

Федеральное Агентство связи

**Государственное образовательное учреждение высшего профессионального
образования**

Г. К. Сосновиков, В. Н. Шакин

**ТЕХНОЛОГИЯ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ УПРАВЛЕНИЯ
РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ**

ЛАБОРАТОРНЫЕ РАБОТЫ

Москва 2012

Данное пособие включает в себя перечень лабораторных работ по использованию СУБД Access 2003. Каждая работа содержит учебный материал и упражнения, которые необходимо изучить и выполнить студенту. Особенностью пособия является включение в работы студента изучение технологии использования СУБД и автоматизации проектирования БД.

Использование СУБД Access более старших версий требует изучения особенностей этих версий с помощью соответствующей литературы (например: [10, 11]) . Порядок выполнения работ остается тот же.

Лабораторная работа 1

ТЕМА: СОЗДАНИЕ И РЕДАКТИРОВАНИЕ ТАБЛИЦ. ИЗМЕНЕНИЕ СТРУКТУРЫ ТАБЛИЦ. СОРТИРОВКА И ПОИСК ДАННЫХ. ФОРМЫ

1.1. Цель работы

Изучить принципы построения баз данных, освоить правила создания и редактирования таблиц в СУБД ACCESS. Освоить приемы редактирования таблиц, поиска и сортировки данных в СУБД ACCESS . Изучение приемов создания и редактирования форм.

1.2. Общие сведения

Основные принципы проектирования баз данных

Основные 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. Название "реляционная" связано с тем, что каждая запись в такой базе данных содержит информацию, относящуюся (related) только к одному объекту. Кроме того, с данными о двух объектах можно работать как с единым целым, основанным на значениях связанных между собой данных. Все данные в реляционной БД представлены в виде таблиц. Каждая строка таблицы содержит информацию только об одном объекте и называется *записью*. Столбец таблицы содержит однотипную для всех записей информацию и называется *полем*.

Для успешного функционирования базы данных важна правильная организация данных в ней. При определении структуры данных в базе выделяют следующие основные понятия [7].

Класс объектов (сущность) - совокупность объектов, обладающих одинаковым набором свойств. Например, в базе данных о ВУЗе классами объектов являются студенты, преподаватели, предметы. Для каждого отдельного объекта из данного класса объектов в таблице создается отдельная запись.

Свойство (атрибут) - определенная часть информации о некотором объекте. Хранится в виде столбца (поля) таблицы. Например, *фамилия, имя, отчество* - это свойства для объекта *Студент*.

Связь (отношение) - способ, которым связана информация о разных объектах.

1.3. Типы связей между объектами

Основным структурным компонентом базы данных, как правило, является таблица. При определении состава таблиц следует руководствоваться правилом: **в каждой таблице должны храниться данные только об одном классе объектов**. Например, в одной таблице нельзя хранить анкетные данные студента и фамилии преподавателей, которым он сдавал экзамены, т.к. это свойства разных классов объектов.

Если в базе данных должна содержаться информация о разных классах объектов, то она должна быть разбита на отдельные таблицы. Связь между таблицами осуществляется с помощью общих полей.

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

Связь типа “один-к-одному” (1:1)

При этом типе связи 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 типа “один-ко-многим” (1:M)

При таком типе связи каждой записи в одной таблице соответствует несколько записей в связанной таблице. Этот наиболее распространенный тип связей. Для его реализации используются две таблицы. Одна из них представляет сторону "один", другая - сторону "много". Например, нужно иметь информацию о студентах и результатах сдачи ими экзаменов (дата сдачи, предмет, оценка и т.д.). Если все это хранить в одной таблице, то ее объем неоправданно возрастет, т.к. в ней для каждой записи об очередном экзамене должны повторяться все анкетные сведения о студенте. Поскольку *Студент* и *Экзамены* - это разные классы объектов, то и свойства их должны храниться в разных таблицах.

Решением этой задачи является создание двух таблиц. Условно назовем их *Студенты* и *Экзамены*. В каждой из них хранятся соответствующие свойства. Для связи этих таблиц нужно использовать только часть информации о студенте, сдающем экзамен. Но она должна однозначно определять каждого студента среди всех. Такой информацией может явиться, например, номер зачетки (он уникален для каждого студента).

В таблице со стороны "один" (в нашем примере *Студенты*) такие поля называются **ключевыми**. **Основное требование к значениям в ключевых полях - это их уникальность для каждой записи (т.е. они не должны повторяться)**.

Связь типа “много-ко-многим” (M:M)

При таком типе связи 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 со стороны "много" и одна со стороны "один". Связь между этими тремя таблицами также осуществляется по общим полям.

1.4. Структура MS Access

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

В Access база данных включает в себя все объекты, связанные с хранимыми данными (таблицы, формы, отчеты, запросы, макросы, модули). Все объекты Access

хранятся в одном файле с расширением **.mdb**. В таблицах хранятся данные, которые можно просматривать, редактировать, добавлять. Используя формы, можно выводить данные на экран в удобном виде, просматривать и изменять их. Запросы позволяют быстро выбирать необходимую информацию из таблиц. С помощью отчетов можно создавать различные виды документов для вывода на печать, макросы и модули позволяют автоматизировать работу с базой данных.

1.5. Начало работы с MS Access. Создание новой таблицы

После запуска Access нужно щелкнуть на кнопке **Новая база данных** в окне *Microsoft Access* и в предложенном диалоговом окне задать имя для файла БД. После этого на экране появляется **окно базы данных**, из которого можно получить доступ ко всем ее объектам: таблицам, запросам, отчетам, формам, макросам, модулям.

Для создания новой таблицы нужно перейти на вкладку **Таблица** и нажать кнопку **Создать**. **Таблицу можно создать с помощью мастера, путем ввода данных (свойства полей можно при необходимости откорректировать позже) и в режиме конструктора.** Если выбрать способ создания таблицы – Конструктор, то после этого Access выводит окно Конструктора таблицы, в котором задаются имена, типы и свойства полей для создаваемой таблицы.

Среди типов данных Access есть специальный тип - **Счетчик**. В поле этого типа Access автоматически нумерует строки таблицы в возрастающей последовательности. Редактировать значения такого поля нельзя.

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

Размер поля - определяется только для текстовых и Метод-полей; указывает максимальное количество символов в данном поле. По умолчанию длина текстового поля составляет 50 символов

Формат поля – определяется для полей числового, денежного типа, полей типа *Счетчик* и *Дата\Время*. Выбирается один из форматов представления данных.

Число десятичных знаков - определяет количество разрядов в дробной части числа.

Маска ввода - определяет шаблон для ввода данных. Например, можно установить разделители при вводе телефонного номера

Подпись поля - содержит надпись, которая может быть выведена рядом с полем в форме или отчете (данная надпись может и не совпадать с именем поля, а также может содержать поясняющие сведения).

Значение по умолчанию - содержит значение, устанавливаемое по умолчанию в данном поле таблицы.

Условие на значение - определяет множество значений, которые пользователь может вводить в это поле при заполнении таблицы. Это свойство позволяет избежать ввода недопустимых в данном поле значений. Например, если стипендия студента не может превышать 250 р., то для этого поля можно задать условие на значение: ≤ 250 .

Сообщение об ошибке - определяет сообщение, которое появляется на экране в случае ввода недопустимого значения.

Обязательное поле - установка, указывающая на то, что данное поле требует обязательного заполнения для каждой записи. Например, поле **Домашний телефон** может быть пустым для некоторых записей (значение **Нет** в данном свойстве). А поле **Фамилия** не может быть пустым ни для одной записи (значение **Да**).

Пустые строки - установка, которая определяет, допускается ли ввод в данное поле пустых строк.

Индексированное поле - определяет простые индексы для ускорения поиска записей.

Для сохранения структуры таблицы нужно ввести команду **Файл\Сохранить** и в окне **Сохранение** ввести имя таблицы.

1.6. Типы данных MS Access

1. Текстовый. Алфавитно-цифровые данные (до 255 символов)
2. Поле Мемо. Алфавитно-цифровые данные - предложения, абзацы, тексты (до 64 000 символов)
3. Числовой. Различные числовые данные (имеет несколько форматов).
4. Дата\Время. Дата или время в одном из предлагаемых Access форматов.
5. Денежный. Денежные суммы, хранящиеся с 8 знаками в десятичной части. В целой части каждые три разряда разделяются запятой.
6. Счетчик. Уникальное длинное целое, создаваемое Access для каждой новой записи
7. Логический. Логические данные, имеющие значения *Истина* или *Ложь*
8. Поле объекта OLE. Картинки, диаграммы и другие объекты OLE из приложений Windows
9. 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.

1.7. Определение ключевых полей

При создании многотабличной базы данных важно правильно определить связи между таблицами и задать ключи.

Создавая таблицы с помощью **Конструктора**, Access может автоматически создать первичный ключ, поставив перед первым указанным в структуре полем поле **Код** и задав для него тип **Счетчик**. Для этого при сохранении таблицы в запросе на автоматическое определение ключа нужно ответить **Да**.

Но далеко не всегда такой автоматически определенный ключ бывает корректен. В большинстве случаев ключи должен определять сам разработчик БД. Для того, чтобы определить какое-либо поле таблицы ключевым, нужно установить курсор в строке с именем этого поля и нажать на кнопку **Определить ключ** на панели инструментов или ввести команду **Правка\Ключевое поле**. После этого в строке рядом с именем этого поля появится изображение ключа.

1.8. Определение связи таблиц

При реализации отношений “один-ко-многим” и “много-ко-многим” данные разбиваются на несколько таблиц, определяются ключевые поля, обеспечивающие связь таблиц. После определения структуры таблиц БД нужно указать Access способ, которым эти таблицы связаны.

Для определения связей нужно перейти в окно базы данных и выполнить команду **Сервис\Схема данных** или на панели инструментов нажать кнопку **Схема данных**. После этого Access откроет окно **Схема данных** и окно **Добавление таблицы**. В окне **Добавление таблицы** нужно выделить имена таблиц, добавляемых в схему данных, и нажать на кнопку **Добавить**. После этого данное окно можно закрыть.

В окне **Схема данных** появятся имена всех указанных таблиц вместе со списками полей. Имена общих полей выделены полужирным шрифтом. Нужно выделить ключевое поле ключевой таблицы, нажать кнопку мыши, перетащить ее курсор на аналогичное поле в связываемой таблице, после чего кнопку мыши отпустить. В результате появится диалоговое окно **Связи**.

В этом окне Access заполнил первую строку именем поля, по которому связывались таблицы. Чтобы в связанных таблицах не нарушалась целостность данных, нужно щелкнуть по флажку **Обеспечение целостности данных**. После этого Access сделает невозможным запись в неключевую таблицу такого значения общего поля, которого нет в таблице ключевой.

После установления целостности данных Access включает две дополнительные опции: **Каскадное обновление связанных полей** и **Каскадное удаление связанных полей**.

Если выбрать первую опцию, то при изменении какого-либо значения ключевого поля в ключевой таблице Access автоматически обновит значения этого поля для соответствующих записей во всех связанных таблицах. Например, если у одного из студентов изменился номер зачетки в таблице **Студенты**, то он автоматически должен поменяться и в таблице **Экзамены**.

Выбор второй опции при удалении одной из записей в ключевой таблице приведет к удалению тех записей в таблице со стороны "много", которые имеют такое же значение ключа. Например, если из таблицы **Студенты** удалить запись об одном из студентов, то записи о результатах сданных им экзаменов будут удалены автоматически.

Для завершения процесса создания связей, нужно щелкнуть по кнопке **Создать**.

1.9. Использование Мастера подстановок при вводе данных в таблицы

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.

Например, в базе данных **Студенты-Экзамены** поле **Номер зачетки** является общим для обеих таблиц. При заполнении этого поля в неключевой таблице **Экзамены** важно, чтобы его значения для каждого студента точно соответствовали значениям в ключевой таблице. Для этого значения поля **Номер зачетки** в таблице **Экзамены** лучше всего представить в виде списка. Это избавит от необходимости печатать данные при заполнении этой таблицы, предотвратит появление ошибок при наборе и сделает ввод данных более наглядным.

Комбинированный список для поля **Номер зачетки** формируется на основе данных связанной таблицы **Студенты**. Для создания комбинированного списка нужно войти в режим конструктора таблицы **Экзамены**, перейти в колонку **Тип данных** для поля **Номер зачетки**. Из списка доступных типов полей выбрать элемент **Мастер подстановок**.

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/запросов следует выбрать таблицу/запрос, являющуюся источником данных для списка. В нашем примере такой таблицей является таблица **Студенты**, т.к. она послужит источником данных для списка номеров зачетов.

В третьем окне Мастера подстановок из списка **Доступные поля** нужно выбрать поля, значения которых фигурируют в списке. В данном случае можно выбрать поля **Фамилия** и **Имя**, которые сделают список более информативным. После этого можно

нажать кнопку **Готово** для завершения процесса проектирования комбинированного списка. Сохранив проект таблицы, можно открыть таблицу **Экзамены** и просмотреть результат работы.

Другая возможность использования Мастера подстановок - это создание списков с фиксированными значениями. Он используется в тех случаях, когда набор значений в поле строго ограничен. Например, поле **Факультет** в таблице **Студенты** может иметь ограниченный набор значений (все факультеты, имеющиеся в данном ВУЗе). Никаких других значений в этом поле быть не может. Чтобы ввести единообразие в записях и избежать ошибок при вводе, это поле также можно представить в виде списка, формируемого Мастером подстановок. Процесс создания списка похож на описанный выше, но на первом шаге Мастера подстановок следует выбрать опцию **Фиксированный набор значений**. На следующем шаге нужно создать список этих значений (в данном примере записать в список все факультеты данного ВУЗа). После чего можно завершить процесс создания списка и сохранить изменения в структуре. Теперь при вводе данных в поле **Факультет** нет необходимости печатать значения, а достаточно просто выбрать из предлагаемого списка нужное.

1.10. Изменение структуры таблиц

В режиме Конструктора таблицы можно изменять имена, типы и свойства полей, порядок их следования, удалять и добавлять поля.

Для изменения имени поля нужно перейти в строку с именем этого поля в окне Конструктора таблицы и внести изменения в столбце Имя поля.

Для вставки нового поля нужно установить курсор на поле, перед которым оно будет вставлено, и выполнить команду Вставка\Строки или нажать клавишу Insert. На появившейся пустой строке нужно ввести имя нового поля и выбрать тип данных для него.

Для удаления поля нужно выделить строку с его именем и выполнить команду Правка\Удалить или нажать клавишу Delete.

Для изменения типа поля или его формата нужно в режиме Конструктора выделить это поле, перейти в столбец Тип данных, раскрыть список типов данных и выбрать нужное значение. Аналогично можно изменять свойства поля, перейдя в бланк Свойства.

Для сохранения всех изменений достаточно просто закрыть окно Конструктора.

1.11. Переименование и удаление таблиц

В случаях, когда одна из таблиц БД содержит устаревшую или ненужную информацию, ее следует удалить из проекта БД.

Для этого нужно выделить имя таблицы в окне базы данных нажать клавишу **Delete** или выполнить команду **Правка\Удалить**. Access откроет диалоговое окно для подтверждения или отмены удаления.

Также для удаления можно использовать команду **Правка\Вырезать**. В этом случае копия таблицы помещается в буфер обмена, откуда ее можно вставить в другую БД.

Для изменения имени таблицы ее также нужно выделить в окне базы данных и выполнить команду **Правка\Переименовать**. После чего Access даст возможность ввести новое имя таблицы.

1.12. Редактирование связей

Если нужно изменить уже существующие связи, то следует перейти в окно базы данных и вызвать окно Схема данных (команда **Правка\Схема данных**).

Перед редактированием связей нужно закрыть все таблицы, которых это касается!

Для редактирования существующей связи нужно щелкнуть правой кнопкой мыши на ее линии. В появившемся контекстном меню всего две команды: **Изменить связь** и **Удалить**. Выбор первой команды дает диалоговое окно **Связи** где нужно изменить существующие установки.

Если в схему данных необходимо добавить новую таблицу, то нужно щелкнуть по кнопке **Добавить таблицу** на панели элементов или выполнить команду **Связи\Добавить таблицу**. После этого будет выведено окно **Добавление таблицы**, где нужно указать имя добавляемой таблицы.

Некоторые из связанных таблиц в схеме данных можно скрыть. Для этого используется команда **Связи\Скрыть таблицу**. Связи при этом не нарушаются, но в схеме данных таблица не отображается. Это можно делать для различных справочных таблиц, создаваемых, например, Мастером подстановок.

Для удаления ненужной таблицы из схемы данных нужно выполнить команду **Связи\Убрать таблицу**.

1.13. Изменение порядка следования столбцов

По умолчанию порядок столбцов в режиме таблицы совпадает с последовательностью расположения полей в структуре таблицы. При просмотре таблицы этот порядок легко изменить, сделав более удобным.

Для этого в режиме таблицы следует полностью выделить нужный столбец, щелкнув на его заголовке. Чтобы переместить выделенный столбец, перенесите область выделения в новое место.

1.14. Другие изменения макета таблицы

Команда **Формат** дает возможность внести в макет таблицы следующие изменения: переименовать столбец, скрыть столбцы - скрыть выделенные столбцы при просмотре таблицы, отобразить столбцы - вывод диалогового окна, в котором предлагается выбрать столбцы, необходимые для отображения на экране, закрепить столбцы - фиксировать выделенные столбцы при просмотре таблицы, освободить все столбцы - отмена фиксации столбцов.

1.15. Сортировка данных

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

Для сортировки таблицы по значениям только одного поля используются кнопки **Сортировка по возрастанию** и **Сортировка по убыванию**.

Сортировка таблицы по нескольким полям выполняется с помощью фильтра. Для этого нужно открыть таблицу и выполнить команду **Записи\Фильтр\Расширенный фильтр**. Появится окно **Фильтр**.

В этом окне в строке **Поле** из списка полей таблицы нужно выбрать то, сортировка по которому должна производиться в первую очередь. В строке **Сортировка**

в этом же столбце бланка выбрать тип сортировки (по возрастанию или убыванию). В следующем столбце бланка выбирается поле, по которому будет проводиться сортировка во вторую очередь, и для него указывается тип сортировки. И так далее для всех полей, по которым будет проведена сортировка. После заполнения бланка сортировки выполнить команду **Записи\Применить фильтр**.

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 Студенты по возрастанию в поле Факультет в первую очередь и по возрастанию в поле Курс во вторую, то по возрастанию курсов будут расположены только студенты одного и того же факультета.

1.16. Поиск данных

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

Для проведения поиска в таблице нужно открыть эту таблицу, перейти на любую запись в поле, по которому предполагается вести поиск и выполнить команду **Правка\Найти**. Появится диалоговое окно **Поиск**.

В поле **Образец** следует ввести искомое значение. В поле **Совпадение** можно установить значение Поля целиком. В этом случае Access находит только те записи, значение которых полностью совпадает с искомым.

Расширить круг поиска можно, применив символы подстановки:

* - заменяет любое количество символов;

? - заменяет только один символ. Например, по образцу ***ова** будут найдены все фамилии, оканчивающиеся на **-ова**: Петрова, Иванова, Смирнова.

Процесс поиска запускается нажатием кнопки **Найти**.

1.17. Основные сведения о формах

Формы являются основным средством организации интерфейса пользователя в приложениях Access. Хорошо разработанные формы позволяют работать с приложением даже неподготовленному 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 – в этом случае формы используются для запуска макросов (они подробно будут описаны в главе 7);

– 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. Формы позволяют задавать два различных набора параметров: один – для вывода формы на экран, а другой – для вывода на печать.

1.18. Способы создания форм

Для создания формы следует в окне базы данных перейти на вкладку **Формы** и нажать кнопку **Создать**, после чего на экране появится окно **Новая форма**. В данном окне предлагается выбрать источник данных для формы и способ ее создания.

Access предлагает следующие способы создания форм:

Конструктор форм. Этот способ позволяет разрабатывать собственные экранные формы с заданными свойствами для просмотра, ввода и редактирования данных.

Мастер форм. Позволяет достаточно быстро создать форму на основе выбранных для нее данных.

Автоформа: в столбец. Позволяет автоматически создать для исходной таблицы/запроса форму, в которой все поля располагаются в один или несколько столбцов.

Автоформа: ленточная. Позволяет автоматически создать для исходной таблицы/запроса форму, в которой все поля расположены в строку. Названия полей расположены сверху, как в таблице, но каждое значение имеет собственное поле для просмотра.

Автоформа: табличная. Позволяет автоматически создать табличную форму, которая аналогична ленточной, но каждая запись в такой форме находится на отдельной странице.

Диаграмма. Позволяет создавать форму, данные в которой представлены в виде диаграммы.

Сводная таблица. Этот мастер использует Microsoft Excel для создания объекта сводной таблицы, и Microsoft Access для создания формы, в которую внедряется объект сводной таблицы.

Для вариантов создания форм **Автоформа: в столбец**, **Автоформа: ленточная**, **Автоформа: табличная** и **Диаграмма** необходимо выбрать источник данных для создания формы. Во всех остальных случаях это делать необязательно.

1.19. Использование Мастера по созданию форм

По сравнению с простыми автоформами формы, созданные с помощью **Мастера**, более разнообразны по стилю оформления, могут содержать выбранные поля, в т.ч. и из нескольких связанных таблиц.

Для запуска **Мастера форм** нужно в окне **Новая форма** выбрать соответствующий вариант создания формы – **Мастер форм**.

На 1 шаге **Мастера форм** необходимо определить поля будущей формы

На 2 шаге предлагается задать внешний вид формы: в один столбец, ленточный, табличный, выровненный.

На 3 шаге выбирается стиль формы из списка вариантов стилей.

На 4 шаге открывается последнее окно **Мастера форм**, где нужно ввести имя создаваемой формы.

1.20. Создание форм в режиме Конструктора

Создавать и редактировать формы любой степени сложности позволяет только **Конструктор форм**. Также в режиме **Конструктора** можно отредактировать формы, созданные **Мастером**, или автоформы.

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 для них свойств, связанных с ними событий и выполняемых действий.

Для создания формы в режиме **Конструктора** нужно перейти на вкладку **Формы** в окне базы данных и нажать кнопку **Создать**. В окне **Новая форма** выбрать таблицу/запрос, являющиеся источником данных для формы, и указать режим создания **Конструктор**. На экране откроется окно **Конструктора форм**.

1.21. Разделы форм

При открытии окна **Конструктора** форма содержит только один раздел - область данных. Помимо этого, она может содержать разделы заголовка и примечания формы, верхний и нижний колонтитулы. Для добавления или удаления разделов формы используются команды **Вид-Заголовок/примечание** формы и **Вид-Колонтитулы**. Каждый раздел отделен горизонтальной линией, на которой написано его название.

Данные, размещаемые в заголовке, области данных и примечании отображаются в форме. Элементы, расположенные в области данных, отображаются для каждой записи базовой таблицы/запроса. Элементы, расположенные в разделах заголовка и примечания формы, отображаются только в заголовке и примечании формы.

Содержимое верхнего и нижнего колонтитулов отображается только при печати, соответственно в начале и в конце каждой страницы.

Для изменения размера какого либо раздела нужно в режиме Конструктора формы установить указатель мыши на верхнюю часть его границы и переместить ее на новое место.

1.22. Панель элементов и Список полей

Панель элементов используется для размещения объектов в форме. Обычно панель элементов появляется в режиме Конструктора форм автоматически. Убрать или восстановить ее на экране можно командой Вид-Панель элементов. Ниже приведены наиболее используемые элементы формы и соответствующие им кнопки на Панели элементов.



Надпись. Размещение в форме произвольного текста.



Поле. Размещение в форме данных из соответствующего поля базовой таблицы/запроса, вывод результатов вычислений, а также прием данных, вводимых пользователем.



Выключатель. Создание выключателя, кнопки с фиксацией.



Переключатель. Создание селекторного переключателя.



Флажок. Создание контрольного переключателя.



Поле со списком. Размещение элемента управления, объединяющего поле и раскрывающийся список.



Список. Создание списка, допускающего прокрутку. В режиме формы выбранное из списка значение можно ввести в новую запись или использовать для замены уже существующего значения.



Кнопка. Создание командной кнопки, позволяющей осуществлять разнообразные действия в форме (поиск записей, печать отчета, установка фильтров и т.п.).



Рисунок. Размещение в форме рисунка, не являющегося объектом OLE.



Свободная рамка объекта. Размещение в форме свободного объекта OLE, который остается неизменным при перемещении по записям.



Присоединенная рамка объекта. Размещение в форме объектов OLE. Используется для объектов, сохраненных в базовом источнике записей формы, поэтому при перемещении по записям в форме отображаются разные объекты.

Линия. Размещение в форме линии для отделения логически связанных данных.



Прямоугольник. Размещение в форме прямоугольника для группировки элементов управления или выделения логически связанных данных.

При размещении в форме полей из базовой таблицы/запроса удобно пользоваться **Списком полей**, который можно вывести командой **Вид-Список полей**. Он представляет собой окно, содержащее перечень всех полей базовой таблицы/запроса. Чтобы поместить какое-либо поле в форму, достаточно просто перетащить его левой кнопкой мыши из **Списка полей** в нужный раздел формы.

1.23. 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 выполнить команду **Вид-Свойства** (либо нажать кнопку **Свойства** на панели инструментов). В результате на экране появится окно со свойствами и событиями выбранного объекта.

При размещении в форме нового объекта для него устанавливаются значения свойств, принятые по умолчанию. Некоторые свойства наследуются из базовой таблицы. Например, для элемента **Поле** свойства **Формат**, **Число десятичных знаков**, **Маска ввода**, **Значение по умолчанию** копируются из структуры таблицы, в которой находится данное поле.

Каждый элемент формы имеет свое имя. Это имя используется в дальнейшем для ссылок на данный элемент. По умолчанию, новым элементам присваивается имя, состоящее из названия элемента и его номера, который определяется порядком создания. Например, Поле1 или Надпись3.

Если элемент управления создается путем переноса поля с помощью мыши из списка полей, то в свойство **Имя** копируется значение свойства **Имя поля**. Например, если из списка полей в форму перенесено поле Факультет, то этому элементу в форме будет задано имя Факультет.

Не допускается существование в одной форме двух элементов управления с одинаковыми именами, однако, в разных формах элементы управления могут иметь одинаковые имена. Нельзя присвоить одинаковые имена элементу управления и разделу в одной форме.

1.24. 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 лишь Мастером по созданию кнопки. Для этого нужно выполнить следующие действия:

- установить режим использования мастера на панели элементов;
- выбрать инструмент Кнопка на панели элементов и указать местоположение и размер кнопки в форме;
- в первом окне Мастера выбрать категорию кнопки и выполняемые ею действия;
- если создаваемая кнопка предназначена для работы с другим объектом базы данных (отчетом, таблицей, другой формой), то на втором шаге нужно выбрать необходимый объект;
- далее следует выбрать текст или рисунок, размещаемый на кнопке. При размещении рисунка можно воспользоваться кнопкой Обзор для выбора рисунка;
- на заключительном шаге работы Мастера задается имя создаваемого элемента.

1.25. Управление элементами формы

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

Для выделения одиночного элемента достаточно просто щелкнуть по нему мышью. Для выделения нескольких объектов можно воспользоваться следующими способами:

- Нажать клавишу SHIFT и, удерживая ее, щелкнуть мышью по всем выделяемым элементам;
- Выбрать инструмент Выбор объектов на панели элементов. Установить указатель мыши на элемент и, не отпуская кнопки мыши, переместить рамку выделения так, чтобы внутри нее оказались все нужные элементы.

Выделив несколько элементов, можно управлять их общими свойствами.

Для изменения размера элемента нужно переместить один из маркеров выделения до достижения нужного размера. Если нужно установить размер для поля или надписи так, чтобы в нем были полностью видны все отображаемые данные, то лучше воспользоваться командой **Формат-Размер-По размеру данных**.

Для перемещения выделенного элемента нужно установить указатель мыши на границу выбранного объекта так, чтобы указатель принял форму открытой ладони. После этого нажать кнопку мыши и переместить элемент на новое место.



Рис. 1. Элементы поля 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 элемента (рис. 1). Если подвести к ним указатель мыши, то он примет форму указательного пальца руки. В этом положении можно переместить каждый из связанных элементов отдельно.

Для удаления элемента формы нужно его выделить. После этого следует нажать клавишу **Delete** или воспользоваться командой **Правка-Удалить**.

1.26. Изменение порядка обхода элементов формы

Во время ввода данных в форме переход от одного элемента к другому, при нажатии клавиши **ТАВ**, осуществляется в соответствии с заданным в экранной форме порядком обхода объектов. Для определения этого порядка используется команда Вид-Последовательность перехода. На экране откроется окно диалога Последовательность перехода. В этом окне нужно расположить поля в том порядке, в котором должен осуществляться их обход в форме. Для перемещения поля на другое место в этом списке следует выделить это поле и, нажав кнопку мыши, переместить его на новое место.

1.27. 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 Экзамены), связанных отношением “один-ко-многим”.

Но можно создать форму, которая позволит вводить, редактировать и просматривать все эти данные одновременно. В таких случаях для отображения записей основной и подчиненной таблиц можно использовать многотабличную иерархическую форму (рис. 2).

Предмет	Дата сдачи	Оценка
философия	15.01.01	3
биология	22.01.01	3
культурология	02.02.01	4

Рис. 2. Пример иерархической формы

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, 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 Экзамены (таблица со стороны “много”).

Для создания такой формы нужно:

- перейти на вкладку **Формы** в окне базы данных, нажать кнопку **Создать** и выбрать в качестве режима создания **Мастер форм**;

- в окне **Новая форма** 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. Опция **Связанные формы** предназначена для представления данных без использования подчиненной формы;

- далее следует выбрать внешний вид подчиненной формы: табличный или ленточный;

- на последующих шагах **Мастера форм** выбрать стиль, задать имена форм, а также указать режим, в котором будет открыта форма после завершения ее создания.

1.28. Порядок выполнения работы

1. Запустить MS Access.
2. Создать новую базу данных в файле с именем BASA1фамилиягруппа, например BASA1IVANOVPO0801.
3. Создать структуру ключевой таблицы БД, определив ключевое поле и индексы; сохранить ее, задав имя **Студенты**.

Для создания структуры ключевой таблицы **Студенты** рекомендуется использовать режим конструктора.

Внимание! В бланке **Свойства** обязательно указать длину текстовых полей, формат числовых полей и дат. Поле **Номер зачетки** в таблице **Студенты** объявить **ключевым и индексированным со значением Совпадения не допускаются**.

Структура таблицы **Студенты** может быть следующей:

Имя поля	Тип поля
НомерЗачетки	Числовой
Фамилия	Текстовый
Имя	Текстовый
Отчество	Текстовый
Факультет	Текстовый
Курс	Числовой
Группа	Числовой
ДатаРождения	Дата\Время
Стипендия	Числовой

4. Ввести в таблицу **Студенты** 10-15 записей и сохранить их.

Вводить данные в таблицу **Студенты** рекомендуется в режиме таблицы. Для сохранения записей достаточно просто закрыть окно таблицы.

5. Создать структуру неключевой таблицы БД и сохранить ее, задав имя **Экзамены**.

Структура таблицы **Экзамены** может быть следующей:

Имя поля	Тип поля
Номер зачетки	Мастер подстановок
Предмет	Текстовый
Оценка	Числовой
ДатаСдачи	Дата\Время

Обязательно определить нужные формат и длину полей в бланке Свойства.

Тип поля **Номер зачетки** определяют **Мастером подстановок** используя для подстановки данные из таблицы **Студенты**. В качестве доступных при подстановке полей выбрать **Фамилию** и **Имя**. Снять галочку **Скрыть ключевой столбец**.

Поле **НомерЗачетки** в таблице **Экзамены** объявить индексированным со значением **Совпадения допускаются**.

Внимание! При сохранении структуры неключевой таблицы Access может предупредить об отсутствии ключевого поля и предложит создать это поле сейчас. В данном случае следует отказаться от этого.

6. Установить связь с отношением один-ко-многим между таблицами **Студенты** и **Экзамены** с обеспечением целостности данных.

Для этого следует вызвать окно **Схема данных** и добавить в схему данных обе таблицы (**Студенты** и **Экзамены**). Установить в окне схемы данных связь между таблицами по полю **Номер зачетки**. Включить переключатель **Определение целостности данных**. После этого нужно указать тип связи: “один-ко-многим” и включить опции **Каскадное обновление** связанных полей и **Каскадное удаление** связанных полей. После этого закрыть окно **Связи**.

7. Заполнить таблицу **Экзамены** данными в режиме таблицы, используя созданный с помощью Мастера подстановок список в поле **НомерЗачетки**.

8. Проверить соблюдение целостности данных в обеих таблицах. Для этого нужно:
· изменить значение ключевого поля (**НомерЗачетки**) для одной из записей в таблице **Студенты**. Перейти в таблицу **Экзамены** и проверить, изменилось ли в ней значение общего поля для соответствующих записей;

9. Изменить структуру ключевой таблицы, добавив новые поля и изменив свойства для некоторых полей, созданных ранее.

В режиме Конструктора таблицы **Студенты** произвести в ее структуре следующие изменения:

- добавить поля **Город, Адрес, Телефон**;
 - для поля **Город** в свойстве **Значение по умолчанию** задать значение: Москва
- Определить тип поля **Факультет** таблицы **Студенты** с помощью Мастера подстановок, взяв в качестве источника данных фиксированный набор значений (список всех факультетов ВУЗа).

Для поля **Курс** таблицы **Студенты** ввести условие на значение: **>0 and <=5** и задать соответствующее сообщение об ошибке.

Данные в поля **Город, Адрес, Телефон** вводить в режиме таблицы.

Отредактировать значения в поле **Факультет**, используя список значений, созданный Мастером подстановок.

Для нескольких записей использовать значение по умолчанию в поле **Город**.

В одной из записей попробовать внести в поле **Курс** значение, большее 5.

10. Произвести сортировку записей в таблицах по одному и нескольким признакам

Отсортировать таблицу **Студенты** по следующим признакам:

- возрастанию в поле **Фамилия**;
- убыванию в поле **Стипендия**;
- возрастанию в поле **Факультет** и убыванию в полях **Курс** и **Группа**, используя для этого расширенный фильтр.

11. Найти в таблицах записи, удовлетворяющие определенным условиям, и заменить в них некоторые значения на другие.

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

- студенты, чьи фамилии начинаются с определенной буквы;
- студенты, обучающиеся на одном курсе определенного факультета.

Найти записи для студентов определенного факультета и заменить для них название этого факультета. Например, **эконом.** на **экономический**.

12. Создать различные формы с помощью Создание/Формы и Создание/Мастер форм.

13. В режиме формы произвести следующие действия:

- с помощью кнопок перемещения по записям перейти на последнюю запись таблицы и ввести 2 новые записи:

- для второй записи изменить значения в полях **Факультет** и **Курс**;

- сохранить форму;

- открыть таблицу **Студенты** и найти в ней изменения, внесенные в режиме формы.

14. Создать форму **Экзаменационная ведомость**, используя режим Мастера форм:

- в качестве источника данных выбрать таблицу **Студенты**;

- включить в форму поля **Фамилия, Имя, НомерЗачетки**;

- завершить построение формы, выбрав ей внешний вид, стиль оформления и название **Экзаменационная ведомость**;

- просмотреть полученный результат на экране.

15. Перейти в режим Конструктора формы **Экзаменационная ведомость** и внести следующие изменения:

- увеличить высоту раздела **Заголовок формы**;

- выделить все подписи полей и перенести их в нижнюю часть области заголовка ;

- в верхней части области заголовка ввести элементы типа **Надпись** и **Линия** по следующему образцу:

Зачетно-экзаменационная ведомость

Факультет _____ **Курс** _____ **Группа** _____

Название предмета _____

Дата проведения зачета/экзамена _____

Ф.И.О. преподавателя _____

- в области заголовка формы за надписью **Номер зачетки** разместить надписи **Оценка** и **Подпись преподавателя**;

- в области данных под вновь созданными надписями разместить поля свободного формата (т.е. пустые поля, не связанные ни с какими данными таблиц/запросов), удалив их присоединенные подписи. Для размещения полей свободного формата нужно на Панели элементов выбрать элемент Поле и указать место его размещения;

- просмотреть созданную форму и при необходимости отредактировать ее, вернувшись в режим Конструктора;

- сохранить форму с именем **Экзаменационная ведомость**.

16. При помощи Мастера форм создать иерархическую форму по двум таблицам. Для этого:

- в качестве режима создания выбрать Мастер форм;

- в качестве источника данных указать таблицу **Экзамены**;

- включить в форму поля **НомерЗачетки, Предмет** и **Оценка** из таблицы **Экзамены** и поля **Фамилия, Имя, Отчество, Факультет, Курс, Группа, Стипендия** из таблицы **Студенты**;

- выбрать тип представления данных **по таблице Студенты**;

- выбрать стиль и внешний вид формы (табличный);

- задать название формы **Данные об успеваемости студента**;

- просмотреть полученную форму и сохранить.

1.29. Требования к отчету

Содержание отчета:

1. тема и цель работы;
2. краткое описание последовательности выполнения;
3. описание структуры создаваемой БД;

1.30. Контрольные вопросы

1. Что понимают под классом объектов? Свойствами объектов?
2. Типы связей "один-к-одному" и "один-ко-многим": как они реализуются в реляционных базах данных?
3. Что такое ключевое поле, и какие требования предъявляются к нему?
4. Как в СУБД Access можно определить ключевое поле?
5. Как и для чего в СУБД Access создается схема данных?
6. Перечислите типы данных в Access.
7. Как при определении структуры таблицы использовать свойства полей?
8. Как в Access определить связи между таблицами?
9. Как используется Мастер подстановок?
10. Каковы возможности Access по изменению структуры таблиц?
11. Как добавить или удалить поле в структуре таблицы?
12. Как отсортировать записи по одному полю?
13. Как отсортировать записи по нескольким полям?
14. Как произвести поиск и замену данных в Access?
15. Как использовать шаблоны при поиске данных?
16. С какой целью создаются формы? Назовите виды форм MS Access.
17. Назовите инструменты создания форм.
18. Назовите виды автоформ.
19. Перечислите основные разделы форм и их назначение.
20. Для чего используется раздел **Данные** в форме?
21. Как разместить в форме элемент управления?
22. Как используется при создании форм **Список полей**?
23. Как изменить порядок обхода элементов в форме?
24. Как построить форму по двум связанным таблицам?

Лабораторная работа 2

ТЕМА: РАБОТА С ДАННЫМИ С ИСПОЛЬЗОВАНИЕМ ЗАПРОСОВ. ЗАПРОСЫ-ВЫБОРКИ.

2.1. Цель работы: изучение работы с данными при помощи запросов.

2.2. Понятие запроса

При работе с таблицами можно в любой момент выбрать из базы данных необходимую информацию с помощью запросов.

Запрос - это обращение к БД для поиска или изменения в базе данных информации, соответствующей заданным критериям.

С помощью Access могут быть созданы следующие типы запросов: запросы на выборку, запросы на изменение (запросы-действия), перекрестные запросы, запросы с параметром.

Одним из наиболее распространенных запросов является запрос на выборку, который выполняет отбор данных из одной или нескольких таблиц по заданным пользователем критериям, не приводящий к изменениям в самой базе данных.

2.3. Создание запроса

Для начала создания запроса следует открыть базу данных, и, перейдя на вкладку **Запросы** нажать кнопку **Создать**. Появится окно Новый запрос для выбора способа построения запроса (Рис 1).

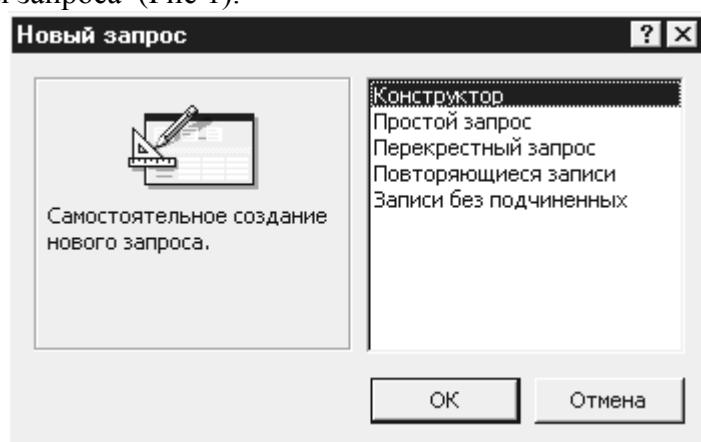


Рис. 1. Диалоговое окно **Новый запрос**

Конструктор - создает запрос на основе пустого бланка запроса.

Простой запрос - создает простой запрос из определенных полей.

Перекрестный запрос - создает запрос, данные в котором имеют компактный формат, подобный формату сводных таблиц в Excel.

Повторяющиеся записи – создает запрос, выбирающий повторяющиеся записи из таблицы или простого запроса.

Записи без подчиненных - создает запрос, выбирающий из таблицы записи, не связанные с записями из другой таблицы.

При выборе **Конструктора** через диалоговое окно **Добавление таблицы** (рис. 2) добавляются имена таблиц в окно конструктора запроса

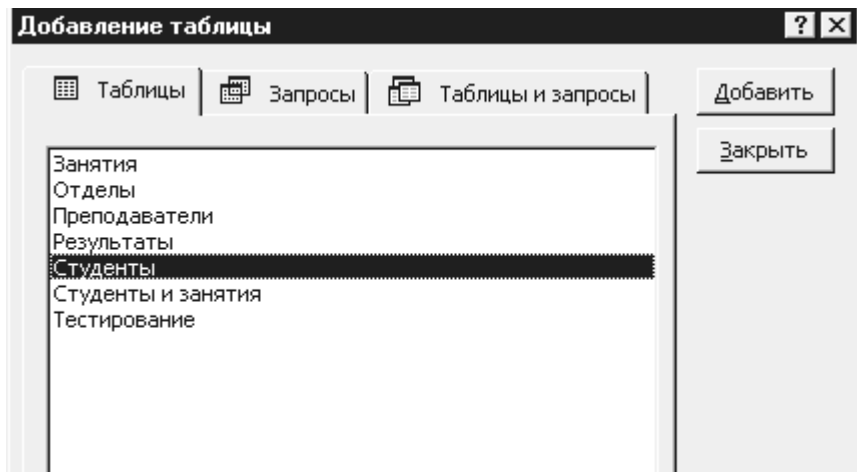


Рис 2. Окно **Добавление таблицы**

Окно **Добавление таблицы** состоит из трех вкладок, содержащих перечни объектов, предлагаемых программой для проектирования запроса: **Таблицы**, **Запросы** и **Таблицы и запросы**. При выборе вкладки **Таблицы** следует выделить нужную нам таблицу из предложенного списка и с помощью кнопки **Добавить** можно добавить несколько таблиц. Например, на рис.2 выбрана таблица **Студенты**.

Имена таблиц должны быть представлены в окне конструктора запроса (рис 3).

2.4. Окно конструктора запроса

Окно конструктора (рис.3) разделено на две части. В верхней части находятся окна таблиц со списками полей. Имя каждой таблицы отображается в строке заголовка такого окна.

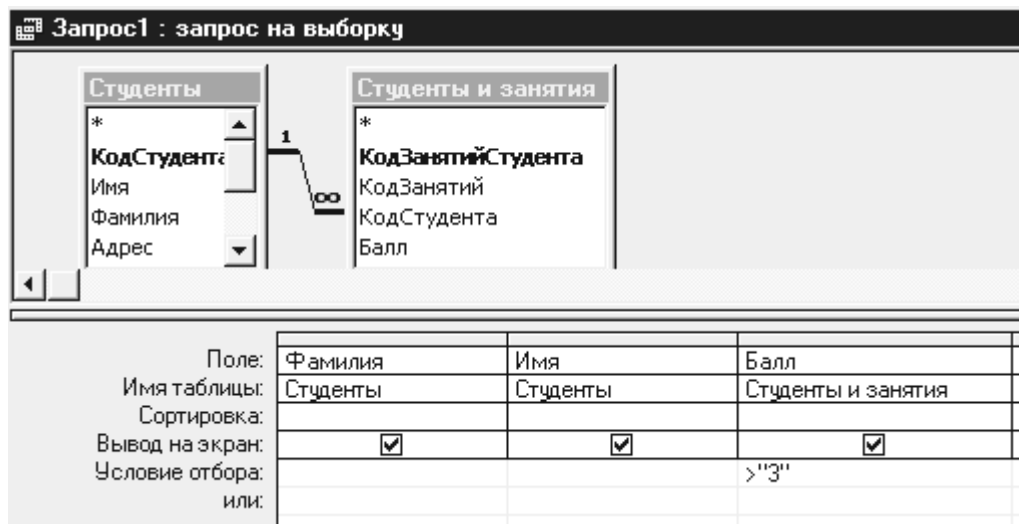


Рис 3. Окно конструктора запроса. Пример ввода условия.

Например, на рис.3 представлены таблицы **Студенты** и **Студенты и занятия**.

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

Для перемещения из верхней панели окна в нижнюю и обратно используется клавиша F6.

2.5. Включение полей в запрос

В запрос не следует включать все поля выбранных таблиц.

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.3 в бланк запроса включены поля *Фамилия, Имя и Город* из таблицы *Студент*.

Примечание:

Если был установлен флажок **Имена таблиц** из меню **Вид**, то во второй строке бланка QBE выйдет на экран имя таблицы, из которой выбрано поле (см. рис.3). В строке **Вывод на экран** флажком помечаются те поля, которые должны быть выведены на экран.

В общем случае поля, вводимые в наборе записей запроса, наследуют свойства, заданные для соответствующих полей таблицы.

Можно определить другие значения свойств, выполнив команду **Свойства** из меню **Вид**: Описание (текст, содержащий описание объекта), Формат поля (представление данных на экране), Число десятичных знаков (для числовых данных), Маска ввода, Подпись (заголовок столбца).

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

2.6. Установка критериев отбора записей

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

Чтобы найти записи с конкретным значением в каком либо поле, нужно ввести это значение в данное поле в строке бланка QBE Условие отбора (см. рис.3).

Критерии, устанавливаемые в QBE – области, должны быть заключены в кавычки. Если ACCESS идентифицирует введенные символы как критерии отбора, то заключает их в кавычки автоматически, а если нет, то сообщает о синтаксической ошибке.

Например, как показано на рис.3, построен запрос, по которому из данных по баллам будут выбраны фамилии и имена студентов с оценками только 4 и 5.

2.7. Виды критериев

Для создания запроса с несколькими критериями пользуются различными операторами.

Можно задать несколько условий отбора, соединенных логическим оператором ИЛИ (or), для некоторого поля одним из двух способов:

1) можно 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.

Например, запись 4 or 5 соответствует тому, что будут выбраны фамилии с оценками 4 или 5.

2) ввести второе условие в отдельную ячейку строки или. И если используется несколько строк или, то чтобы запись была выбрана, достаточно выполнения условий хотя бы в одной из строк или, как, например, показано на рис. 4.

Поле:	Фамилия	Имя	Балл
Имя таблицы:	Студенты	Студенты	Студенты и занятия
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			"4"
или:			"5"

Рис 4. Пример записи условия с использованием оператора *или* (*or*)

При такой записи условия также будут выбраны фамилии с оценками 4 или 5.

Логическая операция И.

Логическая операция И (and) используется в том случае, когда должны быть выполнены оба условия и только в этом случае запись будет выбрана.

Например, записав условие

>2 and <5

будут выбраны только оценки 3 и 4.

Чтобы объединить несколько условий отбора оператором И (and), следует привести их в одной строке.

Например, на рис 5 показано как можно задать условие для выбора фамилий студентов, проживающих в городе Уфе, которые учатся только на 5.

Поле:	Фамилия	Город	Балл
Имя таблицы:	Студенты	Студенты	Студенты и занятия
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		"Уфа"	"5"
или:			

Рис 5. Пример записи условия с использованием оператора *и* (*and*)

Исключить группу данных из состава анализируемых запросом записей позволяет следующий критерий

<> 4

В этом случае можно не использовать кавычки.

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")

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

* — обозначает любое количество (включая нулевой) символов;

? — любой одиночный символ;

— указывает что в данной позиции должна быть цифра.

Например: для выбора фамилии, начинающейся с буквы П и с окончанием "ов" можно записать

like П*ов

Можно ввести дату и время, при этом значения должны быть заключены между символами #. Например:

#10 мая 1998#

>#31.12.96#

В Access используется ряд других функций, которые помогут задать условия отбора для даты и времени, например:

Day(дата) – возвращает значение дня месяца в диапазоне от 1 до 31

Month(дата) – возвращает значение месяца года в диапазоне от 1 до 12

Year(дата) – возвращает значение года в диапазоне от 100 до 9999

2.8. Вычисляемые поля

Можно задать вычисления над любыми полями таблицы и сделать вычисляемое 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.

2.9. 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 по кнопке панели инструментов **Построить**, 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 кнопки, расположенные под областью ввода.


Для начала нужно щелкнуть дважды в левом списке по папке **Таблицы**, и выбрать саму таблицу, а затем в колонке **Код** само поле и щелкнуть по кнопке **Вставить**.

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

2.10. Итоговые запросы

Итоговые запросы значительно отличаются от обычных. В них поля делятся на 2 типа:

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

Для составления итогового запроса, находясь в режиме конструктора, следует нажать кнопку Групповые операции  на панели инструментов или воспользоваться командой Групповые операции из меню Вид.

В результате чего в бланке запроса появится строка Групповая операция. Если для соответствующего поля из списка выбрать функцию Группировка (рис 6), то при выполнении запроса записи по этому полю группируются по значениям в этом поле, но итог не подводится.

Группировка в итоговом запросе производится только по одному полю. Во всех остальных полях вводятся итоговые функции.

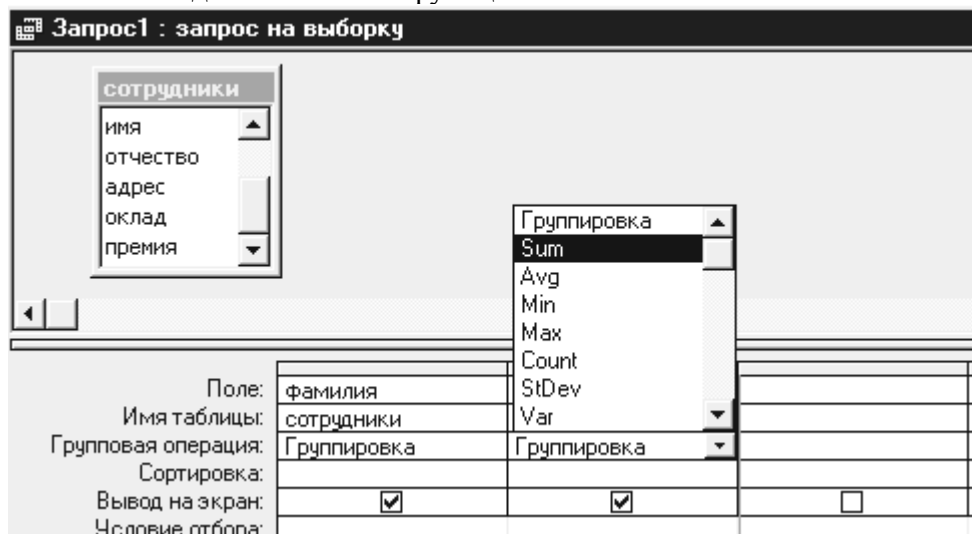


Рис 6. Строка *Групповая операция* в бланке QBE

Access предоставляет ряд функций, обеспечивающих выполнение групповых операций. Можно задать нужную функцию, набрав на клавиатуре ее имя в строке, Групповая операция или выбрав ее из раскрывающегося списка.

Основные групповые функции, которыми можно воспользоваться:

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

AVG - вычисляет среднее значение в тех записях определенного поля, которые отобраны запросом (для числовых или денежных полей);

MIN - выбирает минимальное значение в записях определенного поля, отобранных запросом;


MAX – выбирает максимальное значение в записях определенного поля, отобранных запросом;

COUNT – вычисляет количество записей, отобранных запросом в определенном поле, в которых значения данного поля отличны от нуля;

FIRST - определяет первое значение в указанном поле записей;

LAST - определяет последнее значение в указанном поле записей.

2.11. Выполнение запроса

Готовый запрос выполняется после щелчка по кнопке  панели инструментов в режиме Конструктора запросов или при активизации команды **Запуск** из меню **Запрос**. В результате будет получена таблица с ответом на заданные условия . Например, на рис 7 показан результат запроса, построенного на рис. 3.

	Фамилия	Имя	Балл
▶	САФИН	АЛИК	4
	АХМЕТОВ	РАИС	5
	ГАЛИМОВ	ИЛЬГИЗ	4
	СИДОРОВ	АНДРЕЙ	4
	ГАРИПОВ	ИЛЬДАР	4
*			

Рис.7. Результат выполненного запроса

Запросы можно создавать для отбора данных как из одной, так и из нескольких таблиц. Запросы к нескольким таблицам производятся аналогично запросам к однотабличным БД с той лишь разницей, что в окно конструктора запроса добавляются все таблицы, данные которых нужны в запросе.

При этом следует учитывать наличие связей между таблицами.

2.12. Порядок выполнения работы

1. Создать простой запрос - выбрать несколько произвольных полей из таблицы **Студенты**.

2. С помощью **Конструктора** создать запросы, удовлетворяющие условиям:

- единственное значение факультета;
- два различных факультета;
- фамилии студентов, начинающиеся с определенной буквы (использовать шаблоны);
- фамилии студентов, заканчивающиеся на "ов";
- фамилии студентов одного факультета и одного курса;
- фамилии и имена студентов, проживающие в одном из городов или обучающиеся на одном из факультетов;
- фамилии студентов, у которых стипендия больше 400 рублей;
- фамилии студентов, занимающиеся не в 1-ой группе и стипендия которых в пределах от 200 до 500 р.

Примечание:

В запрос должны быть включены поля *Фамилия*, *Имя*, *Отчество* и те поля, где вводятся критерии.

После того как был задан критерий для запроса, запрос нужно выполнить и сохранить под именем, подходящим по смыслу.

3. Для запросов с полем типа **Дата/время** добавить поле *ДатаРождения* и выбрать записи, удовлетворяющие условиям:

- дата больше 1.1.80;
- дата в интервале значений и задан факультет;
- фамилии и имена студентов, родившихся в 80-х годах;
- вычислить возраст студентов;
- фамилии и имена студентов, родившихся в первой половине месяца;

4. Создать итоговый запрос:

- оставить в запросе поля *Факультет*, *Стипендия*, *НомерЗачетки*, вычислить максимальное значение стипендии для каждого факультета и подсчитать количество студентов на каждом факультете (используя Count).

5. Запрос с вычисляемыми полями:

- включить в запрос вычисляемое поле, которое является результатом сцепления текстовых полей *Фамилия*, *Имя*, *Отчество*. Назвать поле *Ф. И. О. студента*;
- используя построитель выражений, подсчитать надбавку студентам, равную 15% от стипендии;

6. Создать запрос, в котором используются поля из двух ранее созданных и связанных таблиц, задав ему имя **Запрос для 2-х таблиц**

– убрать несколько полей таблицы **Студенты** и добавить поля *Предмет* и *Оценка* из таблицы **Успеваемость**;

– выбрать поле *Фамилия*, *предмет* и *Оценка*, вычислить минимальное значение по полю *Оценка*;

– сгруппировать по номеру зачетки и вычислить среднюю оценку для каждого студента.

Требования к отчету

Отчет о проделанной работе должен содержать:

1. Название работы, цель, последовательность выполнения.
2. Ответы на контрольные вопросы.

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. Как удалить таблицу из запроса?
6. Как удалить поле из запроса?
7. Перечислите основные операторы, используемые в запросе.
8. В каких случаях используется оператор OR?
9. В каких случаях используется оператор AND?
10. В чем различие между операторами OR и AND?
11. Назначение итоговых запросов.
12. Назначение строителя выражений.
13. Как вычислить сумму значений заданного поля?
14. Как осуществить сортировку записей в запросе?
15. Какие виды вычислений можно произвести в итоговых полях?

Лабораторная работа 3

ТЕМА: РАБОТА С ДАННЫМИ С ПОМОЩЬЮ ЗАПРОСОВ-ДЕЙСТВИЙ. ПЕРЕКРЕСТНЫЕ ЗАПРОСЫ

3.1. Цель работы: приобрести навыки по работе с данными с помощью запросов-действий

3.2. Понятие запросов-действий

Для быстрого изменения, вставки, 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, добавление.

Создание таблицы. Этот запрос позволяет сохранить набор данных, вышедших в ответ на запрос, в виде таблицы.

Обновление. Используется для того, чтобы обновить значения некоторых полей для всех записей, вышедших в ответ на запрос.

Удаление. Используется для удаления всех записей, вышедших в ответ на запрос.

Добавление. Используется для добавления данных в имеющуюся таблицу.

ACCESS отмечает запросы-действия специальными значками.

Для того чтобы обезопасить себя от каких-либо случайностей, необходимо перед выполнением запроса-действия создать резервную копию используемой таблицы.

Для создания резервной копии таблицы необходимо перейти в окно базы данных, выбрать таблицу, в которой произойдут изменения, а затем воспользоваться командой **Правка/Копировать**. Затем выполнить команду **Правка/Вставить**, и в диалоговом окне указать имя для копии таблицы.

Создание запросов на обновление, добавление и удаление невозможно без разрушения целостности данных, поэтому упражняться лучше на созданной копии таблице.

Новый запрос желательно создать вначале как запрос на выборку. Так вы сможете проверить правильность отбора записей.

Прежде чем щелкнуть по кнопке **Запуск**, просмотрите результат в режиме таблице, воспользовавшись кнопкой **Вид**.

3.3. Запросы на создание таблицы

Для сохранения результатов выборки в реально существующей таблице используются запросы на создание таблицы. Это ускоряет доступ к данным, полученным по запросу. Сохранение в виде отдельной таблицы полезно и в том случае, если какие-либо данные нужно хранить в течение длительного времени.

Например, сводную информацию по годам необходимо хранить в течение 5 лет. Для этого создается итоговый запрос, и результат его выполнения сохраняется в виде отдельной таблицы. Для этого в режиме Конструктора запросов нужно выбрать команду **Запрос/Создание таблицы**.

Access выведет на экран диалоговое окно «Создание таблицы», где нужно указать имя таблицы, в которой будут сохранены результаты выборки (рис. 1).

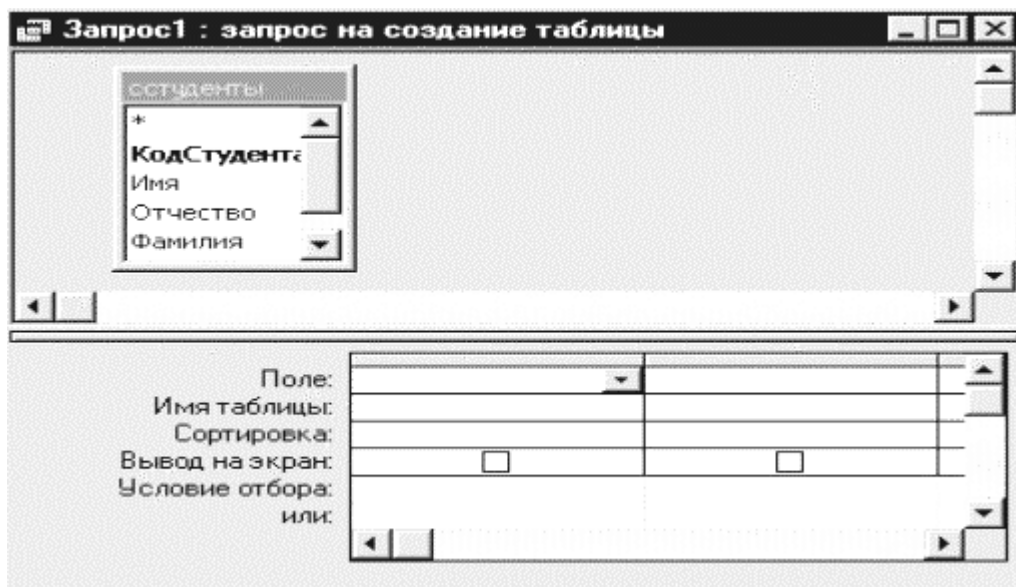


Рис. 1. Вид окна запроса на создание таблицы

3.4. Запросы на обновление записей

Этот тип запросов-действий предназначен для изменения значений некоторых полей для всех выбранных записей. После входа в режим **Конструктора запроса** нужно выбрать команду **Запрос/Обновление**. После этого в бланке запроса появится строка **Обновление**. Эта строка используется для того, чтобы задать новые значения для выбранных записей (рисунок 2).

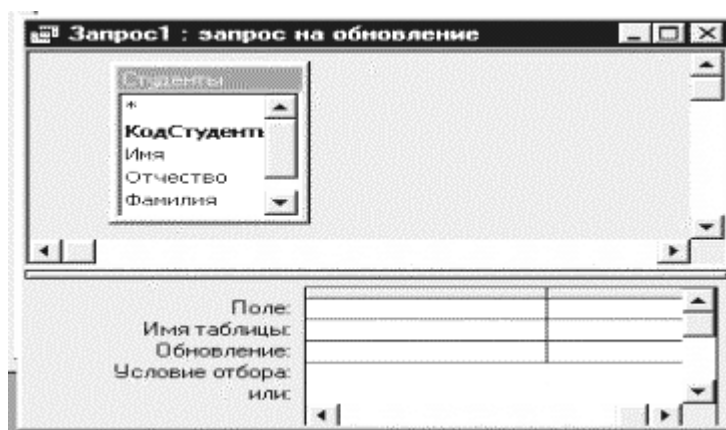


Рис.2. Вид окна запроса на обновление записей

Обновлять значения можно не только в полях критериев, а в любом поле таблицы. Также для выбранных записей можно создавать вычисляемые поля.

3.5. Запрос 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 **Запрос/Удаление** в режиме Конструктора запросов. При этом в бланке запроса появится строка **Удаление**, где можно ввести условие для удаления записей (рис. 3).

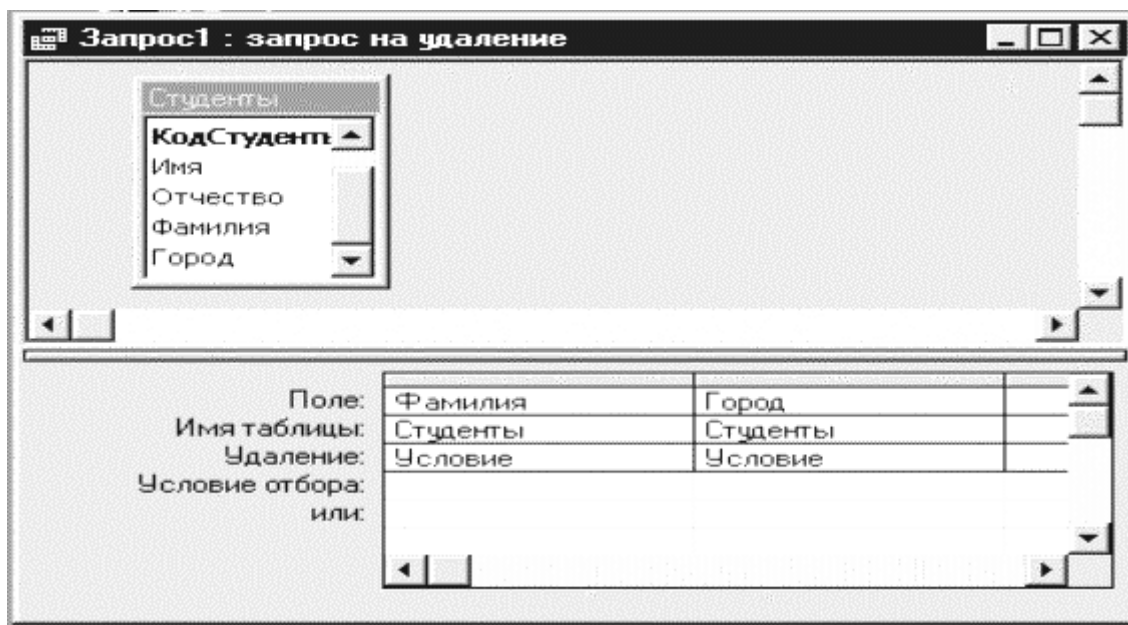


Рис. 3. Вид окна запроса на удаление записей

В режиме **Конструктора запроса** необходимо указать таблицу, в которой будут проводиться удаления. Затем определить критерии и обязательно 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 непредсказуемым результатам, в том числе и к нарушению целостности базы данных, по следующим причинам:

- Таблица является частью отношения “один-ко-многим”.
- В качестве условия обеспечения целостности данных использована опция **Каскадное удаление связанных полей**.

Каскадное удаление связанных полей.

Например, вы решили исключить каких-то студентов и удалить соответствующие записи из таблицы **Студенты**. При этом существует также таблица **Экзамены**, которая связана с таблицей **Студенты** отношением “один–ко-многим”. Если для обеспечения целостности данных выбрана опция **Каскадное удаление связанных полей**, в результате удаления записей о студентах будут удалены записи об их оценках, то есть информация об успеваемости!

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 скажет про каскадное удаление.

3.6. Запрос на добавление записей

С помощью запроса на добавление 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 таблицы).

Выполнение запроса на добавление записей можно начать (так же, как и в других запросах-действиях) с запуска запроса-выборки – для того, чтобы убедиться, что вы копируете необходимые записи.

Если необходимо добавить записи к таблице в другой базе данных, сначала следует присоединить таблицу-источник к базе, содержащей целевую таблицу, с помощью команды **Файл/Внешние данные/Связь с таблицами**. Для отбора добавляемых записей нужно составить запрос на выборку. Затем надлежит выполнить составленный запрос и оценить результат, переключившись в режим таблицы с помощью команды **Вид/Режим таблицы**. После этого необходимо вернуться в режим Конструктора и активизировать команду **Запрос/Добавление**.

В открывающемся окне диалога нужно в поле **Имя таблицы** ввести имя таблицы, к которой требуется присоединить данные из результирующего набора записей запроса. Закрытие диалогового окна кнопкой ОК приводит к тому, что Access добавляет в бланк запроса строку **Добавление** (рис. 4).

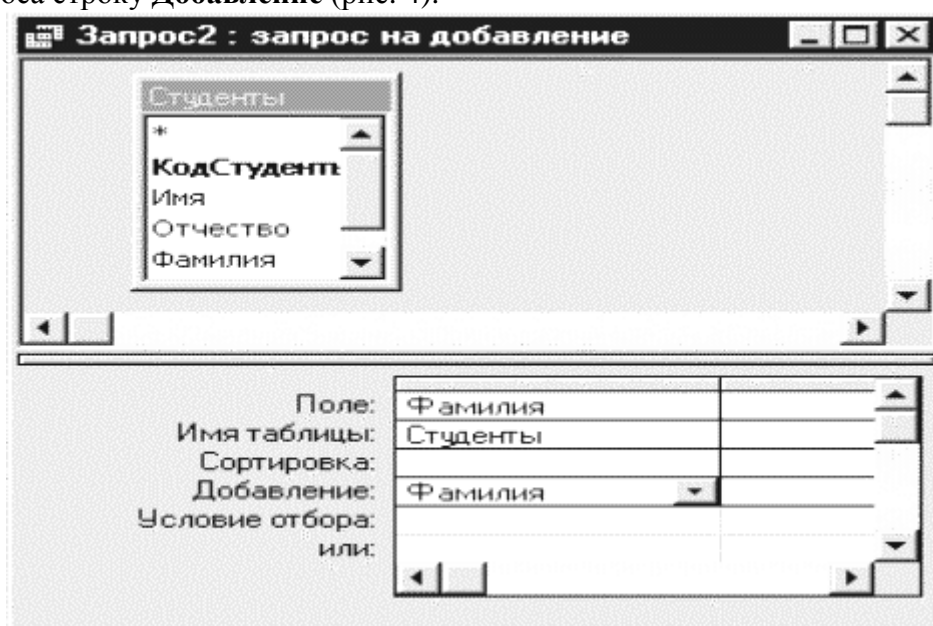


Рис. 4. Вид окна запроса на добавление записей

В строку **Добавление** автоматически (или вручную) вставляются имена тех полей целевой таблицы, которые совпадают с именами полей запроса. Далее следует выполнить запрос. Для этого выполните команду **Запрос/Выполнить** или щелкните по кнопке **Выполнить** на панели инструментов. В специальном диалоговом окне Access укажет сколько записей будет добавлено к целевой таблице, и потребует подтвердить выполнение этой операции.

Однако при переключении на запрос-выборку и обратно Access не сохраняет соответствия имен полей. Так что если перед запуском соответствующего запроса на добавление записей вы хотите выполнить запрос-выборку, приходится заново задавать соответствия имен двух таблиц.

Добавленные по ошибке записи легко находятся и удаляются. Тем не менее, можно сэкономить время, если сделать резервную копию таблицы, в которой будет осуществляться добавление записей.

3.7. 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 преобразования итогового запроса-выборки необходимо перейти в режим его **Конструктора** и выполнить команду **Запрос/Перекрестный**.

После этого в бланке запроса строка **Вывод на экран** будет заменена на строку **Перекрестная таблица**. В этой строке нужно задать, как будет использовано данное поле в перекрестной таблице: в виде заголовков строк, полей или значений (рисунок 5).

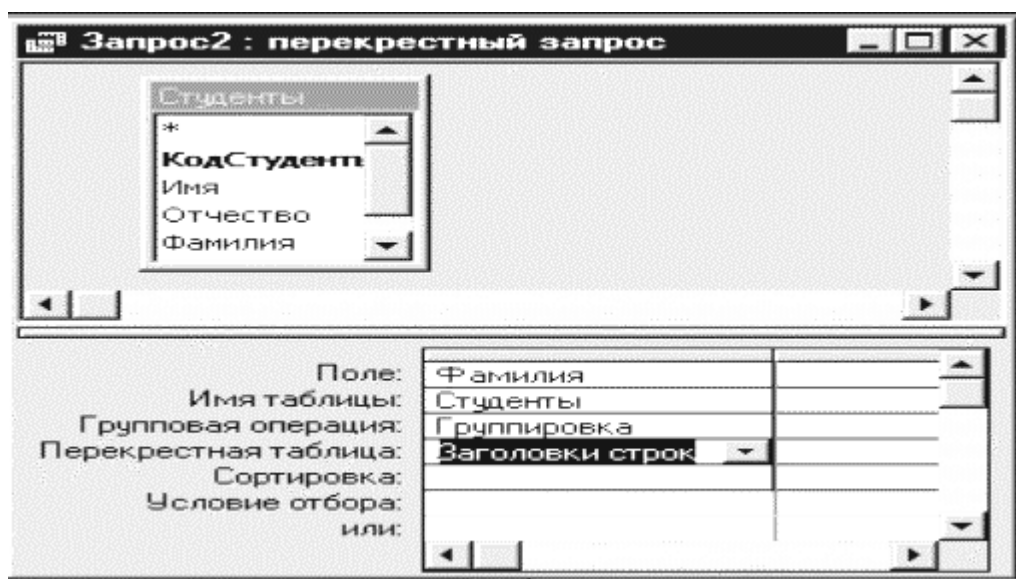


Рис.5. Вид окна перекрестного запроса

Существует возможность вывести данные в перекрестной таблице без создания в базе данных отдельного запроса. Для этого следует использовать Мастер сводных данных. В сводной таблице пользователь имеет возможность изменять заголовки строк или столбцов, что позволяет анализировать данные различными способами.

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.

3.8. Использование Мастера для создания перекрестной таблицы

Можно создать перекрестную таблицу в интерактивном режиме с помощью мастера «Перекрестный запрос». Обращение к **Мастеру** значительно упрощает процесс создания перекрестной таблицы. **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** 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.

При создании перекрестной таблицы предстоит выполнить следующие действия:

1) Перейти на вкладку «**Запросы**» и выбрать кнопку **Создать**, а затем в окне диалога выбрать опцию **Перекрестный запрос** (рис. 6).

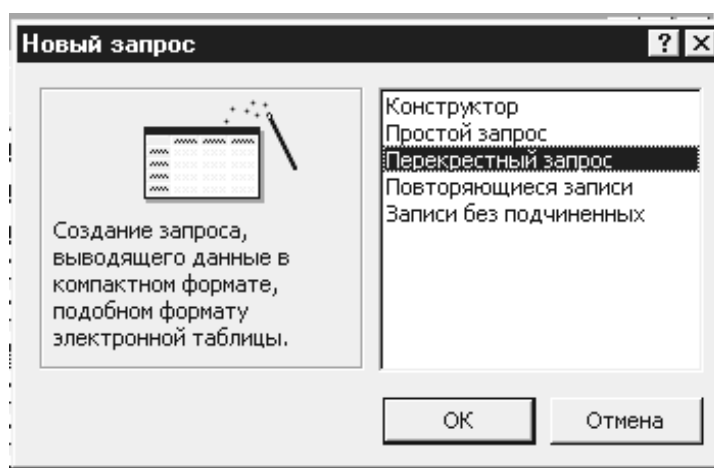


Рис. 6. Вид окна диалога «Новый запрос»

Эти действия запустят **Мастер** создания перекрестной таблицы, который шаг за шагом будет задавать вопросы о параметрах создаваемой таблицы.

На первом шаге Мастер предлагает выбрать исходную таблицу или запрос.

На следующем шаге вам необходимо указать поля, значения которых будут использоваться в качестве заголовков строк. Можно указать от одного до трех полей путем их переноса из списка **Доступные поля** в список **Выбранные поля**. Затем вы переходите в окно диалога определения заголовков столбцов. В качестве заголовка столбцов может использоваться только одно поле.

На предпоследнем шаге создания перекрестной таблицы нужно задать поле, которое будет использоваться в качестве источника данных для итоговых вычислений, и тип итоговых вычислений для каждой пары строка/столбец

В этом же окне диалога можно добавить столбец, который будет содержать итоговое значение по каждой строке результирующей таблицы.

На завершающей стадии создания перекрестной таблицы нужно задать ее имя и выбрать опцию просмотра результатов запроса или изменения структуры запроса в окне конструктора запросов.

Для завершения процесса создания необходимо нажать кнопку Готово, на экране появится построенная перекрестная таблица.

В перекрестной таблице, созданной с помощью Мастера, присутствует дополнительный столбец Итого, в котором отображаются суммарные итоги данных.

3.9. Порядок выполнения работы

При работе с запросами-действиями можно использовать созданные в предыдущих лабораторных работах таблицы (**Студенты, Экзамены**).

Перед выполнением запросов-действий не забудьте создать резервную копию таблицы, чтобы не повредить данные в исходной таблице. Запросы на удаление и обновление записей делайте только в ней!

Запрос 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 для этого запроса и преобразуйте его в запрос для сохранения результатов в виде отдельной таблицы с именем **Студенты без стипендии**. Для этого нужно в режиме **Конструктора** запросов ввести команду **Запрос/Создание таблицы**.

3. Откройте любой итоговый запрос (созданный в предыдущих лабораторных работах) в режиме **Конструктора** и также преобразуйте его в запрос с сохранением данных в отдельной таблице.

Запрос на обновление записей

4. Выберите в копии таблицы **Студенты** всех, кто проживает в определенном городе, например, Уфе. Сделав запрос, проверьте правильность отбора записей.

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

Для просмотра результатов выполнения запроса на обновление записей перейдите в режим просмотра таблицы **Студенты**.

6. Создайте запрос-обновление для таблицы **Студенты**, выбрав всех студентов определенного курса и измените курс на другой.

Запрос на удаление записей

7. Создайте запрос к копии таблицы **Студенты**, в которой будут удалены все записи о студентах определенного курса, не получающих никакой стипендии (например, удалить все записи о студентах III курса, не получающих стипендию).

Перекрестный запрос

8. Создайте перекрестный запрос для таблицы **Студенты**, включив в него поля **Город**, **Факультет** и вычисляемое поле для подсчета количества студентов (функция **Count** по полю **Номер зачетки**). Поле **Факультет** выберите в качестве заголовка строк, поле **Город** - заголовка столбцов, а вычисляемое поле с количеством студентов – как значения.

Выполните запрос и проанализируйте результат.

3.10. Требования к отчету

Отчет о проделанной работе должен содержать:

1. - название работы, ее задачи и последовательность выполнения;
2. - ответы на контрольные вопросы.

3.11. Контрольные вопросы

1. Типы запросов в Access. Как выбрать тип запроса?
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 удаления?
5. Как создать таблицу для постоянного хранения результатов определенного запроса?
6. Как посмотреть результат запроса?
7. Как можно изменить значение выбранного поля по всем записям таблицы, удовлетворяющим определенным условиям?

Лабораторная работа 4

ТЕМА: ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ. РАЗРАБОТКА ER-МОДЕЛИ. СОЗДАНИЕ БД, РАЗРАБОТКА ЗАПРОСОВ К БД.

4.1. Цель данной работы состоит в получении навыков в проектировании БД и построении запросов к ней в соответствии со своим индивидуальным вариантом задания.

4.2. Вопросы, подлежащие изучению

1. Основные понятия и определения баз данных. БД реляционного типа.
2. Принципы построения БД реляционного типа. Инфологическое проектирование и модель «Сущность-связь».
3. Создание реляционной БД в среде СУБД Access.
4. Построение запросов к БД.

4.3. Задание

1. Просмотреть п. 4.4. Предварительно изучить основные понятия, определения и способы работы с БД в среде Access. Изучить на примере способы: создания БД, работы с записями БД, формирования различных запросов в БД, создания формы для ввода данных в БД.
2. Выбрать вариант задания (п. 4.5.) и выполнить следующие действия:
 - изучить предметную область проектируемой БД и уточнить задание, определив данные, которые необходимо хранить в БД;
 - разработать модель «Сущность-связь» - ER-модель проектируемой БД (для этого необходимо выявить необходимый набор сущностей-таблиц, определить требуемый набор атрибутов для каждой сущности, определить ключи и связи между сущностями, формализовать связи);
 - создать новую БД в Access;
 - создать три таблицы, соответствующие сущностям ER модели БД;
 - создать схему БД, определив две связи типа «один-ко-многим» между двумя сущностями-таблицами и таблицей-связкой, а затем заполнить таблицы данными (заполнять необходимо сначала основные таблицы, а затем таблицу сущность-связку в режиме Конструктора таблиц, причем количество данных в таблицах должно обеспечивать выдачу не менее 3-5 записей по запросу задания);
 - создать запросы либо с помощью Конструктора запросов либо с помощью соответствующего Мастера запросов, а затем их выполнить;
 - создать форму для ввода данных в БД и с ее помощью ввести несколько записей в БД, а затем их выполнить;
 - посмотреть запросы в режиме SQL.

Содержание отчета

1. Тема и название работы. Индивидуальное задание на разработку БД.
2. Формализация и уточнение задания.
3. ER модель БД.
4. Структуры таблиц БД. Содержимое таблиц БД.
5. Формы для ввода данных в БД.
6. Запросы и результаты запросов к БД (запросы представить в визуальном виде (в форме QBE) и на языке SQL).

4.4. Пример разработки ER-модели БД «Экзамены»

БД должна содержать список студентов и результаты экзаменов каждого студента.

Уточнение задания.

Данные о каждом студенте должны содержать номер зачетки, фамилию, имя, отчество, название факультета, курс, номер группы, дату рождения, размер стипендии. Данные об экзаменах: название предмета, оценка, дата сдачи экзамена.

Сущности:

Опишем сущности на языке инфологического моделирования:

СТУДЕНТЫ (*НомерЗачетки, Фамилия, Имя, Отчество, Факультет, Курс, Группа, ДатаРождения, Стипендия*)

ЭКЗАМЕНЫ (*Предмет, Оценка, Дата сдачи*)

Ключи:

В сущности **СТУДЕНТЫ** ключевой атрибут – НомерЗачетки.

Связи:

Один студент (один экземпляр сущности **СТУДЕНТЫ**) может сдавать несколько экзаменов, поэтому один студент связан со многими экзаменами. Один факт сдачи одного экзамена (один экземпляр сущности **ЭКЗАМЕНЫ**) связан только с одним студентом. Таким образом, связь между сущностями **СТУДЕНТЫ** и **ЭКЗАМЕНЫ** «один-ко-многим».

Формализация связей:

Класс принадлежности со стороны сущности **ЭКЗАМЕНЫ** – обязательный (факт сдачи экзамена обязательно должен быть связан с соответствующим студентом из сущности **СТУДЕНТЫ**), поэтому ключ Номер зачетки сущности **СТУДЕНТЫ** должен присутствовать как внешний в сущности **ЭКЗАМЕНЫ**. Таким образом, сущность **ЭКЗАМЕНЫ** на языке инфологического моделирования должна быть следующей

ЭКЗАМЕНЫ (*НомерЗачетки, Предмет, Оценка, Дата сдачи*)

Таким образом, ER-модель разрабатываемой БД выглядит как на рис. 4.3-1.

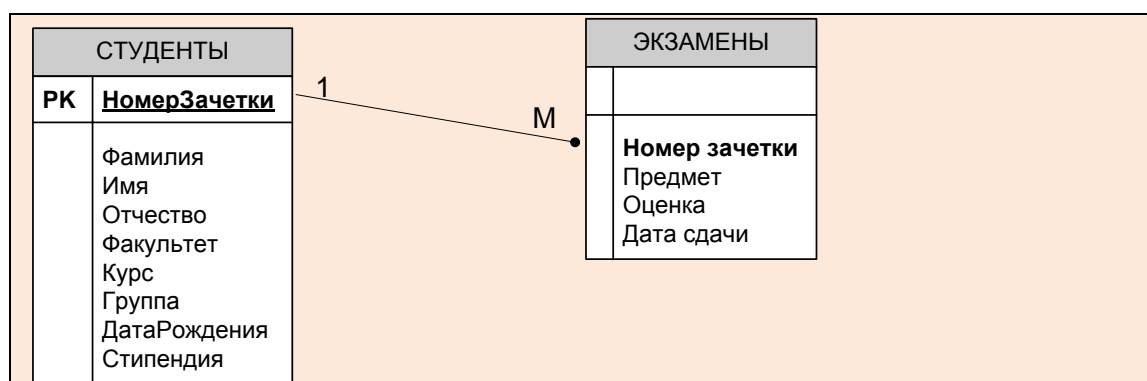


Рис. 4.3-1

4.5. Варианты заданий

Таблица 4.1.

№ вар.	Название БД	Таблицы	Примечание	Запросы
1	Банк	<p>БАНК (<u>код банка</u>, название, адрес)</p> <p>КРЕДИТ (<u>код банка</u>, номер клиента, дата выплаты кредита, размер кредита).</p> <p>КЛИЕНТ (<u>номер клиента</u>, ФИО, телефон клиента).</p>	Банк имеет много клиентов и один клиент может брать кредиты в разных банках.	<p>1. Выбрать банки, которые не имеют клиентов.</p> <p>2. Выбрать клиентов, которые имеют задолженность более месяца.</p>
2	ВУЗ	<p>ПРЕПОДАВАТЕЛЬ (<u>номер преподавателя</u>, ФИО, адрес и должность преподавателя, ученая степень).</p> <p>ЗАНЯТИЯ (<u>номер преподавателя</u>, код дисциплины, номер аудитории, дата)</p> <p>ДИСЦИПЛИНА (<u>код дисциплины</u>, название, количество часов).</p>	Один преподаватель может вести несколько дисциплин и одна дисциплина может вестись несколькими преподавателями.	<p>1. Выбрать преподавателей, которые были "без работы" в определенный период времени.</p> <p>2. Выбрать дисциплину, количество часов которой наибольшее.</p>
3	Склад	<p>ТОВАР (<u>код товара</u>, название товара, количество на складе, единица измерения, стоимость единицы товара).</p> <p>ПОСТАВКА (<u>код поставщика</u>, код товара, дата поставки, количество товаров в поставке).</p> <p>ПОСТАВЩИК (<u>код поставщика</u>, ФИО).</p>	Один и тот же товар может доставляться несколькими поставщиками и один и тот же поставщик может доставлять несколько видов товаров.	<p>1. Выбрать поставщиков, которые поставляют данный товар.</p> <p>2. Определить самый дешевый товар и соответствующего поставщика.</p>
4	Производство	<p>ИЗДЕЛИЕ (<u>код изделия</u>, название изделия, объем выпуска).</p> <p>СПЕЦИФИКАЦИЯ МАТЕРИАЛОВ (<u>код изделия</u>, код материала, количество материала в спецификации изделия, дата установления спецификации).</p> <p>МАТЕРИАЛ (<u>код материала</u>, название, единица измерения материала, цена за единицу).</p>	Одно изделие может содержать много типов материалов и один и тот же материал может входить в состав разных изделий.	<p>1. Определить изделия, в которые входит соответствующий материал.</p> <p>2. Вывести список изделий, которые не производились в 2000 г.</p>

5	Сеть магазинов	ВЛАДЕЛЕЦ (<u>номер владельца</u> , ФИО, адрес, телефон владельца магазина, номер регистрации, дата регистрации). ВЛАДЕНИЕ (<u>номер владельца</u> , номер магазина, дата регистрации). МАГАЗИН (<u>номер магазина</u> , название, адрес магазина, профиль).	Один и тот же магазин может иметь несколько владельцев и один и тот же владелец может иметь в собственности много магазинов.	1.Определить владельцев, имеющих собственность в соответствующем районе. 2.Вывести список профилей магазинов, которыми владеет соответствующий предприниматель
6	Автомас-терские	АВТОМЕХАНИК (<u>номер автомеханика</u> , ФИО). НАРЯД (<u>номер автомобиля</u> , номер автомеханика, название, адрес ремонтной мастерской). АВТОМОБИЛЬ (<u>номер</u> , марка, год выпуска, ФИО автовладельца).	Один и тот же автомобиль может обслуживаться разными автомеханикам и и один и тот же автомеханик может обслуживать несколько автомобилей.	1.Выбрать механиков, которые работают с автомобилями заданной марки. 2.Определить тех владельцев автомобилей, которых обслуживает данный механик.
7	Деканат	ГРУППА (<u>код группы</u> , курс, количество студентов). СЕССИЯ (<u>код группы</u> , <u>код дисциплины</u> , ФИО преподавателя, вид контроля, дата). ДИСЦИПЛИНА (<u>код дисциплины</u> , название дисциплины, количество часов).	Одна группа изучает несколько дисциплин и одна дисциплина может преподаваться нескольким группам.	1.Определить преподавателя, который в сессию принимает экзамены у данной группы по данной дисциплине. 2.Определить группу, у которой число студентов наибольшее.
8	Организация	РАБОТА (<u>шифр работы</u> , название, трудоемкость). ПОРУЧЕНИЕ (<u>шифр работы</u> , <u>табельный номер сотрудника</u> , плановая и реальная даты окончаний). СОТРУДНИК (<u>табельный номер</u> , ФИО сотрудника, должность).	Одна и та же работа может выполняться несколькими сотрудниками и один и тот же сотрудник может участвовать в нескольких работах.	1.Определить те работы в 2000 г., по которым сроки выполнения превышают один месяц. 2.Определить общее количество работ, находившихся на выполнении у некоторого сотрудника.

9	Поликлиника	ПАЦИЕНТ (<u>номер</u> , ФИО, дата рождения, социальный статус) ЛЕЧЕНИЕ (<u>номер пациента</u> , <u>номер врача</u> , диагноз, срок потери трудоспособности, дата начала лечения). ВРАЧ (<u>номер врача</u> , ФИО, должность, специализация).	Один и тот же пациент может лечиться у нескольких врачей и один врач может лечить несколько пациентов.	1.Определить те случаи, когда конкретное заболевание лечилось врачом конкретной специализации. 2.Вывести имена тех врачей, которые работают исключительно с пенсионерами.
10	Спорт	ВИД СПОРТА (<u>название вида спорта</u> , единица измерения, мировой рекорд и его дата). РЕЗУЛЬТАТ (<u>название вида спорта</u> , код спортсмена, показанный результат, дата проведения, место проведения). СПОРТСМЕН (<u>код спортсмена</u> , ФИО спортсмена, год рождения).	Один и тот же спортсмен может заниматься разными видами спорта	1.Вывести виды спорта для каждого спортсмена. 2.Вывести список спортсменов, превысивших мировой рекорд по данному виду спорта.
11	Сельскохозяйственные работы	ПРЕДПРИЯТИЕ (<u>код предприятия</u> , наименование, дата регистрации, вид собственности, число работников, основной вид продукции). ПОСТАВКА (<u>код продукции</u> , <u>код предприятия</u> , дата поставки, объем). ПРОДУКЦИЯ (<u>код продукции</u> , название продукции, единица измерения, закупочная цена).	Одно и то же предприятие может выпускать разные виды продукции и одна и та же продукция может выпускаться разными предприятиями	1.Вывести все данные по конкретному предприятию. 2.Вывести список продуктов и их дату поставки, для которых закупочная цена ниже заданной.
12	Городской транспорт	ВИД ТРАНСПОРТА (<u>название вида транспорта</u> , средняя скорость движения, количество машин в парке, стоимость проезда). МАРШРУТ (<u>номер пути</u> , <u>название вида транспорта</u> , количество остановок на маршруте, количество машин на маршруте). ПУТЬ (<u>номер пути</u> , начальный пункт пути, конечный пункт, расстояние).	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.Вывести данные по маршруту, у которого максимальное количество видов транспорта. 2.Вывести данные по маршрутам трамваев в порядке убывания их протяженности.

13	География	СТРАНА (<u>название страны</u> , регион, столица, площадь территории). НАСЕЛЕНИЕ (<u>название страны</u> , <u>национальность</u> , количество людей данной национальности). НАЦИОНАЛЬНОСТЬ (<u>национальность</u> , язык, общая численность).		1.Выбрать численность населения по всем странам. 2.Определить столицу той страны, где проживает более всего представителей конкретной национальности.
14	Домоуправление	КВАРТИРА (<u>номер квартиры</u> , <u>номер дома</u> , число жильцов, площадь). ОПЛАТА (<u>номер квартиры</u> , <u>номер дома</u> , <u>вид оплаты</u> , сумма оплаты, дата оплаты). ВИД ОПЛАТЫ (<u>вид оплаты</u> , цена за единицу площади, цена за одного жильца).	В одной квартире используются разные виды оплаты.	1.Определить сумму месячной оплаты для данной квартиры конкретного дома. 2.Определить задолженность по оплате конкретного вида конкретной квартиры.
15	Аэропорт	САМОЛЕТ (<u>номер самолета</u> , тип, число мест, скорость полета). РЕЙС (<u>номер самолета</u> , <u>номер маршрута</u> , дата и время вылета, дата и время прилета, количество проданных билетов). МАРШРУТ (<u>номер маршрута</u> , расстояние, пункт вылета, пункт назначения).	Один самолет может летать на разных маршрутах и по одному маршруту могут летать разные самолеты.	1.Определить время полета для конкретного самолета по конкретному маршруту. 2.Выбрать тип самолета, который чаще всего летает по конкретному маршруту.
16	Компьютеры	ФИРМА (<u>номер фирмы</u> , название и место размещения фирмы). РЫНОЧНОЕ ПРЕДЛОЖЕНИЕ (<u>номер фирмы</u> , <u>номер конфигурации</u> , объем партии, цена партии). КОНФИГУРАЦИЯ ПК (<u>номер конфигурации</u> , тип процессора, тактовая частота, объем ОЗУ, объем жесткого диска, дата выпуска).	Одна конфигурация ПК может изготавливаться разными фирмами и одна и та же фирма может собирать ПК разной конфигурации.	1.Определить фирму, которая представляет самую новую модель на базе конкретного типа процессора. 2.Выбрать модель с наибольшей тактовой частотой, которая выпускается в конкретном городе.

17	Личные данные	<p>СТУДЕНТ (<u>номер зачетной книжки</u>, ФИО студента, дата рождения студента).</p> <p>ОБУЧЕНИЕ (<u>название факультета</u>, <u>номер зачетной книжки</u>, номер группы, сумма стипендии студента, год зачисления).</p> <p>ФАКУЛЬТЕТ (<u>название факультета</u>, число мест на факультете).</p>	Один и тот же студент может обучаться на разных факультетах.	<p>1. Определить количество студентов на конкретном факультете.</p> <p>2. Определить суммарную стипендию конкретной группы.</p>
18	Зоопарк	<p>ВИД ЖИВОТНОГО (<u>вид животного</u>, суточное потребление корма, семейство, континент обитания).</p> <p>РАЗМЕЩЕНИЕ (<u>вид животного</u>, <u>номер помещения</u>, количество животных)</p> <p>ПОМЕЩЕНИЕ (<u>номер помещения</u>, наличие водоема, отопления в помещении).</p>	Один и тот же вид животного может в зоопарке находиться в разных помещениях и в одном помещении может находиться несколько видов животных.	<p>1. Определить суточное потребление корма обитателями конкретного помещения.</p> <p>2. Определить общую численность представителей конкретного вида животных.</p>
19	Шахматы	<p>ШАХМАТИСТ (<u>код спортсмена</u>, ФИО, дата рождения, страна, спортивный разряд, рейтинг).</p> <p>УЧАСТИЕ (<u>код спортсмена</u>, <u>код турнира</u>, стартовый номер спортсмена в данном турнире, занятое место).</p> <p>ТУРНИР (<u>код турнира</u>, страна, город, дата проведения, уровень турнира).</p>	Один шахматист может участвовать в разных турнирах.	<p>1. Выбрать турнир с самым высоким рейтингом участников.</p> <p>2. Выбрать тех шахматистов, которые выиграли хотя бы один турнир течение конкретного года.</p>
20	Судоходство	<p>КОРАБЛЬ (<u>название корабля</u>, водоизмещение, порт приписки, капитан).</p> <p>ПОСЕЩЕНИЕ ПОРТА (<u>название корабля</u>, <u>название порта</u>, дата посещения порта, дата убытия, номер причала, цель посещения).</p> <p>ПОРТ (<u>название порта</u>, страна).</p>	Один корабль может посещать несколько портов.	<p>1. Выбрать список кораблей, посещавших конкретный город в конкретное время.</p> <p>2. Определить страны, в которые никогда не приходят корабли с целью 'туризм'.</p>

21	Научные конференции	<p>УЧЕНЫЙ (<u>имя ученого</u>, организация, страна, ученая степень).</p> <p>УЧАСТИЕ (<u>имя ученого</u>, <u>название конференции</u>, тип участия, тема доклада, публикация (да/нет)).</p> <p>КОНФЕРЕНЦИЯ (<u>название конференций</u>, место проведения, дата).</p>	<p>Один ученый может участвовать в разных конференциях и с разными типами докладов.</p> <p>Тип участия: доклад, сообщение, стендовый доклад, оргкомитет и т.д.</p>	<p>1. Выбрать названия конференций, материалы которых не опубликованы.</p> <p>2. Определить количество докторов наук в каждой конференции.</p>
22	Программные продукты	<p>ПРОГРАММНЫЙ ПРОДУКТ (<u>название программного продукта</u>, версия, фирма, дата выпуска, прикладная область, стоимость лицензии).</p> <p>ИНСТАЛЛЯЦИЯ (<u>название программного продукта</u>, <u>название пользователя</u>, стоимость инсталляции, дата инсталляции, количество лицензий при инсталляции).</p> <p>ПОЛЬЗОВАТЕЛЬ (<u>название пользователя</u>, регион, сфера применения).</p>	<p>Один и тот же программный продукт может устанавливаться разными пользователями и один пользователь может установить разные продукты.</p> <p>Прикладная область: делопроизводство, коммерция, бизнес и т.д.</p> <p>Тип программного продукта: ОС, сервер приложений, СУБД, Web-сервер и т.п.</p>	<p>1. Вывести данные по прикладным областям.</p> <p>2. Определить затраты на инсталляции программного обеспечения, сделанные конкретным пользователем в конкретном году.</p>

23	Операционная система	<p>ПРОЦЕСС (<u>идентификатор процесса</u>, идентификатор владельца процесса, название процесса, приоритет, класс, идентификатор владельца).</p> <p>ПЛАНИРОВАНИЕ (<u>идентификатор процесса</u>, <u>название ресурса</u>, запрошено/выделено).</p> <p>РЕСУРС (<u>название ресурса</u>, количество, цена за единицу).</p>	<p>Один и тот же процесс может задействовать много разных ресурсов и один и тот же ресурс может быть задействован разными процессами. Классы процессов : критический, серверный, нормальный.</p>	<p>1. Вывести данные по ресурсам для каждого процесса.</p> <p>2. Определить владельца процесса, у которого "самые большие аппетиты" в ценовом выражении.</p>
24	Добыча полезных ископаемых	<p>ПОЛЕЗНОЕ ИСКОПАЕМОЕ (<u>название полезного ископаемого</u>, единица измерения, годовая потребность, цена за единицу).</p> <p>МЕСТОРОЖДЕНИЕ (<u>название полезного ископаемого</u>, <u>название пункта</u>, название месторождения, запасы, способ разработки, годовая добыча).</p> <p>ПУНКТ (<u>название пункта</u>, пропускная способность путей сообщения данного пункта).</p>	<p>Из одного и того же пункта вывоза могут вывозиться разные полезные ископаемые, одно и то же полезное ископаемое может вывозиться с разных пунктов вывоза. Конкретное месторождение может использовать конкретный пункт вывоза конкретного полезного ископаемого.</p>	<p>1. Определить, какие ископаемые добываются в конкретном месторождении.</p> <p>2. Выбрать список месторождений, в которых добывается конкретное ископаемое заданным способом разработки.</p>
25	Автотранспортное предприятие	<p>АВТОМОБИЛЬ (<u>номер автомобиля</u>, марка автомобиля, его техническое состояние, пробег, расход топлива).</p> <p>РЕЙС (<u>номер автомобиля</u>, <u>табельный номер водителя</u>, дата выезда, дата прибытия, расстояние, расход горючего).</p> <p>ВОДИТЕЛЬ (<u>табельный номер водителя</u>, ФИО, дата рождения, оклад).</p>	<p>Один и тот же автомобиль может использоваться разными водителями и один водитель может использовать разные автомобили.</p>	<p>1. Выбрать автомобиль с наименьшим расходом горючего.</p> <p>2. Вывести данные по всем рейсам, в которых участвовал данный автомобиль.</p>

26	Театр	АКТЕР (<u>номер актера</u> , ФИО, звание, амплуа, пол). ПОСТАНОВКА НА РОЛЬ (<u>номер актера</u> , роль, дата назначения на роль, тип роли, режиссер, номер состава); РОЛЬ (<u>роль</u> , тип роли, название пьесы).	На одну и ту же роль могут назначаться разные актеры. Амплуа: герой-любовник, инженерю, злодей т.д., тип роли: главная, вторая, эпизод.	1.Определить список актеров конкретного режиссера. 2.Выбрать пьесы, в которых исполнителем главной роли является конкретный актер.
27	Справочная аптек	ЛЕКАРСТВО (<u>название лекарства</u> , показания к использованию, противопоказания, производитель). НАЛИЧИЕ (<u>название лекарства</u> , <u>номер аптеки</u> , тип, цена, количество, срок годности). АПТЕКА (<u>номер аптеки</u> , специализация аптеки, район, телефон).	Тип: таблетки, микстура, мазь и т.д.	1.Вывести ассортимент предлагаемых лекарств жителей конкретного района. 2.Определить, в каких аптеках дешевле всего конкретное лекарство.
28	Кулинария	БЛЮДО (<u>название блюда</u> , категория, рецепт, вес порции). СОСТАВ (<u>название блюда</u> , <u>название продукта</u> , количество, обработка). ПРОДУКТ (<u>название продукта</u> , категория, калорийность, цена за ед., ед. измерения).	Одно блюдо состоит из разных продуктов и один и тот же продукт может входить в состав разных блюд. Категория блюда: первое, гарнир, и т.д., продукта: мясо, мучное и т.д.	1.Определить, для каких блюд продукты данной категории предварительно подвергаются заданной обработке. 2.Определить данные по соответствующему блюду.
29	Изучение студентами дисциплин по выбору	СТУДЕНТ (<u>номер зачетной книжки</u> , ФИО студента, адрес, телефон). ВЫБОР (<u>номер зачетной книжки</u> , <u>наименование дисциплины</u> , деканат, адрес деканата). ДИСЦИПЛИНА (<u>наименование дисциплины</u> , количество лекционных часов, семинарских и лабораторных занятий, отметка о сдаче дисциплины).	Каждый студент должен изучить несколько дисциплин по выбору и каждая дисциплина может изучаться несколькими студентами.	1.Получить список студентов, изучающих заданную дисциплину и сдавших ее. 2.Получить список дисциплин, изученных заданным студентом и объем дисциплины.

30	Отдел кадров	СОТРУДНИК (<u>номер сотрудника</u> , ФИО, образование, телефон, дата рождения). ЗАЧИСЛЕНИЕ (<u>номер сотрудника</u> , <u>наименование подразделения</u> , должность, дата зачисления, оклад). ПОДРАЗДЕЛЕНИЕ (<u>наименование подразделения</u> , количество штатных единиц, фонд заработной платы).	Один и тот же сотрудник может числиться в разных подразделениях	1. Выбрать сотрудников, которые занимают инженерные должности, не имея высшего образования. 2. Определить, в каких подразделениях работает данный сотрудник.
----	--------------	--	---	---

Лабораторная работа 5

ТЕМА: СОЗДАНИЕ ОТЧЕТОВ

5.1. Цель работы: освоить приемы создания отчетов и графического представления данных в СУБД ACCESS

5.2. 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 и вычисления промежуточных и общих итогов для больших наборов данных;
- отчеты могут быть использованы для получения красиво оформленных материалов, наклеек и других документов.

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

Для создания отчета в окне базы данных выбираем пункт **Отчет** в раскрывающемся списке кнопки **Новый объект** на панели инструментов, или – кнопку **Создать** на вкладке **Отчеты**. Access выведет на экран окно диалога **Новый отчет**.

Access позволяет выбирать имя источника данных в поле со списком в нижней части окна диалога.

5.3. Выбор мастера по разработке отчетов

В окне диалога **Новый отчет** можно выбрать один из шести вариантов. Конструктор позволяет создавать отчет «с нуля» и редактировать уже созданный отчет.

Мастер отчетов запускает основного мастера для создания отчетов, позволяющего выбрать поля для отчета, задать форматы, условия группировки и итоговые функции.

Автоотчет: в столбец - этот мастер создает отчет, в котором каждая запись базового запроса или таблицы представлена в виде простого списка, содержащего названия и значения полей.

Автоотчет: ленточный - в отчете этого типа данные записи базового запроса или таблицы выводятся в одной строке. Если таблицы базового запроса связаны отношением «один-ко-многим», то мастер создает группу для данных, поступающих со стороны отношения «один», но не вычисляет никаких итоговых значений. При создании отчета используется последний стиль, который был выбран в мастере отчетов или применен с помощью автоформата в конструкторе отчетов.

Мастер диаграмм - этот мастер помогает построить диаграмму и создает в отчете свободную рамку объекта OLE с внедренной диаграммой Microsoft Graph.

Почтовые наклейки - этот мастер позволяет выбрать поля с именами и адресами, отформатировать их и создать отчет для печати почтовых наклеек. Вы можете выбрать любой тип и настроить размер наклеек.

5.4. Разделы отчета

Пустой макет отчета содержит разделы Верхний колонтитул, Нижний колонтитул, между которыми находится Область данных.

Добавить или удалить верхний и нижний колонтитулы можно с помощью команды Вид - Колонтитулы. При создании отчета можно задать заголовок, который

будет напечатан только в самом начале отчета, и примечание, выводимое в конце отчета. Чтобы добавить в отчет эти разделы, выберите команду Вид/ Заголовок/примечание отчета. Линейка с делениями по верхнему и левому краям окна отчета помогают планировать расположение данных на странице. Изменить размер любого отдела можно, перетащив его границу. В каждом разделе Access выводит сетку.

5.5. Окно отчета и его инструменты

Панель элементов содержит все кнопки для всех элементов управления, которые можно использовать при создании отчета. Это поля, надписи, рисунки, флажки и т.д. Чтобы открыть эту панель нужно выбрать команду Вид - Панель элементов, либо щелкнуть на кнопке Панель элементов на панели инструментов. Кнопка Другие элементы предоставляет доступ ко всем элементам управления ActiveX (в том числе к элементу управления Календарь) установленным в системе. Чтобы разместить в отчете конкретный элемент управления, щелкните на соответствующей кнопке на панели элементов, нажмите кнопку мыши, протащите указатель мыши до того места, где должен быть элемент, и отпустите кнопку. Список полей упрощает размещение в отчете присоединенных элементов управления (элементов управления, связанных с полями таблицы или запроса). Он открывается щелчком на кнопке Список полей на панели инструментов или выбором команды Вид - Список полей. В строке заголовка списка полей Access выводит имя базовой таблицы или запроса. Чтобы создать присоединенный элемент управления с помощью списка полей, выберите его тип, щелкнув на соответствующей кнопке на панели элементов. Затем перетащите поле из списка в нужное место отчета. Если кнопка не выбрана, типом элемента управления по умолчанию является Поле

В окне свойств можно установить свойства самого отчета, его разделов и элементов управления. Набор свойств, представленных в нем зависит от объекта. Чтобы открыть окно свойств, выделяем нужный объект и щелкаем на кнопке Свойства на панели инструментов или выбираем команду Вид / Свойства.

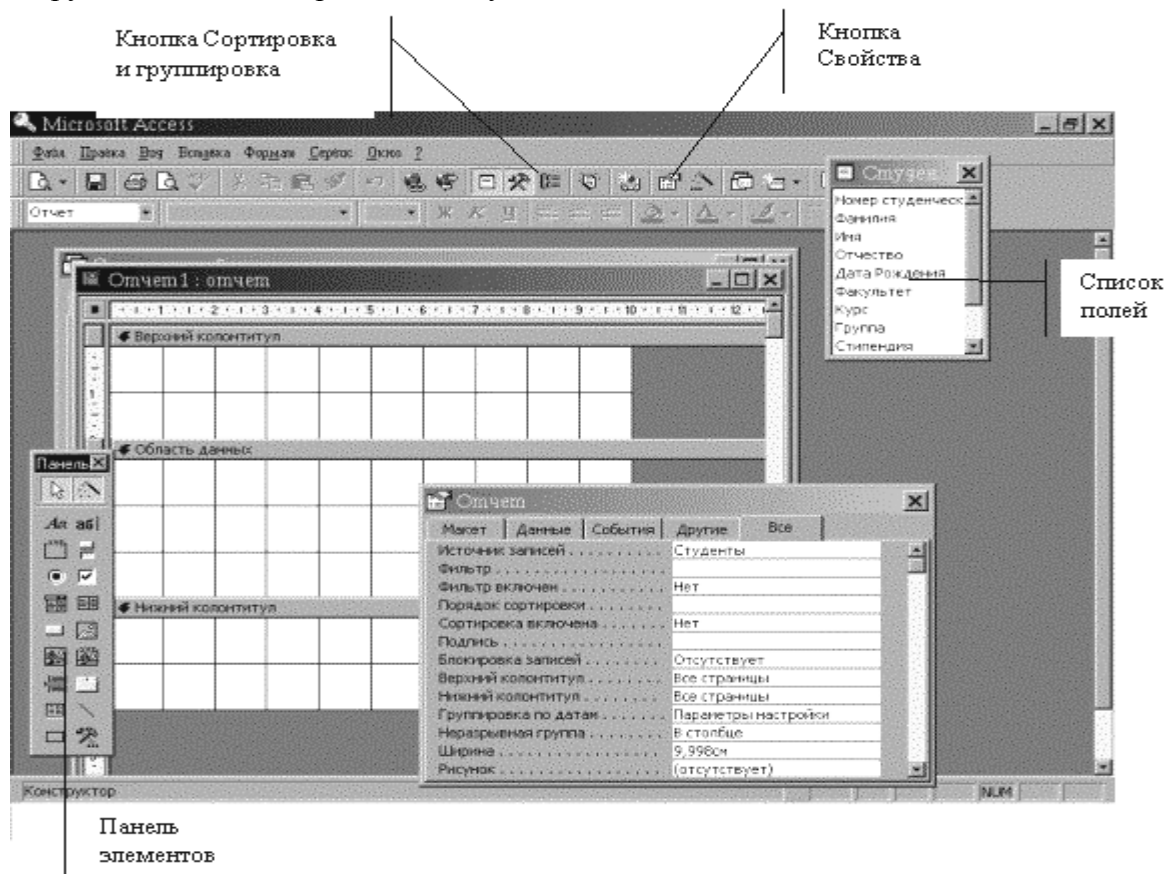


Рис. 1. Окно отчета

5.6. Сортировка и группировка

Отчеты предоставляют возможность группировать выводимую информацию с помощью окна **Сортировка и группировка**. Чтобы открыть его, щелкните на кнопку **Сортировка и группировка** на панели инструментов. В окне **Сортировка и группировка** можно определить до 10 полей или выражений, которые будут использоваться в отчете для группировки данных. Первый элемент списка определяет основную группу, а последующие подгруппы внутри групп.

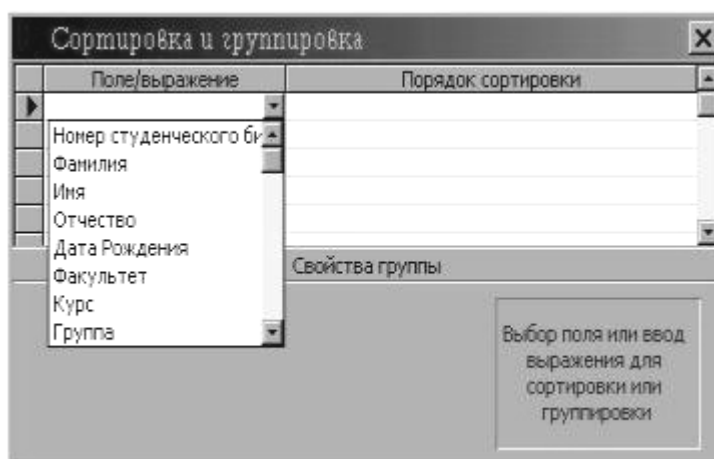


Рис 2. Окно Сортировка и группировка

В первой строке столбца **Поле/выражение** раскрывается список полей базовой таблицы или запроса. Начав ввод со знака равенства(=), можно ввести в этом столбце выражение, содержащее ссылку на любое поле базовой таблицы или запроса.

По умолчанию Access сортирует значения по возрастанию. Для изменения порядка сортировки выберите значение **По убыванию** в раскрывающемся списке, который появляется после щелчка в ячейке столбца **Порядок сортировки**.

В отчете должно быть зарезервировано место для заголовка каждой группы и примечания, в котором могут выводиться вычисляемые итоговые поля. Чтобы добавить в отчет эти разделы, установите для свойств **Заголовок группы** и **Примечание группы** значение **Да**. Кроме того, можно попросить Access выводить все детальные данные для данного уровня группировки на одной странице (если это возможно), выбрав для того же свойства значение **Вся группа**.

5.7. Задание диапазонов группировки

Для каждого поля или выражения, заданного в верхней части окна сортировки и группировки, можно установить свойства **Группировка** и **Интервал**. Обычно данные группируются по каждому значению поля или выражения. Но записи можно группировать не только по отдельным значениям, но и по диапазонам значений. Вид и размер диапазона зависят от типа данных поля или выражения.

При группировке по текстовому полю можно начинать новую группу при изменении первого или нескольких начальных символов значения поля. Например, чтобы образовать по одной группе для каждой буквы алфавита, определите создание новой группы при изменении первой буквы значения поля – таким образом, вы получите группу записей, в которых значение поля начинается с буквы А, группу записей со значением поля, начинающегося с буквы Б, и т.д. Чтобы задать группировку по первым символам, установите для свойства **Группировка** значение **По первым знакам**, а в ячейке свойства **Интервал** введите число начальных символов, по которым вы хотите образовывать группы.

Для чисел можно установить свойство **Группировка** в значение **Интервал**. В этом случае свойство **Интервал** определяет диапазоны группировки. Access вычисляет диапазоны, начиная с 0. Если задать интервал, равный 10, то будут формироваться группы значений от -20 до -11, от -10 до -1, от 0 до 9, от 10 до 19, от 20 до 29 и т.д.

Для полей даты/времени можно установить в качестве значения свойства **Группировка** такие единицы измерения как **По годам, По кварталам, По месяцам, По неделям, По дням, По часам, По минутам**. Для свойства **Интервал** укажите число единиц измерения, если вы хотите проводить группировку по определенному отрезку времени. Например, если вы хотите создать группы с интервалом в два года, установите для свойства **Группировка** значение **По годам**, а в ячейке свойства **Интервал** введите 2.

5.8. 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.

5.9. Добавление текущей даты и номера страницы

Для дат Microsoft Access предоставляет две встроенные функции, которые можно использовать для включения в отчет текущей даты и времени. Функция **Date** возвращает текущую системную дату и время в виде значения даты/времени.

Чтобы добавить текущую дату в отчет, создайте свободное поле в верхнем колонтитуле и введите выражение `= Now()` в ячейке его свойства **Данные**, а для свойства **Формат** поля установите значение **длинный формат даты**. Вернитесь в отчет и введите с клавиатуры в надпись поля содержательный текст или вовсе удалите ее.

Чтобы добавить номер страницы, используйте свойство отчета **Page**. Вы не увидите его в окне свойств, так как оно поддерживается Access на системном уровне. Кроме того, в Access существует свойство **Pages**, содержащее общее число страниц в отчете. Чтобы добавить в отчет номер текущей страницы, создайте свободное поле и в ячейке его свойства **Данные** введите выражение `= "Page " & [Page] & " of " & [Pages]`

5.10. Вычисления в области данных

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, а также функции, которые вы сами определили в модуле. При желании вы можете воспользоваться помощью построителя выражений. Чтобы сообщить Access, что в свойстве **Данные** вы хотите использовать выражение, начните его ввод со знака равенства(=).

Например, чтобы разместить в области данных поле **Надбавка**, выберите на панели элементов кнопку **Поле** и укажите ее размещение в области данных. Вызовите окно **Свойств** и в свойстве **Данные** введите `= [Стипендия]*0,5`, т.е. Надбавка составляет 50% для всех записей.

5.11. 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 используется выражение, построенное 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 Вам не удастся создать вычисляемое поле в области данных и сослаться на него в выражении, использующем итоговую функцию. Но вы можете 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 вы должны 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.

5.12. Объединение текстовых значений и скрытие повторяющихся значений

Когда в нескольких соседних строках отображаются одинаковые значения, отчет трудно читается и выглядит непрофессионально. Свойства поля **Не выводить повторы** (оно доступно только в отчетах) позволяет управлять выводом повторяющихся значений.

Иногда, не выводя повторы, мы можем скрыть нужные данные. Чтобы этого избежать, можно объединить значения соответствующих полей в одном элементе управления. Для объединения служит символ амперсанда(&), обозначающий операцию конкатенации при объединении текстовых значений. Например, =Format([Номер студенческого билета];"000000")&" "&[Фамилия]. Функция Format возвращает ровно шесть цифр. Вторая текстовая строка в этом выражении добавляет несколько пробелов между двумя полями данных.

5.13. 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, можем создать выражение для вычисления процентов от общих сумм или промежуточных сумм, рассчитанных для группировки более высокого уровня. Создав выражение для нахождения процентов от суммы, устанавливаем свойство **Формат поля** в значение **Процентный**. В режиме просмотра можем убедиться в том, что в итогах по группе отразится процент от общего итога.

5.14. 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, установите для него свойства **Данные** – значения поля, для которого вычисляется сумма, а для свойства **Сумма с накоплением** - значение **Для группы**.

5.15. Создание и внедрение подчиненных отчетов

Вывод в отчете детальных данных из нескольких таблиц, связанных отношением “один-ко-многим”, не представляет особого труда. Вы можете внедрять в область данных создаваемого отчета подчиненные отчеты для вывода нескольких детальных строк из таблицы или запроса, находящегося на стороне "многие" по отношению к текущей строке в области данных.

Для создания подчиненного отчета в окне базы данных выделите подчиненную таблицу или запрос и выберите пункт **Отчет** в раскрывающемся списке кнопки Новый объект на панели инструментов. В окне диалога **Новый отчет** выберите любой пункт и откройте отчет в нужном режиме. Обычно в подчиненном отчете не бывает колонтитулов.

Откройте основной отчет, т.е. тот, в который вы хотите внедрить подчиненный отчет, в области данных создайте элемент управления типа "подчиненный отчет". В качестве значения свойства **Объект-источник** введите имя отчета, который нужно внедрить, задайте связанные поля, т.е. выберите значения свойств **Основные поля** и **Подчиненные поля**. Кроме того, нужно установить для свойств расширение и сжатие значение **Да**, чтобы Access мог расширять или сжимать подчиненный отчет. Чтобы настроить размер внедренного отчета, выберите команду **Формат-Размер-По размеру данных** для этого элемента.

Упростить внедрение подчиненного отчета можно с помощью **Мастера** подчиненных отчетов и форм. Все необходимые параметры **Мастер** запрашивает в ходе диалога. Чтобы начать работу **Мастера** нажмите кнопку , а затем кнопку на панели элементов.

5.16. Порядок выполнения работы

1. Открыть базу данных, созданную на предшествующих лабораторных работах
2. Создать отчет №1 для таблицы **Студенты**;
 - В окне БД перейти на вкладку **Отчеты** и нажать кнопку **Создать**;
 - В качестве таблицы для создания отчета указать таблицу **Студенты**;
 - Выбрать режим создания - **Мастер отчетов**;
 - На первом шаге **Мастера отчетов** включить в отчет поля **Фамилия Имя Факультет Курс Группа Стипендия**;
 - На втором шаге задать уровень группировки - по полю **Факультет**;
 - На третьем шаге указать порядок сортировки записей в отчете: в первую очередь по полю **Курс**, во вторую - по полю **Фамилия**;
 - Нажать кнопку **Итоги 1/4** и выбрать в качестве итога сумму по полю **Стипендия**;
 - На четвертом и пятом шагах выбрать стиль оформления отчета;
 - На последнем шаге указать имя отчета «**Начисление стипендий по факультетам**».
3. Просмотреть отчет на экране, анализируя его содержимое (группировка записей, вид итогов, элементы страницы отчета и т.д.).
4. Перейти в режим Конструктора данного отчета (нажав кнопку **Конструктор** на панели инструментов) и произвести в отчете следующие изменения:
 - Отредактировать подпись в примечании группы **Факультет** таким образом, чтобы она приняла вид:
Итого стипендий по факультету [Название факультета]
 - Удалить название итогового поля **Sum**;
 - Перенести само итоговое поле на уровень его подписи.
5. Отредактированный отчет вновь просмотреть на экране и сохранить под именем **Начисление стипендий**.

6. Создать базовый запрос к таблицам **Студенты - Экзамены**, включив в него поля *Номер зачетки*, *Фамилия*, *Предмет*, *Оценка*. Сохранить запрос под именем **Данные для отчета по успеваемости**.
7. Создать отчет № 2 на основе запроса **Данные для отчета по успеваемости**:
 - В качестве режима создания выбрать **Мастер отчетов**;
 - На первом шаге Мастера выбрать все поля исходного запроса для включения в отчет;
 - На втором шаге выбрать тип представления данных - по таблице **Студенты**;
 - На третьем шаге добавить уровень группировки по полю *Номер зачетки*;
 - На четвертом шаге указать сортировку по полю *Предмет* и *Вид итогов* - Среднее (Avg) по полю *Оценка*;
 - На последующих шагах Мастера задать стиль оформления отчета и его название **Результаты экзаменационной сессии**.
8. Просмотреть отчет на экране, проанализировать его содержание (группировка, итоговые поля, области отчета).
9. Перейти в режим Конструктора для данного отчета и отредактировать его так, чтобы:
 - В примечании группы заголовков был таким:
Средний балл за текущую сессию;
 - Результат в итоговом поле находился непосредственно рядом с заголовком.
10. Добавить новое вычисляемое поле *Надбавка*. Для этого
 - В области верхнего колонтитула разместите надпись *Надбавка*;
 - В области данных разместите свободное поле и удалите присоединенную надпись для него;
 - Вызовите окно свойств и с помощью пунктов Построителя выражений - Функции управления в строке Данные введите =If([Avg _ Оценка]=5;150;If([Avg _ Оценка]>4 Or [Avg _ Оценка]=4;100;50)), т.е. надбавка начисляется в зависимости от результатов сессии.
11. Добавьте в базовый запрос **Данные для отчета по успеваемости** поле **Стипендия**.
12. Добавьте новое поле **Удержания**, учитывая, что удерживается 10% от значения Стипендии.
13. Для значения Свойства Не выводить повторы полей Номер студенческого билета, Фамилия установите Да.
14. В окне сортировки и группировки по полю **Оценка** задайте группировку по полному значению и Сортировку по убыванию, а для поля Стипендия группировку по интервалу 10 и сортировку по возрастанию.
15. Создать ленточный автоотчет по таблице **Экзамены** и сохранить его с именем **Экзамены**.
16. Создать ленточный автоотчет по таблице **Студенты**.
17. Открыть его в режиме Конструктора и отредактировать его следующим образом:
 - На панели элементов включить режим мастера и нажать кнопку **Подчиненная форма/отчет**;
 - Укажите указателем мыши место для размещения подчиненного отчета в области данных;
 - На первом шаге Мастера создания Подчиненных форм и отчетов для определения нового объекта укажите отчет **Экзамены**.
 - На втором шаге самостоятельно укажите поле связи “Номер студенческого билета”.
 - На 3 шаге сохраните подчиненный отчет с именем Результаты экзаменов.
 - Отредактируйте полученный отчет таким образом, чтобы все данные о студенте разместились на 1 листе.

5.17. Требования к отчету

Отчет о проделанной работе должен содержать :

1. Название работы, ее задачи и описание последовательности выполнения.

2. Ответы на контрольные вопросы.

5.18. Контрольные вопросы

1. Что представляют собой отчеты в Access?
2. В чем их преимущества перед другими методами вывода данных?
3. С чего начинается создание отчета?
4. Какие существуют мастера для создания отчета?
5. В чем отличие автоотчетов от других способов создания отчета?
6. Чем характеризуется режим Конструктора?
7. Назовите разделы отчета.
8. Назовите инструменты окна отчета.
9. В каком разделе отчета выводятся итоги по группе?
10. В поле какого типа можно задать группировку по интервалу? По годам?
11. Как добавить вычисляемое поле в область данных?
12. Для чего применяется свойство объекта Не выводить повторы?
13. Наличие каких объектов необходимо для внедрения подчиненного отчета?
14. Какие изменения можно внести в отчет в режиме предварительного просмотра?

Лабораторная работа 6

ТЕМА: ИСПОЛЬЗОВАНИЕ МАКРОСОВ В ACCESS

6.1. Цель работы: изучение возможностей Access по созданию и использованию макросов.

6.2. Понятие макрокоманды и макроса

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 – 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.

6.3. Создание макроса

Для создания макроса в окне базы данных нужно выбрать вкладку **Макросы** и нажать на кнопку **Создать**. Появится окно конструктора макросов. Оно состоит из двух столбцов: **Макрокоманда** и **Примечание**. В столбце **Макрокоманда** перечисляются команды, подлежащие выполнению. Чтобы в ячейке столбца **Макрокоманда** задать нужную макрокоманду, ее следует выбрать из списка.

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.

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 расположить окно базы данных и окно макроса рядом на экране с помощью команды **Сверху вниз** или **Слева направо** из меню **Окно**. Затем 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.

6.4. 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.

Варианты запуска макросов:

- Чтобы запустить макрос из окна конструктора макроса, нужно нажать кнопку  **Запуск** на панели инструментов.
- Чтобы запустить макрос из окна базы данных, следует выбрать вкладку **Макросы**, активизировать имя макроса и нажать кнопку **Запуск** в окне базы данных (можно также выполнить двойной щелчок на имени запускаемого макроса).
- Чтобы запустить макрос из любого режима Microsoft Access, нужно выбрать команду **Сервис\Макрос\Запуск макроса** и указать имя макроса.

6.5. Выполнение макроса по шагам

В Access предусмотрена возможность выполнения макроса в пошаговом режиме.

Для этого предназначена кнопка  **По Шагам** на панели инструментов «Конструктор макросов».

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

6.6. Создание кнопки для запуска макроса в режиме формы

Формы являются основным способом организации интерфейса пользователя в приложениях MS Access и создаются с различными целями. Чтобы автоматизировать выполнение некоторой последовательности действий, можно создать формы для работы с макросами.

Существуют несколько способов создания кнопочной формы для запуска макросов.

6.7. Создание кнопки с помощью мастера

Открыть форму в режиме конструктора и воспользоваться инструментом  Кнопка на Панели элементов.

После указания места размещения кнопки на форме запускается **Мастер Создания Кнопок**. Работа с ним состоит из нескольких шагов.

Для перехода к следующему шагу нажимается кнопка **Далее**, а для завершения работы с Мастером - кнопка **Готово**.

На 1 шаге в поле "**Категория**" нужно выбрать параметр "**Разное**", а в поле "**Действия**" - "**Запуск макроса**".

На 2 шаге следует выбрать нужный макрос из списка имеющихся в базе данных.

На 3 шаге - задать рисунок или текст, который должен быть на кнопке.

На 4 шаге - задать имя кнопки для дальнейших ссылок на нее.

6.8. Создание кнопки с помощью мыши

Кнопку, запускающую макрос, можно также создать, перетащив с помощью мыши название макроса из окна базы данных в окно конструктора формы в область заголовка проекта формы.

6.9. Изменение вида кнопки

При необходимости можно изменить размер кнопки с помощью маркировочной рамки. Переместить кнопку можно с помощью манипулятора, расположенного в левом верхнем углу маркировочной рамки.

Имя макроса автоматически появляется внутри кнопки. Если оно слишком длинное, размер кнопки следует увеличить. Чтобы на кнопке была видна вся надпись, можно уменьшить размер шрифта или создать надпись, отличающуюся от имени макрокоманды. Редактирование надписи осуществляется после щелчка по кнопке. Для выхода из режима редактирования достаточно щелкнуть вне кнопки.

6.10. 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 Microsoft 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, выбрать в меню **Правка** команду **Переименовать** и присвоить макросу новое имя - AutoExec.

При следующем открытии базы данных Microsoft Access автоматически запустит этот макрос.

Примечание. Если требуется загрузить базу данных, не выполняя при этом макрос AutoExec, следует открыть базу данных при нажатой клавише SHIFT.

6.11. Порядок выполнения лабораторной работы

1. Для создания макроса, открывающего таблицы, рекомендуется выполнить следующие действия:

- перейти на вкладку **Макросы**, нажать кнопку **Создать**, вследствие чего откроется окно конструктора макроса. Выбрать в меню **Окно** команду **Слева направо**, чтобы на экране отображались и окно базы данных, и окно макроса;
- в окне базы данных перейти на вкладку **Таблицы**. Перетащить с помощью мыши таблицу **Студенты** в окно макроса и разместить в первой ячейке столбца **Макрокоманда**. Таким образом, в макрос включается операция открытия таблицы **Студенты**.
- в столбец **Примечание** той же строки ввести текст: «Открытие таблицы «Студенты».
- с помощью клавиши ENTER перейти во вторую строку столбца **Макрокоманда**.
- в списке макрокоманд выбрать **На Запись** и установить аргументы: таблица **Студенты** и **Последняя запись**. Таким образом, после открытия таблицы текущей станет последняя запись. В столбец **Примечание** ввести текст: «Переход на последнюю запись».
- повторить те же действия для таблицы **Экзамены**.
- щелкнуть в следующей свободной ячейке столбца **Макрокоманда** и открыть список доступных макрокоманд. Выбрать команду **Выполнить команду**. В области аргументы

макрокоманды активизировать поле **Команда** и выбрать в списке элемент **TileVertically**. Таким образом, в макрокоманду будет включена операция разделения экрана в случае открытия нескольких окон. В столбце **Примечание** набрать текст: «Упорядочение окон».

- сохранить макрос с именем «Открытие БД».
- осуществить запуск макроса, воспользовавшись одним из способов, описанных в п.5.3.

Выполните подобную последовательность действий для того, чтобы создать макросы «Просмотр отчета» и «Открытие запроса» для открытия любых ранее созданных отчета и запроса для таблицы **Студенты**.

Примечание

При создании макроса рекомендуется активизировать кнопку панели **По шагам**.

2. Создать на панели инструментов кнопку для запуска макроса «Просмотреть отчет». Выполнить макрос, щелкнув мышью по его кнопке на панели инструментов.

Если на кнопке нет текста, его можно установить щелкнув правой кнопкой мыши и выбрав опцию **Настройка**. При открытом окне «Настройка» вновь щелкнуть правой клавишей на кнопке макроса и установить режим «Значок и текст».

3. Создать кнопочную форму, разместив на ней элементы (кнопки) для открытия базы данных; открытия запроса; просмотра отчета.

Кнопки будут использоваться для запуска ранее созданных макросов.

Изменить размеры кнопок, шрифты для надписей, цвета. Сохранить форму с именем «Главная форма».

4. Создать автоматически выполняемый макрос с именем AUTOEXEC для открытия формы «Главная форма». Чтобы проверить работоспособность макроса, нужно закрыть, а затем вновь открыть базу данных.

6.12. Требования к отчету

Отчет о проделанной работе должен содержать:

1. название и цель работы;
2. описание основных этапов работы и результатов их выполнения;
3. письменные ответы на контрольные вопросы.

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. Как добавить на панель инструментов кнопку, запускающую макрос?
7. Как в режиме формы создать кнопку для запуска макроса?
8. Для чего предназначен макрос AutoExec?
9. Как создать макрос AutoExec?
10. Как загрузить базу данных, не выполняя макрос AutoExec?

Лабораторная работа 7

ТЕМА: ЗАЧЕТНОЕ ЗАНЯТИЕ

Данное занятие состоит из упражнений, которые необходимо выполнить, затем необходимо оформить отчет и показать результаты работы преподавателю.

7.1. Упражнения

Упражнение 1. Тема: Создание таблиц

1. Создать новую базу данных «фамилия_группаТЕСТ», например «ИвановОС0701Тест».
2. В режиме **Конструктор** заполнить столбцы «имя поля» и «тип данных» следующим образом:

Имя поля	Тип данных
Фамилия	Текстовый
Имя	Текстовый
Отчество	Текстовый
Дата рождения	Дата/время
Адрес	Текстовый
Телефон	Текстовый
Группа	Числовой

3. Завершив ввод структуры таблицы, сохраните ее с именем «**Анкета**». В ответ на вопрос «Создать ключевое поле сейчас» нажмите кнопку **Да**. Тип данных у этого поля - «**Счетчик**».
4. Определить свойства полей.
5. Открыть таблицу «**Вид/ режим таблицы**».
6. Заполнить 10 записей.
7. Файл сохранить.
8. На основе созданной таблицы создайте список фамилий студентов Вашей группы, для этого в режиме конструктора для поля «**фамилия**» откройте вкладку «**подстановка**» и установите следующие свойства для поля «**фамилия**»

Тип элемента управления	Поле со списком
Тип источника строк	Таблица/запрос
Источник строк	Таблица «Анкета». Перетащить с помощью мыши поле «фамилия» в «поле» бланка запроса
Ограничиться списком	Нет

В результате выполнения Задания 4 в таблице «**Анкета**» в поле «**фамилия**» при щелчке мышью должна появиться кнопка, обеспечивающая выбор из списка.

Упражнение 2. Тема: Использование форм для ввода и редактирования данных

1. Создать форму (**Автоформа: ленточная**) для заполнения таблицы «**Анкета**».
2. Создайте таблицу Группы для ввода групп, имеющих на третьем курсе:

Имя поля	Тип данных
Группа	Текстовый

В ответ на вопрос «Создать ключевое поле сейчас» нажмите кнопку **Нет**.

3. Создать форму («**Автоформа : в столбец**») для заполнения таблицы «**Группы**».

Введите данные в форму «**Группы**».

Обратите внимание на то, что если вы в поле «**фамилии**» таблицы «**Анкета**» введете новую фамилию, то она не появится в раскрывающемся списке фамилий. Убедитесь в этом!!

4. Создать самозаполняющийся список фамилий, чтобы при вводе новой фамилии эта фамилия автоматически попадала в список и впоследствии эту фамилию можно было бы выбрать из списка, а не набирать заново. Для этого выполните следующие действия:

- создайте форму, основанную на таблице «**Анкета**» с именем «**Форма для добавления записей**». Эта форма будет использоваться для добавления в список новых пунктов.

- создайте группу макросов. Будьте очень внимательны при выполнении этого пункта!!!

- Выбрать вкладку «**макросы**», нажать кнопку **Создать**.
- Ввести макрокоманду «**открыть форму**»
 - ✓ *Имя формы: «Форма для добавления записей»*
 - ✓ *Режим: Форма*
 - ✓ *Режим данных: Добавление*
 - ✓ *Режим окна: Обычное*
- Закрыть окно и сохранить макрос с именем «**Добавление**»
Создать следующий макрос:
 - Ввести макрокоманду «**Выполнить команду**»
 - ✓ *Команда «Сохранить запись (Save Record)»*
 - Ввести макрокоманду «**Выделить объект**»
 - ✓ *Тип объекта: Форма*
 - ✓ *Имя объекта: Анкета*
 - ✓ *В окне базы данных: Нет*
 - Ввести макрокоманду «**Обновление**»
 - ✓ *Имя элемента: Фамилия*

Закрыть окно и сохранить макрос с именем «**Закрытие**»

- переключитесь в форму «**Форма для добавления записей**» в режиме конструктора.

- выберите режим «**вид/свойства**».

-- откройте вкладку «**События**».

- добавьте макрос «**Закрытие**» в свойство «**Закрытие формы**».

- закройте форму.

- откройте в режиме конструктора форму «**Анкета**».

Щелкните «правой мышкой» по полю «**фамилия**» и выберите пункт «**свойства**»:

- Откройте вкладку «**События**»
- Укажите имя макроса «**Добавление**» в поле ввода свойства «**Двойное нажатие кнопки**».

5. Проверьте работу созданных Вами макросов по созданию самозаполняющегося списка.

- Переключитесь в режим формы «**Анкета**»
- Дважды щелкните мышкой по полю «**фамилия**», введите произвольную фамилию.
- Заполните остальные обязательные поля для записи с этой фамилией
- Закройте форму. Вы опять в главной форме «**Анкета**».
- Попробуйте раскрыть список фамилий. Если Вы аккуратно выполнили Задание 5, то в списке Вы обнаружите данную фамилию.

Закройте форму, сохранив внесенные изменения.

Упражнение 3. Тема: Проектирование многотабличных баз данных

Задание 1

1. Создать таблицу «Преподаватели», используя *Мастер таблиц*.

- Выбрать для создания таблицы образец «Сотрудники» и выбрать из предложенного перечня полей следующие:
Фамилия, Имя, Отчество, Должность, Название отдела (переименовать, как «кафедра»)
- Назвать созданную таблицу «Преподаватели». Связи не определять, т.е. нажать кнопку **Далее**.
- Выбрать режим «**непосредственный ввод данных в таблицу**»
- В режиме конструктора определить тип данных для каждого поля **Текстовый**.
 - Определить свойства полей на вкладке «**Общие**» в режиме конструктора следующим образом:
 - Для поля «кафедра» изменить элемент управления следующим образом:
Откройте вкладку «**Подстановка**» и установите следующие свойства для поля «кафедра»:

Тип элемента управления	Поле со списком
Тип источника строк	Список значений.
Источник строк	Введите список кафедр, используя в качестве разделителя точку с запятой.
Ограничиться списком	Да

Задание 2

1. Создайте форму для заполнения таблицы «Преподаватели».

- Выбрать вкладку «**Формы**» и создать форму, используя **Мастер форм** и определив в качестве источника таблицу «Преподаватели».
 - Отметить мышкой все поля для создания формы.
 - Выбрать вид формы «ленточная».
 - Имя формы - «Преподаватели».
 - Дальнейшие действия «Открыть форму для просмотра».
2. Заполните форму. Обратите внимание на то, что столбец «кафедра» заполняется путем подстановки из сформированного Вами списка.

Упражнение 4. Продолжение темы «Проектирование многотабличных баз данных»

Задание 1

1. Создать таблицу «Дисциплины» в режиме конструктора с текстовым полем **Дисциплина**.
2. Определить свойства полей таблицы «Дисциплины» на вкладке «**Общие**» следующим образом:

Поле	Свойства полей			
	Размер	Подпись	Обязательное поле	Индексированное поле
Дисциплина	20	Дисциплина	Да	Да (совпадения не допускаются)

3. Создайте ленточную автоформу для заполнения таблицы «Дисциплины». В качестве «источника данных» выбрать таблицу «Дисциплины».
4. Введите названия дисциплин в форму.

Задание 2

1. Создайте таблицу «Экзамен» в режиме конструктора таблиц с параметрами

Имя поля	Тип данных
Дисциплина	Текстовый
Группа	Числовой
Студент	Текстовый
Результат контроля	Текстовый

2. Определить свойства полей.

У Вас в базе уже имеется таблица с перечнем групп, таблица с перечнем дисциплин, а также таблица «Анкета» с перечнем студентов. Воспользуемся этим для автоматического заполнения полей «группа», «дисциплина» и «студент» в создаваемой Вами таблице «Экзамен» Это можно сделать с помощью элемента управления полем.

Задание 3

1. Для поля «группа» изменить элемент управления:

Откройте вкладку «Подстановка» и установите следующие свойства для поля «группа»

Тип элемента управления	Поле со списком
Тип источника строк	Таблица/запрос
Источник строк	Добавить таблицу «Группы» и перетащить поле «Группа» в «поле» бланка запроса
Ограничиться списком	Да

2. Аналогично изменить элемент управления для полей «дисциплина» и «студент», используя в качестве строк соответственно таблицы «дисциплины» и «анкета».
3. Для поля «результат контроля» изменить элемент управления следующим образом: откройте вкладку «Подстановка» и установите следующие свойства для поля «результат контроля»:

Тип элемента управления	Поле со списком
Тип источника строк	Список значений.
Источник строк	Введите список оценок, используя в качестве разделителя точку с запятой (отл.;хор;удов;неуд.)
Ограничиться списком	Да

Задание 4

1. Создайте ленточную автоформу для заполнения таблицы «Экзамен», выбрав в качестве «источника данных» таблицу «Экзамен».

Задание 5

1. Пользуясь созданной в Задании 4 формой «Экзамен» введите данные по двум дисциплинам на Ваш выбор по всем студентам своей группы в таблицу «Экзамен». Оценки проставлять произвольно (желательно разнообразные).

Упражнение 5. Продолжение темы «Проектирование многотабличных баз данных»

Задание 1

1. Создайте таблицу «Зачет» в режиме конструктора таблиц с параметрами

Имя поля	Тип данных
Дисциплина	Текстовый
Группа	Числовой
Студент	Текстовый
Результат	Логический

2. Определить свойства полей на вкладке «Общие».

У Вас в базе уже имеется таблица с перечнем групп, таблица с перечнем дисциплин, а также таблица «Анкета» с перечнем студентов. Воспользуемся этим для автоматического заполнения полей «группа», «дисциплина» и «студент» в создаваемой Вами таблице «Зачет».

3. Для поля «группа» изменить элемент управления:

Откройте вкладку «Подстановка» и установите следующие свойства для поля «группа»

Тип элемента управления	Поле со списком
Тип источника строк	Таблица/запрос
Источник строк	Таблица «Группы». Перетащите поле «Группа» в бланк запроса.
Ограничиться списком	Да

4. Аналогично изменить элемент управления для полей «Дисциплина» и «Студент», используя в качестве строк соответственно таблицы «дисциплины» и «анкета».

Задание 2

1. Создайте ленточную автоформу для заполнения таблицы «Зачет».

Выбрать в качестве «источника данных» таблицу «Зачет».

2. Пользуясь созданной формой «Зачет» введите данные по всем студентам своей группы по двум дисциплинам на свой выбор.

Зачет отмечайте флажком.

Теперь сделаем так, чтобы рядом с флажком появлялась надпись «зачет», а если Вы не отметили его галочкой, то пусть появляется надпись «незачет».

Задание 3

Для того чтобы посредством флажка ввести в поле одно из двух значений, надо воспользоваться функцией управления *Iif*, а для этого следует выполнить следующие действия:

1. Откройте форму «Зачет» в режиме конструктора.

2. Справа от поля типа флажок добавить элемент типа «поле»:

- Щелкните мышкой по значку **аб** на панели инструментов и справа от флажка еще раз щелкните мышкой, появится поле с произвольной надписью.

3. Нажмите «правую мышку» на этом поле, раскройте пункт «свойства».

Введите в свойство «данные» следующее выражение:

`=Iif([Результат];"Зачет";"Незачет")` .

4. Установить для свойства «Вывод на экран» для надписи (надпись находится слева от поля) поля значение «нет».

Задание 4

1. Просмотрите форму «Зачет» в режиме формы. Теперь установленный флажок будет соответствовать значению «зачет», а снятый флажок – значению «незачет».
2. Пользуясь новым элементом управления «флажок» проставьте результаты по зачету заново, изменения сохранить!!!

Упражнение 6. Продолжение темы «Проектирование многотабличных баз данных»

Задание 1

Установите связь между таблицами «Зачет», «Дисциплины», «Анкета».

Создайте связь «один-ко-многим» между полями Дисциплина таблицей «Зачет» и таблицей «Дисциплина». Аналогично создайте связь между полем «фамилия» таблицы «Анкета» и полем «студент» таблицы «Зачет».

Помните, что поле, по которому устанавливается связь со стороны «один» должно быть индексированным и совпадения не допускаются.

Задание 2

Если Вы правильно выполнили задание 1, то:

После установления связи изменение полей Фамилия и Дисциплина соответственно в таблицах «Анкета» и «Дисциплины» автоматически повлечет изменение в таблице «Зачет». В свою очередь Вы не сможете ввести в таблицу «Зачет» новые фамилии или дисциплины, если они отсутствуют в таблицах «Анкета» и «Дисциплины».

Проверьте это.

Задание 3

1. Установите связь между таблицами «Экзамен» - «Анкета», «Дисциплины». Проверьте, как работают связи.

Упражнение 7. Тема: Подчиненные формы.

Допустим, Вам требуется посмотреть данные результатов экзамена по каждому студенту в отдельности, либо по каждой дисциплине в отдельности, используя уже готовую таблицу, где у Вас занесены общие результаты по всем студентам и по всем дисциплинам. Для этого можно использовать механизм создания подчиненных форм.

Задание 1

Создать подчиненную форму по результатам сдачи экзаменов с выборкой по каждому студенту в отдельности.

Выполнить последовательно действия:

1. Откройте Вашу базу данных
2. Выберите вкладку «формы».
3. Нажмите кнопку **Создать** и выберите опцию **Мастер форм**, не выбирая источник данных, нажмите кнопку ОК.
4. В окне «Создание форм» выберите из списка **Таблицы/запросы** таблицу «Экзамен» (это будет подчиненная форма). Поместите в «выбранные поля» поля «дисциплина», «студент», «результат контроля».
5. Не выходя из окна «Создание форм» выберите из списка «Таблицы/запросы» таблицу «Анкета» (это будет главная форма) и поместите в список «**Выбранные поля**» поле «Фамилия».

6. Нажмите кнопку **Далее**, в результате откроется следующее окно диалога «**Создание форм**». В этом окне Вам предстоит определить главную и подчиненную таблицы.
7. Щелкните мышкой по таблице «**Анкета**».
8. Обратите внимание на правую часть окна: в верхней части отображаются поля *главной* («Анкета»), а в нижней части – *подчиненной* («**Экзамен**») таблицы.
9. Выбрать опцию «**подчиненные формы**»
10. Определив главную и подчиненную таблицы, нажмите кнопку **Далее**.
11. Выберете внешний вид – **табличный**. Выберете стиль на Ваше усмотрение.
12. Форму назвать «**Анкета для экзамена**», название подчиненной формы не изменять.

Созданная форма вполне приемлема для использования.

Попробуйте менять фамилию студента (используйте кнопки перемещения по записям в низу окна), и Вы увидите как меняется содержимое таблицы «**Экзамен**». Причем информация, появляющаяся в таблице относится только к выбранному Вами студенту.

Убедитесь в этом!!! В этой форме явно лишняя информация в подчиненной таблице «**Экзамен**», ведь вполне достаточно фамилии студента, которая появляется вверху над результатами контроля. Это можно исправить в конструкторе форм в следующем упражнении.

Упражнение 8. Тема: Форматирование форм.

Задание 1

Измените стиль формы «**Анкета**» по своему усмотрению, для этого:

- Откройте форму в режиме конструктора.
- Нажмите на панели инструментов кнопку «**Автоформат**».
- Выберете нужный стиль.

Задание 2

1. Измените цвет фона и цвет шрифта для каждого поля по своему усмотрению.

Задание 3

1. Попробуйте изменить размер поля для ввода фамилии по высоте, для этого надо щелкнуть мышкой по этому полю и изменить появившиеся границы поля.
2. Измените шрифт (размер, вид, цвет) для поля «**Фамилия**».

Задание 4

1. Убрать в подчиненной форме «**Экзамен**» поле «**Студент**»:
 - Открыть форму «Анкета для экзамена». (В подчиненной форме «**Экзамен**» желательно убрать поле «**Студент**», так как фамилия уже присутствует в верхней части формы).
 - Убрать поле «**студент**»:
 - ✓ *Открыть форму в режиме конструктора*
 - ✓ *Выделить поле «студент» и «вырезать» это поле.*
2. Выйдите из режима конструктора и посмотрите, как теперь выглядит форма

Упражнение 9. Тема: Выбор данных из таблиц с помощью запросов

Задание 1

1. Добавьте в таблицу «**Экзамен**» поле «**Преподаватель**» со следующими характеристиками:
 - Тип данных – текстовый
 - Подпись – преподаватель
 - Подстановка – используйте поле со списком (источник строк – таблица «Преподаватели»)

Задание 2

У Вас в таблице «**Экзамен**» должно быть не менее 20 записей и эти записи должны содержать информацию:

- Не менее чем по трем дисциплинам.
- Оценки должны быть разные (от отлично до неудовлетворительно).
- Студенты должны быть из разных групп.

Убедитесь в том, что записей в таблице достаточное количество, в противном случае добавьте новые записи.

Задание 3

1. Создайте запрос по выбору записей, содержащих студентов, сдавших все экзамены на «отлично».
2. Откройте созданный запрос и убедитесь в том, что выборка была сделана правильно.

Задание 4

1. Создайте аналогичный запрос по двоечникам и сохраните с именем «**Запрос по двоечникам**»
2. Откройте созданный запрос и убедитесь в том, что выборка была сделана правильно

Задание 5

1. Создайте запрос на выборку записей по результатам контроля по дисциплине, которую преподает конкретный преподаватель. Запрос сохраните с именем «**Запрос по дисциплине**».
2. Откройте созданный запрос и убедитесь в том, что выборка была сделана правильно

Задание 6

1. Создайте на базе таблица «**Анкета**» запрос для формирования списка студентов, которые родились с 1 мая 1979 года по 31 декабря 1979года.

Для формирования запроса Вам следует вызвать построитель выражений (кнопка «построить») и выбрать в списке операторов - оператор сравнения Between, нажать кнопку «Вставить».

Теперь откорректируйте появившееся выражение следующим образом:

Between #01.05.79# And #31.12.79# .

Если вдруг оказалось, что таких студентов в Вашей группе нет, измените условия запроса.

Запрос сохранить с именем «**Запрос по дате рождения**»

2. Откройте созданный запрос и убедитесь в том, что выборка была сделана правильно

Задание 7

Создайте формы для всех запросов (**автоформа ленточная**).

Задание 8

Закрывать доступ к корректировке базы данных в режиме запросов.

Чтобы в режиме запросов нельзя было изменить информацию в базе данных, надо выполнить следующие действия:

1. Открыть форму для запроса в режиме конструктора
2. Пометить поле и правой кнопкой вызвать свойства помеченного поля, затем на вкладке «**данные**» установить свойство «**Блокировка**» – Да.

Аналогично установите Свойство «**Блокировка – Да**» для всех полей во всех формах для запросов.

Упражнение 10. Тема: Построение отчетов.

Задание

Создать отчет по результатам сдачи экзаменов, используя таблицу «**Экзамен**». Используйте следующие поля:

- ◆ **Дисциплина**
- ◆ **Группа**
- ◆ **Студент**
- ◆ **Результат контроля**

Выбрать группировку по дисциплине, сохранить отчет «**Отчет по экзаменам**». Аналогично созданию отчета «**Экзамен**» создайте отчет «**Зачет**»

Упражнение 11. Тема: Создание кнопок управления.

Задание 1

1. Создать кнопки выхода из всех форм, которые Вы создали:

- ✓ **Анкета**
- ✓ **Преподаватели**
- ✓ **Дисциплины**
- ✓ **Группы**
- ✓ **Экзамен**
- ✓ **Зачет**

Начнем с формы «**Анкета**»

- ◆ Открыть соответствующую форму в режиме конструктора.
- ◆ Установите режим использования мастера на панели элементов (кнопка «**Мастера**»)
- ◆ Выберите инструмент **Кнопка** на панели элементов.
- ◆ Установите указатель мыши на место в форме, в котором Вы предполагаете расположить кнопку для выхода из формы и нажмите кнопку мыши. Запускается мастер создания кнопки конструктора форм

В первом окне диалога расположены два списка: Категории и Действия

- ◆ Выберите в списке Категории пункт «**Работа с формой**», а в списке «**Действия**» пункт «**Закрытие формы**».
- ◆ В окне «**Создание кнопок**» введите текст «**Выход**».
- ◆ Задать имя кнопки «**Выход из анкеты**».

Теперь можно выйти из конструктора и попробовать, как работает кнопка.

Задание 2

Создать кнопки выхода для всех форм (в том числе и запросов), которые Вы создали.

Задание 3

Создать кнопки выхода для всех отчетов, которые Вы создали.

ПРИМЕЧАНИЕ. При описании упражнений 12 - 14 предполагалось использование Access 2003. Для создания кнопочной формы в Access более старших версий предполагается создание меню для работы в БД средствами этих версий Access.

Упражнение 12. Тема: Создание кнопочных форм

Кнопочная форма есть не что иное, как Меню для работы в базе данных. В меню может быть несколько вложенных подменю. Вы будете создавать меню следующей структуры:

В этом меню 3 уровня

Задание 1

1. Создать меню 1-го уровня:

- ✓ Закройте все таблицы, формы, отчеты, запросы
- ✓ Выберите пункт меню **«Сервис/ Надстройки/ Диспетчер кнопочных форм»**, в ответ на вопрос **«Создать кнопочную форму?»** нажать кнопку Да. Перед Вами окно диспетчера кнопочных форм.
- ✓ Нажать кнопку **«Изменить»**.
- ✓ Ввести название кнопочной формы - **«Студенты МТУСИ»**.
- ✓ Нажать кнопку **Заккрыть**.
- ✓ Нажать кнопку **Создать**.
- ✓ В окне **«Создание»** ввести текст - **«Формы»**.
- ✓ Нажать кнопку **Создать**.
- ✓ В окне **«Создание»** ввести текст - **«Запросы»**.
- ✓ Нажать кнопку **Создать**.
- ✓ Аналогично создать следующие страницы кнопочной формы :
«Отчеты», «Выход».

2. Создать меню 2-го уровня

Теперь последовательно будем раскрывать каждый пункт меню 1-го уровня (для наглядности можно ориентироваться на схему меню).

- ◆ Отметить курсором главную кнопочную форму **«Студент»** и нажать кнопку **Изменить**.
- ◆ Создать элемент кнопочной формы.
- ◆ Текст - **«Формы»**.
- ◆ Команда - **«Переход к кнопочной форме»**.
- ◆ Кнопочная форма - выбрать из списка **«Формы»**.
- ◆ Создать элемент кнопочной формы.
- ◆ Текст - **«Запросы»**.
- ◆ Команда **«Переход к кнопочной форме»**.
- ◆ Кнопочная форма - выбрать из списка **«Запросы»**.
- ◆ Создать элемент кнопочной формы.
- ◆ Текст - **«Отчеты»**.
- ◆ Команда **«Переход к кнопочной форме»**.
- ◆ Кнопочная форма - выбрать из списка **«Отчеты»**.
- ◆ Создать элемент кнопочной формы.
- ◆ Текст - **«Выход»**.
- ◆ Команда **«Выход из приложения»**.
- ◆ Кнопочная форма - выбрать из списка **«Выход»**.

Закрыв окно **«Изменение страницы кнопочной формы»**, Вы вернулись в окно **«Диспетчер кнопочных форм»**

3. Создать меню 3-го уровня:

Установить курсор на страницу **«Формы»** и нажать кнопку **Изменить**.

Вы находитесь в окне **«Изменение страниц кнопочной формы»**

- ◆ Создать элемент кнопочной формы.
- ◆ Текст - **«Анкета»**.
- ◆ Команда **«Открытие формы в режиме редактирования»**
- ◆ Кнопочная форма - выбрать из списка **«Анкета»**.
- ◆ Создать элемент кнопочной формы.
- ◆ Текст - **«Преподаватели»**.
- ◆ Команда **«Открытие формы в режиме редактирования»**.
- ◆ Кнопочная форма - выбрать из списка **«Преподаватели»**.
- ◆ Создать элемент кнопочной формы.

- ◆ Текст - «Дисциплины».
- ◆ Команда «Открытие формы в режиме редактирования».
- ◆ Кнопочная форма - выбрать из списка «Дисциплины».
- ◆ Создать элемент кнопочной формы.
- ◆ Текст - «Группы».
- ◆ Команда «Открытие формы в режиме редактирования».
- ◆ Кнопочная форма - выбрать из списка «Группы».

Вы создали меню для ввода и корректировки информации в базу данных, теперь создадим меню для вызова запросов

Задание 2

Создать меню 3-го уровня для вызова запросов.

Закрыть окно «Изменение страницы кнопочной формы», если Вы в нем находитесь. Вы опять в диспетчере кнопочных форм

Установить курсор на страницу «Запросы» и нажать кнопку **Изменить**. Вы находитесь в окне «Изменение страниц кнопочной формы»

- ◆ Создать элемент кнопочной формы.
- ◆ Текст - «Запрос по двоечникам».
- ◆ Команда «Открытие формы в режиме редактирования».
- ◆ Кнопочная форма - выбрать из списка «Запрос по двоечникам».

Аналогично создать элементы кнопочной формы для других запросов.

Задание 3

Создать меню 3-го уровня для просмотра отчетов, выполнив действия аналогичные действиям при создании меню для форм и запросов.

Задание 4

1. Закройте диспетчер кнопочных форм.
Теперь в Вашей Базе на вкладке «Формы» появилась новая форма с именем **«Кнопочная форма»**.
2. Откройте эту форму и проверьте, как работает меню.

Упражнение 13. Тема: Использование кнопок управления для перехода из одного уровня Меню в другой.

Задание 1

В Вашем Меню имеется одна кнопка выхода, настроенная на выход из базы данных.

Создайте на каждом уровне кнопки управления для возврата на более высокий уровень (выход в меню «Студенты МГУСИ»).

1. Закройте все таблицы, формы, отчеты, запросы
2. Выберите пункт меню «Сервис/Настройки/Диспетчер кнопочных форм».
3. Отметить страницу «Формы», нажать кнопку **Изменить**.
4. Создать новый элемент кнопочной формы с элементами:
Текст - **Выход**.
Команда – **Переход к кнопочной форме**.
Кнопочная форма - «Студент».
5. Закрыть окно «Изменение страницы кнопочной формы».
6. Отметить страницу «Запросы», нажать кнопку **Изменить**.
7. Повторить действия пункта 4. для создания кнопки выхода из пункта меню **«Запросы»**.
8. Аналогичные кнопки создать для пункта меню **«Отчеты»**.

9. Закрыть диспетчер кнопочных форм.

Задание 2

Открыть кнопочную форму и проверить как работают кнопки перехода с второго уровня на первый.

Упражнение 14. Тема: Оформление Меню для работы с базой

Задание

1. Изменить шрифт надписей пунктов меню:
 - ◆ Открыть кнопочную форму в режиме конструктора.
 - ◆ Выделить мышкой то место, где может быть надпись пункта меню «**Формы**».
 - ◆ Выбрать пункт меню «**Вид\свойства**» (либо просто щелкнуть правой мышкой).
 - ◆ Откройте вкладку «**Макет**» и можете на свое усмотрение изменить:
Размер шрифта, Цвет текста, Цвет границы, Насыщенность.
2. Аналогично измените оформление для всех пунктов меню.
3. Теперь можно выйти из режима конструктора и посмотреть, как выглядит кнопочная форма (это и есть главное меню для работы с базой данных).

7.2. Содержание отчета

1. Название и цель работы.
2. Результаты работы по всем заданиям и упражнениям.

Лабораторная работа 8

ТЕМА: Изучение CASE средства ERWin

8.1. Использование Erwin для разработки модели данных. Логическая модель. Сущности и связи

Цель: Подготовить презентацию для начального обсуждения предметной области, создав хранимые отображения "Определения сущностей" и "Определения связей".

Хранимым отображением [8] называется отображение конкретного аспекта модели с удобным для презентации расположением, масштабом и цветовыми эффектами.

Предварительная подготовка

1. На основании результатов обследования объекта **для своего варианта задания из лабораторной работы 4** подготовить список имен сущностей.
2. Для каждой сущности записать:
 - словесное определение сущности;
 - примеры возможных запросов по отношению к этой сущности;
 - несколько экземпляров сущности, задав их основные свойства - атрибуты, в произвольной форме.
3. Продумать связи между сущностями.
4. Для каждой связи подготовить:
 - словесное определение связи - какие сущности связываются, в каком отношении, для отношения "один ко многим" отметить, какая из сущностей является родительской и характер связи - идентифицирующая или неидентифицирующая;
 - имя связи, записав глагольную фразу, отражающую для связи "один ко многим" - одну фразу, отношение родительской сущности к дочерней; для связи "многие ко многим" - две фразы, отношения слева направо / справа налево.
 - для связей "один ко многим" определить мощность связи: 0, 1, ∞ ; 1, ∞ (P); 0, 1 (Z); точно N (N); правила ссылочной целостности для операций "удаление", "добавление", "обновление".

Установим правила ссылочной целостности для идентифицирующей связи типа P (1 или много). В этом случае на каждый экземпляр родительской сущности должен ссылаться хотя бы один экземпляр дочерней сущности. Воспользуемся сокращениями ERwin для указания характеристик связей на хранимых отображениях:

Сущность	Операция		
	D	I	U
<Имя 1> - дочерняя	R	R	R
<Имя 2> - родительская	C	C	R

D - Delete, удаление записи, **I - Insert**, добавление записи, **U - Update**, обновление записи, **R - Restrict**, ограничение: нельзя удалять/обновлять экземпляры родительской сущности, пока имеется хотя бы один ссылающийся на него экземпляр дочерней сущности.

C - Cascade, тотальное удаление/обновление экземпляров дочерней сущности, ссылающихся на удаляемый экземпляр родительской сущности.

N - None, при удалении экземпляра родительской сущности ссылающиеся на него экземпляры дочерней сущности остаются без изменений, т.е. ссылаются на отсутствующий экземпляр и не могут быть классифицированы.

8.1.1. Порядок выполнения 1-ой части работы


Создание хранимого отображения "Определения сущностей" - закладка 1

1. Открыть диалог Computer Associates Erwin, включить опцию "Create a new model". В окне Create Model/New Model Type выбрать опцию Logical/Physical. Открыть диалог *Stored Display Settings*, выполнив ►Format ►Stored Display... (вариант: выполнить команду Stored Display... из контекстного меню рабочей области Erwin)



2. В диалоге *Stored Display Editor* сделать настройки хранимого отображения

- нажав кнопку Rename, задать имя отображения "Определение сущностей" вместо имени по умолчанию - Display1;
- в закладке General указать авторов работы;
- в закладке Logical выбрать отображение сущностей на уровне определений (рамка Display Level, Definition) с иконками сущностей (рамка Entity Option, опция Entity Icon) и отключить все дополнительные опции отображения связей (рамка Relationship Option).

С помощью панели инструментов конструктора создать сущности будущей модели:

- щелкнуть по кнопке ;
- щелчками по рабочему полю, создать нужное количество сущностей;

3. Вместо имен по умолчанию (E/1, E/2, ...) набрать подготовленные имена сущностей.

4. Выделить родительские сущности, изменив цвет шрифта  и цвет заливки .

5. С помощью контекстного меню сущностей вызвать диалог *Entity Properties* и описать каждую сущность (в закладке Definition набрать определение сущности).

Описав одну сущность, переключиться на другую, выбрав ее из распахивающегося списка Entity и описать ее.

6. Установить нотацию IDEFX1 (►Model ►Model Properties..., закладка Notation) и создать связи: задать тип связи щелчком по соответствующей кнопке; для каждой связи выбранного типа щелкнуть сначала по родительской, а затем по дочерней сущности; для связи many-to-many создать сущность-ассоциацию (контекстное меню связи/диалог Create Association Entity).

Создание хранимого отображения "Определения связей" - закладка 2

Для создания и настройки хранимого отображения вызвать диалог *Stored Display Settings*

(выполнить ► EDIT ► Stored Display... или вызвать контекстное меню рабочей области Erwin и выполнить команду Stored Display...). Выполнить New и ввести название новой закладки "Определение связей)

1. В закладке *Logical* диалога *Stored Display Settings* задать отображение сущностей на уровне сущностей (рамка Display Level - Entity) с иконками сущностей (рамка Entity Option - опция Entity Icon) и включить все дополнительные опции отображения связей (рамка Relationship Option - опции Verb Phrase, Cardinality, Referential Integrity).
2. Изменяя цвета подписей к линиям связи, выделить связи 1-го уровня (родительские), связи 2-го уровня и т.д.: вызвать диалог *Default Fonts & Color* из меню ► Format ► Default Fonts & Colors...
 - в закладке *Relationship* задать шрифт и цвет описания связи, а также указать к каким связям (All Objects) относится сделанная настройка (рамка Apply To)
3. С помощью редактора связей Relationship Editor задать свойства связей:
 - в контекстном меню любой связи выполнить команду ► Relationship Properties... для вызова редактора связей;
 - чтобы задать новую связь, нажать кнопку New... редактора и в появившемся окне New Relationship выбрать из выпадающих списков Parent и Child имена связываемых сущностей, установить переключатель в рамке Type на нужный тип связи и, нажав ставшую доступной кнопку **OK**, вернуться в редактор связей
 - в редакторе связей для каждой связи из выпадающего списка Relationship:
 - на вкладке *General*
 - в окнах Parent-to-Child и Child-to-Parent задать имя связи;
Когда тип связи "многие ко многим", задается только имя связи
 - Для связей один ко многим, кроме того, задать мощность связи (рамка Cardinality) и уточнить тип (рамка Relationship Type).

8.2. Использование Erwin для разработки модели данных. Логическая модель. Атрибуты сущностей. Физическая модель

Цель : Создать полную атрибутивную модель , т.е. модель, содержащую все сущности в 3-ей нормальной форме со всеми атрибутами и связями. Выполнить преобразование логической модели в физическую.

Предварительная подготовка

Для каждой сущности, созданной в 1-ой части лабораторной работы:

1. подготовить имена атрибутов логического и физического уровня;
2. указать домены типов атрибутов и ограничения на их значения;
3. выделить среди атрибутов или добавить искусственный первичный ключ;
4. задать имя роли для внешних - мигрирующих ключей дочерних сущностей;

5. отметить потенциальные - альтернативные ключи;
6. сделать список инверсионных входов.

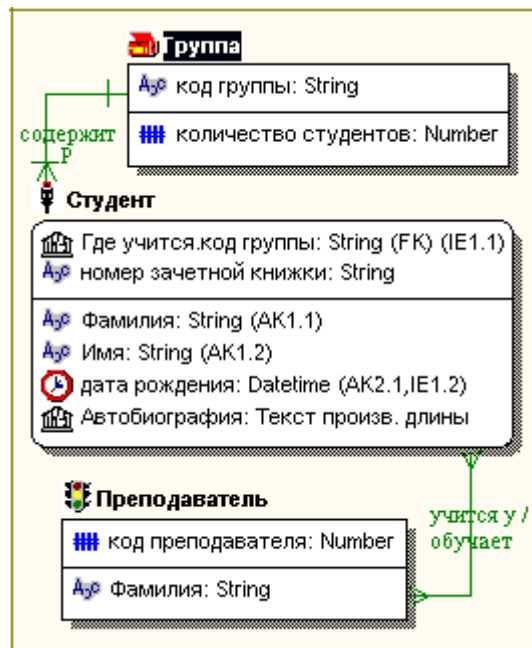
8.2.1. Порядок выполнения 2-ой части работы

1. Открыть модель, созданную ранее - "Сущности и связи".
2. Создать хранимое отображение "Первичные ключи"(закладка 3):
 - вызвать диалог *Stored Display Settingr*, выполнив ► *Format* ► *Stored Display...* или ► *Stored Display...* из контекстного меню рабочей области *Erwin*
 - нажав кнопку **New...** этого диалога, задать имя нового отображения;
 - в закладке *Logical* выполнить настройку отображения:
 - рамка *Display Level* - опция *Primary Key*
 - рамка *Relationship Option* - опция *Verb Phrase*
 - рамка *Entity Option* - опции *Rolename/Attribute*, *Primary Key Designation*, *Foreign Key Designation*, *Show Migrated Attributes*, *Entity Icon*
3. Ввести первичные ключи в родительские сущности. Для их ввода:
 - Выполнить команду *Attributes ...* из контекстного меню любой сущности для вызова редактора *Attributes*
 - В выпадающем списке **Entity** окна редактора выбрать нужную родительскую сущность, например, "Студент"
 - Нажать кнопку **New...** и создать первичный ключ этой сущности в диалоге *New Attribute*:
 - задать логическое - *Attribute Name* имя ключа
 - указать тип щелчком по соответствующему домену (в приведенном примере - *String*)
 - закрыть диалог *New Attribute* нажатием на кнопку *OK*
 - В закладке *General* включить опцию *Primary Key*
 - В закладке *Definition* записать, является ли созданный ключ реальным или автоинкрементным (из-за отсутствия явных уникальных атрибутов)
 - повторить это для других родительских сущностей
4. Задать имена ролей для мигрирующих ключей:
 - Выполнить команду *Relationship Properties* из контекстного меню любой связи.
 - В появившемся окне редактора связей выбрать закладку *RoleName* и набрать имя роли в поле *RoleName*.
 - Проверить ссылочные правила в закладке *RI Actions*

При выходе из этих полей имя роли будет присвоено текущему мигрирующему атрибуту.
5. Отобразить полные имена атрибутов (<имя роли>. <функциональное базовое имя>) в хранимом изображении "Первичные ключи":

- Выполнить команду Entity Display ... из контекстного меню окна ERwin.
 - В появившемся списке опций включить опцию RoleName/Attribute. На экран будут дополнительно выведены имена ролей.
6. Задать первичные ключи для дочерних сущностей.
 7. Расположить сущности на экране так, чтобы они хорошо смотрелись, и изменить, если нужно, масштаб изображения; выполнить ► **File** ► **Save**.
 8. Создать еще одно хранимое изображение - "Атрибуты сущностей" (закладка 4) (в контекстном меню Erwin ► *Stored Displa ...y*, кнопка **New...**)
 9. Сделать настройки для этого изображения, выбрав закладку *Logical*:
 - рамка Display Lever - выбрать **Attribute**
 - рамка Relationship Option - отключить отображения всех опций связей
 - рамка Entity Option - включить все опции
 10. Описать все атрибуты всех сущностей, выполнив команду **Attributes** из контекстного меню любой сущности. В редакторе атрибутов:
 - Пока не кончатся атрибуты текущей сущности:
 - создавать очередной атрибут (кнопка **New...**)
 - в диалоге **New Attribute** задавать логическое (**Attribute Name**) имя атрибута и домен его типа (щелчок мышью)
 - в закладку **Definition** заносить подробное определение атрибута;
 - в закладку **Note** - ограничения на значения атрибута, выявленные в процессе обследования.
 - Сменить сущность, выбрав следующую из распахивающегося списка **Entity**, и повторить создание атрибутов.
 11. Для каждой сущности отметить альтернативные ключи и инверсионные входы.
 - Открыть закладку *Key Group* редактора атрибутов
 - Для создания новой группы нажать кнопку справа от **Key Group Membership**, чтобы вызвать диалог *Key Group*
 - В появившемся диалоге для каждой новой группы ключевых полей, нажимать кнопку **New...** для вызова диалога *New Key Group*
 - в рамке **Key Group Type** этого диалога с помощью переключателя указывать тип группы - альтернативный ключ или инверсионный вход (в приведенном примере - альтернативный ключ)
 - в рамке **Name** задавать имя ключа
 - нажатием кнопки **OK** вернуться из диалога *New Key Group* в *Key Group*
 - При необходимости добавить атрибуты из списка атрибутов сущности - **Available Attributes** в список атрибутов созданной группы - **Key Group Members**.

После создания всех групп и возврата в окно Erwin в модели должны быть отражены все сущности и их атрибуты, а у атрибутов, вошедших в альтернативные ключи или инверсионные входы стоять ссылки на соответствующие группы.



В приведенном рисунке для сущности "Студент" созданы два альтернативных ключа "Фамилия, Имя" (AK1) и "Дата рождения" (AK2), а также инверсионный вход "Код группы, дата рождения"(IE1).

1. Создать новое хранимое изображение "Физическая модель", оставив для него те же настройки, что и для "Атрибутов сущностей".
2. Переключиться на физический уровень.
3. Выполнить Forward Engineer/Generate **предварительно создав в среде Access пустую базу данных !!!!**
4. В окне Access Connection ввести в User Name "ADMIN", а в окне Database – путь к только что созданной пустой базе данных. Выполнить Connect и просмотреть созданную базу данных в Access-e.

8.3. Требования к оформлению отчета

Отчет должен содержать:

1. Индивидуальное задание
2. Порядок выполнения работы.
3. Распечатки всех хранимых изображений.
4. Для каждого атрибута – физическое и логическое имена, домен, словесное описание (Definition), ограничения (Note), значение по умолчанию.

Информация должна быть подготовлена вручную и с помощью созданных в среде ERwin отчетов. Для этого надо нажать кнопку (Report Browser) на панели инструментов ERwin, завести новый отчет File/New ERWin Report, дать ему имя, выбрать категорию и включить соответствующие опции в отчет. Просмотреть отчет.

9. ИТОГОВОЕ ЗАДАНИЕ

Требования к заданиям:

1. Каждая таблица должна содержать как минимум, заданное в квадратных скобках число записей.
2. Каждый запрос должен выдавать минимум 5 записей.
3. Информация, вводимая в таблицы должна быть осмысленной, соответствующей заданию.
4. Для каждой таблицы должна быть создана ленточная форма.
5. Для каждой таблицы и запроса должны быть созданы табличная форма и отчёт.
6. В верхней части каждой формы, отчёта и страницы, должна быть надпись-заголовок с именем таблицы или запроса.
7. На табличных формах должна быть реализована сортировка данных по выбранному полю, поиск информации и фильтрация по заданному полю.
8. Создать кнопочную форму.
9. В верхнюю часть кнопочной формы поместить надпись с названием БД.
10. В центре поместить вкладки "Формы" и "Отчёты".
11. На соответствующие вкладки поместить кнопки для открытия всех форм и отчётов. На кнопках или рядом с ними должны быть поясняющие надписи.
12. В нижней части кнопочной формы поместить кнопки для выхода из программы.
13. Сделать форму "Заставка" стартовой.

Задание №1: БД Ресторана

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Склад (Код ингредиента, Наименование ингредиента, Дата выпуска, Объём, Срок годности, Стоимость, Поставщик)[10 записей].4. Меню (Код блюда, Наименование блюда, Код ингредиента 1, Объём ингредиента 1, Код ингредиента 2, Объём ингредиента 2, Код ингредиента 3, Объём ингредиента 3, Стоимость, Время приготовления)[10 записей].5. Заказ (Дата, Время, ФИО заказчика, Телефон, Код блюда 1, Код блюда 2, Код блюда 3, Стоимость, Отметка о выполнении, Код сотрудника)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Отдел кадров2. Меню3. Заказы4. Сотрудники отдельных должностей5. Ингредиенты определённых поставщиков6. Выполненные и невыполненные заказы

Задание №2: БД Банка

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Вклады (Код вклада, Наименование вклада, Минимальный срок вклада, Минимальная сумма вклада, Код валюты, Процентная ставка, Дополнительные условия)[5 записей].4. Валюта (Код валюты, Наименование, Обменный курс)[3 записи].5. Вкладчики (ФИО вкладчика, Адрес, Телефон, Паспортные данные, Дата вклада, Дата возврата, Код вклада, Сумма вклада, Сумма возврата, Отметка о возврате вклада, Код сотрудника)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Отдел кадров2. Вклады3. Все вкладчики4. Сотрудники отдельных должностей5. Вклады отдельных валют6. Вкладчики с отдельными вкладами

Задание №3: БД Больницы.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Лекарства (Код лекарства, Наименование, Показания, Противопоказания, Упаковка, Стоимость)[5 записей].4. Болезни (Код болезни, Наименование, Симптомы, Продолжительность, Последствия, Код лекарства 1, Код лекарства 2, Код лекарства 3)[10 записей].5. Пациенты (ФИО пациента, Возраст, Пол, Адрес, Телефон, Дата обращения, Код болезни, Код сотрудника, Результат лечения)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Отдел кадров2. Болезни3. Пациенты4. Сотрудники отдельных должностей5. Болезни с одинаковыми симптомами6. Пациенты с одинаковыми болезнями

Задание №4: БД Гостиницы.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Номера (Код номера, Наименование, Вместимость, Описание, Стоимость, Код сотрудника)[5 записей].4. Услуги (Код услуги, Наименование, Описание, Стоимость)[5 записей].5. Клиенты (ФИО, Паспортные данные, Дата заселения, Дата выезда, Код номера, Код услуги 1, Код услуги 2, Код услуги 3, Стоимость, Код сотрудника)
Запросы:	<ol style="list-style-type: none">1. Отдел кадров2. Номера3. Клиенты4. Сотрудники отдельных должностей5. Клиенты, проживающие в разных номерах

Задание №5: БД Отдела внутренних дел.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности, Код звания)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Звания (Код звания, Наименование, Надбавка, Обязанности, Требования)[5 записей].4. Виды преступлений (Код вида преступления, Наименование, Статья, Наказание, Срок)[5 записей].5. Преступники (Номер дела, ФИО, Дата рождения, Пол, Адрес, Код вида преступления, Код пострадавшего, Состояние, Код сотрудника)[10 записей].6. Пострадавшие (Код пострадавшего, ФИО, Дата рождения, Пол, Адрес)[5 записей].
Запросы:	<ol style="list-style-type: none">1. Отдел кадров2. Преступники3. Сотрудники отдельных должностей4. Сотрудники отдельных званий5. Преступники по видам преступлений6. Преступники по состоянию

Задание №6: БД Аэропорта.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Самолёты (Код самолёта, Марка, Вместимость, Грузоподъемность, Код типа, Технические характеристики, Дата выпуска, Налётано часов, Дата последнего ремонта, Код сотрудника)[5 записей].4. Типы самолётов (Код типа, Наименование, Назначение, Ограничения).5. Экипажи (Код экипажа, Налётано часов, Код сотрудника 1, Код сотрудника 2, Код сотрудника 3)[5 записей].6. Рейсы (Код рейса, Дата, Время, Откуда, Куда, Код экипажа, Код самолёта, Время полёта)[5 записей].7. Билеты (ФИО пассажира, Паспортные данные, Место, Код рейса, Цена)
Запросы:	<ol style="list-style-type: none">1. Отдел кадров2. Самолёты3. Экипажи4. Рейсы5. Билеты6. Сотрудники отдельных должностей7. Самолёты разных типов8. Билеты отдельных рейсов

Задание №7: БД Видео проката.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Жанры (Код жанра, Наименование жанра, Описание)[5 записей].4. DVD (Код dvd, Наименование фильма, Год создание, Производитель, Страна, Главный актёр, Дата записи, Код жанра, Цена)[10 записей].5. Клиенты (ФИО, Адрес, Телефон, Паспортные данные, Дата взятия, Дата возврата, Отметка об оплате, Отметка о возврате, Код dvd 1, Код dvd 2, Код dvd 3, Код сотрудника)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Отдел кадров2. Полный список DVD3. DVD на руках4. Сотрудники отдельных должностей5. DVD отдельных жанров6. DVD фильмов за отдельные годы7. DVD фильмов с определёнными актёрами8. DVD на руках отдельных клиентов9. DVD оплаченных и не оплаченных, сданных и не сданных

Задание №8: БД Библиотеки.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Издательства (Код издательства, Наименование, Город, Адрес)[5 записей].4. Жанры (Код жанра, Наименование, Описание) [5 записей].5. Книги (Код книги, Наименование, Автор, Код издательства, Год издания, Код жанра) [10 записей].6. Читатели (Код читателя, ФИО, Дата рождения, Пол, Адрес, Телефон, Паспортные данные) [10 записей].7. Выданные книги (Код книги, Код читателя, Дата выдачи, Дата возврата, Отметка о возврате, Код сотрудника) [10 записей].
Запросы:	<ol style="list-style-type: none">1. Отдел кадров2. Каталог3. Книги на руках4. Сотрудники отдельных должностей5. Книги отдельных издательств6. Книги отдельных авторов7. Книги отдельных годов издания8. Сданные и не сданные книги9. Книги на руках отдельных читателей

Задание №9: БД Радиостанции.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Исполнители (Код исполнителя, Наименование, Описание)[5 записей].4. Жанры (Код жанра, Наименование, Описание)[5 записей].5. Записи (Код записи, Наименование, Код исполнителя, Альбом, Год, Код жанра, Дата записи, Длительность, Рейтинг)[10 записей].6. График работы (Дата, Код сотрудника, Время 1, Код записи 1, Время 2, Код записи 2, Время 3, Код записи 3)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Отдел кадров2. Музыкальный архив3. Сетка вещания4. Сотрудники отдельных должностей5. Записи отдельных исполнителей6. Записи отдельных жанров7. Сетка вещания по отдельным датам8. Сетка вещания по отдельным сотрудникам

Задание №10: БД Таксопарка.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Марки (Код марки, Наименование, Технические характеристики, Стоимость, Специфика)[5 записей].4. Тарифы (Код тарифа, Наименование, Описание, Стоимость)[5 записей].5. Дополнительные услуги (Код услуги, Наименование, Описание услуги, Стоимость)[5 записей].6. Автомобили (Код автомобиля, Код марки, Регистрационный номер, Номер кузова, Номер двигателя, Год выпуска, Пробег, Код сотрудника-шофёра, Дата последнего ТО, Код сотрудника-механика, Специальные отметки)[10 записей].7. Вызовы (Дата, Время, Телефон, Откуда, Куда, Код тарифа, Код услуги, Код автомобиля, Код сотрудника-оператора)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Отдел кадров2. Автопарк3. Список вызовов4. Сотрудники отдельных должностей5. Автомобили отдельных годов выпуска6. Автомобили отдельных марок7. Вызовы по отдельным тарифам8. Вызовы по отдельным датам

Задание №11: БД Туристического агентства.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Виды отдыха (Код вида, Наименование, Описание, Ограничения)[5 записей].4. Отели (Код отеля, Наименование, Страна, Город, Адрес, Телефон, Количество звёзд, Контактное лицо)[10 записей].5. Дополнительные услуги (Код услуги, Наименование, Описание, Цена) [5 записей].6. Клиенты (Код клиента, ФИО, Дата рождения, Пол, Адрес, Телефон, Паспортные данные)[5 записей].7. Путёвки (Дата начала, Дата окончания, Продолжительность, Код отеля, Код вида, Код услуги 1, Код услуги 2, Код услуги 3, Код клиента, Код сотрудника, Отметка о бронировании, Отметка об оплате)[10 записей].
-----------------	--

Запросы:	<ol style="list-style-type: none"> 1. Отдел кадров 2. Список путёвок 3. Сотрудники отдельных должностей 4. Путёвки по отдельным видам отдыха 5. Путёвки в отдельные страны 6. Путёвки в отдельные отели 7. Забронированные и не забронированные путёвки 8. Оплаченные и не оплаченные путёвки 9. Заказанные и не заказанные путёвки

Задание №12: БД Страховой компании.

Таблицы:	<ol style="list-style-type: none"> 1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. 2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. 3. Риски (Код риска, Наименование, Описание, Средняя вероятность)[5 записей]. 4. Виды полисов (Код вида полиса, Наименование, Описание, Условия, Код риска 1, Код риска 2, Код риска 3)[5 записей]. 5. Группы клиентов (Код группы, Наименование, Описание)[5 записей]. 6. Клиенты (Код клиента, ФИО, Дата рождения, Пол, Адрес, Телефон, Паспортные данные, Код группы)[10 записей]. 7. Полисы (Номер полиса, Дата начала, Дата окончания, Стоимость, Сумма выплаты, Код вида полиса, Отметка о выплате, Отметка об окончании, Код клиента, Код сотрудника)[10 записей].
Запросы:	<ol style="list-style-type: none"> 1. Отдел кадров 2. Риски полисов 3. Список клиентов 4. Список полисов 5. Сотрудники отдельных должностей 6. Клиенты отдельных групп 7. Полисы отдельных видов 8. Полисы, по которым производились выплаты и по которым не производились выплаты 9. Оконченные и неоконченные полисы

Задание №13: БД Брачного агентства.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Знаки зодиака (Код знака, Наименование, Описание)[5 записей].4. Отношения (Код отношения, Наименование, Описание)[5 записей].5. Национальности (Код национальности, Наименование, Замечания)[5 записей].6. Дополнительные услуги (Код услуги, Наименование, Описание, Цена)[5 записей].7. Клиенты (Код клиента, ФИО, Пол, Дата рождения, Возраст, Рост, Вес, Количество детей, Семейное положение, Вредные привычки, Хобби, Описание, Код знака, Код отношения, Код национальности, Адрес, Телефон, Паспортные данные, Информация о партнёре)[10 записей].8. Услуги (Код клиента, Дата, Код услуги 1, Код услуги 2, Код услуги 3, Стоимость, Код сотрудника)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Отдел кадров2. Список клиентов3. Список услуг4. Сотрудники отдельных должностей5. Клиенты отдельных знаков зодиака6. Клиенты по отношениям7. Клиенты отдельных национальностей8. Клиенты по хобби9. Клиенты по семейному положению

Задание №14: БД Сервис центра.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Запчасти (Код запчасти, Наименование, Функции, Цена)[5 записей].4. Ремонтируемые модели (Код модели, Наименование, Тип, Производитель, Технические характеристики, Особенности)[5 записей].5. Виды неисправностей (Код вида, Код модели, Описание, Симптомы, Методы ремонта, Код запчасти 1, Код запчасти 2, Код запчасти 3, Цена работы)[5 записей].6. Обслуживаемые магазины (Код магазина, Наименование, Адрес, Телефон)[5 записей].7. Заказы (Дата заказа, Дата возврата, ФИО заказчика, Серийный номер, Код вида неисправности, Код магазина, Отметка о гарантии, Срок гарантии ремонта, Цена, Код сотрудника)[10 записей].
-----------------	---

Запросы:	<ol style="list-style-type: none"> 1. Отдел кадров 2. Список неисправностей 3. Список заказов 4. Сотрудники отдельных должностей 5. Виды неисправностей отдельных моделей 6. Виды неисправностей отдельных типов устройств 7. Виды неисправностей моделей отдельных производителей 8. Заказы отдельных магазинов 9. Заказы отдельных неисправностей 10. Гарантийные и не гарантийные заказы

Задание №15: БД Школы.

Таблицы:	<ol style="list-style-type: none"> 1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. 2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. 3. Ученики (ФИО, Дата рождения, Пол, Адрес, ФИО отца, ФИО матери, Код класса, Дополнительная информация) [10 записей]. 4. Классы (Код класса, Код сотрудника-класного руководителя, Код вида, Количество учеников, Буква, Год обучения, Год создания)[5 записей]. 5. Виды классов (Код вида, Наименование, Описание)[5 записей]. 6. Предметы (Код предмета, Наименование, Описание, Код сотрудника-учителя)[10 записей]. 7. Расписание (Дата, День недели, Код класса, Код предмета, Время начала, Время окончания)[10 записей].
Запросы:	<ol style="list-style-type: none"> 1. Отдел кадров 2. Список учеников 3. Список классов 4. Список предметов 5. Расписание занятий 6. Сотрудники отдельных должностей 7. Классы различных годов обучения 8. Расписания для отдельных классов и дат 9. Отдельные виды классов 10. Предметы отдельных преподавателей

Задание №16: БД Транспортной компании.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Виды автомобилей (Код вида автомобиля, Наименование, Описание)[5 записей].4. Марки автомобилей (Код марки, Наименование, Технические характеристики, Описание) [5 записей].5. Виды грузов (Код вида груза, Наименование, Код вида автомобиля для транспортировки, Описание)[5 записей].6. Грузы (Код груза, Наименование, Код вида груза, Срок годности, Особенности)[5 записей].7. Автомобили (Код автомобиля, Код марки, Код вида автомобиля, Регистрационный номер, Номер кузова, номер двигателя, Год выпуска, Код сотрудника-водителя, Дата последнего ТО, Код сотрудника-механика)[5 записей].8. Рейсы (Код автомобиля, Заказчик, Откуда, Куда, Дата отправления, Дата прибытия, Код груза, Цена, Отметка об оплате, Отметка о возвращении, Код сотрудника)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Отдел кадров2. Транспортировка (Виды грузов и виды автомобилей).3. Перевозимые грузы4. Автопарк5. Сотрудники отдельных должностей6. Отдельные виды грузов7. Отдельные виды автомобилей

Задание №17: БД Проката автомобилей.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Марки автомобилей (Код марки, Наименование, Технические характеристики, Описание) [5 записей].4. Дополнительные услуги (Код услуги, Наименование, Описание, Цена)[5 записей].5. Автомобили (Код автомобиля, Код марки, Регистрационный номер, Номер кузова, Номер двигателя, Год выпуска, Пробег, Цена автомобиля, Цена дня проката, Дата последнего ТО, Код сотрудника-механика, Специальные отметки, Отметка о возврате)[10 записей].6. Клиенты (Код клиента, ФИО, Пол, Дата рождения, Адрес, Телефон, Паспортные данные) [5 записей].7. Прокат (Дата выдачи, Срок проката, Дата возврата, Код автомобиля, Код клиента, Код услуги 1, Код услуги 2, Код услуги 3, Цена проката, Отметка об оплате, Код сотрудника)[10 записей].
-----------------	--

Запросы:	<ol style="list-style-type: none"> 1. Отдел кадров 2. Автопарк 3. Автомобили в прокате 4. Сотрудники отдельных должностей 5. Автомобили отдельных марок 6. Автомобили, находящиеся и не находящиеся в прокате 7. Автомобили выданные и возвращённые в определённую дату 8. Оплаченные и не оплаченные автомобили в прокате
-----------------	--

Задание №18: БД Оптового склада.

Таблицы:	<ol style="list-style-type: none"> 1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. 2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. 3. Товары (Код товара, Код типа, Производитель, Наименование, Условия хранения, Упаковка, Срок годности) [10 записей]. 4. Типы товаров (Код типа, Наименование, Описание, Особенности) [5 записей]. 5. Поставщики (Код поставщика, Наименование, Адрес, Телефон, Код поставляемого товара 1, Код поставляемого товара 2, Код поставляемого товара 3) [5 записей]. 6. Заказчики (Код заказчика, Наименование, Адрес, Телефон, Код потребляемого товара 1, Код потребляемого товара 2, Код потребляемого товара 3) [5 записей]. 7. Склад (Дата поступления, Дата заказа, Дата отправки, Код товара, Код поставщика, Код заказчика, Способ доставки, Объём, Цена, Код сотрудника) [10 записей].
Запросы:	<ol style="list-style-type: none"> 1. Отдел кадров 2. Список товаров. 3. Список поставщиков 4. Список заказчиков 5. Заказы 6. Сотрудники отдельных должностей 7. Отдельные типы товаров 8. Отдельные поставщики 9. Отдельные заказчики 10. Отдельные способы доставки

Задание №19: БД Строительной компании.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Виды работ (Код вида, Наименование, Описание, Цена работы, Код материала 1, Код материала 2, Код материала 3)[5 записей].4. Материалы (Код материала, Наименование, Упаковка, Описание, Цена) [5 записей].5. Бригады (Код бригады, Код сотрудника 1, Код сотрудника 2, Код сотрудника 3) [5 записей].6. Заказчики (Код заказчика, ФИО, Адрес, Телефон, Паспортные данные)[5 записей].7. Заказы (Код заказчика, Код вида работ, Код бригады, Стоимость, Дата начала, Дата окончания, Отметка о завершении, Об оплате, Код сотрудника) [10 записей].
Запросы:	<ol style="list-style-type: none">1. Отдел кадров2. Список работ и бригад3. Список заказов4. Сотрудники отдельных должностей5. Отдельные виды работ6. Заказы на конкретные работы7. Заказы отдельных заказчиков8. Заказы, выполняемые отдельными бригадами9. Завершённые и не завершённые заказы10. Оплаченные и неоплаченные заказы

Задание №20: БД Риэлтерской фирмы.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Виды услуг (Код вида услуги, Наименование, Описание, Цена)[5 записей].4. Виды квартир (Код вида, Наименование, Описание)[5 записей].5. Продавцы (Код продавца, ФИО, Пол, Дата рождения, Адрес проживания, Телефон, Паспортные данные, Код вида квартиры, Адрес квартиры, Количество комнат, Площадь, Отметка о раздельном санузле, Отметка о наличии телефона, Цена, Дополнительная информация)[10 записей].6. Покупатели (Код покупателя, ФИО, Пол, Дата рождения, Адрес проживания, Телефон, Паспортные данные, Код вида квартиры, Количество комнат, Площадь, Отметка о раздельном санузле, Отметка о наличии телефона, Цена, Дополнительные пожелания)[10 записей].7. Договоры (Дата заключения, Код продавца, Код покупателя, Сумма сделки, Стоимость услуг, Код вида услуги, Код сотрудника)[10 записей].
-----------------	--

Запросы:	<ol style="list-style-type: none"> 1. Отдел кадров 2. Продажа 3. Покупка 4. Заключённые договора 5. Сотрудники отдельных должностей 6. Квартиры различных видов 7. Продаваемые квартиры 8. Покупаемые квартиры различных видов 9. Договора, заключённые отдельными сотрудниками
-----------------	--

Задание №21: БД Рекламного агентства.

Таблицы:	<ol style="list-style-type: none"> 1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. 2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. 3. Виды рекламы (Код вида, Наименование, Описание) [5 записей]. 4. Дополнительные услуги (Код услуги, Наименование, Описание, Стоимость) [5 записей]. 5. Места расположения (Код места, Наименование, Расположение, Код вида, Описание, Стоимость) [10 записей]. 6. Заказчики (Код заказчика, ФИО, Адрес, Телефон) [10 записей]. 7. Заказы (Дата заказа, Дата начала, Дата окончания, Код заказчика, Код места, Код услуги 1, Код услуги 2, Код услуги 3, Стоимость, Отметка об оплате, Код сотрудника) [10 записей].
Запросы:	<ol style="list-style-type: none"> 1. Отдел кадров 2. Список мест 3. Список заказов 4. Сотрудники отдельных должностей 5. Заказы, сделанные в определённые даты 6. Оплаченные и неоплаченные заказы 7. Места расположения по видам рекламы

Задание №22: БД Компьютерной фирмы.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Виды комплектующих (Код вида, Наименование, Описание)[15 записей].4. Комплектующие (Код комплектующего, Код вида, Марка, Фирма производитель, Страна производитель, Дата выпуска, Характеристики, Срок гарантия, Описание, Цена)[15 записей].5. Заказчики (Код заказчика, ФИО, Адрес, Телефон)[10 записей].6. Услуги (Код услуги, Наименование, Описание, Стоимость)[5 записей].7. Заказы (Дата заказа, Дата исполнения, Код заказчика, Код комплектующего 1, Код комплектующего 2, Код комплектующего 3, Доля предоплаты, Отметка об оплате, Отметка об исполнении, Общая стоимость, Срок общей гарантии, Код услуги 1, Код услуги 2, Код услуги 3, Код сотрудника)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Отдел кадров2. Список комплектующих3. Список заказов4. Сотрудники отдельных должностей5. Комплектующие отдельных видов6. Заказы отдельных заказчиков7. Заказы по датам заказа

Задание №23: БД ГИБДД.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности, Код звания)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Звания (Код звания, Наименование, Надбавка, Обязанности, Требования)[5 записей].4. Марки автомобилей (Код марки, Наименование, Фирма производитель, Страна производитель, Дата начала производства, Дата окончания производства, Характеристики, Категория, Описание)[10 записей].5. Водители (Код водителя, ФИО, Дата рождения, Адрес, Паспортные данные, Номер водительского удостоверения, Дата выдачи удостоверения, Дата окончания удостоверения, Категория удостоверения, Описание, Код сотрудника)[15 записей].6. Автомобили (Код автомобиля, Код водителя, Код марки, Регистрационный номер, Номер кузова, Номер двигателя, Номер техпаспорта, Дата выпуска, Дата регистрации, Цвет, Технический осмотр, Дата технического осмотра, Описание, Код сотрудника)[15 записей].7. Автомобили в угоне (Дата угона, Дата обращения, Код автомобиля, Код водителя, Обстоятельства угона, Отметка об нахождении, Дата нахождения, Код сотрудника)[5 записей].
-----------------	---

Запросы:	<ol style="list-style-type: none"> 1. Отдел кадров 2. Список автомобилей 3. Список угонов 4. Сотрудники отдельных должностей 5. Сотрудники отдельных званий 6. Автомобили одного владельца 7. Автомобили прошедшие и не прошедшие технический осмотр 8. Найденные и не найденные угнанные автомобили
-----------------	--

Задание №24: БД Кинотеатра.

Таблицы:	<ol style="list-style-type: none"> 1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. 2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. 3. Жанры (Код жанра, Наименование, Описание)[5 записей]. 4. Фильмы (Код фильма, Наименование, Код жанра, Длительность, Фирма производитель, Страна производитель, Актёры, Возрастные ограничения, Описание)[10 записей]. 5. Репертуар (Код сеанса, Дата, Время начала, Время окончания, Цена билета)[10 записей]. 6. Места (Код сеанса, Номер места, Занятость, Код сотрудника)[15 записей].
Запросы:	<ol style="list-style-type: none"> 1. Отдел кадров 2. Список фильмов 3. Билеты 4. Сотрудники отдельных должностей 5. Фильмы отдельных жанров 6. Билетов на сеансы определённых дат 7. Занятые и незанятые места

Задание №25: БД Автосалона.

Таблицы:	<ol style="list-style-type: none">1. Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].2. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].3. Производители (Код производителя, Наименование, Страна, Адрес, Описание, Код сотрудника)[5 записей].4. Дополнительное оборудование (Код оборудования, Наименование, Характеристики, Цена)[5 записей].5. Тип кузова (Код типа кузова, Название, Описание)[5 записей].6. Автомобили (Код автомобиля, Марка, Код производителя, Код типа кузова, Дата производства, Цвет, Номер кузова, Номер двигателя, Характеристики, Код оборудования 1, Код оборудования 2, Код оборудования 3, Цена, Код сотрудника)[10 записей].7. Заказчики (ФИО, Адрес, Телефон, Паспортные данные, Код автомобиля, Дата заказа, Дата продажи, Отметка о выполнении, Отметка об оплате, Процент предоплаты, Код сотрудника)[10 записей].
Запросы:	<ol style="list-style-type: none">1. Отдел кадров2. Каталог автомобилей3. Список заказов4. Сотрудники отдельных должностей5. Автомобили отдельных производителей6. Автомобили с отдельными типами кузова7. Выполненные и невыполненные заказы8. Оплаченные и неоплаченные заказы

ПРИЛОЖЕНИЕ

ПРИМЕРЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ

При создании реляционной базы данных пользователь должен располагать описанием предметной области. На основе такого описания в процессе проектирования БД осуществляется определение состава и структуры данных.

ПРИМЕР 1

Рассмотрим пример проектирования с использованием метода сущность-связь построения **ER**-диаграммы и в последствии – реляционной схемы БД.

Предметная область: информационная система «ОТДЕЛ КАДРОВ».

Исходные данные:

1. Фамилия сотрудника, имя, отчество, домашний адрес, телефон, дата рождения, образование.
2. Должность, дата зачисления, оклад, объем должности (число ставок).
3. Наименование подразделения, количество штатных единиц, фонд заработной платы.

Один и тот же сотрудник может зачисляться на работу на разные ставки по разным должностям; в одном и том же подразделении может работать несколько сотрудников.

Перечень должностей, которые могут быть в каждом отделе, определяется штатным расписанием.

Зачисление сотрудника на должность в отделе является сущностью-ассоциацией, связывающей не сотрудника с должностью и отделом, а сотрудника - со штатным расписанием, так как зачисление может производиться только в соответствии со штатным расписанием.

Один сотрудник может быть зачислен на несколько работ.

После выделения сущностей и определения связей [1] получим следующую **ER**-диаграмму:

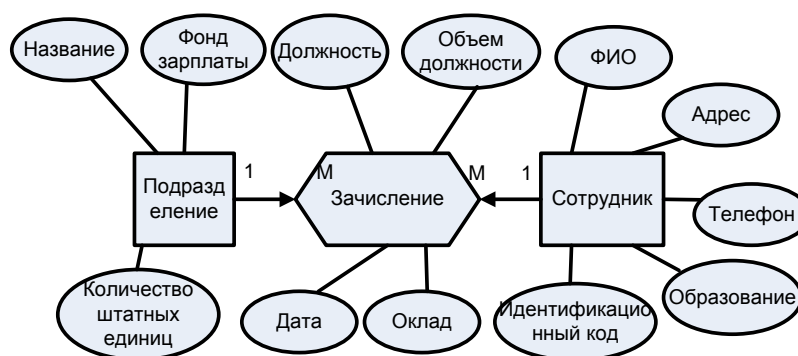


Рис 1. ER-диаграмма предметной области

В результате анализа предметной области получена уточненная **ER**-диаграмма, которая изображена на рис.2. В ней введен ряд новых сущностей и ассоциаций. В полученной **ER**-диаграмме используются обозначения сущностей в соответствии с [1].

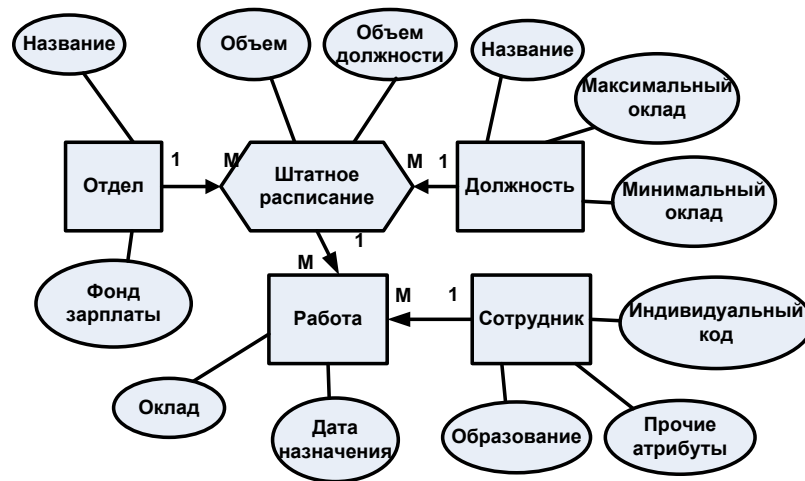


Рис 2. Уточненная ER-диаграмма

Атрибут **Объем должности** сущности **Штатное расписание** – это **Число ставок**, а атрибут **Объем** – **Код строки штатного расписания**.

Должность выбирается из предустановленного набора должностей и не зависит ни от сотрудника, ни от отдела. Поэтому **ДОЛЖНОСТЬ** становится отдельной сущностью.

ШТАТНОЕ_РАСПИСАНИЕ является ассоциативной сущностью, связывающей сущности **ОТДЕЛ** и **ДОЛЖНОСТЬ**. Строка штатного расписания содержит информацию о количестве ставок одной должности в одном отделе. Таким образом, одной должности может соответствовать несколько строк в **ШТАТНОЕ_РАСПИСАНИЕ** (для разных отделов), и одному отделу - тоже несколько строк (для разных должностей).

Одна строка **ШТАТНОЕ_РАСПИСАНИЕ** может быть связана с несколькими строками **РАБОТ**, так как ставок по данной строке может быть несколько и зачисление может происходить на дробную ставку (например, объем ставки 0,5). "Теоретически" общий объем работ (объем должностей), связанных со строкой штатного расписания, не должен превышать указанное в штатном расписании количество ставок, но на практике это правило, по-видимому, может нарушаться. Кроме того, опять-таки "теоретически", сумма окладов по работам, связанным (через штатное расписание) с одним отделом, не должна превышать фонда зарплаты отдела, но это правило тоже не абсолютное.

Среди возможных запросов может быть запрос, подразумевающий наличие информации об образовании сотрудника. Отсюда – необходимость иметь атрибут **Образование** в сущности **СОТРУДНИК**.

Атрибуты сущности **ОТДЕЛ** – *код отдела* (первичный ключ), *наименование* и *фонд* отдела.

Атрибуты сущности **СОТРУДНИК**: идентификационный *Код_сотрудника* (первичный ключ); *ФИО*, *Дата_рождения*, *Адрес*, *Телефон*, *Образование*, *Код отдела*.

Атрибуты сущности **ДОЛЖНОСТЬ**: *Код_должности* (первичный ключ); *Наименование*; граничные оклады для должности (*Мин_оклад* и *Мак_оклад*);

Атрибуты сущности **ШТАТНОЕ_РАСПИСАНИЕ**: *Код_строки шт_расписания* (первичный ключ); *Код_должности*; *Код_отдела*; *Число_ставок*.

Атрибуты сущности РАБОТА (Зачисление на должность): *Код_сотрудника; Код_штатного_расписания; Оклад* ("теоретически" оклад должен лежать в рамках граничных значений для должности); *Дата_зачисления; Объем_должности*.

ПРИМЕР 2

Рассмотрим пример проектирования, основанный на анализе документов, содержащих данные, которые должны быть размещены в БД.

Предметная область: информационная система «ПОСТАВКА ТОВАРОВ»

Фирма осуществляет поставку товаров со своих складов в соответствии с договорами. Каждый товар имеет наименование, единицу измерения и цену. Имеется список покупателей фирмы, который пополняется в процессе работы. Каждый покупатель – юридическое лицо - имеет наименование, идентификационный номер (ИНН), адрес, телефон, номер расчетного счета и наименование банка.

В договоре о поставке товара кроме реквизитов поставщика и покупателя указываются реквизиты товара, срок поставки, минимальная партия поставки, количество поставляемого товара и сумма поставки.

Учетная информация с данными по фактической отгрузке товаров покупателю со склада фирмы в соответствии с договорами содержится в накладных. Каждая накладная содержит кроме реквизитов поставщика и покупателя также и все реквизиты продаваемого товара, цену, ставку НДС, количество и сумму.

Для выделения информационных объектов необходимо сделать следующее:

- выявить документы, используемые фирмой в своей деятельности, и их реквизиты (поля);
- определить функциональную зависимость между реквизитами для каждого документа в отдельности;
- выбрать все зависимые реквизиты и указать для каждого из них все его ключевые (один или несколько), т.е. те, от которых он зависит функционально полно;
- сгруппировать реквизиты, зависимые от одинаковых ключевых реквизитов; полученные группы и будут составлять информационный объект (ИО) - сущность.

Совокупность полей - реквизитов выделенного объекта должна отвечать требованиям нормализации:

- ИО должен содержать уникальный ключ (простой или составной);
- все остальные (описательные) реквизиты должны быть независимыми друг от друга;
- все реквизиты, входящие в составной ключ, также должны быть независимыми;
- каждый описательный реквизит должен функционально полно зависеть от ключа, т.е. каждому значению ключа должно соответствовать только одно значение описательного реквизита;
- при составном ключе описательный реквизит должен зависеть целиком от всей совокупности реквизитов, образующих ключ;
- каждый описательный реквизит не должен зависеть от ключа транзитивно, через другой промежуточный реквизит.

Определим в документе **Справочник Товаров (ТОВАР)** функциональные зависимости между реквизитами и присвоим им сокращенные имена (рис. 4).

Реквизиты	Имя реквизита	Функциональные зависимости
Код товара	<i>Код_тов</i>	
Наименование товара	<i>Наим_тов</i>	
Единица измерения	<i>Ед</i>	
Цена	<i>Цена</i>	
Ставка НДС	<i>Ндс</i>	

Рис. 4. Функциональные зависимости в документе **Справочник Товаров**

Из анализа документа очевидно, что ключом является *Код_тов*, от которого функционально полно зависят все остальные описательные реквизиты. Все реквизиты составляют содержание **ИО ТОВАР**.

Аналогично легко определить **ИО ПОКУПАТЕЛЬ** и **ИО СКЛАД**:

ПОКУПАТЕЛЬ (*Код_пок* – ключевое поле, *ИНН*, *Наим_пок*, *Адрес_пок*, *Ном_расч*, *Банк*).

СКЛАД (*Код_ск* - ключевое поле, *Наим_ск*, *Адрес_ск*, *Отв_лицо*).

Определим состав **ИО ДОГОВОР**, содержащего данные о плановых поставках товара.

Кодом покупателя однозначно определяются описательные реквизиты покупателя – наименование, ИНН, адрес, телефон, расчетный счет, банк. В таблице зависимостей **Договор** эти реквизиты можно не отображать, поскольку информационный объект **ПОКУПАТЕЛЬ**, образованный этими реквизитами, был уже выделен.

Описательные реквизиты товара (наименование, единица измерения, цена) однозначно определены кодом товара. Эти реквизиты также можно не включать в таблицу зависимостей **Договор**, поскольку ранее их взаимосвязи были установлены при анализе **ИО ТОВАР**. Остальные реквизиты одного договора (количество поставки товара, минимальная партия, сумма за товар) однозначно определяются кодом товара. На всем же множестве договоров эти реквизиты будут функционально полно зависеть от составного ключа:

Ном_дог+код_тов. Будем исходить из того, что в договоре для одного товара возможно несколько сроков поставки, тогда срок поставки войдет в составной ключ *Ном_дог+Код_тов+Срок_пост* (рис. 5).

Наименование реквизитов	Имя реквизита	Функциональные зависимости
Номер договора	<i>Ном_дог</i>	
Дата договора	<i>Дата_дог</i>	
Код покупателя	<i>Код_пок</i>	
Сумма по договору	<i>Сумма_дог</i>	
Код товара	<i>Код_тов</i>	
Срок поставки	<i>Срок_пост</i>	
Количество поставки	<i>Кол_пост</i>	
Минимальная партия	<i>Мин_пост</i>	
Сумма поставки	<i>Сумма_пост</i>	

Рис.5. Функциональные зависимости в документе **Договор**

Сгруппируем реквизиты, одинаково зависимые от ключевых, и объединим их с ключевыми реквизитами в один информационный объект. В результате получим следующие ИО:

ДОГОВОР (*Ном_дог* – ключевое поле, *Дата_дог*, *Код_пок*, *Сумма_дог*).

ПОСТАВКА_ПЛАН(*Ном_дог*, *Код_тов*, *Срок_пост*, *Кол_пост*, *Мин_пост*, *Сумма_пост*).

Выполним анализ документа **Накладная**, содержащего информацию об отгрузке товаров по договорам.

Номер накладной не повторяется на одном складе, но может повторяться на разных складах данной фирмы. Поэтому для уникальной идентификации накладной необходим составной ключ: *Ном_накладной+Код_склада*.

Описательные реквизиты товара (*Наим_тов*, *Еи*, *Цена*, *НДС*) однозначно определены *Код_тов*, что уже учтено в **ИО ТОВАР**.

Количество отгруженного товара и сумма за товар определяются *Код_тов* в соответствующей строке, а полная идентификация по всем накладным определяется составным ключом: *Ном_накл+Код_ск+Код_тов* (рис. 6)

Наименование реквизитов	Имя реквизита	Функциональные зависимости
Номер накладной	<i>Ном_накл</i>	
Код склада	<i>Код_ск</i>	
Дата отгрузки	<i>Дата_отгр</i>	
Номер договора	<i>Ном_дог</i>	
Сумма всего	<i>Сумма_накл</i>	
Код товара	<i>Код_тов</i>	
Количество отгруженного товара	<i>Кол_отгр</i>	
Сумма за товар	<i>Сумма_отгр</i>	

Рис. 6. Функциональные зависимости в документе **Накладная**

Из рис.6 видно, что реквизиты *Дата_отгр*, *Ном_дог* и *Сумма_накл* зависят от ключевых атрибутов *Ном_накл + Код_ск*, а реквизиты *Кол_отгр* и *Сумма_отгр* зависят от *Ном_накл+ Код_ск + Код_тов*.

Сгруппируем реквизиты, одинаково зависимые от ключевых, и объединим их вместе с ключевыми реквизитами в соответствующие информационные объекты:

НАКЛАДНАЯ (*Ном_накл*, *Код_ск*, *Дата_отгр*, *Ном_дог*, *Сумма_накл*). Ключ составной– *Ном_накл + Код_ск*.

ОТГРУЗКА(*Ном_накл*, *Код_ск*, *Код_тов*, *Кол_отгр*, *Сумма_отгр*). Ключ составной – *Ном_накл+Код_ск+Код_тов*.

Определение структуры базы данных.

Для определения структуры БД (построения информационно-логической модели) необходимо установить связи между сущностями-ИО. Объекты **ДОГОВОР** и **ТОВАР** имеют отношения М:М, поэтому необходима сущность-связка, в качестве которой выступает сущность **ПОСТАВКА_ПЛАН**. Такая же ситуация и между объектами **ТОВАР** и **НАКЛАДНАЯ** – в качестве связки между ними выступает сущность **ОТГРУЗКА**. Остальные связи – 1:М - между объектами **ПОКУПАТЕЛЬ-ДОГОВОР**, **СКЛАД-НАКЛАДНАЯ**, **ДОГОВОР-НАКЛАДНАЯ**.

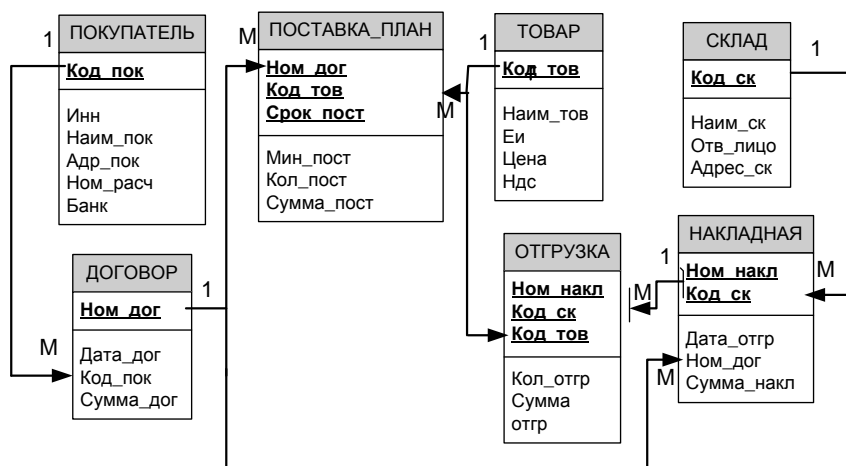


Рис. 7. Логическая структура (реляционная схема) БД

Логическая структура БД изображена на рис.7. Все связи характеризуются отношением 1:М. Имена ключевых полей подчеркнуты.

Список литературы

1. Г.К.Сосновиков, В.Н. Шакин. Основы реляционных баз данных. Учебное пособие. МТУСИ. – М, 2013 г.
2. Шакин В.Н., Сосновиков Г.К, Юскова И.Б. Методические указания по дисциплине ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПОСТРОЕНИЯ БД/ МТУСИ.- М., 2004.
3. Сосновиков Г.К., Шакин В.Н., Юскова И.Б.. Методические указания и контрольные задания по дисциплине ОСНОВЫ ПОСТРОЕНИЯ БД/ МТУСИ.- М., 2004.
4. Сосновиков Г.К., Шакин В.Н.. Практикум по проектированию баз данных средствами СУБД Access./ МТУСИ.- М., 2006.
5. Бекаревич Ю.Б., Пушкина Н.В. Microsoft Access за 21 занятие. – СПб.:БХВ-Петербург, 2005.-544 с.:ил.
6. Пушников А.Ю. Введение в системы управления базами данных. Часть 1. Реляционная модель данных: Учебное пособие/Изд-е Башкирского ун-та. - Уфа, 1999. - 108 с. - ISBN 5-7477-0350-1.
7. Пушников А.Ю. Введение в системы управления базами данных. Часть 2. Нормальные формы отношений и транзакции: Учебное пособие/Изд-е Башкирского ун-та. - Уфа, 1999. - 138 с. - ISBN 5-7477-0351-X.
8. Зверева Н.Н. Методические указания к лабораторным работам по Access 97/Изд. Уфимского авиационного института./Каф. информатики и информационных технологий.-Уфа, 2001.
9. Палеев А.Б. Уч. пособие по курсу: Базы данных / Лаб. работа: Использование Erwin для разработки модели данных. – М: МГУПП, 2002.
10. Маклаков С.В.. ВРwin и ERwin CASE-средства разработки информационных систем - М.: "ДИАЛОГ-МИФИ", 2000.

11. Microsoft Office Access 2007. Русская версия. Серия «Шаг за шагом»/ Ламберт С., Ламберт М.Д. Ш, Преппернау Дж.; пер. с англ.–М:ЭКОМ Паблшера, 2007.- 432 с.: ил
12. Геннадий Гурвиц: Microsoft Access 2010. Разработка приложений на реальном примере (+CD)/ БХВ-Петербург, СПб.2010.
13. Базы данных. Учебник для вузов./Под ред. Хомоненко А.Д. СПб.: КОРОНА ПРИНТ 2009.

Содержание

- 1. Лабораторная работа 1. СОЗДАНИЕ И РЕДАКТИРОВАНИЕ ТАБЛИЦ. ИЗМЕНЕНИЕ СТРУКТУРЫ ТАБЛИЦ. СОРТИРОВКА И ПОИСК ДАННЫХ. ФОРМЫ.**
- 2. Лабораторная работа 2. РАБОТА С ДАННЫМИ С ИСПОЛЬЗОВАНИЕМ ЗАПРОСОВ. ЗАПРОСЫ-ВЫБОРКИ.**
- 3. Лабораторная работа 3. РАБОТА С ДАННЫМИ С ПОМОЩЬЮ ЗАПРОСОВ-ДЕЙСТВИЙ. ПЕРЕКРЕСТНЫЕ ЗАПРОСЫ.**
- 4. Лабораторная работа 4. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ. РАЗРАБОТКА ER-МОДЕЛИ. СОЗДАНИЕ БД, РАЗРАБОТКА ЗАПРОСОВ К БД.**
- 5. Лабораторная работа 5. СОЗДАНИЕ ОТЧЕТОВ.**
- 6. Лабораторная работа 6. ИСПОЛЬЗОВАНИЕ МАКРОСОВ.**
- 7. Лабораторная работа 7. Зачетное занятие.**
- 8. Лабораторная работа 8. Изучение CASE средства ERWin.**
- 9. ИТОГОВОЕ ЗАДАНИЕ.**
- 10. ПРИЛОЖЕНИЕ.**
- 11. СПИСОК ЛИТЕРАТУРЫ.**