите правой кнопкой мыши на **Верхний** или **Нижний Колонтитул** и выберите **Колонтитулы страницы**. После этого в открывшемся окне нажимаем **ДА**.
- Сохраните подчиненный отчет под именем **Студенты**.
- Откройте отчет **Состав групп** и перетяните в его область данных подчиненный отчет **Студенты**. Удалите надпись подчиненного отчета.
- Измените заголовок на **Состав групп**.

- Просмотрите отчет **Состав групп** в режиме **Макета** или **Предварительного просмотра**. На Рис. 5.4 приведен возможный вид отчета в режиме конструктора. Отредактируйте отчет и сохраните.

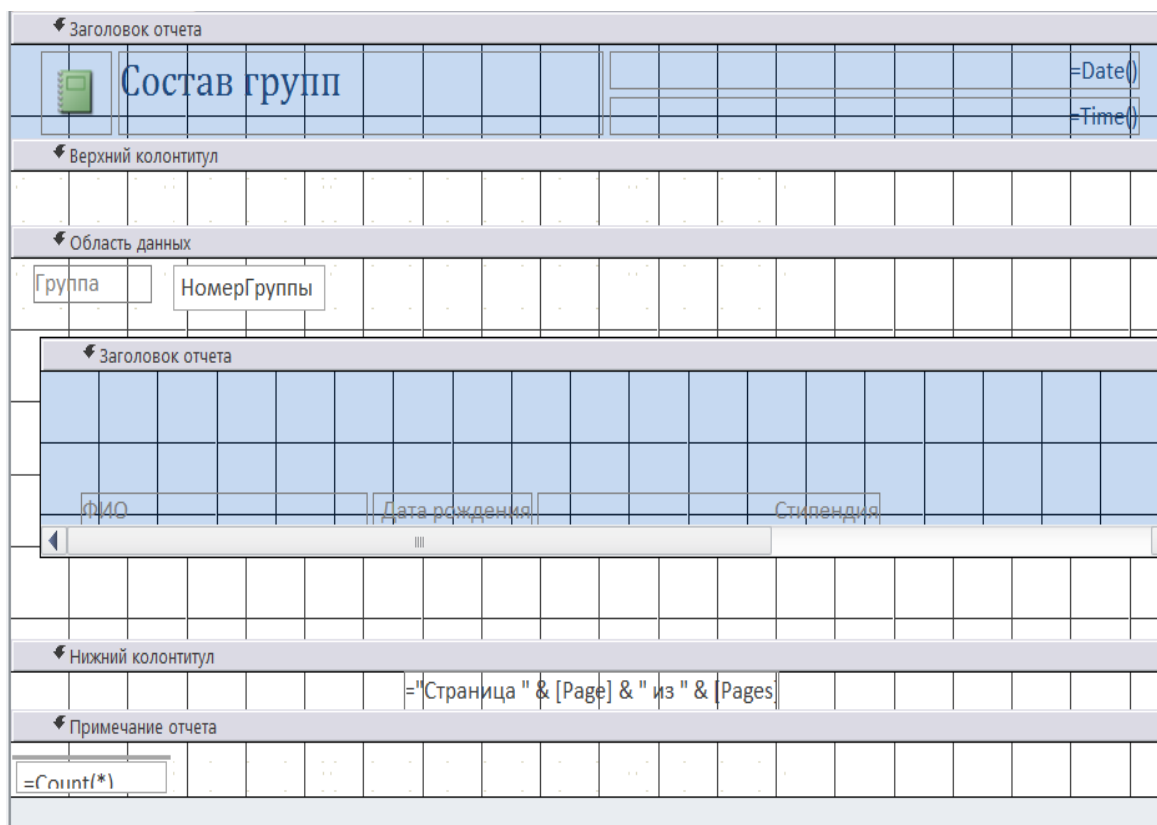


Рис. 5.4 Отчет №3 **Состав групп** в режиме **Конструктора**

Содержание отчета

1. Тема и цель занятия.
2. Скриншоты всех отчетов в режиме просмотра и в режиме конструктора.

Контрольные вопросы

1. Что представляют собой отчеты в Access?
2. В чем преимущества отчетов перед другими методами вывода данных?
3. С чего начинается создание отчета?
4. Чем характеризуется режим **Конструктора**?
5. Назовите разделы отчета.
6. Назовите инструменты окна отчета.
7. Как добавить вычисляемое поле в область данных?
8. Наличие каких объектов необходимо для внедрения подчиненного отчета?
9. Какие изменения можно внести в отчет в режиме **Конструктора**?

Лабораторная работа 6

ИСПОЛЬЗОВАНИЕ МАКРОСОВ В ACCESS

Порядок проведения занятия

Создание изолированного макроса

Создание изолированного макроса, являющегося отдельным объектом базы данных, начинается с выполнения команды **Макрос** на вкладке ленты **Создание** в группе **Макросы и код**. В результате выполнения команды открываются окно макроса и каталог макрокоманд

1. На вкладке **Создание** в группе **Макросы и код** щелкните **Макрос**. Откроется конструктор макросов.

2. Для добавления макрокоманды в макрос в конструкторе макросов выберите макрос **ОкноСообщения** из выпадающего меню **Добавить новую макрокоманду**.

3. Укажите следующие аргументы для макрокоманды (Рис. 6.1.):

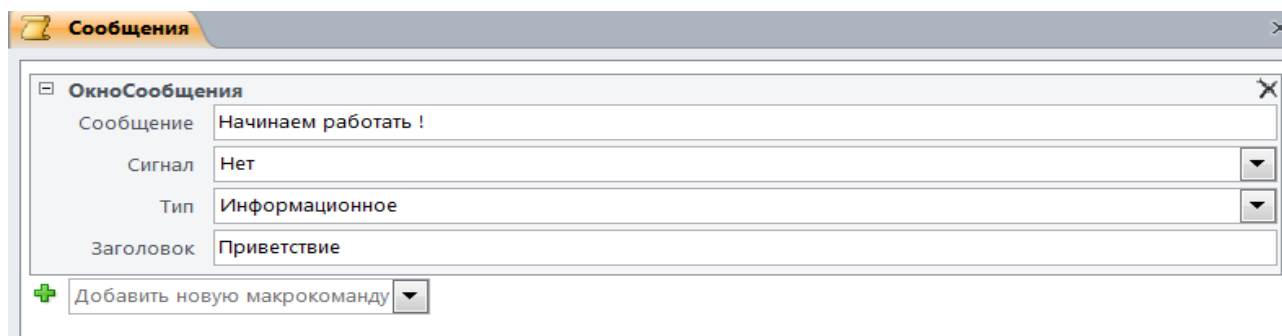


Рис. 6.1 Аргументы макрокоманды

4. Выполните макрос с помощью кнопки **Выполнить**. Подтвердите сохранение макроса, присвоив ему имя **Сообщения**.

5. Переименуйте макрос, дав ему имя **Autoexec**.

6. Закройте базу данных, затем снова откройте ее и убедитесь, что теперь макрос выполняется автоматически при открытии базы.

7. Верните макросу прежнее имя.

Работа с вложенными макросами

Создание вложенных макросов

8. Откройте макрос **Сообщения** в режиме конструктора и добавьте макрокоманду **Вложенный макрос** из каталога макрокоманд.

9. Присвойте имя вложенному макросу «**Окончание**» и добавьте в окно вложенного макроса макрокоманду **ОкноСообщения** со следующими аргументами (рис. 6.2.):

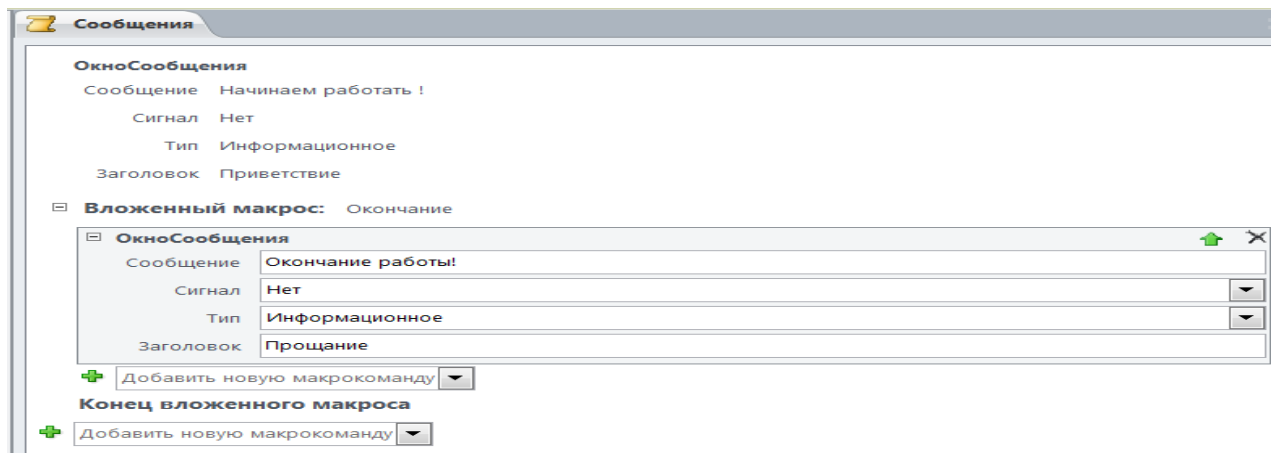



Рис. 6.2. Вложенный макрос

10. Сохраните изменения и закройте окно макроса.
11. Выполните макрос из окна базы данных. Будет выполнен только первый макрос из созданных макросов.

Связь группы макросов с формой (работа с формой **Дисциплины**).

12. Откройте форму **Дисциплины** в режиме конструктора.
13. Откройте окно свойств формы (**Страница свойств**) в группе **Сервис**, выберите вкладку **Свойств формы - События**.
14. Для свойства **Открытие** выберите из списка макрос **Сообщения**.
15. Для свойства **Заккрытие** выберите макрос **Сообщения.Окончание**.
16. Сохраните и закройте форму. Затем откройте форму **Дисциплины** (выполнится основной макрос **Сообщения**), а потом закройте (выполнится вложенный макрос **Сообщения.Окончание**).

Создание внедренного макроса

Для создания внедренного макроса необходимо выбрать в области навигации форму или отчет и выполнить в его контекстном меню команду **Конструктор** или **Режим макета**. Далее открывается окно свойств и выбирается элемент управления, раздел или форма (отчет) целиком, на вкладке **События** выбирается событие, в которое нужно встроить макрос, затем нажимается кнопка построителя . В диалоговом окне **Построитель** выделяется пункт **Макросы** и затем нажимается кнопка **ОК**. В результате откроется окно макроса, точно такое же, как при создании изолированного макроса.

Внедренные макросы отличаются от изолированных макросов тем, что они хранятся в формах и отчетах. Они не отображаются в виде объектов в группе **Макросы** в области навигации. Это упрощает управление базой данных, поскольку не нужно следить за тем, какие макросы относятся к какой форме или отчету. Внедренные макросы сохраняются в составе формы или отчета и при их копировании, импорте или экспорте.

17. Откройте отчет **Состав групп** в режиме конструктора.

18. Разместите на форме отчета элемент управления **Кнопка**.

19. Откройте **Окно свойств** для элемента управления **Кнопка**, вкладку **События**, событие **Нажатие кнопки** и выберите **Построитель** (нажав кнопку с троеточием). В открывшемся диалоговом окне **Построителя** выберите элемент **Макросы**. Откроется окно **Конструктора макросов**.

20. Выберите макрокоманду **ОкноСообщения**, введите текст сообщения: «ЭТОТ ОТЧЕТ ПРЕДНАЗНАЧЕН ДЛЯ ВЫВОДА ДАННЫХ ПО ГРУППАМ», остальные аргументы произвольно назначьте. Выполните макрос и сохраните его. Закройте макрос и отчет **Состав групп** и заново его запустите. Проверьте работу кнопки. Обратите внимание, что созданный внедренный макрос не отобразился в списке макросов.

Создание макроса с условной макрокомандой

Составим макрос, который сначала будет открывать форму, а затем выдавать сообщение с помощью условной макрокоманды **Если**, которая будет выполняться при истинности заданного условия. Для этого следует:

21. Создать макрос и включить в него макрокоманду **Открыть форму**. Выбрать форму **Оценки студентов**.

22. В каталоге макрокоманд справа выбрать команду управления **Если** (рис. 6.3).

23. Выбрать построитель в макрокоманде, нажав на кнопку . Ввести условие MsgBox("Новая запись?";1;"Что делать?")=1

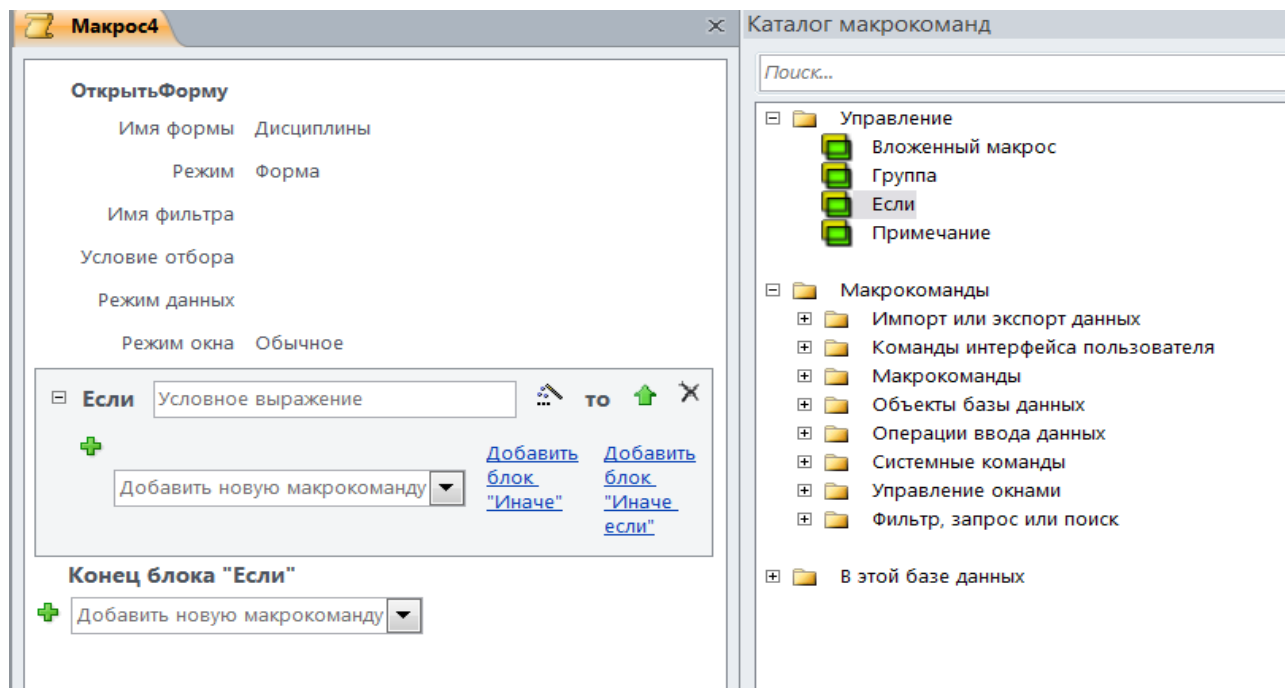


Рис. 6.3. Выбор макрокоманды Если

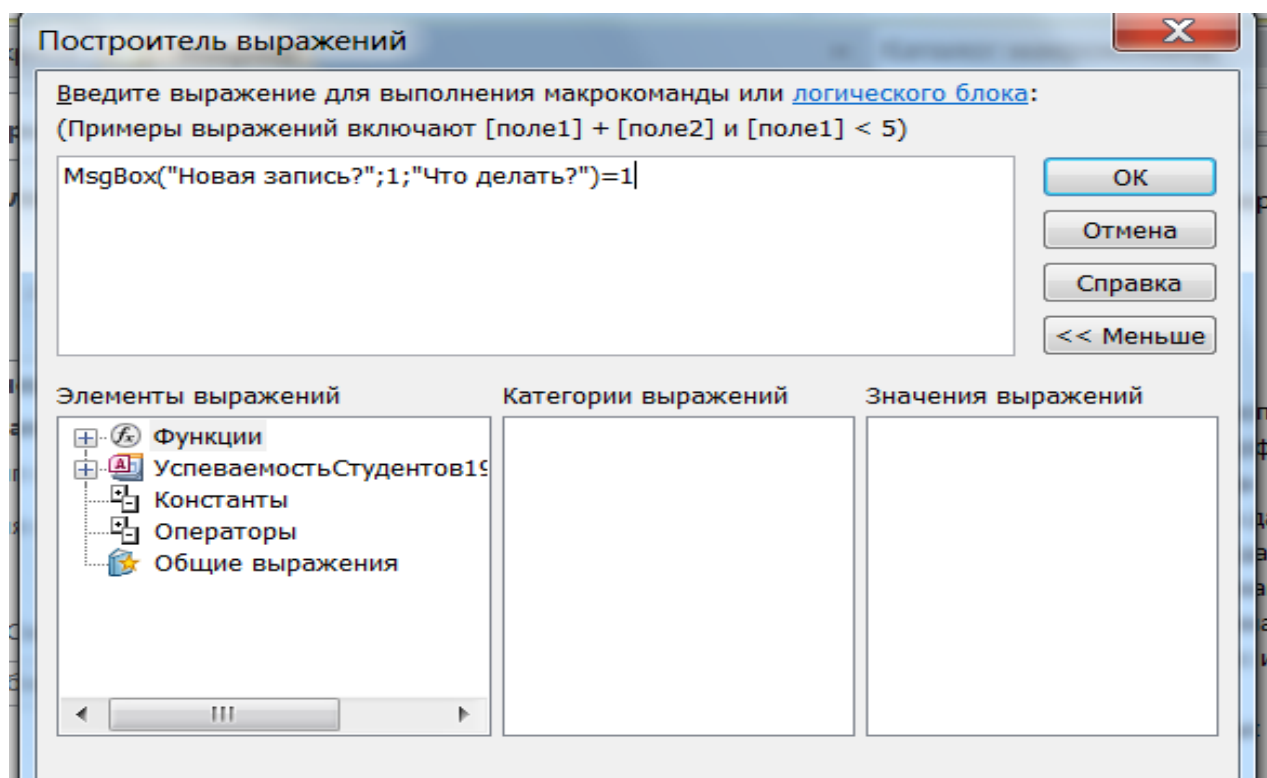


Рис. 6.4. Ввод условной макрокоманды

Смысл условия (рис 6.4.) состоит в том, что при выполнении условной макрокоманды должно выводиться окно сообщения (**MsgBox**), на котором пользователь должен или нажать кнопку **ОК** окна сообщения для перехода к

вводу новой записи или на кнопку **Отмена**. Если нажать кнопку **Отмена** третья макрокоманда, которую мы в дальнейшем введем, не выполнится.

24. Добавить третью макрокоманду **НаЗапись** (рис.6.5).

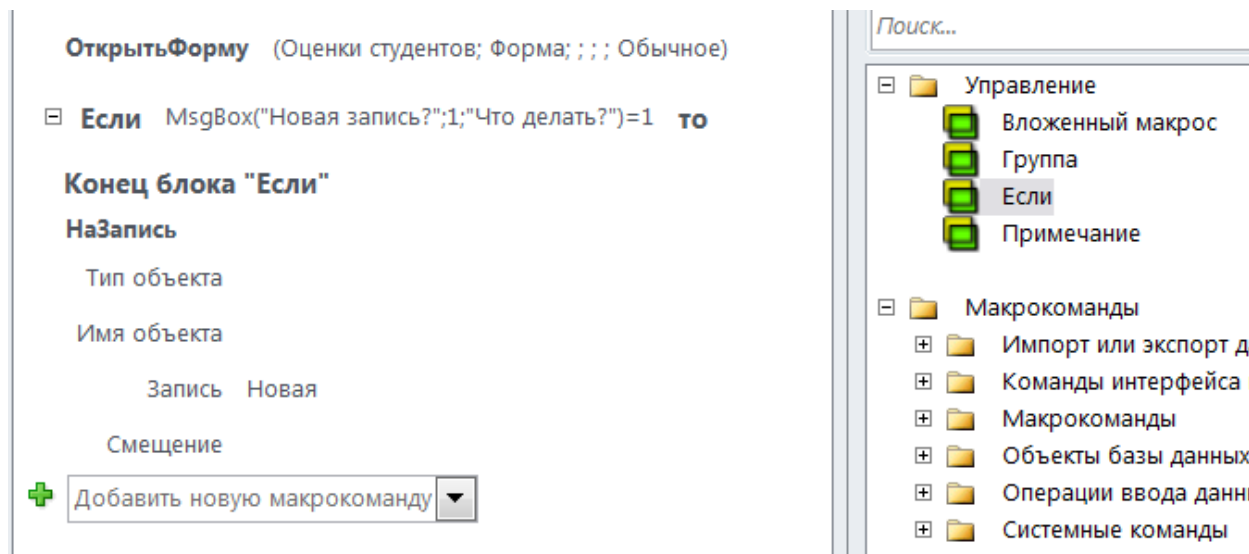


Рис. 6.5. Макрокоманда **НаЗапись**

25. Сохраните макрос и проверьте его выполнение. Результат приведен на рис.6.6.

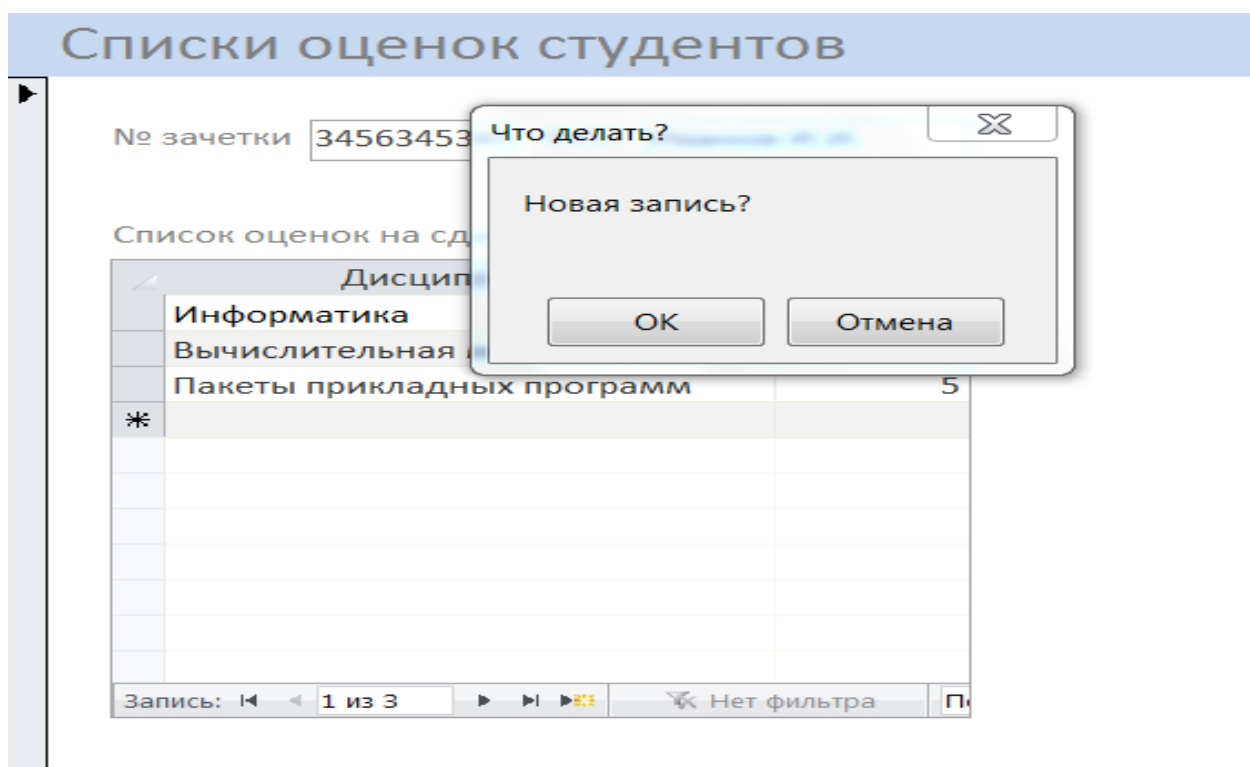


Рис. 6.6. Результат выполнения макроса

Содержание отчета

1. Название и цель работы.
2. Скриншоты всех макросов.

Контрольные вопросы

1. Что такое макрокоманда?
2. Что такое макрос?
3. Как создать макрос?
4. Какие существуют способы запуска макроса?
5. Какие ошибки могут появиться при выполнении макроса и как их исправить?
6. Что такое вложенный макрос, внедренный макрос?
7. Как создать макрос с условной макрокомандой?
8. Как добавить на панель инструментов кнопку, запускающую макрос?
9. Как в режиме формы создать кнопку для запуска макроса?
10. Для чего предназначен макрос AutoExec?
11. Как создать макрос AutoExec?

Лабораторная работа 7

ИЗУЧЕНИЕ ДИСПЕТЧЕРА КНОПОЧНЫХ ФОРМ

Порядок проведения занятия

1. Откройте БД «Успеваемость студентов». Создайте меню 1-го уровня:
 - Закройте все таблицы, формы, отчеты, запросы.
 - Запустите диспетчер кнопочных форм. В ответ на вопрос **Создать кнопочную форму?** нажмите кнопку Да. Перед Вами окно диспетчера кнопочных форм, в котором перечисляются все страницы кнопочной формы. По умолчанию – одна с именем **Главная кнопочная форма**.
 - Нажмите кнопку **Изменить**, введите название кнопочной формы «Успеваемость студентов» и нажмите кнопку **Заккрыть**.
 - Нажмите кнопку **Создать** и введите имена элементов меню 1 уровня.
 - В окне **Создание** введите текст «Формы», нажмите ОК.
 - Нажмите кнопку **Создать** и в окне **Создание** введите текст «Запросы», ОК.
 - Аналогично создайте следующие страницы кнопочной формы: **«Отчеты», «Выход»**.

2. Создайте меню 2-го уровня, последовательно раскрывая каждый пункт меню 1-го уровня.

- Отметьте курсором главную кнопочную форму «Успеваемость студентов», нажмите кнопки **Изменить** и **Создать** и в окне **Изменение кнопочной формы** введите в поле «Текст» - **Формы**, в поле «Команда» - **Переход к кнопочной форме**, в поле «Кнопочная форма» выбрать из списка - **Формы** и нажать **ОК**.
- Точно так же создайте элементы кнопочной формы 2 уровня: **Запросы**, **Отчеты**, **Выход**. В элементе **Выход** выберите команду **Выйти из приложения**.
- Закройте окно **Изменение страницы кнопочной формы** и вернитесь в окно **Диспетчер кнопочных форм**.

3. Создайте меню 3-го уровня:

В окне «Страницы кнопочной формы» установите курсор на страницу **Формы** и нажмите кнопку **Изменить**. Вы находитесь в окне **Изменение страниц кнопочной формы**.

- Нажмите **Создать**, введите в поле «Текст» - «Форма Дисциплины», выберите команду **Открытие формы для изменения**, выберите из списка форму **Дисциплина**, **ОК**. Точно так же создать элементы меню **Формы кнопочной формы 3 уровня** для выбора двух других любых форм базы данных.

Вы создали меню для ввода и корректировки информации в формах базы данных. Точно так же создайте меню 3-го уровня для просмотра отчетов, выполнив действия аналогичные действиям при создании меню для форм.

4. Закройте диспетчер кнопочных форм.

Теперь в Вашей Базе на вкладке **Формы** появилась новая форма с именем **Кнопочная форма**. Откройте эту форму и проверьте, как работает меню.

В Вашем Меню имеется одна кнопка «Выход», настроенная на выход из базы данных.

5. Создайте на каждом уровне кнопки управления для возврата на более высокий уровень (выход в меню **Успеваемость студентов**):

- Закройте все таблицы, формы, отчеты, запросы.
- Запустите **Диспетчер кнопочных форм**.
- Выделите страницу **Формы**, нажмите кнопку **Изменить**.
- Создайте новый элемент кнопочной формы с элементами: текст - **Выход** и командой – **Перейти к кнопочной форме**.
- Выберите кнопочную форму – «Успеваемость студентов», **ОК**.
- Закройте окно **Изменение страницы кнопочной формы**.

- Выделите **Отчеты**, нажмите кнопку **Изменить**.
- Повторите действия для создания кнопки выхода из пункта меню **Отчеты**.
- Закройте диспетчер кнопочных форм.

6. Закройте диспетчер кнопочных форм.

Содержание отчета

1. Название и цель работы.
2. Скриншоты кнопочных форм всех уровней.

Лабораторная работа 8

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗЫ ДАННЫХ (индивидуальное задание №1)

8.1. Цель работы

Получение навыков самостоятельного проектирования базы данных, создания БД и ее таблиц в среде Access, построения запросов к БД и оформления результатов их выполнения в виде отчетов.

8.2. Общее задание

1. Изучить теоретический материал по теме “Инфологическое проектирование БД. Разработка ER-моделей”.
2. Рассмотреть пример разработки ER-модели БД “Учебный процесс”, приведенный в разделе 8.3.
3. Повторить изученные в предыдущих лабораторных работах способы создания БД, работы с таблицами БД, создания форм для ввода и редактирования данных, формирования запросов к БД, разработки отчетов по запросам в среде Access.
4. Выбрать вариант индивидуального задания (п. 8.4).
5. Проанализировать предметную область проектируемой БД и уточнить задание, определив данные, которые необходимо хранить в БД.
6. Разработать модель “Сущность-связь” (ER-модель) проектируемой БД: выявить необходимый набор сущностей, определить требуемый набор атрибутов для каждой сущности, определить ключи и связи между сущностями, формализовать связи.
7. Создать новую БД в Access.
8. Создать таблицы, соответствующие сущностям ER-модели БД. Определить типы и свойства полей, создать необходимые подстановки.

9. Создать схему БД, определив связи между таблицами в соответствии со связями между сущностями в ER-модели. Установить для каждой связи свойства обеспечения целостности данных.

10. Создать формы для ввода и редактирования данных в БД и с их помощью ввести записи в БД. Объем данных в таблицах должен обеспечивать выдачу не менее 3-5 записей по каждому запросу задания.

11. Создать заданные запросы с помощью **Конструктора** или **Мастера** запросов. Для многотабличных запросов установить нужный тип параметров объединения таблиц. Просмотреть запросы в режиме SQL. Выполнить запросы и сохранить их в БД.

12. Создать отчеты по всем запросам.

13. Создать кнопочную форму для работы с БД.

8.3. Пример разработки ER-модели БД «Учебный процесс»

Пусть БД должна содержать списки студентов и изучаемых ими предметов, а также результаты экзаменов студентов по этим предметам.

Данные о каждом студенте должны содержать номер зачетки, фамилию, имя и отчество, название факультета, курс, номер группы. Данные о каждом предмете должны содержать наименование предмета и объем часов, отводимых на его изучение. Результат экзамена студента по некоторому предмету должен содержать дату сдачи и оценку.

Опишем независимые **сущности** БД на языке инфологического моделирования:

СТУДЕНТЫ (Код студента, *Номер зачетки*, *ФИО*, *Факультет*, *Курс*, *Группа*)

ПРЕДМЕТЫ (Код предмета, *Наименование*, *Часы*)

Определим **ключи** независимых сущностей. В сущности **СТУДЕНТЫ** ключом мог бы стать атрибут *Номер зачетки*, а в сущности **ПРЕДМЕТЫ** – атрибут *Наименование*. Однако надежнее ввести для этой цели в число атрибутов этих сущностей специальные атрибуты с уникальными значениями: **Код студента** и **Код предмета**. В соответствующих таблицах базы данных Access этим атрибутам будут сопоставлены поля типа **Счетчик**.

Установим **связи** между сущностями. Один студент (один экземпляр сущности **СТУДЕНТЫ**) может сдавать несколько экзаменов по разным предметам, поэтому связь между сущностями **СТУДЕНТЫ** и **ЭКЗАМЕНЫ** – “один-ко-многим”. Точно так же один предмет может сдаваться на нескольких экзаменах разными студентами, поэтому связь между сущностями **ПРЕДМЕТЫ** и **ЭКЗАМЕНЫ** – “один-ко-многим”.

Проведем **формализацию связей**. **Класс принадлежности** со стороны сущности **ЭКЗАМЕНЫ** – обязательный (факт сдачи экзамена обязательно

должен быть связан с соответствующим студентом из сущности **СТУДЕНТЫ**), поэтому ключ **Код студента** сущности **СТУДЕНТЫ** должен присутствовать как внешний (FK) в сущности **ЭКЗАМЕНЫ**. Класс принадлежности со стороны сущности **ПРЕДМЕТЫ** также обязательный, поэтому ключ **Код предмета** должен присутствовать как внешний (FK) в сущности **ЭКЗАМЕНЫ**. Таким образом, сущность-ассоциация **ЭКЗАМЕНЫ** на языке инфологического моделирования должна быть следующей:

ЭКЗАМЕНЫ (Код студента (FK), Код предмета (FK), Оценка, Дата сдачи)

Ключом этой сущности является составной ключ **Код студента+Код предмета**, обеспечивающий уникальность экземпляров сущности. Заметим, что если бы предметная область БД предполагала возможность неоднократной сдачи экзамена по одной и той же дисциплине (например, в разных семестрах), то для реализации такой возможности ключом сущности **ЭКЗАМЕНЫ** следовало бы сделать составной ключ **Код студента+Код предмета+Дата сдачи**. Будем считать, что в нашем случае каждый предмет может сдаваться лишь однократно.

Таким образом, ER-модель разрабатываемой БД имеет вид, показанный на рис. 8.1.

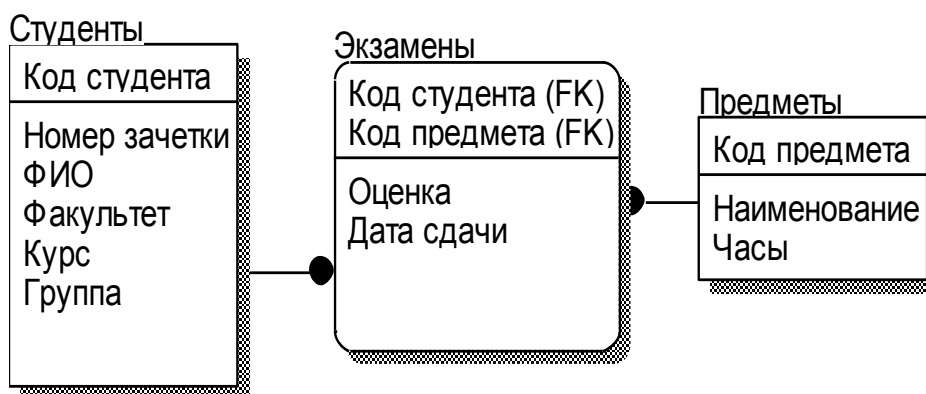


Рис. 8.1. ER-модель БД "Учебный процесс"

8.4. Варианты индивидуальных заданий

№	База данных	Таблицы	Примечание	Запросы
1	Банк	БАНК (код банка, название, адрес) КЛИЕНТ (код клиента, ФИО, телефон клиента) КРЕДИТ (код банка, код клиента, дата взятия кредита, дата выплаты кредита, размер кредита, отметка о выплате)	Банк имеет много клиентов и один клиент может брать кредиты в разных банках.	1. Выбрать банки, которые не имеют клиентов. 2. Выбрать клиентов, которые имеют задолженность более 1 месяца.

№	База данных	Таблицы	Примечание	Запросы
2	ВУЗ	ПРЕПОДАВАТЕЛЬ (код преподавателя, ФИО, адрес, должность, ученая степень) ДИСЦИПЛИНА (код дисциплины, название, количество часов) ЗАНЯТИЯ (код преподавателя, код дисциплины, дата занятия, номер пары, номер аудитории)	Один преподаватель может вести несколько дисциплин и одна дисциплина может вестись несколькими преподавателями.	1. Выбрать преподавателей, которые были "без работы" в заданный период времени. 2. Вывести список дисциплин с количеством преподавателей, ведущих по ним занятия.
3	Склад	ТОВАР (код товара, название товара, единица измерения, стоимость единицы товара, количество на складе) ПОСТАВЩИК (код поставщика, наименование поставщика) ПОСТАВКА (код товара, код поставщика, дата поставки, объем товара в поставке)	Один и тот же товар может поставляться несколькими поставщиками и один и тот же поставщик может поставлять несколько видов товаров.	1. Выбрать поставщиков, которые поставляют заданный товар. 2. Определить самую крупную поставку заданного товара и соответствующего поставщика.
4	Производство	ИЗДЕЛИЕ (код изделия, название изделия, объем выпуска) МАТЕРИАЛ (код материала, название, единица измерения материала, цена за единицу) СПЕЦИФИКАЦИЯ МАТЕРИАЛОВ (код изделия, код материала, количество материала в спецификации изделия, дата установления спецификации)	Одно изделие может содержать много разных материалов и один и тот же материал может входить в состав разных изделий.	1. Выбрать изделия, в состав которых входит заданный материал. 2. Вывести список изделий, которые производились в заданном году, и общую стоимость входящих в них материалов.

№	База данных	Таблицы	Примечание	Запросы
5	Сеть магазинов	ВЛАДЕЛЕЦ (код владельца, ФИО, адрес, телефон) МАГАЗИН (код магазина, название, район расположения магазина, профиль) ВЛАДЕНИЕ (код владельца, код магазина, дата регистрации владения)	Один и тот же магазин может иметь несколько владельцев и один и тот же владелец может иметь в собственности много магазинов.	1. Выбрать владельцев, имеющих собственность в заданном районе. 2. Вывести список профилей магазинов, которыми владеет заданный владелец, с количеством магазинов каждого профиля.
6	Автомас-терские	АВТОМЕХАНИК (код автомеханика, ФИО) АВТОМОБИЛЬ (код автомобиля, регистрационный знак, марка, год выпуска, ФИО автовладельца) НАРЯД (код автомобиля, код автомеханика, дата обслуживания)	Один и тот же автомобиль может обслуживаться разными механиками и один и тот же механик может обслуживать разные автомобили.	1. Выбрать механиков, которые работают с автомобилями заданной марки. 2. Выбрать автомобили, которые обслуживает заданный механик, с количеством их обслуживаний.
7	Деканат	ГРУППА (код группы, номер группы, количество студентов) ДИСЦИПЛИНА (код дисциплины, название дисциплины, количество часов) СЕССИЯ (код группы, код дисциплины, дата контроля, ФИО преподавателя, вид контроля)	Одна группа изучает несколько дисциплин и одна дисциплина может преподаваться нескольким группам.	1. Определить преподавателя, который в сессию принимает экзамены у заданной группы по заданной дисциплине. 2. Вывести список дисциплин с количеством сдач зачетов по ним.

№	База данных	Таблицы	Примечание	Запросы
8	Организация	ПРОЕКТ (код проекта, название, дата начала, дата окончания) СОТРУДНИК (код сотрудника, ФИО сотрудника, должность) УЧАСТИЕ (код проекта, код сотрудника, обязанности участника)	Один и тот же проект может выполняться несколькими сотрудниками и один и тот же сотрудник может участвовать в нескольких проектах.	1. Вывести список проектов в заданном году с количеством участвующих в них сотрудников. 2. Определить общее количество проектов, в которых участвовал заданный сотрудник.
9	Поликлиника	ПАЦИЕНТ (код пациента, ФИО, дата рождения, социальный статус) ВРАЧ (код врача, ФИО, должность, специализация) ЛЕЧЕНИЕ (код пациента, код врача, заболевание, дата начала лечения, срок потери трудоспособности)	Один и тот же пациент может лечиться у нескольких врачей и один и тот же врач может лечить несколько пациентов.	1. Вывести список пациентов с суммарными сроками потери трудоспособности. 2. Вывести список тех врачей, которые работают с пенсионерами.
10	Спорт	ВИД СПОРТА (код вида спорта, название, единица результатов, мировой рекорд и его дата) СПОРТСМЕН (код спортсмена, ФИО спортсмена, год рождения) РЕЗУЛЬТАТ (код вида спорта, код спортсмена, дата проведения, место проведения, показанный результат)	Один и тот же спортсмен может заниматься разными видами спорта	1. Вывести список видов спорта с количеством занимающихся ими спортсменов. 2. Вывести список спортсменов, превысивших мировой рекорд по виду спорта "бег".

№	База данных	Таблицы	Примечание	Запросы
11	Поставки продукции	ПРЕДПРИЯТИЕ (код предприятия, наименование, дата регистрации, вид собственности, число работников) ПРОДУКЦИЯ (код продукции, название продукции, единица измерения) ПОСТАВКА (код предприятия, код продукции, дата поставки, объем, закупочная цена)	Одно и то же предприятие может выпускать разные виды продукции и одна и та же продукция может поставляться разными предприятиями	1. Вывести перечень продукции, поставленной в заданном году с указанием общей стоимости поставок. 2. Вывести список предприятий, поставляющих заданную продукцию по закупочной цене, ниже заданной.
12	Городской транспорт	ВИД ТРАНСПОРТА (код вида транспорта, название, средняя скорость движения, количество машин в парке, стоимость проезда) ПУТЬ (код пути, начальный пункт, конечный пункт, расстояние) МАРШРУТ (код вида транспорта, код пути, количество остановок на маршруте, количество машин на маршруте)	Один и тот же вид транспорта может на разных маршрутах использовать разные пути следования и один и тот же путь может использоваться разными видами транспорта.	1. Определить путь с максимальным количеством видов транспорта. 2. Вывести маршруты трамваев в порядке убывания протяженности путей.
13	Население стран	СТРАНА (код страны, название, регион, столица, площадь территории) НАЦИОНАЛЬНОСТЬ (код национальности, название, язык, общая численность) НАСЕЛЕНИЕ (код страны, код национальности, количество людей данной национальности)	В одной и той же стране могут проживать люди различных национальностей	1. Вывести список стран с общей численностью их населения. 2. Определить столицу той страны, где проживает более всего представителей заданной национальности.

№	База данных	Таблицы	Примечание	Запросы
14	Домо-управление	КВАРТИРА (код квартиры, номер дома, номер квартиры, число жильцов, площадь) ВИД УСЛУГ (код вида услуги, наименование, тариф за единицу площади, тариф за одного жильца) ПОТРЕБЛЕНИЕ УСЛУГ (код квартиры, код вида услуги)	Каждая квартира потребляет хотя бы один вид услуг.	1. Вывести список квартир с общими стоимостями всех потребляемых услуг. 2. Вывести список видов услуг со стоимостями их потребления в заданной квартире.
15	Аэропорт	САМОЛЕТ (код самолета, бортовой номер, тип самолета, число мест, скорость полета) МАРШРУТ (код маршрута, пункт вылета, пункт назначения, расстояние) РЕЙС (код самолета, код маршрута, дата и время вылета, дата и время прилета, количество проданных билетов)	Один самолет может летать на разных маршрутах и по одному маршруту могут летать разные самолеты.	1. Определить максимальное время полета для заданного самолета по заданному маршруту. 2. Определить тип самолета, который чаще всего летает по заданному маршруту.
16	Компьютеры	ФИРМА (код фирмы, название, город размещения) КОНФИГУРАЦИЯ ПК (код конфигурации, название, тип процессора, тактовая частота, объем ОЗУ, объем жесткого диска, дата выпуска) РЫНОЧНОЕ ПРЕДЛОЖЕНИЕ (код фирмы, код конфигурации, объем партии, цена партии)	Одна и та же конфигурация ПК может изготавливаться разными фирмами и одна и та же фирма может собирать ПК разной конфигурации.	1. Определить фирму, которая представляет самую новую конфигурацию на базе заданного типа процессора. 2. Определить конфигурацию с наибольшей тактовой частотой, которая выпускается в заданном городе.

№	База данных	Таблицы	Примечание	Запросы
17	Информация для зрителей	ФИЛЬМЫ (код фильма, название, год выпуска, категория проката) КИНОТЕАТРЫ (код кинотеатра, наименование, адрес, вместимость) КИНОПРОКАТ (код фильма, код кинотеатра, дата начала показа, дата окончания показа)	Один и тот же фильм может идти в разных кинотеатрах.	1. Определить количество фильмов за заданный период в заданном кинотеатре. 2. Определить кинотеатр наибольшей вместимости, в котором демонстрируется заданный фильм.
18	Зоопарк	ВИД ЖИВОТНОГО (код вида животного, название, суточное потребление корма, семейство, континент обитания) ПОМЕЩЕНИЕ (код помещения, наименование, наличие водоема, наличие отопления) РАЗМЕЩЕНИЕ (код вида животного, код помещения, количество животных)	Один и тот же вид животного может в зоопарке находиться в разных помещениях и в одном помещении может находиться несколько видов животных.	1. Определить суточное потребление корма всеми обитателями заданного помещения. 2. Определить общую численность представителей каждого вида животных.
19	Шахматы	ШАХМАТИСТ (код спортсмена, ФИО, дата рождения, страна, спортивный разряд, рейтинг) ТУРНИР (код турнира, страна, город, дата проведения, уровень турнира) УЧАСТИЕ (код спортсмена, код турнира, стартовый номер спортсмена в данном турнире, занятое место)	Один шахматист может участвовать в разных турнирах.	1. Определить турнир с самым высоким средним рейтингом участников. 2. Вывести список шахматистов, выигравших хотя бы один турнир в течение заданного года.
20	Судоходство	КОРАБЛЬ (код корабля, название, водоизмещение, ФИО капитана) ПОРТ (код порта, название, страна) ПОСЕЩЕНИЕ ПОРТА (код корабля, код порта, дата прибытия, дата убытия, номер причала, цель посещения).	Один корабль может посещать несколько портов.	1. Вывести список кораблей, находившихся в заданном порту в заданный день. 2. Определить страны, в которые приходят корабли с целью “туризм”.

№	База данных	Таблицы	Примечание	Запросы
21	Научные конференции	УЧЕНЫЙ (код ученого, фамилия и имя, ученая степень, организация, страна) КОНФЕРЕНЦИЯ (код конференции, название, дата проведения, место проведения) УЧАСТИЕ (код ученого, код конференции, тип участия, тема доклада (сообщения), публикация (да/нет)).	Один ученый может участвовать в разных конференциях и с разными типами участия. Тип участия: доклад, сообщение, стендовый доклад, оргкомитет.	1. Вывести названия конференций, материалы которых не опубликованы. 2. Определить количество докторов наук, участвовавших в каждой конференции.
22	Программные продукты	ПРОГРАММНЫЙ ПРОДУКТ (код программного продукта, название, версия, фирма, дата выпуска, прикладная область, стоимость лицензии) ПОЛЬЗОВАТЕЛЬ (код пользователя, наименование, регион, сфера применения) ИНСТАЛЛЯЦИЯ (код программного продукта, код пользователя, дата инсталляции, количество лицензий при инсталляции)	Программный продукт может устанавливаться разными пользователями. Пользователь может установить разные продукты. Прикладная область: делопроизводство, коммерция, бизнес и т.д. Тип программного продукта: ОС, сервер приложений, СУБД, Web-сервер и т.п.	1. Вывести данные об инсталляциях по программным продуктам. 2. Определить затраты на инсталляции программного обеспечения заданного пользователя в заданном году.

№	База данных	Таблицы	Примечание	Запросы
23	Отдел кадров	СОТРУДНИК (код сотрудника, ФИО, дата рождения, образование, телефон) ПОДРАЗДЕЛЕНИЕ (код подразделения, наименование, количество штатных единиц, фонд заработной платы) ЗАЧИСЛЕНИЕ (код сотрудника, код подразделения, должность, дата зачисления, оклад).	Один и тот же сотрудник может числиться в разных подразделениях	1. Вывести список сотрудников, которые занимают должность “инженер”, не имея высшего образования. 2. Вывести список подразделений с суммарным окладом сотрудников.
24	Добыча полезных ископаемых	ПОЛЕЗНОЕ ИСКОПАЕМОЕ (код полезного ископаемого, наименование, единица измерения, годовая потребность, цена за единицу) МЕСТОРОЖДЕНИЕ (код месторождения, название месторождения, запасы) ВЫВОЗ (код полезного ископаемого, код месторождения, объем вывоза)	В одном и том же месторождении могут добываться различные ископаемые, а одно и то же ископаемое может добываться на разных месторождениях	1. Вывести список ископаемых с общим объемом вывоза по разным месторождениям. 2. Вывести список месторождений, из которых вывозится заданное ископаемое.
25	Автотранспортное предприятие	АВТОМОБИЛЬ (код автомобиля, регистрационный знак, марка, расход топлива на единицу пробега) ВОДИТЕЛЬ (код водителя, ФИО, дата рождения, оклад) РЕЙС (код автомобиля, код водителя, дата выезда, дата прибытия, расстояние)	Один и тот же автомобиль может использоваться разными водителями и один водитель может использовать разные автомобили.	1. Определить автомобиль с наименьшим расходом горючего за все рейсы с заданным водителем. 2. Вывести данные обо всех рейсах с участием заданного автомобиля.

№	База данных	Таблицы	Примечание	Запросы
26	Театр	АКТЕР (код актера, ФИО, звание, амплуа, пол) РОЛЬ (код роли, название пьесы, действующее лицо, тип роли) ПОСТАНОВКА НА РОЛЬ (код актера, код роли, режиссер, дата назначения на роль)	На одну и ту же роль могут назначаться разные актеры. <u>Амплуа</u> : герой-любовник, инженеру, злодей т.д. <u>Тип роли</u> : главная, вторая, эпизод.	1. Вывести список актеров с количеством сыгранных ролей в заданном году. 2. Вывести пьесы, в которых исполнителем главной роли является заданный актер.
27	Справочная аптек	ЛЕКАРСТВО (код лекарства, название, тип, показания к использованию, производитель) АПТЕКА (код аптеки, название аптеки, район, телефон) НАЛИЧИЕ (код лекарства, код аптеки, срок годности, количество, цена)	Одно и то же лекарство продается в разных аптеках. <u>Тип</u> : таблетки, микстура, мазь и т.д.	1. Вывести ассортимент лекарств с наименьшей и наибольшей ценой по всем аптекам в одном запросе 2. Определить аптеку с наибольшим количеством заданного лекарства.
28	Кулинария	БЛЮДО (код блюда, название блюда, категория, вес порции) ПРОДУКТ (код продукта, название продукта, категория, калорийность, единица измерения, цена за единицу) СОСТАВ (код блюда, код продукта, количество продукта)	Блюдо состоит из разных продуктов; один и тот же продукт может входить в состав разных блюд. <u>Категория блюда</u> : первое, гарнир, и т.д. <u>Категория продукта</u> : мясо, мучное и т.д.	1. Вывести список блюд с их ценами, включающими 100% надбавку за приготовление и сервис. 2. Вывести состав и атрибуты продуктов в заданном блюде.

Примечания к заданиям

1. В описании структуры таблиц слово «Код» в наименовании ключа означает порядковый номер записи, не имеющий содержательного значения (поле типа «счетчик»), а слово «Номер» – содержательный ключ записи (обычно текстового типа), формируемый по некоторым правилам соответствующей организации. Например, «номер договора», «номер заказа»

и т.п. могут иметь шаблон вида «nnnn/yy», где yy – последние цифры года, а nnnn – порядковый номер договора, заказа и т.п. в этом году.

2. Слово «заданный» в формулировках запросов означает, что значения соответствующих атрибутов должны вводиться при выполнении запроса (параметрический запрос).

3. В формах в режиме формы, в запросах в режиме таблицы и в отчетах в режиме представления (документах, предназначенных для пользователей БД), не должны фигурировать коды записей (поля типа «счетчик»). Заголовки и другие надписи должны быть содержательными и понятными пользователю.

8.5. Содержание отчета

1. Тема и цель работы.
2. Индивидуальное задание на разработку БД.
3. Формализация и уточнение задания.
4. ER-модель БД.
5. Скриншоты всех таблиц в режиме конструктора и в режиме таблицы.
6. Скриншот схемы БД.
7. Скриншоты всех форм БД.
8. Скриншоты всех запросов в форме QBE и на языке SQL и результатов их выполнения.
9. Скриншоты всех отчетов в режиме конструктора и в режиме представления.
10. Меню всех уровней кнопочной формы.
11. БД на магнитном носителе.

8.6. Контрольные вопросы

1. Основные понятия и определения баз данных.
2. Основные компоненты и принципы построения БД реляционного типа.
3. Инфологическое проектирование и модель “Сущность-связь”.
4. Создание реляционной БД в среде СУБД Access.
5. Виды форм в СУБД Access.
6. Виды запросов и способы их построения в СУБД Access.
7. Назначение и виды отчетов в СУБД Access.

Лабораторная работа 9

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗЫ ДАННЫХ (индивидуальное задание №2)

9.1. Цель работы

Получение навыков самостоятельного проектирования базы данных, создания БД и ее таблиц в среде Access, построения запросов к БД и оформления результатов их выполнения в виде отчетов.

9.2. Общее задание

1. Выбрать вариант индивидуального задания.
2. Проанализировать предметную область проектируемой БД и уточнить задание, определив данные, которые необходимо хранить в БД.
3. Разработать модель “Сущность-связь” (ER-модель) проектируемой БД: выявить необходимый набор сущностей, определить требуемый набор атрибутов для каждой сущности, определить ключи и связи между сущностями, формализовать связи.
4. Создать новую БД в Access.
5. Создать таблицы, соответствующие сущностям ER-модели БД. Определить типы и свойства полей, создать необходимые подстановки.
6. Создать схему БД, определив связи между таблицами в соответствии со связями между сущностями в ER-модели. Установить для каждой связи свойства обеспечения целостности данных.
7. Создать формы для ввода и редактирования данных в БД и с их помощью ввести записи в БД. Объем данных в таблицах должно обеспечивать выдачу не менее 3-5 записей по каждому запросу задания.
8. Создать заданные запросы с помощью **Конструктора** или **Мастера** запросов. Для многотабличных запросов установить нужный тип параметров объединения таблиц. Просмотреть запросы в режиме SQL. Выполнить запросы и сохранить их в БД.
9. Создать отчеты по всем запросам.
10. Создать кнопочную форму для работы с БД.

9.3. Варианты индивидуальных заданий

Задание №1: БД Ресторана

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Ингредиенты (Код ингредиента, Наименование ингредиента, Поставщик).4. Меню (Код блюда, Наименование блюда, Стоимость, Время приготовления).5. Состав блюд (Код блюда, Код ингредиента, Объем ингредиента).6. Заказы (Номер заказа, Код сотрудника, Дата, Время, ФИО заказчика, Отметка о выполнении).7. Состав заказов (Номер заказа, Код блюда, Количество блюд).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. Список ингредиентов, поставляемых заданным поставщиком.3. Список ингредиентов и их объемов, затраченных на выполнения всех заказов на заданную дату.4. Список блюд, включающих заданный ингредиент.5. Список из 3 наиболее популярных блюд.6. Список заказов на заданную дату, выполненных заданным сотрудником, с подсчетом стоимости каждого заказа.

Задание №2: БД Банка

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Виды вкладов (Код вида вклада, Наименование вида вклада, Минимальный срок вклада, Минимальная сумма вклада, Код валюты, Процентная ставка).4. Валюта (Код валюты, Наименование, Текущий обменный курс в рублях).5. Вкладчики (Код вкладчика, ФИО вкладчика, Адрес, Телефон).6. Вклады (Код вкладчика, Код вида вклада, Дата вклада, Дата возврата, Сумма вклада, Сумма возврата, Отметка о возврате вклада, Код сотрудника).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. Виды валютных вкладов с процентной ставкой больше 5%.3. Сумма всех вкладов в рублях по текущему курсу и средний вклад.4. Список вкладов с заданной валютой.5. Список из 3 наиболее популярных видов вкладов.6. Список рублевых вкладов с суммой вклада, большей заданной.

Задание №3: БД Больницы.

Таблицы:	<ol style="list-style-type: none"> 1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности). 2. Должности (Код должности, Наименование должности, Оклад). 3. Лекарства (Код лекарства, Наименование, Фасовка, Стоимость). 4. Болезни (Код болезни, Наименование). 5. Лечение (Код болезни, Код лекарства, Продолжительность приема, Ежедневная доза). 6. Истории болезни (Номер истории, Дата регистрации, Код болезни, Код лечащего врача, ФИО пациента, Дата рождения, Пол).
Запросы:	<ol style="list-style-type: none"> 1. Список сотрудников заданного возраста с указанием всех реквизитов. 2. Список из 3 наиболее часто встречающихся болезней. 3. Список лекарств, используемых при лечении заданной болезни. 4. Список пациентов старше 50 лет, переболевших заданной болезнью. 5. Список пациентов мужского пола, пролечившихся у заданного врача. 6. Список лекарств и их объемов, использованных при лечении пациентов заданного врача.

Задание №4: БД Гостиницы.

Таблицы:	<ol style="list-style-type: none"> 1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности). 2. Должности (Код должности, Наименование должности, Оклад). 3. Номера (Номер, Категория, Вместимость, Суточная стоимость). 4. Виды услуг (Код услуги, Наименование, Стоимость). 5. Заселения (Номер регистрации, Код сотрудника, Дата заселения, Номер, ФИО клиента, Паспортные данные, Дата выезда). 6. Предоставленные услуги (Номер регистрации заселения, Код услуги, Дата предоставления).
Запросы:	<ol style="list-style-type: none"> 1. Список сотрудников заданного возраста с указанием всех реквизитов. 2. Список номеров гостиницы, суточная стоимость которых превышает заданную. 3. Список клиентов, проживавших в гостинице в заданный период. 4. Список номеров гостиницы, занятых на заданную дату. 5. Список регистраций заселения с указанием общих сумм счетов, выставленных клиентам. 6. Список из 3 наиболее популярных категорий номеров, занимаемых клиентами.

Задание №5: БД Отдела внутренних дел.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности, Код звания).2. Должности (Код должности, Наименование должности, Оклад, Код соответствующего звания).3. Звания (Код звания, Наименование, Надбавка).4. Виды преступлений (Код вида преступления, Наименование, Номер статьи, Наказание).5. Преступники (Код преступника, ФИО, Дата рождения, Пол, Адрес).6. Дела (Номер дела, Код вида преступления, Код следователя, Данные о пострадавших, Дата открытия дела, Дата закрытия дела).7. Состав преступников в делах (Номер дела, Код преступника).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. Список сотрудников, звания которых не соответствуют занимаемым должностям.3. Список незакрытых дел, расследуемых заданным следователем.4. Список преступников-рецидивистов с указанием количества совершенных преступлений.5. Список следователей с указанием количества закрытых дел.6. Список из 3 наиболее частых видов преступлений.

Задание №6: БД Авиакомпаний.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Типы самолётов (Код типа, Наименование, Вместимость).4. Самолёты (Код самолёта, Бортовой номер, Код типа, Дата выпуска, Налётано часов, Дата последнего ремонта).5. Экипажи (Код экипажа, Номер экипажа).6. Состав экипажей (Код экипажа, Код сотрудника).7. Расписание рейсов (Номер рейса, Откуда, Куда, Время вылета).8. Рейсы (Номер рейса, Код самолёта, Дата рейса, Код экипажа).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. Список самолетов, у которых дата последнего ремонта меньше заданной.3. Список типов самолётов с указанием их количества.4. Состав экипажа, наиболее часто летающего на самолетах заданного типа.5. Список рейсов по заданному маршруту, выполненных в заданный период.6. Список из 3 наиболее используемых типов самолетов.

Задание №7: БД Видео-проката.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Жанры (Код жанра, Наименование жанра).4. DVD (Код DVD, Наименование фильма, Год создания, Страна, Главный актёр, Код жанра, Количество экземпляров, Цена).5. Прокат (Номер проката, Код сотрудника, ФИО клиента, Адрес, Телефон, Дата выдачи, Планируемая дата возврата, Отметка о возврате).6. Состав DVD в прокате (Номер проката, Код DVD).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. Список DVD на руках с указанием количества.3. Список не востребовавшихся DVD.4. Список DVD с фильмами за заданный год.5. Список из 3 наиболее популярных DVD.6. Перечень сотрудников с невозвращенными из проката DVD с указанием количества DVD и их общей стоимости.

Задание №8: БД Библиотеки.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Издательства (Код издательства, Наименование, Город, Адрес).4. Жанры (Код жанра, Наименование).5. Книги (Код книги, Наименование, Автор, Код издательства, Год издания, Код жанра, Количество экземпляров).6. Читатели (Код читателя, ФИО, Адрес, Телефон).7. Выдачи (Номер выдачи, Код читателя, Код сотрудника, Дата выдачи, Планируемая дата возврата, Отметка о возврате).8. Состав выдач (Номер выдачи, Код книги).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданной должности с указанием всех реквизитов.2. Список книг библиотеки заданного автора с указанием всех реквизитов.3. Перечень читателей, у которых на руках книги с просроченной датой возврата, с перечнями этих книг.4. Список книг на руках с указанием количества экземпляров.5. Список из 3 наиболее популярных книг.6. Список сотрудников с указанием количества выданных книг и количества возвращенных книг.

Задание №9: БД Радиостанции.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Исполнители (Код исполнителя, Наименование).4. Жанры (Код жанра, Наименование).5. Записи (Код записи, Наименование, Код жанра, Код исполнителя, Год записи).6. Сетка вещания (День недели, Время суток, Код сотрудника, Код записи).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых превышает заданный, с указанием всех реквизитов.2. Список записей с заданным годом записи.3. Список из 3 записей, наиболее часто звучащих в эфире в заданный день недели.4. Список сотрудников, работающих в эфире в заданный день недели.5. ФИО сотрудника, работающего чаще всех при данной сетке вещания.6. Список записей наиболее востребованного жанра.

Задание №10: БД Таксопарка.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Модели автомобилей (Код модели, Наименование, Мощность, Объем двигателя).4. Тарифы (Код тарифа, Наименование, Цена за километр пробега).5. Автомобили (Код автомобиля, Регистрационный номер, Код модели, Год выпуска, Код сотрудника-шофёра).6. Вызовы (Код автомобиля, Дата, Время, Телефон заказчика, Расстояние, Код тарифа, Код сотрудника-оператора).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых превышает заданный, с указанием всех реквизитов.2. Автопарк с указанием всех реквизитов автомобилей.3. Список автомобилей заданной модели.4. Список вызовов за заданную дату с вычислением стоимостей отдельных вызовов и их общей суммы.5. Список сотрудников-операторов с указанием количества оформленных вызовов.6. Перекрестный запрос, вычисляющий количество вызовов, принятых сотрудниками-операторами по каждому конкретному автомобилю.

Задание №11: БД Туристического агентства.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Отели (Код отеля, Наименование, Страна, Город, Адрес, Телефон, Количество звёзд, Контактное лицо).4. Дополнительные услуги (Код услуги, Наименование, Цена).5. Клиенты (Код клиента, ФИО, Дата рождения, Пол, Адрес, Телефон).6. Путёвки (Номер путевки, Код отеля, Код клиента, Код сотрудника, Дата начала, Дата окончания, Отметка о бронировании, Отметка об оплате).7. Состав услуг по путевкам (Код путевки, Код услуги, Дата предоставления).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых превышает заданный, с указанием всех реквизитов.2. Список путёвок с указанием всех реквизитов.3. Список сотрудников с указанием количества проданных путевок.4. Список из 3 наиболее популярных стран.5. Список отелей с указанием количества проданных путевок.6. Список забронированных, но неоплаченных путёвок.

Задание №12: БД Страховой компании.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Риски (Код риска, Наименование).4. Виды полисов (Код вида полиса, Наименование).5. Риски по видам полисов (Код вида полиса, Код риска).6. Клиенты (Код клиента, ФИО, Дата рождения, Пол, Адрес, Телефон).7. Полисы (Номер полиса, Код вида полиса, Код клиента, Код сотрудника, Дата начала, Дата окончания, Стоимость, Сумма выплаты, Отметка о выплате).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых превышает заданный, с указанием всех реквизитов.2. Список действующих полисов.3. Список оконченных полисов, по которым производились выплаты.4. Список сотрудников с указанием общего количества и стоимости оформленных полисов.5. Список клиентов с указанием номеров и суммы выплаченных полисов.6. Список из 3 наиболее популярных видов полисов.

Задание №13: БД Брачного агентства.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Знаки зодиака (Код знака, Наименование).4. Национальности (Код национальности, Наименование).5. Виды услуг (Код вида услуги, Наименование, Цена).6. Клиенты (Код клиента, ФИО, Пол, Дата рождения, Рост, Вес, Количество детей, Код знака, Код национальности, Адрес, Телефон).7. Оказанные услуги (Код клиента, Код вида услуги, Дата оказания услуги, Код сотрудника).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых превышает заданный, с указанием всех реквизитов.2. Список сотрудников с указанием количества оказанных услуг.3. Список клиентов с указанием всех реквизитов.4. Список клиентов заданной национальности, родившихся под заданным знаком зодиака.5. Список национальностей с указанием соответствующего количества клиентов.6. Список из 3 наиболее популярных видов услуг.

Задание №14: БД Сервис-центра.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад)3. Запчасти (Код запчасти, Наименование, Цена).4. Ремонтируемые модели (Код модели, Наименование, Тип изделия, Производитель).5. Виды неисправностей (Код вида, Наименование неисправности, Код модели, Цена работы).6. Наборы запчастей (Код вида неисправности, Код запчасти, Количество).7. Обслуживаемые магазины (Код магазина, Наименование, Адрес, Телефон).8. Заказы (Номер заказа, Дата заказа, Серийный номер изделия, ФИО заказчика, Код вида неисправности, Код магазина, Код сотрудника, Отметка о гарантии).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. Список заказов с указанием всех реквизитов и общей стоимости с учетом цены работы и стоимости запчастей.3. Список из 3 наиболее частых видов неисправностей.4. Наименование магазина, в котором приобретено наибольшее количество ремонтируемых изделий.5. Список из 3 наиболее надежных моделей.6. Магазины, в которых были куплены ремонтировавшиеся изделия заданной модели..

Задание №15: БД Школы.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования).3. Ученики (Код ученика, ФИО, Дата рождения, Пол, Адрес, Код класса).4. Классы (Код класса, Год обучения, Индекс класса, Код сотрудника-классного руководителя, Код вида).5. Виды классов (Код вида, Наименование).6. Предметы (Код предмета, Наименование, Код сотрудника-учителя).7. Расписание (Код класса, Код предмета, День недели, Номер урока).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых превышает заданную величину, с указанием всех реквизитов.2. Список учеников заданного возраста.3. Расписание занятий с указанием всех реквизитов, упорядоченное по дням недели и номерам уроков.4. Список классов заданного года обучения с указанием всех реквизитов.5. Количество уроков в неделю каждого учителя по соответствующим предметам.6. Список из 3 наиболее загруженных учителей.

Задание №16: БД Транспортной компании.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Модели автомобилей (Код модели, Наименование, Грузоподъемность).4. Грузы (Код груза, Наименование груза).5. Автомобили (Код автомобиля, Регистрационный номер, Код модели, Код сотрудника-механика).6. Рейсы (Код автомобиля, Код сотрудника-водителя, Заказчик, Дата отправления, Дата прибытия, Код груза, Вес груза, Цена, Отметка об оплате).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых меньше заданного, с указанием всех реквизитов.2. Автопарк с указанием всех реквизитов автомобилей.3. Суммарный вес грузов, перевезенных заданным автомобилем.4. Список рейсов, выполненных с перегрузкой автомобиля.5. Список рейсов, выполненных заданным водителем, с указанием веса груза и цены.6. Список из 3 наиболее часто перевозимых грузов.7. Перекрестный запрос, вычисляющий количество рейсов по грузам и автомобилям.

Задание №17: БД Проката автомобилей.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Модели автомобилей (Код модели, Наименование, Мощность).4. Автомобили (Код автомобиля, Регистрационный номер, Код модели, Год выпуска, Пробег, Цена автомобиля, Цена дня проката).5. Клиенты (Код клиента, ФИО, Пол, Дата рождения, Адрес, Телефон, Паспортные данные).6. Прокат (Код автомобиля, Код клиента, Дата выдачи, Планируемая дата возврата, Отметка о возврате, Код сотрудника).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых больше заданного, с указанием всех реквизитов.2. Автопарк с указанием всех реквизитов автомобилей.3. Список автомобилей, находящихся в прокате.4. Список невозвращенных автомобилей с указанием всех реквизитов проката и автомобиля.5. Список возвращенных автомобилей с указанием всех реквизитов автомобиля и стоимости проката.6. Список из 3 наиболее популярных моделей автомобилей.7. Распределение количества прокатов по сотрудникам и автомобилям.

Задание №18: БД Оптового склада.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования).3. Товары (Код товара, Наименование, Единица отпуска товара).4. Поставщики (Код поставщика, Наименование, Адрес, Телефон).5. Поставляемые товары (Код поставщика, Код товара).6. Заказчики (Код заказчика, Наименование, Адрес, Телефон).7. Продажи (Номер накладной, Код заказчика, Код сотрудника, Дата выдачи).8. Состав продаж (Номер накладной, Код товара, Объем, Цена).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. Список заказчиков с указанием количества продаж за заданный период.3. Список товаров, выданных по накладной с заданным номером, с указанием всех реквизитов продажи и товаров в накладной.4. Список сотрудников с указанием количества продаж и общей цены проданных товаров.5. Список из 3 поставщиков, чьи товары пользуются наибольшим спросом.6. Распределение количества продаж по сотрудникам и товарам.

Задание №19: БД картинной галереи

Таблицы:	<ol style="list-style-type: none">1. Сотрудники галереи (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Залы галереи (Номер зала, Площадь зала, Код ответственного за зал сотрудника).4. Картины (Код картины, Наименование, Номер зала, Автор, Страховая стоимость, Дата поступления).5. Экскурсии (Код экскурсии, Тема экскурсии, Время проведения, Продолжительность, Цена билета, Код экскурсовода).6. Содержание экскурсий (Код экскурсии, Код картины).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых меньше заданного, с указанием всех реквизитов.2. Список сотрудников заданной должности.3. Список залов с указанием количества картин в них.4. Каталог картин галереи, упорядоченный по номерам залов, авторам и наименованиям.5. Список экскурсий с указанием количества посещаемых залов.6. Список из 3 картин, осматриваемых в наибольшем количестве экскурсий.

Задание №20: БД Спортивного клуба

Таблицы	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Помещения (Код помещения, Номер помещения, Назначение, Площадь).4. Клиенты (Код клиента, ФИО, Адрес, Телефон).5. Абонементы (Код абонементов, Название, Цена).6. Используемые помещения (Код абонементов, Код помещения).7. Продажи абонементов (Номер продажи, Код абонементов, Код клиента, Дата продажи, Дата начала, Дата окончания, Код сотрудника).
Запросы	<ol style="list-style-type: none">1. Получить список из 3 сотрудников, которые оформили наибольшее число абонементов.2. Вычислить количество абонементов по помещениям и общее количество абонементов.3. Определить название наиболее востребованного абонементов.4. Определить количество продаж абонементов за заданный период времени.5. Определить суммарную стоимость абонементов, проданных за заданный период времени.6. Получить распределение числа проданных абонементов по сотрудникам и абонементов.

Задание №21: БД Рекламного агентства наружной рекламы

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Рекламные конструкции (Код конструкции, Номер конструкции, Адрес, Стоимость дня аренды одной стороны конструкции).4. Рекламные места (Код места, Код конструкции, Сторона конструкции).5. Рекламодатели (Код рекламодателя, Наименование рекламодателя, Адрес, Телефон).6. Договоры аренды (Номер договора, Дата начала, Дата окончания, Код рекламодателя, Код сотрудника, Отметка об оплате).7. Списки арендуемых мест (Код договора, Код рекламного места, Описание рекламы).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. Список рекламных мест с указанием всех реквизитов.3. Список договоров с указанием всех реквизитов и общей стоимости аренды.4. Список арендуемых по заданному договору рекламных мест с указанием всех реквизитов.5. Список неоплаченных договоров с указанием всех реквизитов.6. Список из 3 наиболее популярных рекламных конструкций.

Задание №22: БД Компьютерной фирмы.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Виды комплектующих (Код вида, Наименование).4. Комплектующие (Код комплектующего, Код вида, Марка, Цена).5. Заказчики (Код заказчика, ФИО, Телефон).6. Услуги (Код услуги, Наименование, Стоимость).7. Заказы (Номер заказа, Код заказчика, Дата заказа, Отметка об оплате, Отметка об исполнении, Код сотрудника).8. Состав комплектующих в заказах (Номер заказа, Код комплектующего, Количество).9. Состав услуг в заказах (Номер заказа, Код услуги).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников, возраст которых меньше заданного, с указанием всех реквизитов.2. Список комплектующих с указанием всех реквизитов.3. Список неоплаченных заказов за заданную дату.4. Список комплектующих конкретного заказа с заданным номером.5. Список услуг с указанием их общего количества в исполненных заказах.6. Список из 3 заказчиков с наибольшим количеством заказов.

Задание №23: БД ГИБДД.

Таблицы:	<ol style="list-style-type: none"> 1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности, Код звания). 2. Должности (Код должности, Наименование должности, Оклад). 3. Звания (Код звания, Наименование, Надбавка). 4. Модели автомобилей (Код модели, Наименование, Мощность двигателя, Рабочий объем двигателя). 5. Владельцы (Код владельца, ФИО, Дата рождения, Адрес, Паспортные данные). 6. Регистрация (Регистрационный знак, Дата регистрации, VIN автомобиля, Код модели, Год выпуска, Номер кузова, Номер двигателя, Цвет, Номер техпаспорта, Код владельца, Код сотрудника). 7. Автомобили в угоне (Регистрационный знак, Дата угона, Дата обращения, Дата нахождения, Код сотрудника).
Запросы:	<ol style="list-style-type: none"> 1. Список сотрудников с указанием всех реквизитов. 2. Список моделей автомобилей с указанием соответствующего количества зарегистрированных машин. 3. Список владельцев угнанных автомобилей. 4. Список регистраций заданным сотрудником за заданный период. 5. Список угнанных автомобилей, находящихся в угоне более 3 месяцев и до сих пор не найденных. 6. Список из 3 наиболее часто угоняемых моделей автомобилей.

Задание №24: БД Кинотеатра.

Таблицы:	<ol style="list-style-type: none"> 1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности). 2. Должности (Код должности, Наименование должности, Оклад). 3. Жанры (Код жанра, Наименование). 4. Фильмы (Код фильма, Наименование, Код жанра, Длительность, Страна-производитель, Возрастные ограничения). 5. Сеансы (Код сеанса, Дата, Время начала, Код фильма, Цена билета). 6. Билеты (Код сеанса, Номер ряда, Номер места, Код сотрудника).
Запросы:	<ol style="list-style-type: none"> 1. Список сотрудников, возраст которых больше заданного, с указанием всех реквизитов. 2. ФИО сотрудника, продавшего наибольшее количество билетов за заданный период. 3. Список фильмов с указанием общего количества просмотревших зрителей. 4. Список проданных билетов на определенный сеанс с заданной датой и временем начала. 5. Список из 3 фильмов с наибольшей посещаемостью. 6. Количество проданных билетов по сеансам на заданную дату.

Задание №25: БД Автосалона.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Дата рождения, Пол, Телефон, Код должности).2. Должности (Код должности, Наименование должности, Оклад).3. Марки автомобилей (Код марки, Наименование, Страна–производитель).4. Дополнительное оборудование (Код дополнительного оборудования, Наименование, Цена).5. Тип кузова (Код типа кузова, Название).6. Автомобили (VIN автомобиля, Код марки, Год выпуска, Код типа кузова, Номер кузова, Номер двигателя, Цвет, Базовая цена).7. Продажная комплектация (VIN автомобиля, Код дополнительного оборудования).8. Продажи (Номер договора продажи, VIN автомобиля, Дата продажи, ФИО покупателя, Адрес, Телефон, Паспортные данные, Код сотрудника).
Запросы:	<ol style="list-style-type: none">1. Список сотрудников заданного возраста с указанием всех реквизитов.2. ФИО сотрудника, продавшего наибольшее количество автомобилей за заданный месяц и год.3. Каталог автомобилей автосалона с указанием всех реквизитов.4. Список продаж за заданный период с указанием всех реквизитов и стоимости с учетом продажной комплектации.5. Список сотрудников с указанием количества проданных ими автомобилей.6. Список из 3 марок автомобилей, лидирующих по объему продаж.

Примечания к заданиям

1. В описании структуры таблиц слово «Код» в наименовании ключа означает порядковый номер записи, не имеющий содержательного значения (поле типа «счетчик»), а слово «Номер» – содержательный ключ записи (обычно текстового типа), формируемый по некоторым правилам соответствующей организации. Например, «номер договора», «номер заказа» и т.п. могут иметь шаблон вида «nnnn/yy», где yy – последние цифры года, а nnnn – порядковый номер договора, заказа и т.п. в этом году.

2. Слово «заданный» в формулировках запросов означает, что значения соответствующих атрибутов должны вводиться при выполнении запроса (параметрический запрос).

3. В формах в режиме формы, в запросах в режиме таблицы и в отчетах в режиме представления (документах, предназначенных для пользователей БД), не должны фигурировать коды записей (поля типа «счетчик»). Заголовки и другие надписи должны быть содержательными и понятными пользователю.

9.4. Содержание отчета

1. Тема и цель работы.
2. Индивидуальное задание на разработку БД.
3. Формализация и уточнение задания.
4. ER-модель БД.
5. Скриншоты всех таблиц в режиме конструктора и в режиме таблицы.
6. Скриншот схемы БД.
7. Скриншоты всех форм БД.
8. Скриншоты всех запросов в форме QBE и на языке SQL и результатов их выполнения.
9. Скриншоты всех отчетов в режиме конструктора и в режиме представления.
10. Меню всех уровней кнопочной формы.
11. БД на магнитном носителе.

9.5. Контрольные вопросы

1. Основные понятия и определения баз данных.
2. Основные компоненты и принципы построения БД реляционного типа.
3. Инфологическое проектирование и модель “Сущность-связь”.
4. Создание реляционной БД в среде СУБД Access.
5. Виды форм в СУБД Access.
6. Виды запросов и способы их построения в СУБД Access.
7. Назначение и виды отчетов в СУБД Access.

Литература

1. Сосновиков Г.К., Шакин В.Н. Основы реляционных баз данных: Учебное пособие / МТУСИ.-М., 2013. – 106 с.
2. Хомоненко А. Д., Цыганков В. М., Мальцев М. Г. Базы данных. Учебник для высших учебных заведений / Под ред. А. Д. Хомоненко. 6-е изд., доп. - СПб.:КОРОНА-Век,2009. - 736 с.

Оглавление

1. ОБЩИЕ СВЕДЕНИЯ О РЕЛЯЦИОННЫХ БАЗАХ ДАННЫХ	- 3 -
2. СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ТАБЛИЦ	- 6 -
3. СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ФОРМ	- 18 -
4. СОЗДАНИЕ И ВЫПОЛНЕНИЕ ЗАПРОСОВ-ВЫБОРОК	- 28 -
5. СОЗДАНИЕ И ВЫПОЛНЕНИЕ ЗАПРОСОВ-ДЕЙСТВИЙ	- 43 -
6. СОЗДАНИЕ ОТЧЕТОВ НА ОСНОВЕ ТАБЛИЦ И ЗАПРОСОВ	- 49 -
7. ИСПОЛЬЗОВАНИЕ МАКРОСОВ В ACCESS	- 60 -
8. ГЛАВНАЯ КНОПОЧНАЯ ФОРМА	- 66 -

Лабораторный практикум

ЛАБОРАТОРНАЯ РАБОТА 1 . СОЗДАНИЕ И РЕДАКТИРОВАНИЕ ТАБЛИЦ, СОРТИРОВКА И ПОИСК ДАННЫХ.....	- 68 -
ЛАБОРАТОРНАЯ РАБОТА 2 . СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ФОРМ	- 73 -
ЛАБОРАТОРНАЯ РАБОТА 3 . РАБОТА С ДАННЫМИ С ИСПОЛЬЗОВАНИЕМ ЗАПРОСОВ. ЗАПРОСЫ-ВЫБОРКИ.....	- 77 -
ЛАБОРАТОРНАЯ РАБОТА 4 . РАБОТА С ДАННЫМИ С ИСПОЛЬЗОВАНИЕМ ЗАПРОСОВ. ЗАПРОСЫ-ДЕЙСТВИЯ.....	- 79 -
ЛАБОРАТОРНАЯ РАБОТА 5 . СОЗДАНИЕ ОТЧЕТОВ НА ОСНОВЕ ТАБЛИЦ И ЗАПРОСОВ.....	- 85 -
ЛАБОРАТОРНАЯ РАБОТА 6 . ИСПОЛЬЗОВАНИЕ МАКРОСОВ В ACCESS	- 91 -
ЛАБОРАТОРНАЯ РАБОТА 7 . ИЗУЧЕНИЕ ДИСПЕТЧЕРА КНОПОЧНЫХ ФОРМ.....	- 96 -
ЛАБОРАТОРНАЯ РАБОТА 8 . ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗЫ ДАННЫХ (ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ №1).....	- 98 -
ЛАБОРАТОРНАЯ РАБОТА 9 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗЫ ДАННЫХ (ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ №2).....	- 111 -